Book of Abstracts

ODYSSEUS 2024

Carmona, 19-24 May, 2024

# Table of Contents

# An Efficient Large Neighborhood Search for the Daily Drayage Problem with Synchronization

**Diana Abi-Nader**[1,2], **Nabil Absi**[1], **Dominique Feillet**[1], **Thierry Garaix**[1], **Taki Eddine Korabi**[2]

[1]Mines Saint-Etienne, CNRS, UMR 6158 LIMOS, 13541 Gardanne France
Emails: {diana.serhal,absi,feillet,garaix}@emse.fr

[2]DMS Logistics, 7 rue de la Republique, 13002 Marseille, France
Email: takieddine.korabi@gmail.com

## 1 Introduction

As an essential component of global trade economies, intermodal container transportation is expected to continue growing despite the challenging economic conditions the world is currently facing [1]. This mode of transportation relies on at least two different modes to move containers from their origins to destinations. It is composed of three main segments: main-haulage involving maritime container movements, and pre-/end-haulage for inland container movements. Inland container movements by trucks are called drayage operations.

Combining multiple means of transportation may help reducing container transportation costs. However, drayage operations represent the least optimized segment, accounting for up to 80% of the total costs despite traveling shortest distances [2]. This leads to higher shipment costs for the different logistic actors, customer shipment delays, and congestion at terminals with an increase in $CO_2$ emissions. Therefore, the challenge is to provide a planning for transportation companies to optimize drayage operations, taking into consideration multiple complex real-world constraints.

Drayage operations consist in moving full and empty containers between different predetermined locations to fulfill customer requests in the local area of a truck transportation company within specific time windows while respecting the driver working hours. In the case of import requests, customers receive a full container before releasing an empty container to be picked-up. Conversely, in the case of export requests, customers must be supplied with an empty container to be filled before delivery for export. Hence, import and export requests consist of two sequentially ordered requests that must be synchronized taking into account the service time and the time needed to fill or empty a container. To our knowledge, no existing study has solved the drayage operations problem with request synchronization proposing an efficient heuristic solution to solve real-world large instances.

The existing literature on drayage operations often models the problem in a simplified way, affecting its industrial applicability. While Moghaddam et al. [3] proposed a generalized model that account for composite requests, they forced the execution of both sub-requests, did not consider the time offset between two sub-requests, and did not solve the synchronization problem. In this work, we address a real-world Daily Drayage Problem with Time windows and Composite Requests Synchronization (DDPTW-CRS) where a transportation company owns a heterogeneous fleet of trucks and serves all types of container requests with heterogeneous container sizes. The real network is composed of a yard from which trucks start and end their shifts, depots where empty containers are stocked, terminals where ships and trucks load and unload containers, and customers with import and export requests. The main objective of the planning is to maximize the number of fulfilled requests while minimizing the total truck traveling time with a limited fleet.

The DDPTW-CRS can be seen as a generalization of the Pickup and Delivery Problem with Time Windows (PDPTW). In addition to the classical PDP constraints for picking and delivering requests, in the DDPTW-CRS, one should satisfy a precedence constraint between two linked requests, introducing a minimum time-lag constraint. When using neighborhood search based heuristics, these precedence constraints make the computations more complex since it introduces inter-route dependencies. In fact, if two linked requests are satisfied with two different trucks, modifying a route may impact other routes.

While the Large Neighborhood Search (LNS) is the most used heuristic to solve similar problems with success, efficient pre-processing for reducing time evaluations, when used, are based on the forward time slack principle introduced by Savelsbergh [4]. For example, Masson et al. [5] extended [4] to enable efficient evaluations for requests with transfers. While this approach can handle the synchronization of two nodes in two different routes, further adaptations are required to handle operations synchronization with a minimal time lag as well as the synchronization of two requests in the same route as in the DDPTW-CRS. Thus, to solve our problem efficiently with real-world large instances, we adapt the work of Vidal et al. [6] introducing efficient pre-processing procedures based on the sub-routes concatenations concept integrated within a dedicated LNS heuristic.

## 2 Solution methodology

The solution method proposed in this work is based on a LNS heuristic involving constant-time feasibility checks. Our heuristic follows the classical LNS scheme using destroy and repair operators. It consists of iteratively removing and inserting a set of requests in a solution. Our destroy and repair operators are based on feasible solutions that is, when destroying a solution, if the first request of a composite request is removed its complementary request is also removed. Similarly, when repairing a solution if the first request is not inserted, we do not allow the insertion of the second request. To enhance diversification, we integrated a simulated annealing mechanism for solution acceptance and we apply local search (LS) as an improvement procedure.

The efficiency of the LNS and LS is directly linked to the number of evaluations they can perform within the allotted time. Repair operators are the most time consuming. In our problem, the main difficulty comes from the precedence constraints between requests that may link several routes. Inserting a location within a given route can impact the timing of linked routes, and thus, affect the feasibility of the whole solution. Thus, it is critical to evaluate efficiently the feasibility of inserting a new request so as to be able to explore a larger part of the solution space without affecting the computational time.

For that, we introduce pre-processing procedures based on the concept of sub-routes concatenations introduced in Vidal et al. [6]. It relies on the concept that any route can be obtained as the concatenation of sub-sequences and that the characteristics of the route can be efficiently computed based on the characteristics of every sub-sequence. Thus, pre-processing essential information on those sub-sequences and defining an adequate concatenation operator can speed up the evaluation process. For any route sub-sequence, we maintain four parameters: the sum of the travelling and service time for the sub-sequence $T(\rho)$, the earliest completion time of its last node $E(\rho)$, the latest feasible starting time of its first node $L(\rho)$ and an indicator of the sub-sequence feasibility $isF(\rho)$. Sub-sequences with a single location are first initialized. Sub-sequences with multiple locations are than derived by applying sub-sequences concatenations in $O(1)$. Based on pre-processed information, concatenations are later applied to evaluate the feasibility of a new insertion in $O(1)$ based on time constraints and route duration.

When evaluating the insertion of a new request, the route is segmented into different distinct sub-sequences based on the pick-up and delivery nodes, testing various positions. Concatenation operations are then applied between these divided sub-sequences and the pick-up and delivery nodes using pre-processed values. However, existing pre-processing procedures do only rely on the evaluation of a single route independently from the others. In our problem, the request synchronization constraint implies that any insertion feasibility must be additionally checked with respect to all the inter-related routes that can be affected. Consequently, our pre-processed information must include additional timing information on the linked routes. Thus, we adapted the work of Vidal et al. [6] to account for the time-lag constraint between two sub-requests of a composite request when initializing single and multiple locations sub-sequences. We managed the evaluation of a new insertion by considering the feasibility of the already inserted composite requests, whether inserted into the same route or in different routes, without affecting the constant time evaluation.

## 3 Results and Discussions

Experiments were conducted using real-life data provided by two transportation companies in the region of Marseille-Fos, France, to demonstrate the efficiency of our approach. To study the impact of different real-world constraints on our solution method, we generated a set of realistic instances of different sizes and parameters using the provided data. We observed that as the percentage of composite requests increase, the computational time increases if the fleet is sufficient otherwise the computational time decreases. Additionally, using our pre-processing technique on instances which are entirely constituted of composite requests, the computational time can be divided by three compared to a LNS applied without pre-processing. Table 1 presents the average computation time and the average inserted requests for 500 iterations across instances of varying size (small, medium and large), where each category is composed of 750 instances. Further comprehensive results on key performance indicators will be presented during the conference.

| Instance size | Computation time (secs) | Number of served requests |
|---|---|---|
| Small (26 requests) | 0 | 22.33 |
| Medium (76 requests) | 31.28 | 74.63 |
| Large (150 requests) | 394.36 | 145.54 |

Table 1: Average results on a set of 2250 instances

## References


[1] UNCTAD, "Review of Maritime Transport 2023", Technical Report, United Nations Publication New York and Geneva, 2023.

[2] T. E. Notteboom, J. P. Rodrigue, "Port regionalization: towards a new phase in port development", *Maritime Policy & Management* 32(3), 297-313, 2005.

[3] M. Moghaddam, R. H. Pearce, H. Mokhtar, C. G. Prato, "A generalised model for container drayage operations with heterogeneous fleet, multi-container sizes and two modes of operation", *Transportation Research Part E: Logistics and Transportation Review* 139, 101973, 2020.

[4] M. W. Savelsbergh, "The vehicle routing problem with time windows: Minimizing route duration", *ORSA journal on computing* 4(2), 146-154, 1992.

[5] R. Masson, F. Lehuédé, O. Péton, "Efficient feasibility testing for request insertion in the pickup and delivery problem with transfers", *Operations Research Letters* 41(3), 211-215, 2013.

[6] T. Vidal, T. G. Crainic, M. Gendreau, C. Prins "Timing problems and algorithms: Time decisions for sequences of activities", in *Networks*, 65(2), 102-128, 2015.

# Heuristic and exact algorithms for a vehicle routing problem with route cost equity constraints

**Maxime Agius**
IMT Atlantique, LS2N, UMR CNRS, Nantes, France
Email: maxime.agius@imt-atlantique.fr

**Nabil Absi, Dominique Feillet, Thierry Garaix**
Mines Saint-Etienne, Univ Clermont Auvergne, INP Clermont Auvergne, CNRS, UMR 6158 LIMOS, F - 42023 Saint-Etienne, France

## 1 Motivation

The vehicle routing literature is rich, and many variants are derived from real-life applications. Among the most emerging variants, equity has gained interest in the last two decades. Typically, efficient solutions usually lack equity, thus, the challenge is to propose models and algorithms capable of finding solutions that distribute workload equitably among drivers with a low impact on the routing cost.

In the literature, equity is usually measured with an inequality function minimized along with the min-cost objective [1]. Alternatively, equity can also be handled with constraints while minimizing only the routing cost. This approach has rarely been investigated in the literature despite its relevance. In this work, we adopt a constraint-based modeling approach.

We use a test-bed problem selected from healthcare logistics, the Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD) introduced in [2].

## 2 Measure of equity and solution methods

### 2.1 Equity constraints

The proposed equity constraints consist in limiting the deviation of each route cost from the average route cost. For a constant-sum equity metric (the sum of workload assigned to drivers remains constant for any feasible solution), the average is known. However, these constraints are more complex to manage with a variable-sum equity metric (the sum of workload assigned to drivers differs between solutions) as the average is unknown and depends directly on the solution. Route cost corresponds to the driving time (service times are not considered) *i.e.*, $c_{ij} = t_{ij}$ and is a variable-sum equity metric.

Let $\mathcal{M}$ denotes the set of drivers and $K$ the number of drivers. Let $\Omega$ be the set of feasible routes. Let $c_r$ be the cost of a route $r \in \Omega$ and $\theta_r$ be a variable that indicates if this route is selected in the solution. Equity constraints impose for each route, a limited deviation above the average routing cost as follows:

$$c_r\theta_r \leq \frac{\alpha}{K} \times \sum_{s\in\Omega} c_s\theta_s, \ \forall r \in \Omega \tag{1}$$

In a previous work [3], we showed the efficiency of considering similar equity constraints for a constant-sum equity metric. The contribution of this new work is to consider a variable-sum equity metric which is harder to manage as the total workload is not known in advance.

## 2.2 Algorithms

The MTMPD-RD (problem without equity constraints) is formulated as a set partitioning problem and is solved efficiently with a dedicated branch-and-price algorithm [2]. However, we show that integrating equity constraints (1) complicates the pricing problem. The argument is that the dual cost $\lambda_r$ associated with (1) cannot easily be counted when evaluating the reduced cost of route $r$. Indeed, if the route is already in the restricted master problem, $\lambda_r$ might be not null and the coefficient $c_r$ on the left-hand side of the constraint could only be considered once it is known that the route under evaluation is $r$. Before, the associated labels may dominate other promising routes and therefore prevent the column generation method to converge. Hence, we propose three new solution methods: two branch-and-price algorithms and a heuristic.

### 2.2.1 Driver-indexed branch-and-price (DI-BP)

The originality of this branch-and-price algorithm is that columns in the column generation are indexed by route and driver. We distinguish the sets of routes between drivers and denote $\Omega_{(mk)}$ the set of feasible routes of driver $m_k$. Thus, given a driver $m_k$ and a route $r \in \Omega_{(mk)}$, the decision variable $\theta_r^k$ equals 1 if the route is selected for this driver, and 0 otherwise. The equity constraints are managed in the master problem:

$$\sum_{r\in\Omega_{(m_k)}} c_r\theta_r^k \leq \frac{\alpha}{K} \times \sum_{m_l\in\mathcal{M}} \sum_{r\in\Omega_{(m_l)}} c_r\theta_r^l, \ \forall m_k \in \mathcal{M} \tag{2}$$

This requires adaptations at each level of the algorithm: at the master problem level with new constraints, while generating columns with one pricing problem solved per driver (due to different counting of the dual variables) and with new reduced cost formulas, and within the branching scheme with new flow computation. The main drawback of this model is that it contains symmetries and requires to solve a pricing problem per driver.

### 2.2.2 Node-based branch-and-price (NB-BP)

Equity concerns drivers so, intuitively equity constraints are expressed on drivers in the previous models. However, they can indifferently be expressed on nodes (customers) instead. The principle of these constraints is that the cost of any route served by a driver is limited to $\alpha\times$ the average route cost. This constraint can similarly be stated on the nodes: given a node, the cost of the route serving this node is limited to $\alpha\times$ the average route cost:

$$\sum_{r\in\Omega} a_i^r \theta_r c_r \leq \frac{\alpha}{K} \times \sum_{r\in\Omega} \theta_r c_r,\ \forall i \in \mathcal{N} \tag{3}$$

Expressing equity constraints that way allows solving a single pricing problem at each step of the column generation instead of one per driver. Only the computation of the reduced cost in the pricing problem necessitates adaptations.

### 2.2.3 Heuristic (Dicho-Heu)

The heuristic is based on a dichotomic search where at each step, a simplified version of the problem is solved. We denote this problem $Q(Z)$ where $Z$ is a fixed value. The single difference with the initial problem is that in $Q(Z)$, the cost in the right-hand term in constraints (1) is fixed to $Z$ which compensates the difficulty of the unknown average route cost. Also, in this formulation, $c^k$ denotes the cost of the route of driver $m_k$:

$$c^k \leq \frac{\alpha}{K} \times Z,\ \forall m_k \in \mathcal{M}\ (\alpha > 1) \tag{4}$$

As $Z$ is known, constraints (4) are managed in the pricing problem, and $Q(Z)$ is solved efficiently to optimality with a branch-and-price algorithm. The structure of the solution guides the search for the next step. Although heuristic in general, the algorithm proves optimal in some specific cases.

## 3 Results and discussion

Experiments are conducted on instances of [2] which defines a benchmark of realistic instances extracted from the city of Aix-en-Provence, France. Instances are divided into two sets: $S_{25}$ and $S_{50}$, each containing 30 instances of 25 and 50 customers respectively. To evaluate the impact of equity on cost, different $\alpha$ values are tested: $\alpha \in$

$\{1.1, 1.08, 1.06, 1.04, 1.02, 1.01\}$. Each run is limited to two hours. In addition, for the heuristic, the running time of each iteration is limited to 15 minutes.

Table 1 shows results on instance set $S_{50}$. Each line corresponds to the results on instances with a parameter value $\alpha$. Columns "feas", "opt" and "inf" state respectively, the number of feasible solutions found, the number of optimal solutions found, and the number of instances proved to be infeasible (out of 30 instances). Columns "cost" and "time" state respectively the average cost increase (compared to the cost without equity), and the average running time (in seconds) for instances solved to optimality only.

Table 1: Results on instances of set $S_{50}$

| $\alpha$ | DI-BP | | | | | NB-BP | | | | | Dicho-Heu | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | feas | opt | inf | cost | time | feas | opt | inf | cost | time | feas | opt | inf | cost | time |
| 1.1 | 15 | 4 | 0 | 0.08 | 2693 | 28 | 24 | 0 | 0.24 | 532 | 30 | 29 | 0 | 0.41 | 357 |
| 1.08 | 8 | 3 | 0 | 0.11 | 1591 | 26 | 20 | 0 | 0.17 | 495 | 30 | 29 | 0 | 0.54 | 272 |
| 1.06 | 2 | 1 | 0 | 0.32 | 3459 | 24 | 18 | 0 | 0.38 | 1168 | 29 | 22 | 0 | 0.56 | 306 |
| 1.04 | 0 | 0 | 0 | - | - | 13 | 8 | 0 | 0.45 | 1960 | 24 | 17 | 0 | 0.90 | 501 |
| 1.02 | 0 | 0 | 0 | - | - | 0 | 0 | 0 | - | - | 15 | 11 | 0 | 1.59 | 418 |
| 1.01 | 0 | 0 | 0 | - | - | 0 | 0 | 0 | - | - | 3 | 3 | 0 | 2.85 | 412 |
| Total | 25 | 8 | 0 | 0.12 | 2375 | 91 | 70 | 0 | 0.28 | 848 | 131 | 111 | 0 | 0.73 | 354 |

Results indicate that it is harder to solve the problem when $\alpha$ is small which makes sense as the problem is more constrained. Instances might become unfeasible and proving infeasibility is the main drawback of considering equity with constraints. Also, for the exact methods, the quality of the bound decreases for small values of $\alpha$. Dicho-Heu performs better than DI-BP and NB-BP; it finds more feasible and optimal solutions and is much faster. Note that among the exact methods, NB-BP performs better than DI-BP. Results show that managing equity this way permits to find equitable solutions with a small impact on the total routing cost. The cost increase never exceeds 3% even for small values of $\alpha$.

## References


[1] P. Matl, R. Hartl and T. Vidal, "Workload Equity in Vehicle Routing Problems: A Survey and Analysis", *Transportation Science* 52(2), 239–260, 2018.

[2] M. Agius, N. Absi, D. Feillet and T. Garaix, "A Branch-and-Price algorithm for a routing problem with inbound and outbound requests", *Computers & Operations Research*, 105896, 2022.

[3] M. Agius, N. Absi, D. Feillet and T. Garaix, "Workload equity in vehicle routing with a medium-term perspective", *Available at SSRN 4537437*, 2023

# The multi-period vehicle routing problem with consistency and arrival time spread

**Emilio Jose Alarcon Ortega**
Department business decisions and analytics,
University of Vienna, Vienna, Austria
Email: emilio.jose.alarcon.ortega@univie.ac.at

**Margaretha Gansterer**
Department of Operations, Energy and Environmental Management
University of Klagenfurt, Klagenfurt, Austria

**Karl F. Doerner**
Department business decisions and analytics
University of Vienna, Vienna, Austria

## 1 Motivation and Problem Description

During the last Covid-19 pandemic, many new logistic problems arose to mitigate the different effects derived from the emergency situation. Governments, health institutions, and companies dealt with a wide range of challenges to provide society with the necessary means to undergo shortages in basic goods, create replenishment plans during a lockdown situation, and propose effective testing and control programs. Within this framework, mass testing was proposed as an intervention strategy for pandemic control in the general population. The mass testing strategies for Covid-19 involve the collection of samples for PCR analysis. In addition, mass testing is considered to be an effective strategy to deal with such emergencies, as it potentially identifies asymptomatic cases, which can then be isolated in the early stages of infection. By this, the risk of virus transmission can be considerably reduced ([3]).

In 2021, such a mass testing program was established in the city of Vienna, Austria ("Alles Gurgelt" mass testing program). This program consists of a pulling testing strategy in which citizens can perform PCR gargle tests at home. These test kits are available free of charge in supermarkets, drugstores, gas stations, etc, where they need to be returned. From there, they are collected by a logistics provider and delivered to a central laboratory. One of the critical factors to ensure an effective application of the "Alles Gurgelt" testing program is how fast the results are available after the tests are performed. In this real world application, test results are expected to be available within 24 hours after the test was picked up. To serve this purpose, pick ups from the different locations are scheduled twice a day (once in the morning and once in the afternoon). Furthermore, in order to offer a reliable service to pick up locations, and to facilitate better coordination between the pick up vehicles and the locations, pick up times are required to be consistent over the whole planning horizon.

In the study at hand, we present the problem faced by the logistics providers that deal with the pick up routes. The logistics providers need to pick up test kits from a large number of locations that present different opening times and require consistency in the vehicle visits. We denote the problem hereby presented as the multi-period vehicle routing problem with consistency and arrival time spread constraints at the delivery location (MPVRPCAS). The problem aims at minimizing the total costs related to routing due to the "free of charge" nature of the service provided in the city of Vienna. However, at the same time, the problem aims at maximizing the time arrival spread of the vehicles to the laboratory. This second part of the objective responds to the requirement of the laboratory, which expects a continuous arrival of vehicles with similar loads to avoid unloading bottlenecks and, consequently, delays in test result deliveries. Furthermore, it is possible to reduce the pick up times at the customer locations by increasing the number of drivers on each vehicle from one to two drivers.

In Figure 1, we depict an example of the underlying decision problem, where three routes are performed by three vehicles. At the final destination (i.e. the laboratory), the vehicles arrive spread over time (10.30am, 11.00am, and 11.30am). The amount of test kits that they deliver is 8, 10, and 10 units. As mentioned above, workload balance is needed in order to avoid bottlenecks. Particularly, bottlenecks would be caused by schedules, where vehicles with low loads arrive in the morning and vehicles with high loads in the afternoon, as this might exceed laboratory capacity towards the end of the day.

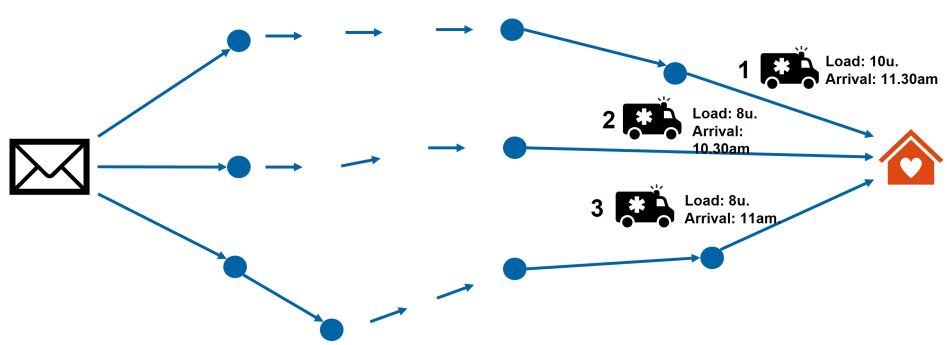



Figure 1: Example arrival time spread and load balancing

Related work in the field of consistent vehicle routing for medical applications is discussed in [1]. A recent study [2] elaborates on arrival time diversification. However, to the best of our knowledge, the problem at hand has not been tackled in the literature so far.

## 2 Solution Approach

To solve the problem presented herein, we propose a matheuristic solution approach that uses the concept of template routes. These routes are then adapted using the real amount of test kits that need to be picked up from the locations. The proposed algorithm combines a constructive heuristic with two versions of an adaptive large neighborhood search (ALNS) metaheuristic, and the solution of a mathematical subproblem.

The first step of the algorithm generates the template routes. We first calculate the average demand that each pick up location experiences over the planning horizon in each subperiod (i.e., morning and afternoon). Then, based on this calculation, we generate feasible template routes that consider the estimated demand by means of the cheapest insertion algorithm combined with a 2-opt operator and ALNS to improve the quality of the template routes. The second step of the algorithm adapts the template routes using the real demands of each period and subperiod. Once the real demands are disclosed, there are two possible situations. The first of which is that all routes remain feasible in terms of vehicle capacity. In this case, we take the template routes as fixed for the current period and its subperiod and we continue with the next subperiod. In the other case, i.e. due to unexpectedly high demands the template routes become infeasible in regards to vehicle capacity, we identify customers within infeasible routes, that would fit in other routes, and we reassign these customers accordingly. This reassigning of customers is performed such that the delivery times are affected the least possible and the consistency constraints are still respected. We proceed with relocating customers until the routes with the real demands become feasible. After applying this procedure, we yield an initial set of feasible routes, with low routing costs. However, the arrival time spread is, most likely, suboptimal. This initial set of feasible routes is used for an additional ALNS that aims at iteratively modifying the solution in regards to both routing cost reduction and optimization of arrival time spread. Furthermore, in order to obtain a better arrival time spread for each set of routes obtained during the ALNS algorithm, we solve a mathematical subproblem to optimize the delivery times for these routes.

## 3 Computational Study

We present an extensive computational study to evaluate the efficiency of the proposed solution approach as well as to gain insights on the problem characteristics. This computational study was performed using real world data provide by the logistics provider. This data includes information of pick up locations in the City of Vienna, depot and laboratory locations, and samples of demands picked up in several days. Then, based on the obtained information we created three sets of instances. The first set of instances correspond to small instances (10, 20, 30, 40 and 50 locations). We use this first set of instances to prove the efficiency of the solution approach and compare the obtained results against solutions generated by a commercial solver. The second set of instances consider medium size instances (100, 150 and 200 locations). Using the set of instances of medium size, we evaluate the impact on overall costs, when the arrival time spread is a core characteristic of the problem, as well as the different requirements for balancing the loads within the vehicles. More precisely, we evaluate from a managerial perspective, whether these additional requirements, which are claimed by the laboratory, have a considerable impact on costs. The third set considers large instances (400 and 500 locations). We use the third set instances to show the quality and the computational performance of the proposed solution approach, when instances of real world size are considered. This study is of particular insterest for future pandemic mitigation endeavours as, to the best of our knowledge, no mass testing program in cities of the size of Vienna was proposed in other countries. Finally, we also solved the original real instance set provided by the logistic provider. This set contains information for 553 locations and observed demands over several months. We can show that the pick up routes applied in real life are of comparable quality if demands are well-predictable and additional constraints by the laboratory are not considered. However, these routes are sensitive to volatile demands and strict requirements regarding balanced vehicle loads and arrival time spread. Detailed results will be presented during the conference.

## 4 Acknowledgements

This work is supported by the Austrian Science Fund (FWF): P 34151. The authors appreciate the support by the Austrian Postal service in providing data instances and valuable managerial insights.

## References

[1] P. Campelo, F. Neves-Moreira, B. Almada-Lobo, "Consistent vehicle routing problem with service level agreements: A case study in the pharmaceutical distribution sector", *Europena Journal of Operational Research* 273(1), 131-145, 2019.

[2] M. Hoogeboom, W. Dullaert, "Vehicle routing with arrival time diversification"*Europena Journal of Operational Research*, 275(1), 93-107, 2019.

[3] M. Shen, Y. Xiao, G. Zhuang, Y. Li, L. Zhang "Mass testing-an underexplored strategy for covid-19 control", in *Innovation (Camb)*, 2(2):100114, 2021.

# Anytime optimization approach for online dial-a-ride problem

**Elahe Amiri***, **Antoine Legrain**, **Issmaïl El Hallaoui**
Mathematics and Industrial Engineering Department,
Polytechnique Montréal, Montréal, Canada
Emails: *elahe.amiri@polymtl.ca

## 1 Introduction.

Commercial ride sourcing services such as Uber have decreased reliance on personal vehicles by introducing on-demand transportation solutions. While these innovations have significantly altered urban mobility, they can have negative impacts on traffic congestion. Promoting ride-sharing (RS) can alleviate this issue by utilizing vehicles more efficiently. However, its implementation faces a variety of challenges including the *high degree of dynamism* and the *large amount of trip requests* to handle. However, advanced optimization techniques are rarely used in practice limiting the potential benefits of these systems. The significance of systematic RS was first demonstrated in [1] who indicated that 98% of ride requests in New York city(NYC) could be served with 15% of the taxi fleet with a wait time of 2.8 minute. Riley et al. [2] improved over that by presenting a myopic solution for real-time RS system (M-RTRS) based on column generation (CG) that ensures service for all requests with a lower waiting time. In their follow-up work [3], they enhance efficiency by introducing a method (A-RTRS) that integrates machine learning techniques for idle vehicle relocation. From the methodical point of view, RS system is commonly modeled as the Dial-a-Ride Problem (DARP). Dispatching decisions should be made online without prior information of requests and fast enough to ensure prompt passenger response. As per [4], existing research classifies into: *sequential methods* which handle requests one at a time and covered mainly greedy methods; and *batch solutions* that involve dividing the time horizon into time epochs to batch requests, and employ optimization-based techniques. Examples include utilizing graph-based matching [1], or CG [2, 3]. Greedy solutions are faster, but they often compromise quality. In contrast, batch-based strategies may provide higher quality solutions, but are subject to a delay in responses. The challenge in reducing the epoch size lies in how quickly the underlying algorithm can compute solutions. If the time interval is short enough to consider the delay negligible, the algorithm is referred to as an *anytime* algorithm. This study proposes a novel tractable rolling-horizon optimization strategy with flexible time periods for real-time RS system, that can be adapted to a range of applications requiring dynamic optimization. We propose a CG based method and employ several acceleration techniques to cut down re-optimization time at each period, thus providing a fast, anytime procedure with strong mathematical foundation. This approach combines the strengths of optimization batch methods for high-quality solutions and smaller batch sizes, ensuring a rapid response similar to sequential methods.

## 2 Problem Statement.

To formulate the problem, we borrowed the model proposed in [2]. The model is represented as a set partitioning formulation with the set of vehicles requests denoted by $V$ and set of ride requests denoted by $P$. The model comprises a master problem (MP) to select optimal routes and pricing subproblems (SP) to generate feasible routes. Let $R_v$ be the set of possible routes for vehicle $v$. A route determines a sequence of pickups and drop-offs, and is feasible if it meets constraints on ride duration and vehicle capacity. For each route $r$, $c_r$ denotes the cumulative waiting time for served customers. The parameter $a_i^r$ signifies whether route $r$ serves request $i$. A penalty $p_i$ is considered for each request $i$ if not served in the current solution. Binary variables $y_r$ and $z_i$ indicate route selection and unserved requests, respectively. In the MP formulation (1)–(4), the objective (1) minimizes waiting time for planned service and penalties for unserved requests. Constraints (2) set $z_i$ to 1 for unserved requests, and (3) ensure each vehicle is assigned exactly to one route. To generate feasible routes, SPs are modeled as resource-constrained shortest-path problem with the aim of minimizing the reduced cost (the reader is referred to [2] for details of formulation). The reduced cost associated with route $r$ is calculated as $c_r - \sum_{i \in P} a_i^r \pi_i - \sigma_v$ where $\pi_i$ and $\sigma_v$ are dual variables associated with constraints (2) and (3) respectively.

$$Z^*_{MP} = \min \sum_{r \in R} c_r y_r + \sum_{i \in P} p_i z_i \tag{1}$$

$$\text{s.t.} \quad \left( \sum_{r \in R} y_r a_i^r \right) + z_i = 1 \qquad \forall i \in P \qquad (\pi_i) \tag{2}$$

$$\sum_{r \in R^v} y_r = 1 \qquad \forall v \in V \qquad (\sigma_v) \tag{3}$$

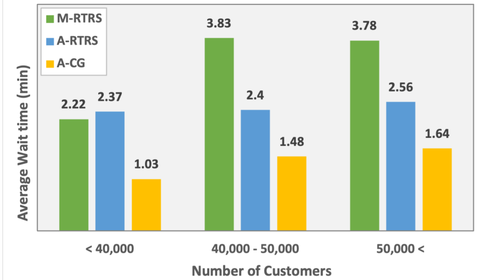
$$z_i \in \{0,1\}\ ,\ y_r \in \{0,1\} \qquad \forall i \in P\ ,\ \forall r \in R \tag{4}$$

## 3 Methodology

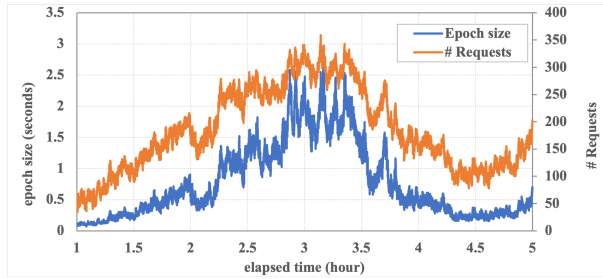
Our proposed approach is based on the rolling horizon strategy. The time is divided into small epochs, requests are batched within each epoch, and a static DARP is solved for the batch using a modified version of the CG. Unlike the traditional approach that starts by solving linear relaxation of restricted MP, our method begins by solving SPs. We initialized the duals of the new arrival requests with penalties and the rest based on the duals obtained from prior solutions. To solve SPs, we implemented a modified version of the label setting algorithm proposed in [5]. This method builds labels that are partial paths originating from the source node. It starts with an initial label at the source, and recursively extends it to successor nodes while updating the reduced cost and resource consumption to ensure feasibility. Dominated labels are also identified and removed to speed up the algorithm. We also employed three acceleration techniques to speed up the process for real-time optimization. The first, known as *truncated labeling* [5], involves retaining a restricted number of labels at each node for potential extension. Another technique is to avoid extending labels to pick-up nodes if their corresponding partial paths have already visited a drop-off node. Additionally, we have established a maximum limit on the number of pick-ups, which significantly reduces the total number of labels generated during the process. Unlike previous studies that often use fixed intervals [2, 3], this work employs a rolling horizon approach with flexible time epochs. At the start of each epoch, the state of the system is updated which involves determining the departing stop for the vehicles, batching the set of incoming and unserved requests from prior epochs, and updating the penalty of unserved requests. Penalties for unserved requests are increased exponentially based on the elapsed time from their release time to promote planning them in following intervals. Then, DARP is solved with CG for the batch of requests to determine the dispatching plan for next epoch. To adapt the method in real-time framework, pricing subproblems which are the bottleneck of the method, are solved just once per epoch and generated routes are saved in a pool. Then linear relaxation of restricted MP is solved with CG using columns in the pool, and when there is no further improve, restricted MP is solved using a MIP solver and the system proceeds to the next epoch, repeating the entire process again.

## 4 Experimental Results and Conclusion

We assessed the efficiency of our method using instances from [3], derived from real data in the NYC dataset. Additionally, we generated 24 larger instances from the same dataset, with the number of customers ranging from 100,899 to 137,178. Figure 1(a) depicts the improved average wait time in our method (A-CG) compared to prior studies using fixed epochs of 30(s) (M-RTRS [2], A-RTRS [3]). Figure 1(b) signifies the fluctuations in epoch size during runtime which depends on the time required to solve DARP. Table 1 summarizes the outcomes of solving 24 large instances, with columns for number of customers, wait time and trip delay per request, percentage of time used to solve MPs and SPs, and average epoch size. We successfully addressed all instances with the average wait time of 1.79 min, marking a 52% improvement compared to prior studies. The average epoch size does not exceed 0.52 (s), underscoring our method's ability to generate anytime solutions.




(a)




(b)

Figure 1: (a) Compare with prior studies; (b) re-optimization intervals over time

We introduced and effectively implemented an anytime re-optimization framework for a large-scale RS system. The resulted improve in average wait time led to the conclusion that a full optimization cycle may not always be beneficial due to its computational expense.

Table 1: Evaluation of the method in large-scale (4 hours simulation)

| #Customers | WT/Req. | TD/Req. | % MP | %SP | Avg. epoch |
|---|---|---|---|---|---|
| **< 120,000** | 1.26 (min) | 1.30 (min) | 41.24% | 53.10% | 0.12 (s) |
| **120,000 - 130,000** | 1.47 (min) | 1.65 (min) | 28.76% | 68.28% | 0.23 (s) |
| **130,000 <** | 1.79 (min) | 1.93 (min) | 24.59% | 73.99% | 0.58 (s) |

## References


[1] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, D. Rus, *On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment*, *Proceedings of the National Academy of Sciences*, 114(3), 462-467, 2017.

[2] C. Riley, A. Legrain, P. Van Hentenryck, *Column generation for real-time ride-sharing operations*, *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, 472-487, 2019.

[3] C. Riley, P. Van Hentenryck, E. Yuan, *Real-time dispatching of large-scale ride-sharing systems: Integrating optimization, machine learning, and model predictive control*, *arXiv preprint arXiv:2003.10942*, 2020.

[4] M. Lowalekar, P. Varakantham, P. Jaillet. *Zone Path Construction (ZAC) based approaches for effective real-time ridesharing. J. Artif. Intell. Res.*, 70:119–167, 2021

[5] V. Ghilas, J.-F. Cordeau, E. Demir, T. Van Woensel, *Branch-and-price for the pickup and delivery problem with time windows and scheduled lines*, *Transp. Sci.*, 52(5), 1191-1210, 2018.

# The Air Transport Unit Consolidation Problem

**Enrico Angelelli**

Dipartimento di Economia e Management,

University of Brescia, Brescia, Italy

Email: lorenzo.peirano@unibs.it

**Claudia Archetti**

ESSEC Business School in Paris, Department of Information Systems, Decision Sciences and Statistics, Cergy, France

Email: archetti@essec.edu

**Lorenzo Peirano**

Dipartimento di Economia e Management,

University of Brescia, Brescia, Italy

Email: lorenzo.peirano@unibs.it

## 1 Overview

Freight forwarders handle international shipments for their customers and focus their activity on offering the most complete logistic service from origin to destination.

One of the activities performed is the consolidation of loose packages into transport units (TU). For air shipments TUs are usually pallets of various dimensions, or crates. Since most of the times shipper companies are not able to autonomously consolidate the goods composing the shipment, freight forwarders spend considerable time in performing this operation. However, choosing the number and type of TUs to be used is not a simple task and it requires taking into account different aspects. Indeed, the way in which loose packages are consolidated directly affects the shipment cost.

The problem tackled in this study is the consolidation of loose packages (boxes) in TUs to be transported through air transportation services. According to the definition provided by Bortfeldt and Wäscher (2013), the problem can be classified as a Multiple Bin-Size Bin Packing Problem, since highly heterogeneous boxes will be packed in a weakly heterogeneous set of containers (TU). The objective of the problem is the one of identifying the set of TUs that minimizes an objective function composed by three terms. The first goal is the minimization of the total volume generated by the resulting set of TUs utilized. Indeed, in air transportation the unitary freight is applied to the *taxable weight*, which is linked to the volume of the shipment. Moreover, the higher is the weight density of the shipment (i.e. the nominal weight divided by the volume), the lower is the freight applied by companies, and thus minimizing the volume is a key objective.

The second objective is the optimization of the position of the center of gravity of the packages loaded into TUs. A center of gravity positioned towards the center and in a low position leads to compact, stable and safer layouts.

The third objective is the minimization of the total number of TUs used in the solution. Most of the handling costs, such as x-rays check, transshipment, and customs and security inspections, depends on the quantity of items to be shipped. Minimizing the number of TUs to be shipped leads to lower shipping costs.

The solution space is defined by a set of constraints. As it is standard for three-dimensional bin packing problems, boxes must be contained within the boundaries of the TU used, and cannot overlap, i.e. cannot occupy the same space. Limitation can be applied to the orientations of boxes. In a similar fashion, boxes can be used as base for other boxes to lean on, and in this case are treated as stackable boxes, or, on the contrary, the space above them should be left empty, i.e. the boxes are not stackable. In the study boxes can, or cannot, be stackable and can, or cannot, be oriented facing particular directions.

The core features of the study can be summarized as follows:

- Inspired by the Extreme Point (EP) technique introduced by Crainic et al. (2008), we develop a Three-Dimensional Bin Packing (BP) algorithm taking into consideration operational aspects related to the air transportation field. In particular, boxes are first sorted according to their weight, then to their width and length dimensions, and lastly by their height. Heavier, taller boxes with a wide base should be placed first. The evaluation of the best position, the EP, where boxes should be placed, as long as their orientation, is based on an objective function taking into account the three terms mentioned above. Aspects taken under consideration rely on finding the EP placed in the position ensuring the better occupation of the available space in all three dimensions, considering the possibility to have multiple boxes of similar dimensions. Moreover, it is preferred to place boxes in lower vertical position, with orientations minimizing the height, to improve horizontal and vertical stability.
- The most efficient mix of TU is explored through an iterated local search algorithm embedding the bin packing algorithm. The local search operates on two levels.

  The first level local search is based on exploring the solution space by applying a set of simple moves which evaluate if transferring boxes in different positions between the same, or a different TU, could lead to improvements. The objective is to check if any enhancements can be applied to the solution found by the BP algorithm when the set of TUs used is not modified.

The second level local search is based on identifying TUs with undesirable packing, destroy them, and explore the solution space by re-applying the BP algorithm making use of TUs of different dimensions. The objective is exploring solutions using different types of TUs.

- A procedure is developed to give an evaluation of the results obtained by the ILS algorithm. The procedure is based on identifying instances for which an optimal solution is known in terms of number and dimensions of TUs that should be used to accommodate boxes of a given total volume and weight. Then, three partitioning algorithms are used to generate boxes of uniform density. Potentially, under perfect conditions, the ILS algorithm could be capable of recreating the original TU, and the optimal solution represent a benchmark value that can be used for comparing the results obtained.

  The first partitioning scheme divide the volume of TUs according to a layer-based procedure. First, layers are identified along the height dimension, and then these layers are divided, sequentially, along the width and length dimensions. The second partitioning algorithm begins by generating a box with random dimensions in the origin of the axis representing the physical space of the TU. Once the box is created, the partitioning scheme cuts the ”tunnel” generated by projecting the selected face in the direction of the projection, generating new boxes. The procedure is iterated until the whole volume is partitioned. In the third partitioning scheme, a perfect partition of each type of TU using boxes of standard dimensions is identified. For each TU in the optimal solution, a set number of boxes compliant to the perfect partition is generated.
- To further test the effectiveness of the procedure, an agreement was made with an Italian freight forwarding company with the objective of comparing the solutions of the ILS algorithm with those referring to the company’s best practice. Results were validated by the company’s experts.

Extensive experiments are presented to compare the performance of the proposed approach in comparison with known optimal solutions, showing promising results.

When the first partitioning scheme is applied, the ILS algorithm is capable of exploiting the natural layered structure of the boxes, presenting high utilization rates of the volume of the

TUs used, and overall the center of gravity is placed in a desirable position in every instance. The second partitioning scheme results in the creation of multiple boxes of unique dimensions. The effect is that the instances generated following this approach are complex to solve, and prove to be particularly challenging. The solutions generated by the BP algorithm, however, present satisfying results in terms of volume utilization rate, the presence of empty spaces between boxes is fairly limited and the position of the center of gravity is, on average, good. The high number of boxes lead to longer solution times. The ILS is particularly effective in improving the starting solution, but in few cases it is capable of identifying the optimal solution.

Solutions of the instances generated by the third partitioning scheme are found in fast times and the TUs built are compact, balanced and with a high utilization rate of the TU volume (approximately 95% on average).

The ILS proves to be particularly effective in tackling the real world instances. The solutions found by the ILS algorithm lead to cost savings and better packing layouts, both in terms of stability and compactness, when compared with the solution adopted by the company. Moreover, solutions are found in matter of seconds, while operators often require long times to effectively identify the packing layouts and the best set of TUs to use.

## References

[1] Bortfeldt, A. and W¨ascher, G. (2013). *Constraints in container loading–a state-of-the-art review*. European Journal of Operational Research, 229(1):1–20.

[2] Crainic, T. G., Perboli, G., and Tadei, R. (2008). *Extreme point-based heuristics for three-dimensional bin packing*. Informs Journal on Computing, 20(3):368–384.

# Choice-based crowdshipping. A dynamic task display problem for next-day delivery services

**Alp Arslan**
Department of Management Science, Lancaster University,
Email: a.arslan1@lancaster.ac.uk

**Firat Kilci**
McCombs School of Business,
University of Texas Austin

**Shih-Fen Cheng, Archan Misra**
School of Computing and Information Systems,
Singapore Management University

## 1 Motivation

The growth of e-commerce has resulted in substantial demands for field operation workforce, particularly for last-mile delivery. While technologies such as drones, droids, or autonomous vehicles may well underpin the delivery infrastructure in the future, many e-tailers are presently adopting a "crowd workforce" model. In crowdshipping, the last-mile delivery tasks are delegated to a pool of willing individuals with the help of online platforms. [1] show that crowdshipping has the potential to reduce the overall delivery cost by lowering the barriers for individual "workers" to voluntarily utilize their resources (time and vehicles) to deliver such packages. Such a crowdsourced workforce also provides a more elastic labour supply that efficiently responds to demand variations, e.g., during holiday season peaks.

The role of the online platform is critical to the efficiency and viability of this crowd workforce model, as its task assignment mechanisms directly affect the participation rate of the crowd workforce. Very broadly speaking, the task assignment mechanism can either be centralized (where the platform *assigns* tasks to available individual workers) or decentralized (where individual workers *select* available tasks independently from a pool). The two strategies illustrate a broad tension between *efficiency and autonomy*.

In practice, the choice of the task assignment mechanism depends heavily on the nature of the tasks. For domains such as on-demand transportation or meal delivery, where tasks have short expiration times, most online platforms (e.g., Uber, DoorDash) centralize task assignment, pairing drivers and delivery requests quickly without elaborate drivers' consultation. In such scenarios, because assignments are driven by the worker's instantaneous location, the chance of worker rejection of centralized assignments is lower. The situation is, however, reasonably distinct for **next-day delivery tasks**, a dominant fraction of the e-commerce market, where delivery tasks are known in advance, and the task-worker allocation usually happens over a longer time window. In such a scenario, the decentralized assignment mechanism could be more appealing as it allows crowd workers to choose preferred tasks based on their anticipated itinerary.

This study focuses on a next-day last-mile delivery platform and explores how to design the decentralized task assignment mechanism to perform like a centralized mechanism. More specifically, we propose a mechanism that will grant task-selection autonomy to workers while retaining the advantage of central-planing by showing crowd workers carefully curated subsets of available tasks. Such a strategy helps with global performance since the planner can nudge crowd workers to focus on service regions that have higher chances of being cleared. This in turn enables task consolidation and reduces the cost of engaging contract drivers, which is proportional to the number of service regions with unselected tasks. To stay focused on the strategic goal of having fewer service regions with unselected tasks, we incorporate a realistic cost approximation method to estimate the cost of engaging contract drivers. An illustrative example of how this idea works is presented next.

## 2 An illustrative example

Assume that we are deciding on the set of tasks to show to the last incoming driver in order to minimize the platform's fulfilment cost for Zones 1 and 2, where there are three and one remaining tasks respectively. We also assume that this driver is choosing at most one task and probabilistically prefers Zone 1 over Zone 2. There are four possible display scenarios: $D_1$ shows tasks from both zones, $D_2$ shows tasks from Zone 1, $D_3$ shows the task from Zone 2, and $D_4$ shows no tasks. Let $P_i(D_j)$ be the probability that the driver would choose a task from Zone $i$ given a display set $D_j$. $P_0(D_j)$ is the case where the driver chooses not to serve.

The platform aims to minimize the total cost, which includes the cost of engaging contract drivers. For this example, we assume that the fixed cost of engaging a contract driver is \$100 for any zone with unselected tasks remaining, and the variable cost for serving each additional task in the same zone is \$10/task. The expected costs of contract workers for the three display options are computed below:

- $D_1$: If the driver picks a task from Zone 1, there are 2 and 1 remaining tasks in Zones 1 and 2 respectively, resulting in a cost of $(100 + 2 \cdot 10) + (100 + 10) = 230$. Similarly,

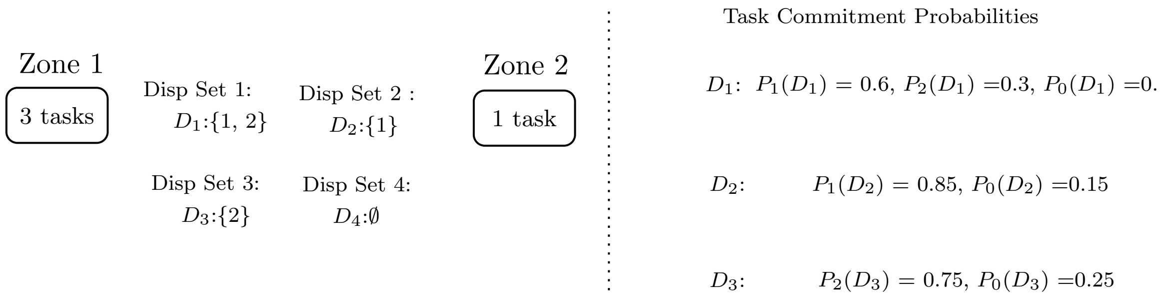



Figure 1: An example of display sets.

if the driver picks a task from Zone 2, the resulting cost will be 130. Finally, if the driver chooses not to serve, the resulting cost will be 240. The expected cost is thus: $0.6 \cdot 230 + 0.3 \cdot 130 + 0.1 \cdot 240 = 201$.

- $D_2$: The expected cost is: $0.85 \cdot 230 + 0.15 \cdot 240 = 231.5$.
- $D_3$: The expected cost is: $0.7 \cdot 130 + 0.3 \cdot 240 = 157.5$.
- $D_4$: The cost of fulfilment is 240.

In the above example, the use of display set $D_3$ (i.e., showing the crowd driver a task only from Zone 2) results in the lowest expected cost. This is even though the driver prefers Zone 1 and thus has a significantly higher probability of not choosing any task in the display set $D_3$. This is caused by the high fixed cost of engaging a contract driver. Consequently, whenever possible, the platform would desire to adopt a display set that helps to reduce the number of zones with residual tasks.

Of course, this is an overly simplified example designed only to demonstrate the benefits of display set customization. The complete model captures the complexity of having more zones, heterogeneous driver preferences, and the non-linear cost of engaging contract drivers. Moreover, we also have to consider the sequential nature of the decision-making process; i.e., drivers make their selection asynchronously and sequentially, and thus, earlier display decisions impact the pool of tasks available for selection by future drivers' preferences and the display set.

## 3 Summary of contributions

We refer to this model of 'centralized customization, autonomous selection' as *choice-based crowdshipping for next-day delivery*. Thus, this paper's central theme is to develop and quantify the significance of such display policies, which strategically incorporate both the choice behaviours of crowd drivers and the platform's cost-minimizing objective. To analyze choice-based crowdshipping in the next-day delivery service, we introduce the Dynamic Task Display Problem (DTDP). In DTDP, (a) there is a finite duration (Selection Horizon) over which crowd drivers arrive randomly and request the platform/App to display tasks from which to make a selection, (b) the platform dynamically determines the subset of displayed tasks for each individually arriving driver, and (c) the platform hires contract drivers to make the delivery of residual tasks. We make the following contributions:

- We introduce a dynamic task display problem (DTDP) in the crowdsourced next-day delivery setting that considers crowd drivers' stochastic choice behaviour, probabilistic knowledge of future crowd driver arrivals and the generalized cost components for tasks contract drivers, which enables to consider both delivery resources without prioritizing one over other one.
- Given that an exact solution to the DTDP is computationally intractable, we propose a stochastic look-ahead strategy constructed on two main pillars: (i) Value Function Approximation and (ii) Efficient Display Sets. Our solution approach enables us to solve real-life problems.
- Instances inspired by Singapore's geographical properties, we numerically show that enabling choice-based crowdshipping decreases the fulfilment cost of overnight delivery tasks up to 16.9% by balancing the workload between the crowd and contracted drivers.
- The experiments exhibit that the chosen display policy significantly influences cost savings that could be obtained from choice-based crowdshipping. The proposed customized task display policy consistently outperforms other benchmarks representing fully decentralized (display-all tasks) and centralized strategies (priority on cost-saving).
- We observe the amount of cost-savings is sensitive to the reward paid to drivers and the number of arriving crowd drivers. We also observe that the fully decentralized display policy may increase the total fulfilment cost if the reward amount of the drivers is particularly high or the crowd drivers become picky.
- Our method of customized display policy shows additional benefits when the crowd driver's task choice behaviour is less predictable. Also, the cost-saving results remain robust against varying contract expense profiles.

## References


[1] Sampaio, A., Savelsbergh, M., Veelenturf, L., & Van Woensel, T. (2019). Crowd-based city logistics. In Sustainable transportation and smart logistics (pp. 381-400). Elsevier.

# Hub Transportation Problem with Chance Constrained Due Dates

**Öykü Naz Attila**
Department of Mathematics and Industrial Engineering,
Polytechnique Montréal, CIRRELT & GERAD, Montreal, Canada
Email: oyku-naz.attila@polymtl.ca

**Antoine Legrain**
Department of Mathematics and Industrial Engineering,
Polytechnique Montréal, CIRRELT & GERAD, Montreal, Canada

**Quentin Cappart**
Department of Computer and Software Engineering
Polytechnique Montréal, CIRRELT, Montreal, Canada

## 1 Introduction

The French railway company (SNCF) provides public transportation services between train stations integrated with shuttle services to increase passenger mobility and satisfaction. This involves providing shuttles to serve passengers who require a ride either from a custom location to the train station, or from the train station to a custom location to help passengers continue or complete their itinerary. A key aspect to operate this service successfully is to ensure that shuttles arrive at the train station before the time that passengers are due to take their train. Consequently, finding an optimal routing plan for shuttles is a highly time-sensitive problem. This is crucial to consider from a mathematical modeling perspective, since inaccurate representation of travel times can easily lead to infeasible routing decisions with respect to meeting the due time that passengers are required to be at the train station. This is especially important for practical problems as the one addressed in this study, given that travel times often vary due to factors such as traffic congestion, weather disruptions and accidents.

In this study, we propose a *hub transportation problem with chance constrained due dates*. Our problem involves optimizing routing decisions for shuttles while ensuring their timely arrival at the train station under uncertain travel times. To achieve this, a chance constrained approach is utilized where a routing plan is deemed feasible if shuttles arrive at the train station before passengers' due time. To model travel time uncertainty, a discrete and finite uncertainty set is used to represent travel time scenarios. Under this setting, employing a chance constrained approach implies that the final decision is guaranteed to remain feasible in terms of respecting passengers' due time for at least a given proportion of travel time scenarios.

Our contributions in this study are threefold: Firstly, our problem tackles a novel setting where chance constraints are applied to assess the feasibility of shuttle routing decisions with respect to due times that arise from passengers' need to catch their train. Secondly, we show that the problem can be solved using column generation, where problem-specific rules on domination and feasibility checks for the pricing problem are presented to improve computational performance without compromising on chance constraint optimality. Finally, we provide insights on the computational performance of the proposed methods in terms of solution time and quality.

## 2 Solution method

Our problem ensures that shuttles return to the train station before a predefined due time to ensure that passengers catch their train on time. It typically faces a large number of decision variables and constraints due to the high number of passenger requests that arise from trains transporting a large number of passengers with each arrival and departure. Consequently, the problem suffers in terms of computational tractability. To address this challenge, a column generation procedure is utilized to solve the linear relaxation of the problem, where the pricing problem is responsible for returning columns that respect the chance constraints. Due to the high complexity of the pricing problem, our contribution is related to increasing its computational efficiency both in terms of solution time and quality.

### 2.1 Pricing problem

The pricing problem can be modeled as a resource constrained shortest path problem as it involves determining a path that does not exceed resource limits associated with passengers' due time at the train station and shuttle capacity (e.g. [1]). Problems that fall under this class can be solved using dynamic programming approaches through labelling algorithms. In the hub transportation problem with chance constraints, this involves searching a large set of partial routes dynamically, which have associated labels to track travel time, vehicle load and the due time under time deviations that occur while serving passengers. The latter is particularly of interest for our problem, as predictable deviations can be utilized to encourage shorter routes to further improve passenger satisfaction. Although performing feasibility and dominance checks on the aforementioned resources is computationally tractable in a deterministic setting, the solution procedure becomes severely complex and intractable under a stochastic setting where chance constraints are taken into account. This is due to the fact that travel times implied by each scenario is required to be tracked to assess solution feasibility with respect to whether a given route respects passengers' due time at the train station. This naturally increases the size of the resources in labels severely. To tackle this issue, we follow a procedure where the number of dominance checks performed between labels, as well as the size of feasibility checks are reduced. Here, feasibility checks refer to checking whether the number of scenarios that respect chance constraints remains above the desired level of feasibility across the scenarios given in the uncertainty set. Therefore, reducing the size of feasibility checks imply enforcing chance constraint feasibility on a subset of scenarios in the uncertainty set. Likewise, reducing dominance checks implies performing it only on a subset of resources instead of all resources. Although reducing the size of feasibility checks is desirable in terms of improving solution time, it may lead to cases where chance constraints are violated. To maintain the feasibility of chance constraints, our approach filters the resulting routes where they undergo a final evaluation check that ensures that the chance constraint holds for the solution based on the non-reduced uncertainty set. Since this procedure is performed post-optimization, it leads to a considerable improvement in terms of the computational effort needed to solve the pricing problem without compromising on the desired level of feasibility implied by the chance constraints.

Another important challenge associated with the pricing problem is to increase the quality of the final solutions obtained, especially since the column generation procedure is only applied at the root node of the problem. To do so, we integrate two additional methods into our framework: a diving heuristic and an NG-path approach, both of which are used to generate additional routes that are of high quality.

## 3 Computational results

Computational results have been obtained by solving an instance class that has been generated based on a train hub location in France containing 100 passenger requests. The column generation algorithm was executed in Python, where the master problem was solved using the library `Python-MIP` and the pricing problem was solved with `cspy`.

As Figure 1 suggests, with increasing size of scenarios considered in feasibility checks (denoted as $F$), the overall solution time required for the pricing problem increases considerably regardless of the rate of infeasibility (denoted as $\beta$) allowed. Therefore, it is necessary to solve the pricing problem under reduced sizes of the feasibility set to maintain tractability. Our approach is motivated by this observation, as it allows us to solve the problem under reduced feasibility sets without compromising on the feasibility of chance constraints due to final evaluation checks.

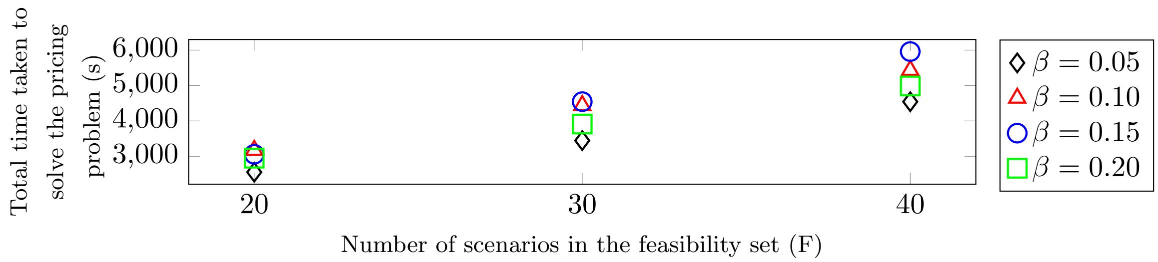



Figure 1: Time performance of the pricing problem under different feasibility set sizes and rates of allowed infeasibility ($\beta$).

To assess the quality of solutions, we analyze the percentage gap between the optimal objective function value of the final MIP solution and its value when its linear relaxation is solved. The diving heuristic when used alone with $F = 20$, led to a gap of $[11.06\% - 16.22\%]$ under different configurations of the allowed rate of infeasibility; as opposed to $[15.96\% - 20.69\%]$, which was observed when diving and NG-path were not utilized. Using NG-path along with the diving heuristic significantly improves this performance, where the gap was observed as $[3.33\% - 3.64\%]$ among different NG-set size configurations.

Figure 2 shows the minimum level of feasibility observed in an optimal route under different allowed levels of infeasibility ($\beta$). The feasibility rates have been calculated under an evaluation set size of 200, where chance constraint feasibility is assured, and a test set of size 1000. As the results suggest, the minimum level of feasibility observed among a chosen route stays well above the desired level of feasibility for both the evaluation set and the test set under the proposed approach.

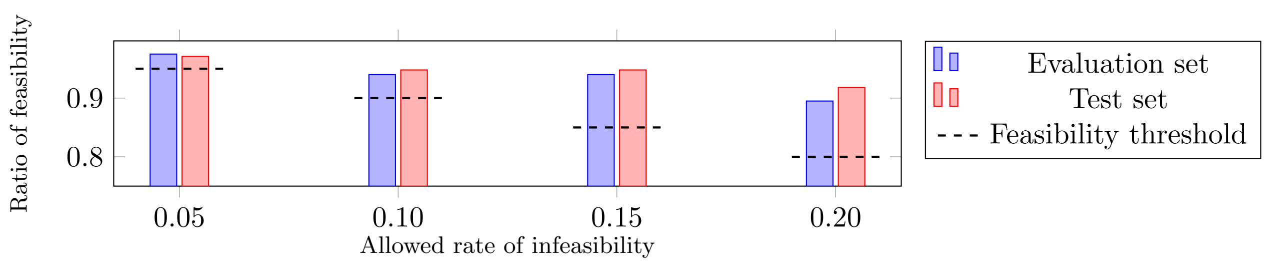



Figure 2: Minimum ratio of feasibility among chosen routes across different allowed rates of infeasibility (diving heuristic and NG-set enabled, where NG set size is set to 1).

## References


[1] Agius, M., Absi, N., Feillet, D., and Garaix, T. (2022). A branch-and-price algorithm for a routing problem with inbound and outbound requests. *Computers & Operations Research*, 146:105896.

# Breaking down The Amazon Routing Challenge: A heuristic approach for the clustered TSP

**Andreas Barstad** **Peter Schütz** **Mohamed Kais Msakni**
Department of Industrial Economics and Technology Management,
Norwegian University of Science and Technology, Trondheim, Norway
Email: kais.msakni@ntnu.no

**Lars Magnus Hvattum**
Faculty of Logistics,
Molde University College

## 1 Introduction

Traditionally, last-mile delivery problems are modeled as traveling salesman problems (TSP) with different variants. This area of study is well established, and advancements have led to high quality solutions in efficient computational time. The TSP is typically formulated to minimize costs, focusing on travel distance or time. However, in practical applications, the efficiency of a route is not solely determined by its theoretical cost. Other factors can also be considered, such as how a driver behaves when faced with real-life conditions for a route. For instance, Li and Phillips [1] observed that drivers of a soft drinks company follow the suggested optimized routes in only one out of four deliveries. Drivers often deviate from proposed routes because of their knowledge of local traffic, parking conditions, or personal preferences. This form of knowledge, referred to as tacit knowledge, is generally acquired through experience and practice.

In 2021, Amazon collaborated with MIT's Center for Transportation & Logistics to launch "The Amazon Routing Challenge," a research competition [2]. This initiative was motivated by the tacit knowledge acquired by Amazon's drivers, who frequently deviated from the planned routes despite the company's optimization software that was designed to optimize routes considering various factors such as safety and efficiency. The Amazon challenge's objective was to develop innovative methods that leverage tacit knowledge to produce delivery routes that surpass traditional optimization techniques in both quality and cost.

Several studies have been conducted to address this challenge. The proposed solution approaches range from adapted classical optimization methods, e.g., solving a TSP problem with a transformed distance matrix [3], to machine learning models, such as a reinforcement learning algorithm aiming to learn a reward function from expert behavior [4]. In this paper, we present an exploratory data analysis of the Challenge dataset and propose a two-stage solution approach that combines a classifier with a local search procedure. Despite its simplicity, the computational study shows that our proposed method performs remarkably well, requiring an extremely short CPU time.

## 2 Exploratory data analysis

The Amazon Routing Challenge provided an initial dataset of 6,112 historical routes for the purpose of training and preparation, followed by a second set of 3,012 routes for evaluating the proposed methods [5]. Our analysis, conducted post-competition, focused on the first dataset. Essentially, a route is a sequence of stops, with each stop characterized by an identifier, location, type, Zone ID, and package(s) to deliver. Each route is executed by a single driver in one of five U.S. cities. The number of routes varies by city, ranging from 214 in Austin to 2,888 in Los Angeles. For the latter, there are six different depots serving as starting points for routes. Most importantly, we observed that throughout the data set, stops are generally exclusive to a single route, posing challenges in directly learning from this dataset.

A visualization of the routes shows that the sequence of visited stops follows a particular pattern, for which all the stops within the same zone are visited before proceeding to the next. While there are cases of revisiting zones, these are relatively infrequent. Further analysis reveals a hierarchical structure within the 'Zone ID', comprising three levels: (1) a 'super-super-cluster', the largest geographical entity, usually encompassing one or two such clusters per route, (2) a 'super-cluster' within the super-super-cluster, with 1 to 8 super-clusters per super-super-cluster, and (3) a 'cluster', representing a smaller area with a maximum of 3 clusters in each 'super-cluster'. As an example, the stop with 'Zone ID' '`B-4.1G`' has '`B4`' as the super-super-cluster, '`G`' as the super-cluster, and '`1`' as the cluster.

Moreover, the data analysis reveals a consistent pattern. Upon entering a super-super-cluster, a driver typically visits all super-clusters within it before moving to the next super-super-cluster. This sequence often follows an increasing or decreasing alphanumeric order. Similarly, within a super-cluster, clusters are visited in numerical order before moving to the next super-cluster. Since there are two possible directions at each of the three levels, there are eight possible ways to order these 'ZoneIDs'. It is important to note that while this pattern is applicable for most routes, there are exceptions, particularly in densely populated super-super-clusters, where the pattern does not apply systematically.

## 3 Problem definition and solution method

From the data analysis, the challenge presents a specific variant of the TSP, for which the task is to deliver to all stops in a single tour, departing and arriving at the same depot. The stops are partitioned into disjoint clusters, and the requirement is that when a driver visits a stop belonging to a particular cluster, they must visit all other stops in that cluster before going to a new cluster. Besides, the clusters must be visited in a specific order, following the hierarchical structure of 'super-clusters' and 'super-super-clusters', as well as the alphanumeric order of these clusters, as described in the previous section. The distance between any two stops is asymmetric, and the objective is to minimize the total distance for completing a tour. This problem is a special case of the Clustered Traveling Salesman Problem (CTSP) and is NP-hard.

To tackle typical challenge instances ranging from 100 to 200 stops, we develop a two-stage heuristic for solving the CTSP in a very short computational time. The first stage involves a classifier that determines the sequence of clusters, primarily by identifying the ascending or descending order of visits across the three hierarchical levels. The second stage determines the sequence of stops within each cluster before moving to the next. This is achieved by simply selecting the nearest stop from the current one. Subsequently, an improvement step is applied that consists of applying the 3-opt heuristic.

## 4 Results

The data analysis and the solution method were conducted in Python 3.10, except for the 3-opt heuristic, which was coded in C++. Computational experiments were performed on a standard computer. In the competition, the degree of similarity between the actual driven route and the solution of our proposed method is evaluated using a scoring metric. This score comprises two components: Sequence Deviation, which assesses the disparity between the proposed and actual sequences (ranging from 0 to 1), and Edit Distance With Real Penalty, which measures the number of operations needed to transform the proposed route into the actual driven one. A combination of these two metrics yields a lower score for higher similarity between the routes.

Our solution method was initially applied to the first dataset for training and validation, followed by its application to the test set, consisting of 3,012 instances. On average, the solution method generates a route in 0.41 seconds, and the obtained scoring metric is 0.03105. Figure 1 illustrates the distribution of scores on the test data. The best and worst scores are 0.00073 and 0.26529, respectively. The distribution is left-skewed, indicating that most of the generated solutions have a very low score. Additionally, Figure 2 shows that our solution method ranked second among all submissions to the challenge. This significant achievement is attained despite the relative simplicity of our method compared

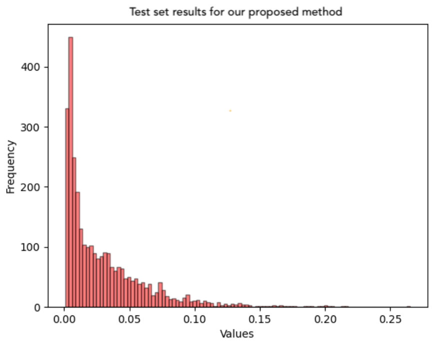



Figure 1: Histogram showing the distribution of scores obtained on test data using the proposed solution method.

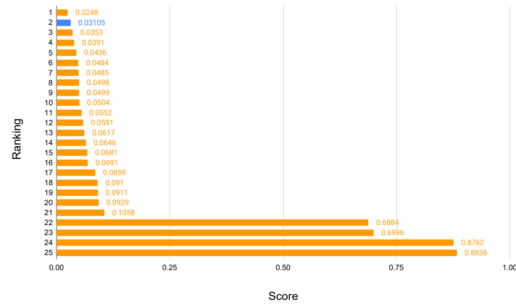



Figure 2: Ranking of participating works in The Amazon Routing Challenge, highlighting our method in blue.

to the submitted works.

Analysis of the results reveals that higher scores are typically associated with instances where clusters are visited multiple times, a scenario challenging to predict due to the lack of a discernible pattern. Furthermore, the results suggest that the tacit knowledge within the dataset is more closely associated with the predefined zones assigned to stops.

## References


[1] J. Olsson, D. Hellström, H. Pålsson, "Framework of last mile logistics research: A systematic review of the literature", *Sustainability* 11(24), 7131, 2019.

[2] D. Merchán, J. Arora, J. Pachon, K. Konduri, M. Winkenbach, S. Parks, J. Noszek, "2021 Amazon last mile routing research challenge: Data set", *Transportation Science*, 2022.

[3] O. Arslan, R. Abay, "Data-Driven Vehicle Routing in Last-Mile Delivery", in *Amazon Last-Mile Routing Research Challenge Supported by the MIT Center for Transportation & Logistics*, M. Winkenbach, S. Parks, J. Noszek (eds), DSpace@MIT, 2021.

[4] A.R. Pitombeira-Neto, "Route Sequence Prediction Through Inverse Reinforcement Learning", in *Amazon Last-Mile Routing Research Challenge Supported by the MIT Center for Transportation & Logistics*, M. Winkenbach, S. Parks, J. Noszek (eds), DSpace@MIT, 2021.

[5] D. Merchán, J. Arora, J. Pachon, K. Konduri, M. Winkenbach, S. Parks, J. Noszek, "2021 Amazon last mile routing research challenge: Data set", *Transportation Science*, 2022.

# Selecting Delivery Patterns in a Two-Echelon Distribution Problem with Load Balancing

**Lucas Baussay**
IMT Atlantique, LS2N, CRC Services, Nantes, France,
Email: l.baussay@crc-services.com

**Fabien Lehuédé, Audrey Cerqueus**
IMT Atlantique, LS2N, Nantes, France

**Juliette Medina**
CRC Services, Nantes, France

**Mike Hewitt**
Quinlan School of Business, Chicago, United States

In the retail industry, suppliers often deliver goods in small quantities (a few pallets), several times a week, to multiple retailer warehouses and stores. To reduce costs, suppliers rely on third party carriers by making use of consolidation on regional hubs, collaborating to consolidate their final delivery with other suppliers. This transport system can be described as a two-echelon distribution network, where the first echelon represents the transportation from suppliers to hubs, and second echelon hubs to customers. In this context, we consider a tactical problem that involves selecting delivery days for each supplier-customer pair in the network, which are consistent over several weeks, considering a varying demand. In addition, each actor in the network has load balancing consideration over the maximum quantity of product shipped, transferred or received each day of the week. We seek to minimize transportation costs, considering FTL transportation at the first echelon (between suppliers and cross-docks) and the cost of delivery routes that can change every day between the cross-docks and customers.

Different aspects of this problem have been tackled separately in the literature. First, the optimization of transportation in two-echelon networks has received a lot of attention from the vehicle routing community [1]. The survey of [2] mentions contributions integrating cyclic schedules and regular visit interval in inventory routing problems, which relates to the consistency of deliveries in our problem. The contributions in the literature that are the nearest to our work are the periodic inventory routing problem of [3], in which a weekly delivery schedule is determined for the delivery of stores from a central distribution center, and, in a similar context, the assignment of repetitive delivery patterns to store deliveries [4]. To our knowledge, no previous work studied the choice of delivery patterns for several commodities in a two-echelon distribution networks with load balancing considerations.

In this paper, we propose a matheuristic to solve this problem and we evaluate it on a real case study taken from a do-it-yourself retailer collaborative network in France.

# 1 Problem description

The considered network has three types of facilities: a set of customers $\mathcal{N}$, a set of suppliers $\mathcal{S}$, and a set of hubs $\mathcal{H}$. Regarding transportation in this network, each customer is delivered from a single pre-assigned hub. On a given day, products are transported from suppliers to hubs using point-to-point transportation moves. On the next day, short distribution routes (two or three customers) are designed to deliver the customers with all products from the different suppliers. Each customer is delivered at most once per day. Each route, either supply or distribution route, has a cost and a capacity. We assume that the set of feasible distribution routes $\mathcal{R}$ can be enumerated a priori because of the low number of customers per route. In this way, the cost of all routes can be pre-calculated, taking into account, the route distance, the number of stops, and the vehicle specific costs. This two-echelon distribution network is illustrated in Figure 1.

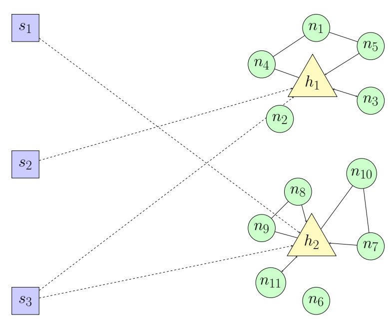



Figure 1: Two-Echelon distribution network

The set of commodities $\mathcal{K} = \{(s, n) \;\; \forall s \in \mathcal{S} \; \forall n \in \mathcal{N}\}$ is defined as all pairs between a supplier and a customer. We consider a time horizon $\mathcal{T}$, representing a repeated number of time periods. As an example, $\mathcal{T}$ can consist of four weeks, each including five delivery days. We consider a historical data set, specifying a demand quantity for each commodity $k \in \mathcal{K}$ for each day of $\mathcal{T}$. Each commodity is also associated with a frequency and a corresponding set of feasible delivery patterns. A pattern is defined as a set of delivery days. For example, a commodity $k \in \mathcal{K}$ with frequency two is associated with a set $\mathcal{P}_k = \{\{Mon, Wed\}, \{Mon, Thu\}, \{Tue, Thu\}, ...\}$ of feasible patterns. An important requirement for the network organization is that a given commodity $k \in \mathcal{K}$ is always delivered according to the same delivery pattern over the entire time horizon $\mathcal{T}$. The delivered quantity is considered to satisfy the demand until the next delivery, so for each pattern and each commodity $k$, we can calculate the quantity to be transported if the pattern is selected.

An additional requirement from the hubs operators is that the transferred quantity at each hub needs to be balanced over the delivery days of each week. This is expressed as a constraint imposing that a difference between the quantity received each delivery day and the average quantity over a week must be less than a given threshold. Also to balance their workload over week-days, we define a maximum loading or unloading product capacity for each supplier or customer.

We define the Periodic Two-Echelon Capacitated Distribution Problem, P-2E-CDP, which involves determining the delivery pattern for each commodity, the supply and distribution routes for each time period of the horizon $\mathcal{T}$, such that the commodities' demands and balancing constraints are satisfied, while minimizing the transportation costs.

## 2 Solution approach

To solve this problem, we propose an integer programming formulation (denoted "the model" in the following), a two-step framework, and several first step strategies for this framework. This framework consists in decomposing the model into two sub-problems which are solved sequentially. Its principle is the following:

- **Pattern Selection:** First, a relaxation of the model is solved to select one pattern for each commodity, respecting the load balancing constraints. This determines the quantities to be transported to each hub for each period, from which we can determine the minimum cost transport on the first echelon.
- **Second Echelon Routing:** Second, for each period in $\mathcal{T}$ and each period, we determine the delivery routes by solving a Vehicle Routing Problem (VRP), which can be expressed as a Set Partitioning Problem (SPP) since we consider an enumerated set of feasible routes.

We devise four strategies for the first step:(i) in the *Relax distribution capacity* strategy, we relax the distribution route capacities; (ii) in *Relax distribution*, we consider a linear relaxation of the route distribution variables; (iii) in *Relax supply*, the variables representing the number of trucks between the suppliers and the hubs are supposed to be real instead of integer; (iv) in *Minimize supply cost*, we simplify the objective function to integrate only first echelon costs and delete each variable and constraint on distribution routes.

## 3 Experiments

We compare the solving of the model and the four strategies implemented in our framework on a set of instances based on historical data from a retail distribution network. In addition to the full-size instance, we generated five groups of instances of different sizes, each containing five instances. Each experiment was carried out with a two hours time limit using Gurobi. Figure 2 shows the solution value, the transport cost, of each method, for each group of instances. For each group, we indicate the number of suppliers ($| \mathcal{S} |$), hubs ($| \mathcal{H} |$), customers ($| \mathcal{N} |$) and weeks ($| \mathcal{W} |$).

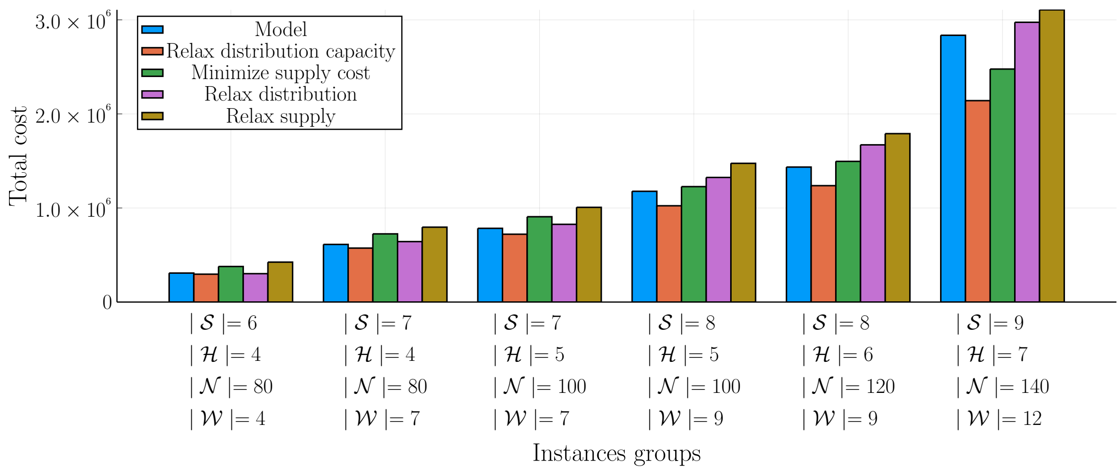



Figure 2: Average transportation cost for each method and each group of instances

As a result the framework with the *Relax distribution capacity* strategy for pattern selection performs as well, if no better, than the model and other strategies for each instance group. We also note that the heuristic finds optimal solutions on smaller instances that can be solved to optimality by the solver.

In our presentation, we will present how the proposed framework can be used to draw managerial insights into this tactical planning problem, comparing different delivery frequencies, load balancing requirements and their impact on transportation cost. We will analyze different trade-offs between cost and quality of service or load balancing considerations and show how these impact on the chosen delivery patterns for each commodity.

## References


[1] N. Sluijk, A. Florio, J. Kinable, N. Dellaert and T. Van Woensel, "Two-echelon vehicle routing problems: A literature review", *EJOR* 304(3), 865-886, 2023.

[2] A. A Kovacs., B. L. Golden, R. F. Hartl, and S. N. Parragh, "Vehicle routing problems in which consistency considerations are important: A survey", *Networks*, 64(3), 192-213, 2014.

[3] V. Gaur and M. Fisher, "A Periodic Inventory Routing Problem at a Supermarket Chain", *Operations Research* 52(6), 813–822, Dec. 2004.

[4] A. Holzapfel, A. Hübner, H. Khun and M. Sternbeck, "Delivery pattern and transportation planning in grocery retailing", *EJOR* 252(1), 54–68, 2016.

# Price-and-Branch for a Real-Life Multidepot Pickup and Delivery Problem with Scheduled Lines

**Casper H. Bazelmans, Albert H. Schrotenboer**
School of Industrial Engineering,
Eindhoven University of Technology, Eindhoven, The Netherlands
c.h.bazelmans@tue.nl, a.h.schrotenboer@tue.nl

November 20, 2023

## 1 Introduction

Optimizing the pickup-and-delivery of less-than-truckload (LTL) shipments in a local transportation network is a canonical activity of logistics service providers. Such local transportation networks are typically part of a larger international network, see Figure 1. Here, vehicles make local trips for the collection and distribution of shipments at customers, and scheduled lines offer consolidation opportunities to save costs. Whereas the local trips are optimized on a daily basis, the planning of the scheduled lines cannot be changed on a daily basis as they are part of the larger international network. However, the local trips should be optimized jointly with the assignment of shipments to internal scheduled lines; to maximize consolidation and thus reduce cost.

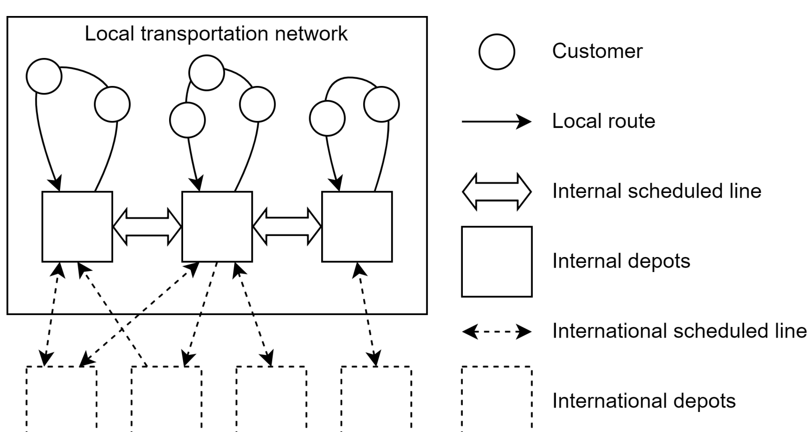



Figure 1: An example of a local transportation network.

In this paper, we introduce the multidepot pickup-and-delivery problem with scheduled lines (MDPDPSL) based on the operations taking place *within* the local transportation network. It considers; i) a set of pickup shipments located at customers that need to be collected and transported to their associated internal depots, ii) a set of delivery shipments arriving at internal depots that need to be delivered to customers, and iii) the use of internal scheduled lines to transport shipments between internal depots. The MDPDPSL aims to minimize the total routing costs while considering time synchronization between scheduled lines and local trips, loading constraints associated with pickup and delivery requests, and customer time windows.

The contributions of this research are fourfold. First, we introduce a novel column generation approach where columns simultaneously define local trips and customer assignment to scheduled lines, whereas the extant literature synchronizes this in the master problem (see, e.g., [1]). We specifically consider multiple vehicle trips to fully utilize scheduled lines, and so-called mixed-loading constraints that impose that deliveries can only be collected if the truck has ample space (e.g., 30% of the truck capacity) to reduce handling operations in practice. Second, we consider a discrete-time horizon inspired by the observation that in practice trips only depart at a limited number of times to ease the synchronization complexity. Third, to solve truly large-scale instances (>1000 customers), we introduce a fast heuristic pricing strategy based on searching the neighborhood of relatively *low* reduced cost columns, whereas traditionally this is done on a full zero-reduced cost solution. Fourth, using real-life data from DB Schenker, we solve instances up to 3 depots, 33 scheduled lines, and 2238 customers - which will be publicly available to further stimulate research in this domain. In the remainder of this abstract, we present our model, the essential details of our column generation procedure, and some exemplary results.

## 2 Model

The MDPDPSL is defined on a graph $G = (N, A)$ where $N$ is the set of nodes and $A \subseteq \{(i, j) \in N \times N : i \neq j\}$ is the set of arcs. We consider a discrete-time horizon $T$. The set of nodes parcel comprises the depot nodes $D$ and the customer nodes $C$, so $N = D \cup C$. Each customer $j \in C$ has an associated weight $q_j > 0$, and an associate depot $d^j \in D$. In case customer $j$ is a delivery, the depot represents the origin of the shipment and its earliest pickup time at the depot equals $g^j$ as a result of the incoming international scheduled lines. In case customer $j$ is a pickup, the depot represents the destination of the shipment and there is a deadline $g^j$ at which the shipment should be at $d^j$ because it is scheduled for further transport in the international network.

Let $K$ represent the set of vehicle types. Each vehicle type has capacity $Q_k$. Each depot $d \in D$ has $m_{ik}$ vehicles of type $K \in K$. Let $S$ denote the set of internal scheduled lines. Each scheduled line $s \in S$ has a given departure time $a^s$ at $e^s \in D$ and arrives at time $b^s$ at destination $f^s \in D$.

Let $\Omega_{ik}$ be the set of feasible trips originating from depot $i \in D$ using vehicle type

$k \in K$, and let $\Omega := \bigcup_{i \in D, k \in K} \Omega_{ik}$. Each trip $r \in \Omega$ incurs a cost $c_r$, representing its travel cost. For each trip $r \in \Omega_{ik}$, $\theta_j^r$ equals 1 if customer $j$ is visited on trip $r \in \Omega$ and equals 0 otherwise. The parameter $\delta_{ikt}^r$ equals 1 if vehicle type $k$ from depot $i$ is utilized at time $t \in T$ by trip $r \in R$. This allows us to model multiple trips by the same vehicle. Similarly, $\gamma_{js}^r$ is 1 if customer $j \in C$ is assigned to internal scheduled line $s \in S$ for trip $r \in \Omega$ and 0 otherwise. The binary decision variable $x_r$ is 1 if the trip $r \in \Omega$ is selected and 0 otherwise. The master problem (MP) is presented below:

$$
\begin{aligned}
\text{(MP)} \quad \min_{x^r \in \{0,1\}} \quad & \sum_{i \in D} \sum_{k \in K} \sum_{r \in \Omega_{ik}} c_r x_r & & & (1) \\
\text{subject to} \quad & \sum_{i \in D} \sum_{k \in K} \sum_{r \in \Omega_{ik}} \theta_j^r x_r \geq 1 & \forall j \in C \quad & (\pi_j) & (2) \\
& \sum_{r \in \Omega_{ik}} \delta_{ikt}^r x_r \leq m_{ik} & \forall i \in D, \forall k \in K, t \in T \quad & (\eta_{ikt}) & (3) \\
& \sum_{i \in D} \sum_{k \in K} \sum_{r \in \Omega_{ik}} x_r \sum_{j \in C} \gamma_{js}^r q_j \leq w_s & \forall s \in S \quad & (\phi_s) & (4)
\end{aligned}
$$

The Objective (1) minimizes the total cost of the selected trips. Constraints (2) assure that every customer is visited, constraints (3) restrict the vehicle fleet over time at each depot, and constraints (4) enforce the capacity of the internal scheduled lines.

## 3 Price-and-Branch and Results

As $\Omega$ cannot be enumerated, we employ column generation to solve the linear relaxation of (MP). Let $\pi_j \geq 0$, $\eta_{ikt} \leq 0$, and $\phi_s \leq 0$ be the dual variables associated with constraints (2), (3), and (4), respectively. The pricing problem $\text{PP}_{ik}$ (for each depot $i$, vehicle type $k$) is an Elementary Shortest Path Problem with Resource Constraints and Time Synchronization (ESPPRC-TS). That is, we need to solve:

$$
\arg\min_{r \in \Omega} c_r - \sum_{j \in C} \theta_j^r \pi_j - \sum_{i \in D} \sum_{k \in K} \sum_{t \in T} \delta_{ikt}^r \eta_{ikt} - \sum_{s \in S} \sum_{j \in C} \gamma_{js}^r q_j \phi_s
$$

We develop a labeling algorithm to solve the pricing problem, in which three main constraints are considered. First, we need to respect customer time windows. Second, trips can only visit a pickup customer if at least $\alpha$% of the truck capacity is empty, where $\alpha$ is an input parameter of our problem. And third, requests within trips need to be assigned to a scheduled line.

The difficulty of our labeling algorithm comes from including the dual variables $\eta_{ikt}$ and $\phi_s$. Both depend on the start and end time of the trip, which dynamically changes while extending partial paths in the labeling algorithm. These duals cannot be simply converted to arc costs in the pricing problem graph. Therefore, we keep track of a so-called dynamic time window and the total partial path duration. The dynamic time window consists of the earliest and latest possible departure time at the last customer of the partial trip.

Then, we can find the optimal trip starting time and associated assignment of requests to scheduled lines (as they limit the start and end time of a partial path) by solving the *request-on-scheduled-line assignment problem* for each partial trip. Based on the dynamic time window and the trip duration we can derive a time-window on the trip starting time We solve this assignment problem by enumerating all starting times that could lead to an optimal solution. Note that the construction of $T$ is essential, as it should comprise at least the moments that linehauls arrive and depart. Any extra time point can be considered as possible starting points for multiple trips. These extreme trip start times consist of time points $t \in T$ and scheduled line arrival times within the trip start time window, as well as the start and end of the trip start time window. We also provide a heuristic pricing strategy that searches the neighborhood of relatively *low* reduced cost trips. The heuristic method destroys part of the trip, then repairs it by adding multiple customers (not included in the trip), and finally performs a small local search over the trip.

We compare our exact approach versus a compact formulation in Gurobi (called MIP) and our heuristic pricing approach. Table 1 presents the results for 3 instances with 15, 30, and 60 customers. Upper bounds are obtained by solving the Reduced Master Problem (RMP) after generating columns with Gurobi. The time limit is two hours.

| | MIP | | | Exact pricing | | | Heuristic pricing | | |
|---|---|---|---|---|---|---|---|---|---|
| Instance | LB | UB | Time (s) | LB | UB | Time (s) | LRMP* | UB | Time (s) |
| T_2_6_15 | 429.24 | 470.30 | 7200 | 433.89 | 447.58 | 1 | 433.89 | 447.58 | 1 |
| T_2_6_30 | Out of memory | | | 659.33 | 668.87 | 220 | 665.16 | 679.55 | 23 |
| T_2_6_60 | Out of memory | | | - | 1527.80 | 7200 | 1180.51 | 1251.47 | 53 |

Table 1: Computational results. * linear relaxation of the master problem with only heuristic pricing.

We also use our heuristic pricing strategy for a real-world instance of DB Schenker comprising 2238 customers. After running for 23h, 75000 columns with the lowest reduced cost are included in the original master problem, and a solution is obtained that shows a gap of around 10% between the LRMP and a valid upper bound.

We are currently finalizing our heuristic search procedure and experiment with setting $T$ to obtain a good balance for having multiple trips and computational efficiency for our real-world data. Furthermore, we will embed our column generation procedure in a branch-and-price scheme. All these results will be available at the time of the conference.

## References

[1] V. Ghilas, J.-F. Cordeau, E. Demir, T. Van Woensel, "Branch-and-Price for the Pickup and Delivery Problem with Time Windows and Scheduled Lines", *Transportation Science* 52(5), 1191-1210, 2018.

# Stochastic Scheduled Service Network Design Problem with Flexible Schedules: mathematical formulations and exact approaches

**Simon Belieres**
TBS Business School, Toulouse, France

**Mike Hewitt**
Quinlan School of Business, Loyola University Chicago, Chicago, USA

## 1 Introduction

We focus on a two-stage stochastic Scheduled Service Network Design (SSND) problem with uncertain demand. The related literature often assumes that the decision-maker determines the location and timing of services before knowing shipment volumes, leaving shipment routing and outsourcing decisions as the only recourse options. Such assumptions are appropriate for carriers that must commit to a schedule to external providers, e.g., scheduled ships in maritime transportation. Still, they may be too restrictive for transportation modes such as trucking, which enables scheduling services once volumes are revealed, allowing for greater consolidation and lower costs. Belieres and Hewitt [1] introduced the Stochastic SSND Problem with Flexible Schedules (SSSNDFS), a variant that reflects the *scheduling flexibility* offered by trucking transportation. Through extensive computational experiments on small-sized instances, they illustrate that scheduling flexibility can often generate savings greater than 3%. However, the authors do not propose an algorithm for solving the SSSNDFS.

So-called "deterministic equivalent" formulations of two-stage stochastic programs repeat second-stage decision variables and constraints for each considered scenario, yielding large instances of models that are intractable for general-purpose mixed-integer programming solvers. Thus, most algorithmic strategies for solving these problems rely on decomposition methods like Benders decomposition. However, the guaranteed convergence of a Benders-based method requires that second-stage subproblems be linear programs. At the same time, the recourse actions modeled by the SSSNDFS are naturally formulated with binary and integer variables. Our main contribution is to propose - to the best of our knowledge - one of the only known exact algorithms for a variant of the stochastic SSND problem with integer recourse. Specifically, we adapt the Unified Branch and Benders Cut

(UB&BC) framework proposed by Mahéo et al. [3] and illustrate multiple techniques for enhancing the performance of that algorithm. Through an extensive computational study, we demonstrate both the superior performance of that algorithm to classical benchmarks and the necessity of those acceleration techniques for that level of performance.

## 2 Mathematical formulations

Let $\mathcal{N}, \mathcal{A}$, and $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ denote the set of nodes, directed arcs, and the directed network, respectively. Associated with each arc $(i, j) \in \mathcal{A}$ is a per-service cost $f_{ij}$, a capacity $u_{ij}$, a per-unit of shipment measurement cost, $c_{ij}$, and a travel time, $\tau_{ij}$. The multiplier $\xi > 1$ defines the premium paid when executing a capacity from $i$ to $j$ greater than the initial commitment. Let $\mathcal{K}$ denote the set of commodities. Each commodity $k$ has origin and destination nodes $o_k$ and $d_k$, an available time at $o_k, e_k$, and a due time at $d_k, l_k$. We use a path-based formulation and let $\mathcal{P}_k$ denote the set of candidate paths for commodity $k$. The per-unit shipment measurement cost of a path $p$ is referred to as $c_p$.

Two sets of first-stage decisions are considered. The first, $y_{ij} \in \mathbb{N}, (i, j) \in \mathcal{A}$, denote the number of vehicles that travel from terminal $i$ to terminal $j$ and reflect aggregate vehicle capacity levels in the network. The second, $x_p^k \in \mathbb{B}, k \in \mathcal{K}, p \in \mathcal{P}_k$ indicate whether or not path $p$ is selected for commodity $k$. We presume a joint distribution $\Omega$ of shipment sizes is known. Associated with realization $\omega \in \Omega$ is a random vector $s(\omega)$ of shipment sizes consisting of elements $s(\omega)^k, \omega \in \Omega, k \in K$. Given this notation, the Stochastic Scheduled Service Network Design Problem with Flexible Schedules (SSSNDFS) can be formulated as follows.

$$\text{minimize} \sum_{(i,j)\in\mathcal{A}} f_{ij} y_{ij} + \sum_{k\in K} \mathbf{E}_\omega[s(\omega)^k] c_p x_p^k + \mathbf{E}_\omega[Q(y, x, s(\omega)]$$

*subject to*

$$\sum_{p\in\mathcal{P}_k} x_p^k = 1 \qquad \forall k \in \mathcal{K}, \tag{1}$$

$$x_p^k \in \mathbb{B} \qquad \forall k \in \mathcal{K}, \forall p \in \mathcal{P}_k, \tag{2}$$

$$y_{ij} \in \mathbb{N} \qquad \forall (i, j) \in \mathcal{A}, \tag{3}$$

The objective function seeks to minimize the sum of committed-to capacity costs and the expectation of both path per-unit-of-size costs and the costs incurred routing shipments given the committed-to capacity. Constraints (1) ensure a single path is chosen for each commodity. Constraints (2) and (3) define the first stage variables and their domains.

Given $y$ and $x$, the recourse cost function $Q(y, x, s(\omega))$ aims to minimize second-stage costs under scenario $\omega$. Second-stage decisions must ensure that (i) commodities are delivered from their origins to their destinations within their time windows, (ii) sufficient capacity is assigned, whether committed-to vehicles acquired in the first stage or extra vehicles acquired in the second stage, (iii) commodities follow the physical paths prescribed by $x$, and (iv) committed-to vehicles are used on the legs prescribed by $y$. The recourse problem is an SSND variant, and the corresponding second-stage scheduling decisions can be modeled on a time-expanded network. We refer to this formulation as $TEN$. On the other hand, an alternative formulation of the recourse problem consists of modeling second-stage scheduling decisions with variables that represent dispatch times, as proposed in Hewitt and Lehuédé [2]. In this case, vehicle capacity needs are modeled with an a priori enumeration of the potential consolidations on an arc, and a consolidation is defined as a set of shipments that can be dispatched simultaneously on the same transportation move. When a consolidation consisting of more than one shipment is chosen, Big-M type constraints synchronize the values of dispatch time variables to ensure the shipments in that consolidation dispatch simultaneously. We refer to this formulation as $CONS$.

## 3 A tailored Unified Branch-and-Benders-Cut

We adapt the UB&BC framework - a recently proposed Benders-based decomposition algorithm for solving two-stage stochastic programs with integer recourse. It applies classical Benders decomposition to the two-stage stochastic program to be solved, albeit with integrality requirements relaxed in the second-stage subproblems. Thus, duality theory can be leveraged. During the course of this Benders-type solution process, UB&BC carefully maintains a list of candidate solutions, and upper and lower bounds on the objective function values of those solutions in the context of the stochastic program with integrality constraints restored. UB&BC bounds these objective function values because the exact evaluation of a candidate solution requires solving second-stage subproblems that are MIPs, something the framework seeks to limit doing. At the end of the Benders process, if a provably optimal solution has not been found, each of the candidate solutions is evaluated by solving the corresponding second-stage subproblems as MIPs in a *post-processing* phase. Because UB&BC uses modified fathoming rules, it guarantees that master solutions yielding the global optimum are included in the candidate solutions.

As UB&BC relies on solving a relaxation of the stochastic program, its performance depends significantly on how much that relaxation can be strengthened to produce tight bounds. Thus, we reinforce the master problem with information derived from the second-stage subproblems by leveraging the consolidation-based formulation. Also critical to the performance of UB&BC is its ability to derive tight bounds, both above and below, on the objective function values of candidate solutions. Tight bounds on these objective function values enable UB&BC to identify better whether a solution is a candidate to be an optimal solution without having to evaluate it exactly. We thus use the second-stage subproblem consolidation-based formulation, which provides better bounds. We also derive valid optimality Benders cuts for that formulation, which has not yet been done.

## 4 Results and discussion

We produce 324 instances based on a portion of the network of a United States-based LTL carrier, with $|\mathcal{K}|$ varying from 25 to 150 and $|\Omega|$ varying from 50 to 100, both with an increment of 25. For these instances, we solve the $TEN$ and the $CONS$ formulations with $CPLEX$, and refer to those results as *CPLEX-TEN* and *CPLEX-CONS*. We also solve the instances with UB&BC. The time limit is set to two hours. Table 1 provides an overview of the results.

| | | *CPLEX-TEN* | | *CPLEX-CONS* | | *UBBC_CMP_CONS* | |
|---|---|---|---|---|---|---|---|
| | No. instances | No. instances solved | Optimality gap (%) | No. instances solved | Optimality gap (%) | No. instances solved | Optimality gap (%) |
| Total | 320 | 83 | 64.98 | **188** | 25.56 | 152 | **1.50** |

Table 1: Comparison of the mathematical formulations and the UB&BC algorithm

*CPLEX-CONS* substantially outperforms *CPLEX-TEN*. However, both deterministic equivalent formulations become intractable as the number of commodities and scenarios increases. On the other hand, while the UB&BC algorithm solves fewer instances than *CPLEX-CONS*, it scales more effectively to larger-scale instances. It yields the lowest average gap at termination, an order of magnitude lower than the benchmarks. This can be explained by the fact that UBBC leverages subproblem separability. Further investigations show that strengthening the master problem and using the consolidation-based subproblem formulation allows UB&BC to converge without resorting to the *post-processing* phase, such that no second-stage subproblems have to be solved as MIPs.

## References


[1] Belieres, S, & Hewitt, M. (2023). Hedging Against Uncertainty in Transportation Network Design Through Flexible Scheduling. working paper or preprint: *https://hal.science/hal-04106912*

[2] Hewitt, M., & Lehuédé, F. (2023). New Formulations for the Scheduled Service Network Design Problem. *Transportation Research Part B: Methodological*, 172, 117-133.

[3] Mahéo, A., Belieres, S., Adulyasak, Y., & Cordeau, J. F. (2020). Unified branch-and-Benders-cut for two-stage stochastic mixed-integer programs. GERAD HEC Montréal.

# Urban Network Design with Ship-from Store

**Imen Ben Mohamed**
AIM Research Center on Quantitative Methods in Business (Quant),
Emlyon Business School, Lyon, France
Email: benmohamed@em-lyon.com

**Walid Klibi**
The Centre of Excellence in Supply Chain (CESIT),
Kedge Business School, Bordeaux, France

**Florian Bertrand**
The Centre of Excellence in Supply Chain (CESIT),
Kedge Business School, Bordeaux, France

## 1 Introduction and problem description

We consider the context of an omnichannel retailer looking for the design of an efficient urban delivery network that considers the ship-from-store option to leverage tight responsiveness requirements. Ship-from-store (SFS) is a novel omnichannel distribution strategy that allows retailers to use their brick-and-mortar stores to fulfill and deliver online orders. At the design level, the consideration of a large network of stores in the distribution schema to customers poses additional challenges. First, the integration of stores as an intermediate layer promotes the design of 2-echelons distribution networks [3, 2], which are more difficult to tackle than traditional distribution schema. Second, the high number of stores spread across a given city increases the options for online orders' assignment to stores and their delivery to a large customer base, and thus the combinatorial of the problem. Third, the specificities of the omnichannel retail with an explosion of online orders from a significant number of various ship-to points, challenge the scalability of the urban design model. Finally, the time lag between the decisions to select a subset of stores to act as ship-from locations and the fulfillment and delivery decisions when online orders occur, raise the uncertainty related to the decisional process of the urban network design.

This paper contributes to the literature by jointly studying the ship-from store locations, in-store fulfillment optimization, and delivery decisions under uncertainty. We formulate the problem as a two-stage stochastic program and we propose an integer l-shaped based decomposition solution approach using an alternating cut strategy.

## 2 The two-stage stochastic program

In the urban delivery network design with ship-from store option, the retailer should decide which subset of stores should be selected to complement the tier-1 distribution center taking into account fixed costs related to store preparation with specific equipment, and variable costs related to order fulfilling the order, and factors such as inventory levels at stores. Multiple delivery options are considered : a carrier service at lower cost and courier service for urgent deliveries at higher cost. The pickup time by a carrier from stores and cannot be changed at an operational level, which calls for anticipating such decision at a design level. Further, the design decisions depend on several parameters as store capacities to online fulfillment, in-store demand and online orders that are uncertain. We model such uncertainty by a set of scenarios.

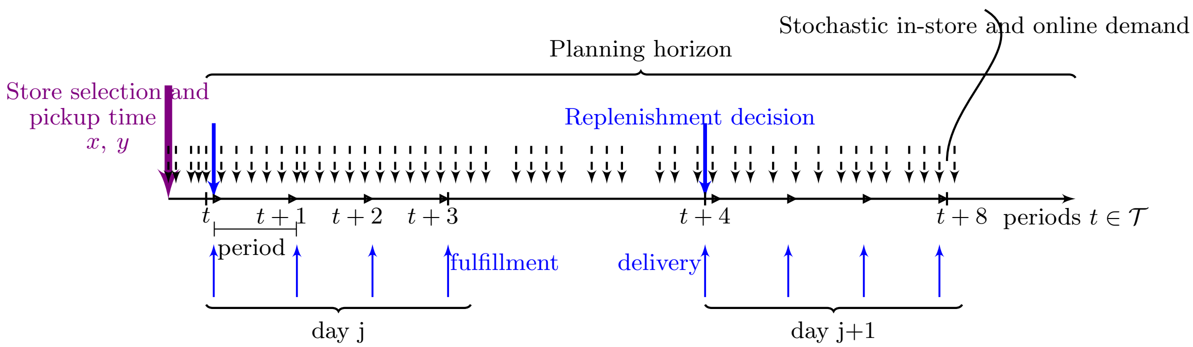



Figure 1: Decisions, time-lag and hierarchy

Figure 1 depicts our problem decisions, the time-lag and hierarchy in decisions. The horizon covers multiple planning periods (days) where each day is divided into four operational periods. Each period $t$ marks a specific time hour in a day, namely 9am, 12pm, 15pm and 6pm respectively. Such periods are defined in accordance with the evolution of the uncertain in-store and online demand over time. Online orders are placed overall the day and accumulated (batched) up to a period $t$. Design decisions on the store selection ($x$) and the carrier pickup time $y$ are taken at the beginning planning horizon before the realization of uncertainty. After revealing the uncertain in-store demand, multi-item online orders and store capacities at $t$, operational decisions on the in-store replenishment, inventory, the order preparation, and delivery are determined. Worth noting that replenishment from the distribution warehouse is made daily, at the beginning of the day.

Our problem is then a hierarchical decision problem. This stems for the temporal hierarchy between the design decisions and the operational decisions. To catch the temporal hierarchy, we formulate the problem as a two-stage scenario-based stochastic model where the scenarios represent the realizations of the uncertain set of multi-item online orders, in-store demands, and the store capacities over the planning horizon. In the first-stage, design decisions are taken here-and-now at the beginning planning periods. Then, when uncertainty revealed, second-stage decisions are determined for every scenario.

## 3 The integer l-shaped based decomposition approach

Due to the inherent uncertainty of the ordering process and the combinatorial complexity of the problem, the deterministic equivalent model along with a sample average approximation becomes computationally intractable for large scale instances. We therefore present an integer l-shaped based decomposition approach to optimally solve the problem. Figure 2 synthesizes the main steps of our approach.

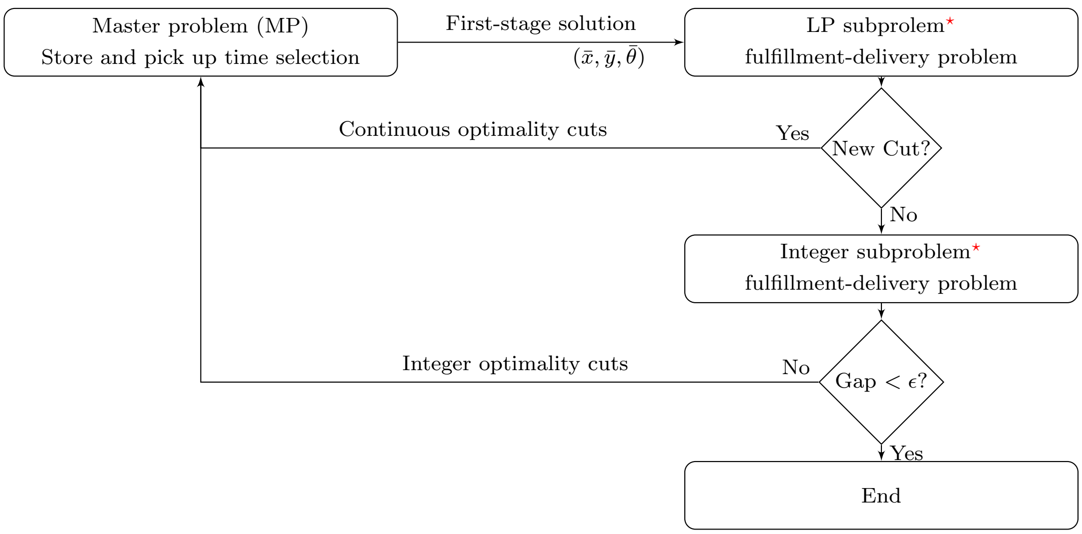



Figure 2: Integer l-shaped based decomposition main steps

* may be solved with parallel computing

The algorithm explores the first-stage solution space and decomposes the second-stage problem into subproblems for fixed first-stage solutions $(\bar{x}, \bar{y}, \bar{\theta})$, i.e., store selection with in-store fulfillment process and pickup time. The resulting problem subproblem is a fulfillment-delivery mixed-integer program. Using the alternating cut strategy [1], we generate optimality cuts from thesubproblem. The linear relaxation (LP) of subproblem is first solved allowing to generate a continuous optimalitty cuts using optimal dual solutions to cut off the current first-stage solution. If this is the case, the algorithm returns adding the continuous cut to the master. If the cut is not violated, then the integer subproblem is solved and the algorithm explores the current solution to see if the integer optimality cuts proposed by [4] can be added. If this is the case, the algorithm proceeds by adding the integer optimality cut to the master problem. Otherwise, the first-stage candidate solution is accepted as an incumbent solution.

## 4 Preliminary results and future work

We consider instances of 3 stores and 1 warehouse under an horizon of 28 periods (7 days) with 5 scenarios of online orders where an average of 180 multi-item online orders is expected per day. The number of products in instances is fixed to 70 products. The solution of the deterministic equivalent model (DE) with Gurobi on theses instances shows a high computing time as it takes 6 hours in average to solve. In light of these results, as the instance size, the number of orders and the number of scenarios grow, the DE becomes untractabe and the integer l-shaped based decomposition algorithm will present an efficient exact solution method.

In addition, the analysis of these results underlines a shift in the fulfillment of online orders. Indeed, two stores are selected promoting the two-echelon structure to leverage tight responsiveness: the results show that up to 80% of online orders are fulfilled from the 2 selected stores. The model further enlightens the retailer about the best time slot for the carrier to collect orders from the stores for delivery.

Accordingly, our study introduces and formulates an emerging problem in the urban context, namely the urban delivery network design with ship-from store option under multi-period and stochastic setting. The integer L-shaped algorithm is presented to effectively solve the problem. This study will offer a detailed analysis of the results on larger instances considering more stores, products and online orders, and derive managerial insights regarding the network structure, the in-store fulfillment and delivery.

## References


[1] G. Angulo, S. Ahmed, and S. S. Dey. Improving the integer l-shaped method. *INFORMS Journal on Computing*, 28(3):483–499, 2016.

[2] I. Ben Mohamed, W. Klibi, and F. Vanderbeck. Designing a two-echelon distribution network under demand uncertainty. *European Journal of Operational Research*, 280(1):102–123, 2020.

[3] T. G. Crainic, J. G. Feliu, N. Ricciardi, F. Semet, and T. Van Woensel. 10. operations research for planning and managing city logistics systems. *Handbook on City Logistics and Urban Freight: 0*, page 190, 2023.

[4] G. Laporte and F. V. Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

# A Machine Learning-assisted Algorithm for Solving the Unsplit Capacitated Vehicle Routing Problem

**Luca Bertazzi**

Department of Economics and Management,

University of Brescia, Brescia, Italy

Email: luca.bertazzi@unibs.it

**Demetrio Laganà**

Department of Mechanical Engineering, Energy Engineering and Management,

University of Calabria, Rende, Italy

Emai: demetrio.lagana@unical.it

**Pantelis Z. Lappas**

Department of Statistics and Actuarial-Financial Mathematics,

University of the Aegean, Samos, Greece

Email: pzlappas@aegean.gr

## 1 Motivation and Innovation

The widespread use of Artificial Intelligence (AI) and Machine Learning (ML), along with big data and new computing hardware, has drawn attention to challenging combinatorial optimization problems in Operations Research (OR) [1, 2]. ML algorithms can aid in routing modeling and algorithm development, initially used as supervised models to handle supply chain uncertainties and enhance existing algorithms [3, 4]. They assist in route construction, hyper-heuristic selection, neighborhood adaptation, heuristic generation, and decomposition strategy definition through unsupervised learning.

While AI&ML solutions gain traction, integration between OR and ML communities remains limited [1, 5]. ML solutions often lack generalization, data efficiency, standardized evaluation, and interpretable structures, contrasting with OR's approach. This work proposes a novel ML-assisted routing algorithm leveraging established ML techniques and Vehicle Routing Problem (VRP) modeling practices. It aims to model the unsplit Capacitated Vehicle Routing Problem (CVRP) using concepts related to the Capacitated Concentrator Location problem (CCLP) and employ a hybrid approach combining Genetic Algorithm (GA), Fuzzy C-Means Clustering

(FCMC), and Deep Learning (DL) to solve CVRP instances. New hyperparameters will be introduced for ML algorithms, optimized during training to minimize routing costs.

## 2 Problem Description and Formulation

Unsplit CVRP is defined on a complete undirected graph $G = (V, E)$, where $V = \{0\} \cup V'$ is the set of vertices, containing the depot $\{0\}$, and a set of $n$ customers $V' = \{1,2,3, \dots, n\}$. $E = \{(i, j): i, j \in V, j > i\}$ is the set of edges between vertices. A nonnegative cost $c_{ij}$, defined as the Euclidean distance, is associated with each edge $(i, j) \in E$ and represents the travel cost spent to go from vertex $i$ to vertex $j$, $\forall i, j \in V$. The usage of the loop edge $(i, i)$ is not allowed, whereas the cost matrix $[c_{ij}]$ is symmetric, i.e. $c_{ij} = c_{ji}$, and satisfies the triangle inequality, $c_{ij} \leq c_{ik} + c_{kj}, \forall i, k, j \in V$. Let integral variables $Q$ and $d_i$ be the vehicle capacity and demand of a customer $i \in V'$, respectively. The fleet of vehicles can be limited or unlimited. In the case of a limited fleet of vehicles, there are $K$ identical vehicles available. The main goal of the unsplit CVRP is to find a set of routes that minimize the total distance.

Considering the work of Bertazzi and Wang [6], the unsplit CVRP is solved by using the CCLP concept in the clustering phase, where a set of concentrators with known capacities and costs to open, a set of terminals with known demands, and the cost of connecting a terminal to a concentrator are given. Then, a Travelling Salesman Problem (TSP) solution is computed for each cluster to find the set of routes starting and ending at the depot. The main objective is to determine the set of concentrators to open and the allocation of terminals to concentrators so that the total cost is minimized and each terminal is associated with an open concentrator without overloading any concentrator. Mapping to the unsplit CVRP, each customer can be both a concentrator and a terminal. Selecting a customer as a concentrator means that a set of customers (i.e., terminals) is allocated to him to create a cluster with total demand not greater than the vehicle's capacity. Therefore, each cluster can be managed as a TSP. Let $\hat{c}_{ij} = c_{0i} + c_{ij} - c_{oj}$ an estimate of the connection cost of associate customer $i \in V'$ to concentrator $j \in V'$, computationally proved by Bertazzi and Wang [6] that outperforms among others cost estimators. Additionally, let $x_{ij}$ be a binary variable equal to 1 if customer $i \in V'$ is associated with concentrator $j \in V'$ and 0 otherwise, as well as $y_j$ be a binary variable equal to 1 if concentrator $j \in V'$ is used to create a cluster and 0 otherwise. Then, the unsplit CVRP can be modeled as follows in the clustering phase:

$$\min z = \sum_{j=1}^{n} 2c_{0j}y_j + \sum_{i=1}^{n}\sum_{j=1}^{n} \hat{c}_{ij}x_{ij} \quad (1)$$

$s.t.$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i \in V' = \{1,2,3,\dots,n\} \quad (2)$$

$$\sum_{i=1}^{n} d_i x_{ij} \leq Q y_j, \qquad j \in V' = \{1,2,3,\dots,n\} \quad (3)$$

$$x_{ij} \in \{0,1\}, \qquad i,j \in V' = \{1,2,3,\dots,n\} \quad (4)$$

$$y_j \in \{0,1\}, \qquad j \in V' = \{1,2,3,\dots,n\} \quad (5)$$

The objective function (1) minimizes an estimate of the total cost which is the sum of (i) costs obtained traveling from the depot to a selected concentrator and vice versa, and (ii) costs visiting all customers within a cluster created by a selected concentrator. Constraints (2) ensure that each customer is related to one concentrator, whereas constraints (3) guarantee that the total demand of a cluster is not greater than the vehicle's capacity. Furthermore, constraints (4) and (5) are the domain constraints. In the case of a limited fleet of vehicles, constraints (6) can be considered to ensure that the maximum number of concentrators cannot be greater than the available number of homogeneous vehicles.

$$\sum_{j=1}^{n} y_j \leq K, \qquad j \in V' = \{1,2,3,\dots,n\} \quad (6)$$

# 3 Solution Approach

Considering the formulation presented in Section 2, essential characteristics of the unsplit CVRP were simplified, indicating candidate ML solutions for approaching the problem. It is worth noticing that the CCLP seems to be directly associated with a clustering problem. In this direction, a soft clustering method such as the FCMC algorithm is suggested for grouping customers into $C$ clusters, where the cluster centers correspond to concentrators and the rest customers within the cluster correspond to terminals. The probability of one terminal belonging to a cluster can take any value between 0 and 1.

Due to the NP-hard nature of the unsplit CVRP, an in-depth exploration of potential concentrators is essential for seeking a near-optimal solution. Thus, the GA drives the search process, beginning with a population of initial solutions generated with the FCMC algorithm. A repetitive procedure, using crossover and mutation operators to investigate clustering schemas, is then applied without violating problem constraints. The fitness function, associated with routing costs, is managed by the 2-opt heuristic algorithm.

As to the main contribution and novelty of this work, we introduce a new ML-assisted VRP algorithm consisting of training and inference phases. The training phase is enriched by generating a large amount of VRP instances sharing the same characteristics per problem category (small, medium, and large problems). Generated data aid in selecting the optimal clustering schema during training. To enhance ML model generalization, we regularize the objective function (1) using factors $\alpha$ and β for the first and second terms of the total transportation cost, respectively. These factors are assumed as hyperparameters and tuned through the training phase to find better cost approximations concerning cost estimators reported by Bertazzi and Wang [6]. In the inference phase, trained deep learning models, combined with GA, assess the solution algorithm's performance on public CVRP benchmark instances. Testing includes varied fleet properties to thoroughly evaluate algorithmic performance, with reported computational results.

## References


[1] R. Bai, X. Chen, Z.-L. Chen, T. Cui, S. Gong, W. He, X. Jiang, H. Jin, J. Jin, G. Kendall, J. Li, Z. Lu, J. Ren, P. Weng, N. Xue and H. Zhang, “Analytics and machine learning in vehicle routing research”, *International Journal of Production Research* 61(1), 4-30, 2023.

[2] Y. Bengio, A. Lodi and A. Prouvost, “Machine learning for combinatorial optimization: A methodological tour d’horizon”, *European Journal of Operational Research* 290, 405-421, 2021.

[3] A. Lodi and G. Zarpellon, “On learning and branching: a survey”, “*TOP* 25, 207-236, 2017.

[4] Q. Wang and C. Tang, “Deep reinforcement learning for transportation network combinatorial optimization: a survey”, *Knowledge-Based Systems* 233, 107526, 2021.

[5] L. Accorsi, A. Lodi and D. Vigo, “Guidelines for the computational testing of machine learning approaches to vehicle routing problems”, *Operations Research Letters* 50, 229-234, 2022.

[6] L. Bertazzi and X. Wang, “Matheuristics with performance guarantee for the unsplit and split delivery capacitated vehicle routing problem”, *Networks*, 1-20, 2022.

# A Column-Generation Algorithm for the Intercity Electric Bus Scheduling Problem

**Malte Billen, Jörg Kalcsics, Sergio García, Gonçalo Dos Reis**
School of Mathematics,
The University of Edinburgh, Edinburgh, United Kingdom
Email: m.d.billen@sms.ed.ac.uk

**Adrian Grütter**
Ember Core Ltd, Edinburgh, United Kingdom

24 November 2023

## 1 Introduction

As transportation emissions are one of the main contributors to global greenhouse gas emissions, many governments seek to increase sales in zero-emission vehicles. Consequently, investments in battery-powered electric vehicles and public charging infrastructure are growing. However, while light-duty electric vehicles are well established, the market share of heavy-duty electric trucks and buses remains low. In response, grants for transport operators are available to incentivise the transition towards zero-emission fleets [1]. While many urban public transport operators are receiving attention in their transition towards an electric fleet, intercity operators have also started to incorporate electric buses into their fleets. The increased range of battery powered long-distance electric coaches, market availability, potential to save on fuel cost and government incentives have led to interest in zero-emission intercity public transportation [1, 5].

Ember Core Ltd, established in 2019, is the first intercity operator in the United Kingdom to offer public transportation across Scotland using an all-electric fleet of battery-powered coaches. Our joint research focuses on optimising the utilisation of Ember's electric fleet and charging infrastructure.

Electric bus scheduling was introduced by [6] with a focus on the optimisation of fleets with limited range and recharging requirements. It is classified as a modification of single-depot or multi-depot vehicle scheduling problems with route constraints [2]. Studies generally focus on urban transportation and use column generation to find a solution [7].

While sharing many similarities, electric intercity bus scheduling differs in various ways: operational requirements, sparse charging infrastructure, frequent and lengthy charging sessions, and schedule-based trips that include 24-hour service. While [8] developed a subgradient model to solve a scheduling model for conventional intercity transport, [5] investigates the challenges of incorporating electric buses in overnight intercity travel. However, to the best of our knowledge, a solution method to optimally schedule electric intercity buses has not yet been investigated. Therefore, we introduce the Intercity Electric Bus Scheduling Problem (IEBSP). It aims to find an optimal selection $x$ of feasible schedules $\mathcal{Q}_v$ for a fleet of plug-in electric intercity buses $\mathcal{V}$ of a transport operator. Given a charger infrastructure $\mathcal{C}$, a set of overnight timetabled trips $\mathcal{I}$ and service slots $\mathcal{S}$, buses must serve all trips without exceeding battery limitations, noting that charger capacities are restricted and cannot be violated. They must also routinely attend limited service slots to enhance battery lifespan and uphold cleanliness. The objective of the IEBSP is to maximise vehicle utilisation by minimising deadheads and the number of buses in rotation. Therefore, we present a column generation algorithm with a variable-fixing heuristic that can support the operational decision-making of an electric intercity bus operator.

## 2 Iterative Column Generation Heuristic

Given the work on electric bus scheduling, and resulting from a Dantzig-Wolfe decomposition [3], we define two minimisation problems: a restricted master problem and a pricing problem. We embed the resulting column generation heuristic with variable-fixing in an iterative solution method to find schedules for a long planning horizon.

Considering any fixed planning horizon with a set of trips $\mathcal{I}$ and service slots $\mathcal{S}$ commencing within the time bounds, we propose the following set covering formulation that describes the restricted master problem for the IEBSP:

$$\min \quad \sum_{i\in\mathcal{I}} y_i F_i^{Penalty} + \sum_{v\in\mathcal{V}} \sum_{q\in\bar{\mathcal{Q}}_v} x_{vq}(F_v^{Vehicle} + F_{vq}^{Schedule}) \tag{2.1a}$$

$$\text{s.t.} \quad \sum_{q\in\bar{\mathcal{Q}}_v} x_{vq} \leq 1 \qquad \forall v \in \mathcal{V}, \tag{2.1b}$$

$$\sum_{v\in\mathcal{V}} \sum_{q\in\bar{\mathcal{Q}}_v : i\in\mathcal{I}(q)} x_{vq} + y_i = 1 \qquad \forall i \in \mathcal{I}, \tag{2.1c}$$

$$\sum_{v\in\mathcal{V}} \sum_{q\in\bar{\mathcal{Q}}_v : i\in\mathcal{S}(q)} x_{vq} \leq 1 \qquad \forall s \in \mathcal{S}, \tag{2.1d}$$

$$\sum_{v\in\mathcal{V}} \sum_{q\in\bar{\mathcal{Q}}_v} x_{vq}(R_{vqct}^S - R_{vqct}^E) + b_{c,t-1} - b_{ct} = 0 \qquad \forall c \in \mathcal{C}, t \in \mathcal{T}, \tag{2.1e}$$

$$0 \leq b_{ct} \leq K_c \qquad \forall c \in \mathcal{C}, t \in \mathcal{T}, \tag{2.1f}$$

$$y_i \geq 0 \qquad \forall i \in \mathcal{I}, \tag{2.1g}$$

$$x_{vq} \in \{0,1\} \quad \forall v \in \mathcal{V}, q \in \bar{\mathcal{Q}}_v \tag{2.1h}$$

Objective (2.1a) minimises the operational cost of selecting a feasible schedule $x$ from an eligible subset of generated schedules $\bar{\mathcal{Q}}_v \subseteq \mathcal{Q}_v$ for a vehicle $v$, while not leaving any trip $i$ unassigned $y_i = 1$. (2.1b) ensures that a vehicle is assigned no more than one schedule. (2.1c) and (2.1d) restrict the number of vehicles assigned to a specific trip or service slot, respectively. (2.1e) tracks the number of coaches $b_{ct}$ at charger $c$ in any time-interval $t$ of the planning horizon $\mathcal{T}$. It monitors the start $R^S$ and end $R^E$ of selected charging breaks, while ensuring in (2.1f) that the charger capacity limits $K_c$ are never violated.

Note that an integer feasible solution always exists, even with $\bar{\mathcal{Q}}_v = \emptyset$ and all trips selected as unassigned. This solution gives an upper bound to the formulation and can be used to start the column generation algorithm. To improve the upper bound, new cost-reducing feasible schedules $q \in \mathcal{Q}_v \setminus \bar{\mathcal{Q}}_v$ must be created. By relaxing the integrality constraint (2.1h) and switching it with a non-negative bound for all variables $x$, we can obtain a primal and dual solution from the linear relaxation of the restricted master problem. Given the dual solution with $\alpha_v \leq 0$, $\beta_i \in \mathbb{R}$, $\delta_s \leq 0$ and $\gamma_{ct} \in \mathbb{R}$, respectively, for (2.1b)-(2.1e), we can define the pricing problem for any electric bus $v$ as follows:

$$\begin{aligned}\arg\min_{q \in \mathcal{Q}_v} \{\widehat{F}^{Reduced}_{vq}(\alpha, \beta, \gamma, \delta) \ :=\ & (F^{Vehicle}_v + F^{Schedule}_{vq}) - \\ & (\alpha_v + \sum_{i \in \mathcal{I}} \beta_i W_{vqi} + \sum_{s \in \mathcal{S}} \delta_s P_{vqs} + \sum_{c \in \mathcal{C}} \sum_{t \in \mathcal{T}} \gamma_{ct}(R^S_{vqct} - R^E_{vqct})) < 0\}\end{aligned} \tag{2.2}$$

For every dual solution, the pricing problem (2.2) returns a minimum reduced cost schedule that can improve the incumbent solution when added to the restricted master problem (2.1). If the pricing problem returns the empty set for every bus, no schedule can further reduce the objective and thus an optimal fractional solution has been found [4].

While the pricing problem has a linear objective, the set of feasible schedules $\mathcal{Q}_v$ is non-convex. The various route restrictions regarding the battery range, charging duration, and cleanliness of the vehicle, result in a non-linear set of constraints. Overall, this is a resource constraint shortest path problem that is solved with a label-setting algorithm on a time-space network. To find integer optimal solutions, we use a heuristic variable fixing strategy. As commonly practised, we branch on the original variables prior to the Dantzig-Wolfe decomposition. After obtaining a fractional optimal solution, we fix the variable representing a trip or a service slot to the vehicle that in sum has the largest fractional solution value. Afterwards, we restart the column generation algorithm. As a result, an integer solution is always found in a finite number of steps, since the algorithm either fixes all trips and service slots to a vehicle or leaves them unassigned.

Finally, to construct a schedule for a long period, we decompose the long planning horizon into shorter intervals. To mitigate the impact of the decomposition, we allow for overlaps between the shorter intervals. This procedure also helps to balance the impact of degeneracy within the restricted master problem.

## 3 Computational Results and Conclusions

Conducting research in collaboration with Ember gave us valuable insights in the operational scheduling of intercity electric buses. The company currently serves three cities in Scotland on two intercity lines, running around 80 trips per day. The number of coaches and chargers is small; however, the company's plans for a larger network indicate our research has a real impact on its operation. We have implemented the column generation algorithm in C++ using only open-source libraries. With a focus on operational feasibility, the algorithm provides quality schedules that meet real-world business constraints. As a result, Ember is deploying our software to dispatch their electric fleet. In our forthcoming presentation, we will provide a comprehensive analysis of the algorithm's performance and overall quality, offering a detailed insight into its effectiveness and capabilities. Early results indicate that limitations arise from solving the non-convex pricing problem and prolonged planning horizons.

## References


[1] International Energy Agency. *Global EV Outlook 2023: Catching up with climate ambitions.* IEA Publications International Energy Agency, 2023.

[2] S. Bunte and N. Kliewer. An overview on vehicle scheduling models. *Public Transport*, 1(4):299–317, 2009.

[3] G.B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.

[4] G. Desaulniers, J. Desrosiers, and M.M. Solomon. *Column generation.* Springer, New York, 2005.

[5] N.A. El-Taweel and H.E.Z. Farag. Incorporation of battery electric buses in the operation of intercity bus services. In *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 1–6. IEEE, 2019.

[6] J.-Q. Li. Transit bus scheduling with limited energy. *Transportation Science*, 48(4):521–539, 2014.

[7] S.S.G. Perumal, R.M. Lusby, and J. Larsen. Electric bus planning & scheduling: A review of related problems and methodologies. *European Journal of Operational Research*, 301(2):395–413, 2022.

[8] S. Yan and H.-L. Chen. A scheduling model and a solution algorithm for inter-city bus carriers. *Transportation Research Part A: Policy and Practice*, 36(9):805–825, 2002.

# Dynamic Service Network Design

**Alexander Bode**
Decision Support Group,
Technische Universität Braunschweig, Germany
Email: alexander.bode@tu-braunschweig.de

**Mike Hewitt**
Quinlan School of Business,
Loyola University Chicago, USA

**Dirk C. Mattfeld**
Decision Support Group,
Technische Universität Braunschweig, Germany

**Marlin W. Ulmer**
Chair of Management Science
Otto-von-Guericke-Universität Magdeburg, Germany

## 1 Introduction

We consider a dynamic version of the service network design problem, where commodities become visible over time and have to be sent from an origin to their destination. We assume that there is no particular demand distribution. This can be the case with spare parts logistics. In this area, it is generally not possible to predict when and where new parts will be needed. After the spare parts have been delivered to the customer from the nearest warehouse, the parts must be replenished in the warehouse. While the spare parts must reach the customer as quickly as possible, more time can be taken to replenish the spare parts in the warehouse. We consider the problem of replenishing the warehouse with spare parts.

The term Dynamic Service Network Design is introduced in [1] for the first time. Our research differs from that approach in various respects. We define transportation services without an a priori baseline plan and guarantee that the commodities reach their destination on time. Each transportation service in our problem is carried out by one or more vehicles, which transport commodities that are small in relation to the vehicle capacity. We take the capacity of vehicles into account, but assume an unlimited fleet size, which for example can be provided by a third party provider, such as Uber Freight. The capacity is therefore required to estimate the number of vehicles and calculate the costs. The aim is to minimize the total cost of all vehicles used on the transport services required.

We model the problem as a temporal decomposition resulting in sub-problems that are solved in a rolling horizon fashion. Decisions are made without any information about future commodities. Once decisions have been made, they cannot be changed and are potentially sub-optimal. It would therefore make sense to estimate the future and incorporate it into the decision-making process. For this reason, a cost function approximation is considered, which estimates the expected future costs. The preliminary results show significant cost savings by using the cost function approximation presented by us.

## 2 Problem

We define a transportation network consisting of transportation services (edges) between transshipment facilities (nodes). The transshipment facilities act as the origin, destination and transshipment points for the commodities. The transport services between the facilities are the subject of the decision. It is decided which transportation services are to be used over the next decision period (e.g. the next 24 hours). New commodities appear during the day, but are batched for consideration of next day planning. They become known at the time of planning (e.g. 6 p.m.) of the next decision period. As already described in the introduction, the parts do not have to reach their destination as fast as possible, instead they have a few days before they are due. Each commodity is defined by weight, origin, destination and due date. The number of vehicles used on a transport service depends on the sum of the weights of the commodities that are to use the service. Since the vehicles of a third party provider are used, the number of trucks available for a service are unlimited. Operational complexities associated with the decisions (e.g. regulations regarding driver itineraries) are not considered. Optimization takes place over a longer period of time (e.g. 21 days), during which new commodities appear every day that need to be delivered. The aim is to minimize the costs for the vehicles used.

## 3 Model

The dynamic service network design problem can be defined as a sequential decision process, which goal it is to minimize transportation costs. It exists a decision point at which decisions must be made for the following decision period. Each decision point is associated with a state. A state contains information about all commodities that appear at the various transshipment facilities or reach them on their route during the decision period. The decisions that are made for a state include all transportation services that are used to transport all commodities in the decision period. To find the decisions, the physical transportation network for each state is converted into a time-expanded network and a MIP is set up. This is a formulation of the service network design problem, which is based on the model of [2].

As not all commodities reach their destination within the decision period and planning is only carried out for the decision period, the MIP ends at the end of this decision period. This is the case because no information about future commodities is known and the computational effort would be very high if the simulation were always carried out to the end. Commodities that have not reached their destination require an intermediate destination for solving the MIP. Such an intermediate destination node (IDN) is a node in the time-expanded network from which the destination of the commodity can still be reached on the shortest path before it is due. Each commodity has a set of IDNs in each decision period. Once the decision has been made, the transition to the post-decision state takes place. The post-decision state contains all commodities that have reached their destination or are in one of their IDNs. The post-decision state is followed by the stochastic transition to the next decision point and the revelation of the new commodities entering the network.

## 4 Method

From the perspective of the MIP, which is solved in each state, the IDN that causes the lowest costs is always selected. The solution is therefore myopic for the Sequential Decision Processes, as the future commodities are not taken into account. However, the decisions made in the current state are decisive for future consolidation options, which in turn lead to cost savings. To solve this problem, we propose a cost function approximation that extends the objective function of the MIP (MIP-1) by two additional terms. The objective function thus comprises three terms. The first term comprises the costs caused by the decisions in the current state. It is therefore the classic objective function of the MIP. The second term is intended to estimate the costs in the near future and include future consolidation options in the decision. Thus, it will be examined which consolidation opportunities may arise with the known commodities in the future. For this purpose, a second MIP (MIP-2) is set up, which begins at the end of MIP-1 and extends into the next planning period. This means that part of the next planning period (e.g. the next 8 hours) is planned using the known commodities. In this way, the decisions of the current planning period are also planned with a view to the future. In MIP-2, as in MIP-1, there are IDNs, which are referred to as temporary IDNs (tIDN) in the following paragraph.

The third part of the objective function evaluates the consolidation potential of the commodities if they end in a certain tIDN. The costs of the far future are thus estimated. The consolidation potential is expressed in costs. First, to calculate the potential, the costs are calculated which are incurred if the commodity is sent from the tIDN alone (without consolidation) to its destination. These costs are then divided by the slack that the commodity has in the tIDN. If a commodity has a high slack in a tIDN, it is likely that a consolidation opportunity will arise for the commodity in the future. This makes the tIDN more attractive for the MIP.

## 5 Computational results

As part of our research, we conducted several experiments to test our cost function approximation (cfa-complete). We compared our solution method with three other heuristic solution methods. The first heuristic solution method was the approach of sending the commodities to their destination as fast as possible (gafap). The second was the approach of making the commodities wait as long as possible at their starting location (walap). The third approach is to solve the model step-wise but without the help of our solution approach (myopic). Thus, a MIP is solved and implemented without considering the future. We also tested the individual costs terms of our CFA (estimating the near future, cfa-near, and the far future, cfa-far). We use a real data set from a freight forwarder in North America as the data basis for our experiments. Our preliminary results show that cfa-complete achieves the best results on larger instances with, for example, 460 commodities over two weeks. A clear cost saving can also be seen. cfa-near causes 8.26% higher costs than cfa-complete and cfa-far 8.74%. myopic causes 29.29% higher costs. galap and walap cause costs that are twice as high as those of cfa-complete. However, the difference between these two methods is only slight. Only on small instances with 100 commodities, for example, does the cfa-complete not perform as well as its individual components. However, the difference is slight. Overall, it can be seen in all instances that the CFA, regardless of its form, brings added value and allows the problem to be solved much better than the other heuristic approaches.

## References

[1] Baubaid, A., Boland, N., and Savelsbergh, M., “The Dynamic Freight Routing Problem for Less-Than-Truckload Carriers”, *Transportation Science*, 57(3), 717-740, 2023.

[2] Boland, N., Hewitt, M., Marshall, L., and Savelsbergh, M., “The continuous-time service network design problem”, *Operations research*, 65(5), 1303-1321, 2017.

# Integrated Load Bundling and Pricing for Decarbonized Freight Operations

**Maxime Bouscary**, **Alexandra Li**, **Mazen Danaf**, **Aron Brenner**, and **Saurabh Amin**
MIT, Cambridge, MA, USA
{mbscry, lialex}@mit.edu, mdanaf@uberfreight.com, {abrenner, amins}@mit.edu

***Keywords***— Dynamic programming, Pricing, Load bundling, Freight operations, Decarbonization

## 1 Introduction

In the ever-evolving landscape of freight transportation, the dynamics between shippers and carriers are being reshaped by digital brokerage platforms. These intermediary platforms facilitate freight movement by centralizing offers to a decentralized fleet of carriers. At the core of their operations lies the challenge of load recommendation and pricing, a critical undertaking that directly affects the efficiency, economy, and environmental footprint of the entire system.

Here we propose to leverage today's digital brokerage platforms by jointly designing and pricing bundles of loads i.e. offering multiple loads as a single package. Crucially, we account for the strategic behavior of carriers in terms of their accept/reject decisions and spatial positioning to be matched with desirable loads. We also account for the reliability preferences of shippers (e.g., tight versus flexible appointment windows). Suggesting carefully selected bundles can encourage carriers to take smarter routes. In particular, when priced adequately, bundles can help re-balance an unbalanced carrier fleet and leverage carrier preferences to reduce platform costs. We overcome the complexity of the joint bundling and pricing problem through a weighted set packing algorithm that efficiently selects non-overlapping bundles.

Significant literature on multi-item inventory management developed heuristics that focus on scenarios with large inventory levels [1] or asymptotic results [2], which are not applicable in scenarios with extremely sparse inventories as is the case in our problem since each load is a unique product. Although mixed bundling (suggesting both bundles and all the separate loads in the bundles) has already been tackled in the literature, prior works are often restricted to the design of a single bundle [3]. In contrast, our work extends to the selection of a potentially large set of bundles. Complementing studies already suggesting that bundling can reduce freight emissions and increase truck utilization [4], we seek to provide key insights that can improve decision-making on freight platforms and drive decarbonization in the transportation industry, reducing empty miles for greater sustainability and efficiency.

## 2 Problem statement

We consider the joint bundling and pricing problem for a freight transportation platform that serves both shippers (the demand side) and carriers (the supply side). On this platform, shippers post information about goods to be transported, commonly referred to as loads in the freight industry. Once this information is gathered, and before the delivery season starts, the platform can decide to form one or multiple bundles by grouping loads. We use the term bundle to refer to all of the options available to a carrier, including bundles of multiple loads and bundles of a single load. The delivery season then starts: carriers sequentially access the platform, the platform prices bundles, and each carrier decides to book one of the bundles or leave the platform without any booking.

We formulate the problem as a discrete-time, discrete-state Markov process. The delivery season runs over a finite time interval $[0, T]$. The demand consists of a set of $N$ loads: $\mathcal{L}$. Each load $i$ has an expiration date

$\tau_i \in [0,T]$. We assume that the demand is known in advance and remains unchanged throughout the delivery season. We assume carrier arrivals follow a Poisson process, with $\lambda$ being the probability of one arrival within a period. With a sufficiently small time interval as one period, we can ignore the probability of more than one arrival within a period and the probability of no arrival is $1-\lambda$. Carriers can arrive at $k$ different locations $x_1, ..., x_k$ (centroids of major cities) with respective probability $p_1, ..., p_k$. By assuming independent arrivals from one period to another, the arrival locations are i.i.d. discrete random variables and we denote $\mu$ their distribution (i.e. $X \sim \mu$ implies $\mathbb{P}(X = x_i) = p_i$). If a load is not delivered on its expiration date, a penalty $\xi_i$ is applied and the load can no longer be booked. Finally, our model assumes a population of homogeneous (identical preferences and independent) carriers making mutually exclusive choices from a set of bundles. We adopt the multinomial logit (MNL) model with known parameters $\beta$ (including the MNL constant $\beta_0$, distance sensitivity $\beta_d$, price sensitivity $\beta_p$, and 12 other parameters) to account for carrier choice preferences as available bundles vary and prices are adjusted. Thus, given the prices of suggested bundles, we can compute the probability that a carrier will accept a particular bundle.

Our goal is to minimize the freight costs of the platform over the delivery season formed by the operating costs (prices of accepted bundles) and penalties for failed deliveries. This is done by (1) selecting bundles out of the set of feasible bundles $\mathcal{B}$ containing all possible bundles satisfying time window and carrier idle time constraints; (2) dynamically pricing bundles at each carrier arrival. Single-load bundles (denoted by the set $\mathcal{B}_1$) are always suggested, and the number of additional bundles can be no more than $K$ to avoid unrealistic solutions that suggest a number of bundles too large for the carrier to consider all the options. In addition, the number of loads contained in any bundle cannot be larger than 3. Due to the intrinsic relationship between expected freight costs and expected empty miles resulting from the delivery of the loads, we can expect that the minimization of the former leads, to some extent, to a reduction in the latter. Still, we introduce later a refinement of the problem considering both expected freight cost and expected empty miles to further explore the trade-offs between these two quantities.

# 3 Formulation

The problem is defined as follows:

*State*: A state $s_t = (t, l, \tau, x)$ consists of the remaining loads $l \in \{0,1\}^N$ at time $t$, the remaining times before expiration $\tau \in [0,T]^N$, and a carrier location $x \in \{x_1, ..., x_k\}$. We denote the space of all states as $\mathcal{S}$.

*Decision*: Before the delivery seasons begins, a set of bundles $B \subseteq \mathcal{B}$ is selected. Let $n$ be the size of $B$. For each state $s_t \in \mathcal{S}$, the pricing decision is derived by a policy $\pi_B : \mathcal{S} \to \mathbb{R}^n$. It is associated with the probability vector $\rho(\pi_B(s_t)) \in [0,1]^n$ that a carrier accepts a bundle. As a special case, the policy pricing single load bundles only is denoted $\pi_S$ (specifically, the price of any bundle with more than one load is set to be $-\infty$ under policy $\pi_S$).

*Transition*: At period $t$, the $i$-th bundle $b_i$ is selected with probability $\rho_i(\pi_B(s_t))$. Thus the transition probability from state $s_t$ to state $s_{t+1}^{-b_i} = (t+1, l-b_i, \tau, x_j)$ is $\mathbb{P}(s_{t+1}|s_t; \pi_B) = \rho_i(\pi_B(s_t)) \cdot p_j$. Similarly, the transition where no bundle is selected, i.e. the transition from state $s_t$ to state $s_{t+1} = (t+1, l, \tau, x_j)$, happens with probability $\mathbb{P}(s_{t+1}|s_t; \pi_B) = (1 - \sum_{i=1}^n \rho_i(\pi_B(s_t))) \cdot p_j$.

*Cost function*: The cost function $C_t^{\pi_B} : \mathcal{S} \to \mathbb{R}$ assigns each state $s_t$ the expected cost-to-go under the pricing policy $\pi_B$. With probability $\rho_i(\pi_B(s_t))$, bundle $b_i$ is selected at price $\pi_B^i(s_t)$ and the next state is $s_{t+1}^{-b_i}$. Hence the cost function satisfies the following Bellman equation:

$$C_t^{\pi_B}(s_t) = \sum_{\{j: l_j=1,\ \tau_j \le t\}} \xi_j + \mathbb{E}_{s_{t+1}}\left[C_{t+1}^{\pi_B}(s_{t+1})\right] + \lambda \sum_{\{i\in[n]:\ B_i \le l\}} \rho_i(\pi_B(s_t))\left(\pi_B^i(s_t) - \Delta_i C_t^{\pi_B}(s_t)\right),$$

where $\Delta_i C_t^{\pi_B}(s_t) = \mathbb{E}_{s_{t+1}}\left[C_{t+1}^{\pi_B}(s_{t+1}) - C_{t+1}^{\pi_B}(s_{t+1}^{-i})\right]$ is the marginal cost of bundle $i$ in state $s_t$.

*Objective*: Minimize the expected freight cost over the season. The objective can be formulated as a bilevel optimization problem:

$$\min_{\substack{B\subseteq\mathcal{B}\\ \mathcal{B}_1\subseteq B,\ |B|\leq N+K}} \quad \min_{\pi_B} C_0^{\pi_B}(s_0)$$

The first stage selects bundles under bundle feasibility constraints (time window and carrier idle time) and the constraint on the maximum number of bundles. The second stage dynamically prices the bundles selected in the first stage.

# 4 Technical approach

The problem described above is a complex combinatorial problem as evaluating a candidate set of bundles requires the use of dynamic programming, making the problem hard to solve even for small instances. It is well known in revenue management [5] [6] that, under the multinomial logit model, the pricing problem of the form $\min_p \sum_i \rho_i(p)\,(p_i - \Delta_i)$ has a near-closed-form solution function of the Lambert $W$ function which in our case is (simplified version): $-\frac{1}{\beta_p} W\left(\sum_i e^{\beta_d d_i + \beta_p \Delta_i C_t^{\pi_B}(s_t) - \beta_0 - 1}\right)$ where $d_i$ is the distance to travel to deliver the loads contained in the $i$-th bundle. A key technical contribution is that we use this result to not only efficiently solve each step of the dynamic program, but also to provide a method for selecting bundles that minimize the expected cost: non-overlapping bundles that maximize $\beta_d d_i + \beta_p \Delta_i C_t^{\pi_B}(s_0)$. We approximate this quantity using policy $\pi_S$ as evaluating each policy is intractable. This leads us to develop the following weighted set packing algorithm selecting bundles of two or more loads to suggest in addition to single loads bundles:

$$\begin{aligned} \max_z \quad & \sum_{b\in\mathcal{B}\setminus\mathcal{B}_1} z_b \left[\beta_d d_b + \beta_p \Delta_b C_0^{\pi_S}(s_0)\right] \\ \text{s.t.} \quad & \sum_{b\in\mathcal{B}\setminus\mathcal{B}_1\,:\,l\in b} z_b \leq 1 \quad \forall l \in \mathcal{L} && (1) \\ & \sum_{b\in\mathcal{B}\setminus\mathcal{B}_1} z_b \leq K && (2) \\ & z_b \in \{0,1\} \quad \forall b \in \mathcal{B}\setminus\mathcal{B}_1 && (3) \end{aligned}$$

Variables $z_b$ are equal to 1 if bundle $b$ is selected, 0 otherwise. Constraint (1) ensures that the selected bundles are not overlapping (no load included in two different bundles). Constraint (2) ensures that no more than $K$ bundles are selected.

Dynamic pricing is handled differently depending on the size of instances. For medium-size instances, we use piecewise static prices (*Approx. DP*), a tractable approximation to the original pricing problem where the delivery season is divided into multiple time segments and each segment is given a fixed price. We show that optimal piecewise static prices can be computed by solving a series of convex problems. For the largest instances, we perform a one-step lookahead and estimate future costs based on average prices and times to expiration.

# 5 Results

Our results demonstrate that jointly recommending and pricing load bundles can reduce emissions (or equivalently, empty miles) while minimizing freight costs. The results are divided into two categories: results obtained on small-scale synthetical randomized scenarios, and results obtained on a real case scenario in the Texas triangle. Small-scale scenarios can be solved to optimality using exact dynamic pricing methods, whereas approximate dynamic pricing is necessary for real-case scenarios.

Figure 1 shows the effects of bundling in terms of expected cost and expected distance for small-scale scenarios. We compare the sets B1, B2, and B3 (sets of all bundles containing respectively no more than 1, 2, and 3 loads) with the set $P$ (set of bundles selected by the weighted set packing algorithm in addition to single loads). The right extremity of the curves represents the nominal case where emissions are not penalized. Increased weight of empty miles in the objective function leads to higher costs and lower empty miles (thus lower emissions), with a better tradeoff when bundles are suggested ("flatter curve", but quick saturation).

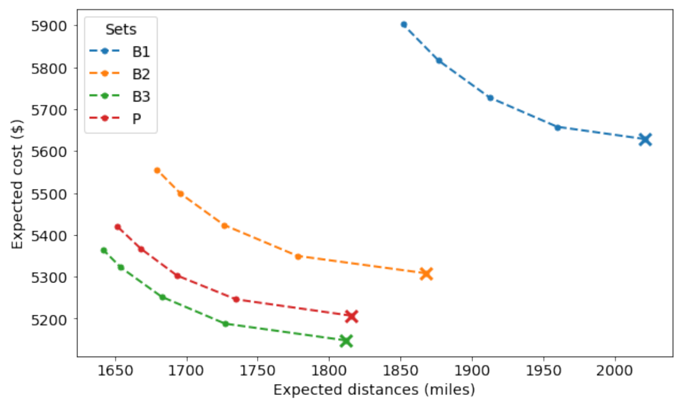



Figure 1: Costs/emissions tradeoff for different bundling strategies

The real-case scenario consists of two months of historical data provided by our industry partner. This data includes loads, carrier location distribution, number of arrivals, price ranges, as well as the parameters of the MNL defining the behavior of carriers. We compare 3 bundling methods: *Single* suggests single loads only, *Pairwise* suggests all pairwise bundles in addition to single loads, and *Packing* suggests the bundles selected by our weighted set packing algorithm in addition to single loads - and 4 pricing methods: *Low*, *Avg.*, *High* static prices, and *Approx. DP*. The results are summarized in the table below:

| | Total costs | | | Percentage of booked loads | | |
|---|---|---|---|---|---|---|
| | Single | Pairwise | Packing | Single | Pairwise | Packing |
| Low | 46507 | 45582 | 45410 | 71.66 | 73.70 | 73.76 |
| Avg. | 46465 | 44760 | 44541 | 81.32 | 85.36 | 86.86 |
| High | 48673 | 47430 | 46963 | 91.06 | 95.16 | 97.06 |
| Approx. DP | 39500 | 38629 | 37222 | 89.46 | 92.22 | 97.92 |

# References


[1] Wallace J. Hopp, Xiaowei Xu, (2008) A Static Approximation for Dynamic Demand Substitution with Applications in a Competitive Market. Operations Research 56(3):630-645.

[2] Awi Federgruen, Daniel Guetta, Garud Iyengar, Xujia Liu, (2023) Scalable Approximately Optimal Policies for Multi-Item Stochastic Inventory Problems.

[3] Markus Ettl, Pavithra Harsha, Anna Papush, Georgia Perakis (2020) A Data-Driven Approach to Personalized Bundle Pricing and Recommendation. Manufacturing & Service Operations Management 22(3):461-480.

[4] Kilian Heilmann, (2020) Information Frictions, Load Matching, and Route Efficiency in the Trucking Industry.

[5] Hongmin Li, Woonghee Tim Huh, (2011) Pricing Multiple Products with the Multinomial Logit and Nested Logit Models: Concavity and Implications. Manufacturing & Service Operations Management 13(4):549-563.

[6] Lingxiu Dong, Panos Kouvelis, Zhongjun Tian, (2008) Dynamic Pricing and Inventory Control of Substitute Products. Manufacturing & Service Operations Management 11(2):317-339.

# The Workforce Scheduling and Routing Problem with Park-and-loop

**Nicolás Cabrera**
**Jean-François Cordeau**
**Jorge E. Mendoza**
Department of Logistics and Operations Management
HEC Montréal
Email: nicolas.cabrera-malik@hec.ca

## 1 Introduction

Inspired by a real-world problem faced by a service company operating in France, we study the workforce scheduling and routing problem with park-and-loop (WSRP-PL). Every day, this company performs on-site tasks (e.g., connection to the electricity grid, troubleshooting, meter reading) which vary in difficulty. Each task has an associated duration and an associated time window indicating possible visit times. In addition, depending on the nature of the task, it may require one or more skills at potentially different levels of proficiency. Accordingly, the company usually forms teams of workers and assigns them tasks. Unfulfilled tasks are outsourced to a third party. Each team departs from and returns to one of the company's facilities within working hours using a vehicle. However, in most cases, customers are located in densely populated areas, in which access to parking may be limited. Thus, workers can walk between nearby locations. While considering these features, the company develops daily plans with the objective of minimizing the total cost of the operations, which is composed of the outsourcing cost of unfulfilled tasks and the variable cost associated with the total driving distance by all the teams assembled. A plan is composed of the assignment of workers to teams and the routing of the vehicles that the teams use. The resulting problem is closely related to two types of problems, namely, the park-and-loop routing problem (PLRP) and the workforce scheduling and routing problem (WSRP) which are both NP-Hard.

The WSRP-PL can be formally defined on a complete and directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of directed arcs. The set of nodes comprises a start depot $\overline{0}$, an end depot $\underline{0}$, and the set of tasks $\mathcal{C} = \{1, ..., n\}$. Note that, $\overline{0}$ and $\underline{0}$ can represent the same or distinct geographical locations. Arcs in $\mathcal{A}$ represent the connections between two tasks or between a task and the depot. Each arc $(i, j) \in \mathcal{A}$ has four attributes:

the driving distance $\mu_{ij}$, the driving time $\tau_{ij}$, the walking distance $\delta_{ij}$, and the walking time $\eta_{ij}$. To perform the tasks, a set of workers $\mathcal{W} = \{1, ..., m\}$ are assigned to teams in the set $\mathcal{T}$. A maximum of $\omega$ workers can be assigned to a team. Each team $t \in \mathcal{T}$ departs and arrives at the depot after performing its route. Each route has a maximum duration $\phi$. The maximum distance that can be traveled on foot between two points is $\theta$. Moreover, the maximum distance that can be traveled by a team on foot in one day is $\zeta$. Driving the car involves a variable cost $c^v$ per unit of distance.

In addition to the above, each task $i \in \mathcal{C}$ has a duration $s_i$ and a time window. Let $[a_i, b_i]$ be the earliest and latest starting time of task $i \in \mathcal{C}$. Also, let $f_i$ be the outsourcing cost of task $i \in \mathcal{C}$. Skill requirements are represented by $\nu_{iql}$, an integer parameter stating the number of workers with the skill $q \in \mathcal{Q}$ with at least a proficiency level $l \in \mathcal{L}$ that the task $i \in \mathcal{C}$ needs. Worker qualifications are represented by $\xi_{kql}$, which is a binary parameter equal to 1 if worker $k$ has at least a proficiency level $l$ for skill $q$. The objective of the WSRP-PL is to minimize the total cost while ensuring that: each task is fulfilled precisely once; the total duration of all routes performed by any team does not exceed the working day duration; and the total walking distance by each team does not exceed the distance limit.

## 2 Solution method

To solve the WSRP-PL we introduce a compact arc-based formulation as well as a path-based formulation with an exponential number of variables. To efficiently solve the latter, we developed a branch-price-and-cut algorithm (BPC). A BPC algorithm is a branch-and-bound algorithm in which at each node of the tree, the linear relaxation (RMP) of an integer formulation is solved using column generation and tightened by adding valid inequalities (i.e., cuts). In brief, our BPC algorithm works as follows:

*Step 1:* Generate an initial solution using a constructive heuristic.

*Step 2:* Select an unprocessed node in the branch-and-bound tree. If the lower bound is greater than or equal to the (global) upper bound, prune the node.

*Step 3:* Solve the RMP. If it is infeasible, prune the node and go back to *Step 2*.

*Step 4:* Solve the pricing problem using the pulse algorithm [3]. If routes with negative reduced cost are found, add them to the pool of routes and go back to *Step 3*.

*Step 5:* Solve the separation procedure. If any cut is separated, go back to *Step 3*.

*Step 6:* If the solution is fractional, mark the node as processed, branch, add the two child nodes to the set of unprocessed nodes in the branch and bound tree, and go back to *Step 2*. If the solution is integer, update the upper bound (if possible) and prune the node.

A key component of our procedure is the pulse algorithm used to solve the pricing problem (*Step 4*). The latter extends the procedure introduced in [1] to handle the presence of time windows and the skills compatibility between teams and tasks. To further improve its performance, we implemented a multi-thread version of the pulse algorithm which allows for testing different team configurations almost simultaneously. We also modified the pre-processing step proposed by [3] to significantly increase the strength of the resulting lower bounds.

## 3 Results

To analyze the performance of our BPC algorithm, we created a set of instances based on the testbed designed by [2]. All the experiments were conducted on the Beluga cluster of the Digital Research Alliance of Canada using eight threads and 20GB of RAM in a Linux environment. The time limit for all the experiments is 2 hours. Table 1 presents detailed results of the comparison between the BPC algorithm and the arc-based formulation (AF) using the new set of instances and setting the maximum number of workers in a team to 2. Column 1 denotes the set of workers (`E`), namely, complete (C) and reduced (R). Column 2 provides information regarding the length of the planning horizon (`T`) and the percentage of tasks with time windows (`m`). Two types of planning horizons (`T`) were considered, namely, short (1) and long (2). In addition, (`m`) can take two values, where 01 is used to indicate 100% and 03 to specify 50%. Column 3 gives the number of tasks. The remaining columns give the number of optimal solutions found, the average optimality gap, and the computational time in seconds used by each algorithm.

Table 1 shows that BPC can solve 140 out of 162 instances to optimality, while the arc-based formulation can only solve 59. Remarkably, BPC can solve all the instances with 25 tasks to optimality. As expected, instances in which the percentage of time windows is lower (i.e., `Tm` = 103) are harder to solve. This is especially the case for the arc-based formulation, as it can only solve 1 out of 54 instances with this setting. A similar argument can be used for the subset of instances with wider time windows (i.e., `Tm` = 201). As routes are longer, BPC decreases its performance. With regard to the number of workers available, it seems that AF can better handle instances with a lower number of workers (i.e., `E` = R), as it can solve 15 more instances (37 vs 22) compared with the subset of instances with the complete set of workers. Concerning the computational times, on average BPC takes 1568.83 seconds to solve the WSRP-PL, while AF takes 4680.01 seconds. BPC is particularly fast on the subset of instances with 25 customers.

We also tested our method on the service technician routing and scheduling problem

Table 1: Comparison of arc-based formulation (AF) vs branch-price-and-cut (BPC) on the WSRP-PL instances with $\omega = 2$.

| E | Tm | n | AF | | | BPC | | |
|---|---|---|---|---|---|---|---|---|
| | | | #Opt. | Avg. Δ | CPU (s) | #Opt. | Avg. Δ | CPU (s) |
| C | 101 | 25 | 7/9 | 3.63% | 1122.25 | 9/9 | 0.0% | 293.01 |
| | 103 | 25 | 0/9 | 27.94% | 7200.00 | 9/9 | 0.0% | 254.34 |
| | 201 | 25 | 6/9 | 5.54% | 2469.41 | 9/9 | 0.0% | 204.36 |
| | 101 | 50 | 4/9 | 12.20% | 3792.39 | 8/9 | 0.53% | 1222.96 |
| | 103 | 50 | 0/9 | 18.97% | 7200.00 | 7/9 | 12.69% | 2115.40 |
| | 201 | 50 | 2/9 | 18.53% | 5911.66 | 9/9 | 0.00% | 1188.62 |
| | 101 | 75 | 2/9 | 31.13% | 6126.59 | 9/9 | 0.00% | 1784.40 |
| | 103 | 75 | 0/9 | 54.91% | 7200.00 | 6/9 | 13.64% | 3618.37 |
| | 201 | 75 | 1/9 | 32.69% | 6411.02 | 5/9 | 22.83% | 3788.10 |
| R | 101 | 25 | 9/9 | 0.00% | 199.75 | 9/9 | 0.00% | 76.25 |
| | 103 | 25 | 1/9 | 8.88% | 6859.72 | 9/9 | 0.00% | 834.51 |
| | 201 | 25 | 9/9 | 0.01% | 244.87 | 9/9 | 0.00% | 172.01 |
| | 101 | 50 | 5/9 | 0.08% | 3134.59 | 9/9 | 0.00% | 498.68 |
| | 103 | 50 | 0/9 | 9.18% | 7200.00 | 7/9 | 2.27% | 2399.45 |
| | 201 | 50 | 7/9 | 2.95% | 2161.92 | 7/9 | 5.04% | 1628.97 |
| | 101 | 75 | 4/9 | 21.81% | 4172.81 | 7/9 | 0.23% | 1825.47 |
| | 103 | 75 | 0/9 | 21.34% | 7200.00 | 5/9 | 13.04% | 3999.66 |
| | 201 | 75 | 2/9 | 0.63% | 5633.11 | 7/9 | 22.22% | 2334.37 |
| Total/Avg. | | | 59/162 | 15.02% | 4680.01 | 140/162 | 5.14% | 1568.83 |

without team building (STRSP) introduced in [2]. In this variant of the WSRP, routes are carried out by each worker individually. Due to its complexity, most of the work on the STRSP has focused on heuristic algorithms. For this problem, our algorithm produced 12 new best known solutions and is the first to prove optimality for 24 out of 54 instances.

## References


[1] Cabrera, N., Cordeau, J. F., and Mendoza, J. E. Solving the Park-and-loop Routing Problem by Branch-price-and-cut. *Transportation Research Part C: Emerging Technologies*, 157, 104369. 2023.

[2] Kovacs, A. A., Parragh, S. N., Doerner, K. F., and Hartl, R. F. Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of scheduling*, 15, 579-600. 2012.

[3] Lozano, L., Duque, D., and Medaglia, A. L. An exact algorithm for the elementary shortest path problem with resource constraints. *Transportation Science*, 50(1), 348-357. 2016

# A Metaheuristic Approach for the Dynamic Berth Allocation Problem

**Letícia Caldas dos Santos**[1]
Email: leticia.caldas@aluno.puc-rio.br

**Rafael Martinelli**[1]
Email: marinelli@puc-rio.br
[1]Departamento de Engenharia Industrial,
Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil

**Michel Gamache**[2]
Email: michel.gamache@polymtl.ca
[2]Département de Mathématiques et de Génie Industriel
Polytechnique de Montréal, Montréal, Canada

## 1 Introduction

Berths are scarce resources in port operations. Optimizing the use of this resource is a determining factor in handling the growing volume of bulk transportation. Thus, using reliable and efficient methods and approaches provides managers with solutions to allocate ships to berths while minimizing queuing time.

The static berth allocation problem in the literature aims to schedule and allocate ships optimally, available at the beginning of the planning horizon, to port terminal positions. In contrast, the vessels arrive during the planning horizon in the dynamic berth allocation problem (DBAP). To solve the problem, we propose a metaheuristic approach based on the Iterated Local Search for a DBAP. We consider a discrete case with a finite set of berthing locations and fixed length and compare our results with benchmark instances.

For these instances, the best results in the literature are reported in works [1], [2], [3], [4] and [5]. The main works considered tabu search combinations [2], algorithms like ALNS heuristic [4] and Iterated Greedy [5] with destroy and repair phases.

## 2 Iterated Local Search

Iterated Local Search (ILS) uses local search to explore the local minima of a function and, at each iteration, generates a perturbation to the previously visited locally optimal solution [6].

For the constructive, vessels are ordered by the beginning of the service time window availability. Each of them goes to the earliest berth they can be allocated to. Thus, the solution structure consists of a vector of ships for each berth, representing a queue.

The local search used classic neighborhoods, such as Shift, Relocate, Swap, Swap2-2, and Swap2-1. As an acceptance criterion, simulated annealing was used, where the current solution is replaced by the candidate solution given a certain probability.

Finally, for the perturbation, the destroy and repair strategy was used. A parameter determined the strength of the perturbation, indicating the percentage of ships that would be randomly removed from the solution. Then, a greedy algorithm allocated the ships removed from the solution to the best possible position.

## 3 Results and Discussion

This section presents the computational experiments carried out to assess the performance of the proposed method. The metaheuristic was coded in Julia 1.9 and solved on a computer with an Intel Core i7-8700K CPU @ 3.70GHz and 64 GB of RAM, running Ubuntu Linux in a single thread.

### 3.1 Benchmark Instances

In this work, we use the problem instances proposed in the literature by [2] and [1]. The set proposed by [2] contains 90 instances with up to 60 vessels and seven berths, and the set proposed by [1] contains 20 instances having up to 250 vessels and 20 berths.

### 3.2 Results

In this section, we report and discuss the results of the proposed metaheuristic (see Section 2) and compare them with results from the literature. For the instances proposed by [2], we present in Table 1 a summary of the results for each size of the 90 instances. Column n presents the number of ships, and column m the number of berths. The Average Gap Literature column presents the average gap between the best result reported in the literature and the optimal solution. Finally, in the Average Gap ILS column, there is the average gap between the results of our method and the optimal solution.

For the instances proposed by [1], we present the results of the 20 instances in Table 2. Column # presents the instance number. The LB and UP report the best lower and upper bounds found for each instance, considering all mathematical programming approaches presented in [1]. Literature MH column presents the results obtained by the metaheuristic by [3] and the gap between the result reported and the upper bound. Finally, we present our results with the ILS method and compare the gap with the upper bound.

Table 1: Results for the small instances set

| n | m | Average Gap Literature | Average Gap ILS |
|---|---|---|---|
| 30 | 3 | 0.00% | 0.35% |
| 30 | 5 | 0.00% | 0.00% |
| 40 | 5 | 0.00% | 0.19% |
| 40 | 7 | 0.00% | 0.01% |
| 55 | 5 | 0.00% | 0.00% |
| 55 | 7 | 0.00% | 0.00% |
| 55 | 10 | 0.00% | 0.00% |
| 60 | 5 | 0.00% | 0.00% |
| 60 | 7 | 0.00% | 0.01% |

Table 2: Results for the large instances set

| n | m | # | LB | UB | Literature MH | Gap | ILS | Gap |
|---|---|---|---|---|---|---|---|---|
| 200 | 15 | 01 | 12604 | 12609 | 12709 | 0.79% | 12669 | 0.48% |
| 200 | 15 | 02 | **10319** | **10319** | 10407 | 0.85% | 10376 | 0.55% |
| 200 | 15 | 03 | 11296 | 11355 | 11558 | 1.79% | 11407 | 0.46% |
| 200 | 15 | 04 | **15441** | **15441** | 15647 | 1.33% | 15573 | 0.85% |
| 200 | 15 | 05 | 18166 | 18352 | 18352 | 0.00% | 18218 | -0.73% |
| 200 | 15 | 06 | **16869** | **16869** | 16961 | 0.55% | 16942 | 0.43% |
| 200 | 15 | 07 | 13025 | 13226 | 13226 | 0.00% | 13114 | -0.85% |
| 200 | 15 | 08 | 14182 | 14259 | 14537 | 1.95% | 14317 | 0.41% |
| 200 | 15 | 09 | **18118** | **18118** | 18198 | 0.44% | 18164 | 0.25% |
| 200 | 15 | 10 | 17102 | 17118 | 17263 | 0.85% | 17191 | 0.43% |
| 250 | 20 | 01 | 15633 | 15769 | 15769 | 0.00% | 15743 | -0.16% |
| 250 | 20 | 02 | 15776 | 15915 | 15915 | 0.00% | 15903 | -0.08% |
| 250 | 20 | 03 | 16519 | 16606 | 16724 | 0.71% | 16669 | 0.38% |
| 250 | 20 | 04 | 16423 | 16481 | 16509 | 0.17% | 16505 | 0.15% |
| 250 | 20 | 05 | 15661 | 15837 | 15837 | 0.00% | 15778 | -0.37% |
| 250 | 20 | 06 | **20060** | **20060** | 20193 | 0.66% | 20145 | 0.42% |
| 250 | 20 | 07 | 14284 | 14362 | 14514 | 1.06% | 14446 | 0.58% |
| 250 | 20 | 08 | 16305 | 16383 | 16498 | 0.70% | 16428 | 0.27% |
| 250 | 20 | 09 | 15864 | 15917 | 16121 | 1.28% | 16039 | 0.77% |
| 250 | 20 | 10 | 16283 | 16371 | 16428 | 0.35% | 16388 | 0.10% |

### 3.3 Conclusions

In this work, we considered the dynamic berth allocation problem (DBAP) using a metaheuristic as a solution method. Computational experiments on benchmark instances were performed to evaluate the results of the proposed method and compare them with the best results of the literature. It was observed that for instances that seek to get closer to a real scenario, dealing with a number of ships larger than 200, the proposed metaheuristic improved the results for all instances, and in five cases, it achieved better results than the upper bound reported in the literature.

It is essential to highlight that all results were obtained in less than four minutes of method execution, showing the efficiency and potential of the method.

## References

[1] A. Kramer, E. Lalla-Ruiz, M. Iori, and S. Voß, "Novel formulations and modeling enhancements for the dynamic berth allocation problem," *European journal of operational research*, vol. 278, no. 1, pp. 170–185, 2019.

[2] E. Lalla-Ruiz, B. Melián-Batista, and J. M. Moreno-Vega, "Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 6, pp. 1132–1141, 2012.

[3] E. Lalla-Ruiz, B. Melián-Batista, and J. M. Moreno-Vega, "A cooperative search for berth scheduling," *The Knowledge Engineering Review*, vol. 31, no. 5, pp. 498–507, 2016.

[4] G. R. Mauri, G. M. Ribeiro, L. A. N. Lorena, and G. Laporte, "An adaptive large neighborhood search for the discrete and continuous berth allocation problem," *Computers & Operations Research*, vol. 70, pp. 140–154, 2016.

[5] S.-W. Lin, K.-C. Ying, S.-Y. Wan, *et al.*, "Minimizing the total service time of discrete dynamic berth allocation problem by an iterated greedy heuristic," *The Scientific World Journal*, vol. 2014, 2014.

[6] H. R. Lourenco, O. Martin, and T. Stutzle, "Iterated local search. handbook of metaheuristics. f. glover and g. kochenberger," 2003.

# Synchronized Deliveries with a Bike and a Self-Driving Robot

**Diego Cattaruzza**
Univ. Lille, CNRS, Inria UMR 9189 - CRIStAL
Centrale Lille, Lille, France
Email: diego.cattaruzza@centralelille.fr

**Yanlu Zhao**
Durham University Business School,
Durham University, Durham, UK

**Ningxuan Kang**
Department of Intelligent Supply Chain Y
JD.com, Beijing, China

**Roberto Roberti**
Department of Information Engineering
University of Padova, Padova, Italy

## 1 Introduction

The growing number of Internet users and the opportunity to easily order their preferred products through mobile devices, such as tablets and phones, are boosting e-commerce sales. Indeed, online retailing sales in the three major e-commerce global markets (i.e., the U.S., China, and Europe) are estimated to increase at an annual rate of over 6% in the next three years [1]. Moreover, a recent survey about customers' expectations on e-commerce markets [2] indicates that almost half of online shoppers expect to benefit from speedy deliveries and delivery cost reductions in the coming years. The growth rate of e-commerce sales combined with such high customer expectations calls for smarter ways to run e-commerce businesses.

Planning last-mile delivery operations is one of the most crucial tasks faced by e-commerce companies [3]. If last-mile delivery is carefully designed, not only can customers' expectations be fulfilled, but also important related issues such as urban congestion and toxic emissions can be addressed. However, smart planning of last-mile delivery operations is particularly challenging because e-commerce companies can rely on razor-thin marginal profits.

To tackle all these challenges posed by e-commerce markets, e-commerce giants, such as Amazon or JD.com, are investigating new paradigms to run last-mile deliveries. One of the most promising ways to improve the current practices in last-mile delivery is to adopt *Unmanned Autonomous Vehicles* (UAV), such as drones or *Self-Driving Robots* (SDR), to complement or replace conventional vehicles. Deliveries with UAVs can allow companies to decrease delivery costs and represent an environmentally-friendly transport mode.

In this paper, we investigate a last-mile delivery problem faced by JD.com, which is the largest (in terms of revenue) Chinese business-to-consumer online retailer company, headquartered in Beijing, with about 550,000 employees (as of 2022) and a total annual revenue of about 151.7 billion dollars in 2022. Given the lack of professional truck drivers and strict regulations, imposed by local authorities, to reduce noise and carbon emissions in urban areas, JD.com is already using a mixed delivery force of conventional vehicles and cargo bikes (see the left panel of Figure 1) in last-mile delivery operations. JD.com is also considering the adoption of SDRs (such as the one displayed in the right panel of Figure 1) to use in combination with cargo bikes (which we refer to as bikes in the following) to deliver parcels to customers.

## 2 Problem definition

JD.com would like to gain insights on the economic feasibility and overall impact of associating a robot with a bike in the following setting. A set of customers must be served with a bike and its robot within a given planning horizon. At the beginning of the planning horizon, the bike and its robot are located at a depot, where all parcels to deliver are also stored. The bike has enough capacity to store all these parcels at once whereas the robot has limited capacity: it can carry just a subset of the parcels at the same time because it features a limited number of containers (of various volumes) to store the parcels. The customers can be served by one of the two vehicles that travel through two different routes, which start and end at the depot. Whenever the robot is empty along its route, it can join the bike at a customer, where some parcels that are on the bike can be moved to the robot for further deliveries. The costs to serve a customer with the bike or the robot are known. The goal of the problem is to find a distribution plan for the bike and the robot so that all customers are served within the given planning horizon and the total distribution costs are minimized. In the following, we refer to this decision-making problem as the *Traveling Salesman Problem with Bike-and-Robot* (TSPBR), which, to the best of our knowledge, has not been studied in the literature so far ([4]).

## 3 Scientific contributions

The main contributions of this work are the following:

Figure 1: A cargo bike (in the left panel) and self-driving robots (in the right panel)

- We formally describe and formulate the TSPBR with two *Mixed-Integer Linear Programming* (MILP) models. The first model is a compact MILP featuring a polynomial number of variables and constraints. The second MILP builds upon the first model but features an exponential number of constraints. We also propose a set of valid inequalities to tighten the linear relaxation of these MILP models and embed these cuts into branch-and-cut algorithms.

- We present a genetic algorithm, based on dynamic programming recursions to explore a large neighborhood of TSPBR solutions, to find high-quality primal solutions to the problem.

- We test the proposed branch-and-cut and the genetic algorithms on real-life instances provided by JD.com. We show that the branch-and-cut methods can find optimal solutions for most of the TSPBR instances with up to 60 nodes and the genetic algorithm can find high-quality solutions in a few minutes of computing time.

- We assess the economic impact of performing last-mile deliveries with the bike and the robot operating in tandem. We show that deploying the robot can attain significant cost reductions and time savings to fulfill all customer requests.

## 4 Computational results

The company that inspired our study on the TSPBR (i.e., JD.com) has provided us with a data set of information that has allowed us to generate a set of realistic instances to test our algorithms. In particular, they have provided us with information about the customers and their orders, the robot and its containers, and the delivery costs. This information has allowed us to generate 900 instances that will act as a set of benchmark instances for the TSPBR.

We have tested the formulations and the genetic algorithm on the set of benchmark instances based on real data provided by JD.com. The formulations can often find optimal solutions for instances with up to 60 nodes in some ten minutes of computing time. The genetic algorithm can yield high-quality solutions, within $\sim 3\%$ to optimality, in a few minutes of computing time. Finally, the analysis of the computational results show that significant cost savings can be achieved by deploying the robot along with the bike rather than assigning all deliveries to the bike alone.

## 5 Conclusion

Motivated by the challenges faced by the Chinese e-commerce giant JD.com in last-mile delivery, we have addressed a delivery problem where a bike and a self-driving robot work in tandem to deliver parcels to customers in urban areas. We called this new problem the Traveling Salesman Problem with Bike-and-Robot (TSPBR). The decisions entailed by the TSPBR are (*i*) partitioning the customers to serve between the bike and the robot, (*ii*) deciding upon the locations where the two vehicles synchronize so that the robot is replenished, and (*iii*) routing the two vehicles. The main challenge in mathematically modeling the TSPBR is the needed spatial and temporal synchronization of the two vehicles. We have introduced two mixed-integer linear programming formulations as well as a genetic algorithm to tackle the TSPBR. Computational results showed the efficiency of the proposed methods.

## References


[1] Statista, *eCommerce Report 2021* `https://www.statista.com/study/42335/ecommerce-report/`, 2021, Accessed on November 21st, 2022.

[2] Statista, *Leading Aspects that Global Shoppers Would Change about the Delivery of Products Purchased Online as of April 2021*, `https://www.statista.com/statistics/1274950/global-online-shoppers-wished-changes-on-delivery/`, 2021, Accessed on November 21st, 2022.

[3] C. Archetti, L. Bertazzi, *Recent Challenges in Routing and Inventory Routing: E-commerce and Last-mile Delivery*, Networks, 77(2), pp. 255–268, 2021.

[4] S. Srinivas, S. Ramachandiran, and S. Rajendran, *Autonomous Robot-driven Deliveries: A Review of Recent Developments and Future Directions*, Transportation Research Part E: Logistics and Transportation Review, 165, 2022.

# Maximizing Electrifiability of Commercial Fleets Through Optimization

C. Ceccherelli[1,2], L. Bravi[1], S. Caprasecca[1], L. Taccari[1], D. Di Lorenzo[1], F. Schoen[2]
[1]Verizon Connect, Florence, Italy
[2]University of Florence, Italy
Email: carlo.ceccherelli@edu.unifi.it, luca.bravi@verizonconnect.com

## 1 Introduction

In a world increasingly concerned with the impact of climate change and growing environmental issues, the need for sustainable transportation systems has never been more evident. As a result, major companies in the transportation industry have recognized the potential of Electric Vehicles (EVs) over Internal Combustion engine Vehicles (ICVs) and have set the goal of electrifying at least part of their fleet.

Works like [1] and [2] focus on Electric Vehicle Routing Problems (EVRP), trying to determine the optimal fleet composition while considering time windows, carbon emission cap constraints and partial recharge possibilities, exploiting metaheuristics like Ant Colony Optimization and Advanced Large Neighborhood Search. Other works such as [3] and [4] focus more on a scheduling problem using genetic algorithms and time based decomposition schemes as solution methods.

Fleet managers are often assisted by fleet management software, like Verizon Connect Reveal, which collect a huge amount of data on vehicle operations and provide them useful decision support tools. One important feature of these tools is recommending which vehicles of their fleet can be converted to EV. This is typically done using very simple statistics, such as the average travel time of a vehicle, leading to suboptimal solutions.

The objective of this work is to propose an optimization-based procedure capable of identifying the maximum number of vehicles which can be profitably converted into electric ones, without affecting the fleet's operations, and allowing for control over charging infrastructure availability. We first developed a pre-processing pipeline to extract stops, depots and trips from historical data, then we addressed the optimization problem, starting with a general model considering an entire week of data, and then exploring alternative methods to find good solutions more efficiently.

## 2 Method

We built a pipeline that leveraging historical data from a commercial vehicle fleet (GPS coordinates with a timestamp and possibly an event code describing vehicle status, e.g. engine on/off), detects fleet depots (by means of a DBSCAN clustering over the vehicle stops), maps each vehicle to its depot (which is the starting and ending point of each working day) and extracts the daily trips for each vehicle. We analyzed 4 weeks and 22 different depots with an average of 40 associated vehicles (from 22 to 94). Thanks to the fact that, in our data, vehicles only perform one daily trip starting and ending at the same depot, we use the extracted information to formulate separately for each depot an optimization problem where the objective is to determine the maximum number of vehicles that can be converted to electric, reassigning trips among the EV and ICVs under the constraints given by the residual battery charge from the previous day. We allow only for overnight recharging, and we consider the constraints arising by the limited availability of charging infrastructure in the depots.

### 2.1 Optimization models

We first designed a Mixed Integer Linear Programming model (M1W) to assign trips to vehicles during one week along with managing recharging decisions to be taken between consecutive days. We define variables to: decide if a vehicle is electric, assign trips to vehicles, decide if a vehicle recharges between days, and model the battery state of charge. The objective is to maximize the number of electrifiable vehicles, subject to the following constraints: a vehicle can perform at most one trip per day; every trip of a given day is covered by one vehicle; the number of recharges between two consecutive days is at most $C$ (limited charging infrastructure); EVs must have enough available range to cover the trip assigned to them. Setting a CPU time limit of 2 hours, only a fraction of instances could be solved to optimality; this led us to search for a more efficient approach.

Thus, we set up an iterative procedure to find the optimal EV/ICV split, that starts by setting the number $b$ of ICVs to its lower bound given by the maximum number of daily trips longer than $R$ (EV range at maximum charge) over the week - trips that necessarily need to be covered with an ICV. We check if the current EV/ICVs split yields a feasible solution to the weekly problem; otherwise, we increase $b$ until it does. Given that, even after fixing the number of EVs and ICVs, the weekly model is still challenging, we build a model M2DS that tries to assign trips to vehicles considering a 2 days period, and use it in a rolling-window fashion. EV ranges at the start of a given day are initialized depending on assignments and recharges made by the previous run. The objective is set to maximize the residual range at the start of the second day of assignments to better drive recharging decisions throughout the chain of runs. The M2DS showed below uses two sets $E$ and $B$ representing respectively electric and conventional vehicles; $J_1$ and $J_2$ include trips belonging to the two days under analysis; parameters $c_j$ and $r_e$ represent respectively the length of the trip and the range of the EV at the start of the first day. The decision variables $res_e$ represent the EV's residual range at the start of the second day; $x_{e,j}$ and $y_{b,j}$ equals 1 when the vehicle (EV or ICV respectively) is assigned to the trip; $z_e$ equals 1 if the vehicle is scheduled for recharging at the end of the first day.

$$\max \quad \sum_{e\in E} res_e \tag{1a}$$

$$\text{s.t.} \quad \sum_{j\in J_1} y_{b,j} \leq 1 \quad \forall b, \quad \sum_{j\in J_2} y_{b,j} \leq 1 \quad \forall b, \tag{1b}$$

$$\sum_{j\in J_1} x_{e,j} \leq 1 \quad \forall e, \quad \sum_{j\in J_2} x_{e,j} \leq 1 \quad \forall e, \tag{1c}$$

$$\sum_{b\in B} y_{b,j} + \sum_{e\in E} x_{e,j} = 1 \quad \forall j \in J_1, \forall j \in J_2, \tag{1d}$$

$$\sum_{e\in E} z_e \leq C, \tag{1e}$$

$$c_j x_{e,j} \leq r_e \quad \forall e, j \in J_1, \quad c_j x_{e,j} \leq R \quad \forall e, j \in J_2, \tag{1f}$$

$$\sum_{j\in J_1} c_j x_{e,j} + \sum_{j\in J_2} c_j x_{e,j} \leq r_e + R \cdot z_e \quad \forall e, \tag{1g}$$

$$res_e \leq R \cdot z_e + r_e - \sum_{j\in J_1} c_j x_{e,j} \quad \forall e, \quad res_e \leq R \quad \forall e, \tag{1h}$$

$$x, y, z \in \{0,1\}, \quad res_e \geq 0 \quad \forall e \tag{1i}$$

The objective (1a) is to maximize the residual range at the beginning of the 2nd day; constraints (1b) and (1c) state that a conventional or electric vehicle can perform at most one trip per day; constraints (1d) ensure that every trip of a given day is covered by one vehicle (EV or ICV); constraint (1e) limits the number of recharges that can be done between the two consecutive days; constraints (1f) and (1g) ensure that a trip on the first day can be assigned to an EV only if its cost is within the starting range $r_e$, while trip assignments on the second day also need to consider whether the vehicle was recharged; constraints (1h) define the residual range at the start of the second day.

## 3 Results and Conclusion

All the experiments were executed on a laptop with a core i5-1135G7 and 16GB of RAM, using a state of the art MILP solver. In Fig. 1 we compare execution of M1W and M2DS presented above, together with results yielded by other two models M1D and M2D obtained as rolling-window variants of M1W; the first one makes assignments on a single day, recharging the $C$ most discharged vehicles between runs, while the latter makes assignments on a two days period but with a myopic recharging strategy. We can see how the M2DS yields the best results within reasonable times, which is important in business contexts where there could be thousands of vehicles distributed among hundreds of depots.

In conclusion, we studied methods to solve the problem of determining the fleet composition with most EVs by assigning trips over an entire week. Our results show that our proposed iterative procedure, that divides a general formulation into smaller 2-days problems, proves to be significantly more efficient. The next steps of this work could include a robustness analysis on costs and number of trips, or the relaxation of the assumption constraining vehicles to perform at most one trip per day.

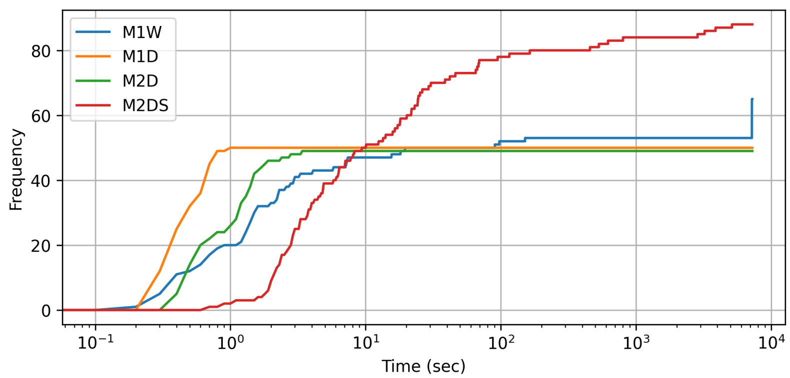



Figure 1: **Performance profile**: shows the number of instances (y axis) where each method reached the best known solution within a certain time (x axis)

## References


[1] M. A. Islam and Y. Gajpal, “Optimization of Conventional and Green Vehicles Composition under Carbon Emission Cap,” *Sustainability*, vol. 13, no. 12, p. 6940, Jan. 2021.

[2] S. Dönmez, Ç. Koç, and F. Altıparmak, “The mixed fleet vehicle routing problem with partial recharging by multiple chargers: Mathematical model and adaptive large neighborhood search,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 167, p. 102 917, 2022.

[3] M. Rogge, E. Van der Hurk, A. Larsen, and D. U. Sauer, “Electric bus fleet size and mix problem with optimization of charging infrastructure,” *Applied Energy*, vol. 211, pp. 282–295, 2018.

[4] M. Rinaldi, E. Picarelli, A. D’Ariano, and F. Viti, “Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications,” *Omega*, vol. 96, p. 102 070, 2020.

# Deep Controlled Learning for the Dynamic Time Window Assignment Vehicle Routing Problem with Stochastic Travel Times

**Sifa Celik, Albert H. Schrotenboer, Layla Martin, Tom van Woensel**
School of Industrial Engineering,
Eindhoven University of Technology, Eindhoven, The Netherlands
Email: s.celik@tue.nl

## 1 Introduction

In today's world, businesses and consumers increasingly demand swift and dependable service. In this context, reliable delivery is essential for attended next-day deliveries. It plays a pivotal role in ensuring customer satisfaction, operational efficiency, and cost-effectiveness for service providers. Reliable delivery consists of two main elements. First, deliveries should be *timely*, i.e., they should arrive in an upfront communicated time window. Second, the communicated time windows should not be too wide; otherwise, they will not enhance the customers' perceived satisfaction. However, finding the right balance between timely deliveries and time window communication for next-day deliveries is challenging. After a customer places an order (for the next day), it is most convenient for the customer to receive a time window directly. From a cost perspective, however, it is better to delay the time-window communication to the customer, as more information about other next-day deliveries can be gathered. In this work, we study the fundamental question of how to balance best the cost-service trade-off associated with direct time-window communication versus delayed time-window communication for home-attended next-day deliveries.

To answer this question, we introduce the *Dynamic Time Window Assignment Vehicle Routing Problem with Stochastic Travel Times (DTWAST).* It considers dynamically arriving next-day delivery requests, which requires a time window to be assigned. We can either do that directly, or we can wait until more customer orders are revealed at the expense of a customer inconvenience cost. Compared to the extant literature, this is the first study that considers the possibility of consolidating orders before we assign time windows – thereby investigating a crucial cost-service trade-off (see, e.g., [1] for related work).

From a methodological perspective, we propose an online one-step look-ahead heuristic algorithm based on scenario sampling methods and propose a novel offline deep reinforcement learning approach called Deep Controlled Learning (DCL) which combines Monte Carlo tree search with approximate policy iteration. From a managerial perspective, we offer practical insights by comparing various time window assignment approaches – varying in computational requirements –, and show under which system settings what approaches are most valuable.

In the remainder of this abstract, we model the DTWAST as a Markov decision process (MDP) and provide some exemplary results of the performance of our methods.

## 2 Model

We model the DTWAST as a Markov decision process (MDP) with a finite time horizon $[0, T]$. Customers order during the day and for each customer, the service provider needs to assign and communicate a time window for the order delivery next day. Whenever an order arrives, the service provider can choose to (i) postpone the time window assignment or (ii) immediately assign a time window to the customer and construct/update the route plan for the next day accordingly. In the second case, the service provider also checks whether to now plan some postponed orders. The two choices that the service provider faces provide a trade-off between customer service regarding to early communication toward the customers, on-time delivery, and transport costs. For the customer, it is beneficial to know the time of delivery early. Ensuring on-time delivery remains a primary concern for the service provider, primarily due to the inherently stochastic nature of travel times between customers and not having perfect information on customers locations and demands.

We denote the state variable as $S_k,\ k \in \{1, \ldots, K\}$. A decision epoch $k$ occurs when the $k^{th}$ customer orders. Note that $K$ is stochastic. The overall state variable definition is as follows:

$$S_k = (t_k, \mathcal{C}_k, \Delta_k, P_k, \mathcal{C}_k^{new}) \tag{1}$$

where $t_k \in [0, T]$ is the point of time when a decision is induced, $\mathcal{C}_k = \{C_{1k}, \ldots, C_{n_k k}\}$ denote the existing customers with a time window and $\Delta_k = \{\delta_{1k}, \ldots, \delta_{n_k k}\}$ present the relevant information associated with the existing customers. For each existing customer $C_{ik}$, we have the information of starting time $t^s_{ik}$ and ending time $t^e_{ik}$ of the assigned time window, denoted by $\delta_{ik} = (t^s_{ik}, t^e_{ik})$. We assume that once a time window is assigned, we cannot change the values of $t^s_{ik}$ and $t^e_{ik}$. $P_k$ is the set of planned routes for the fleet. $\mathcal{C}_k^{new}$ is the new customers set without time window assignment.

For each state variable $S_k$, we define a set of decision variables $\mathcal{X}(S_k)$. We denote a decision variable by $x_k \in \mathcal{X}(S_k)$. A decision determines whether or not to assign a time window to each customer in $C_k^{new}$. If we assign a time window, we employ a straightforward cheapest insertion heuristic considering the total travel time and a measure of time window exceedance (due to inserting the new customer). After inserting, we estimate the arrival time of the customer on which we will base a time window assignment (at the time of the conference, we will have investigated more accurate arrival time estimations). The starting and ending time of the time window is calculated based on the expected arrival time and the variation of the arrival time ensuring an on-time reliability of at least 95%. We consider minimum time window lengths of 30 mins, 45 mins, and 60 mins. The direct cost of assigning a time window is the incremental routing cost after inserting a customer, i.e., expected total travel time cost and time window exceedance cost. If we decide not to assign a time window at state $S_k$, we update the waiting time of the customer and incur the inconvenience cost.We denote the *expected* cost given a state variable $S_k$ and a decision variable $x_k$ as $\mathbb{E}R(S_k, x_k)$.

A solution to DTWASCT is given by a decision policy $\pi \in \Pi$ and a decision rule $\mathcal{X}^\pi : S_k \to \mathcal{X}(S_k)$. We can write the decision $x_k$ under policy $\pi$ as $x_k = \mathcal{X}^\pi(S_k)$. Then, the objective of the problem DTWASCT is to find a policy that minimizes the expected total penalty cost associated with time window exceedance, inconvenience cost of waiting, and travel cost, over the complete time horizon

We propose using Deep Controlled Learning (DCL), a Deep Reinforcement Learning framework [4]. DCL operates as an approximate policy iteration algorithm, enhancing policies by framing reinforcement learning as a classification problem. We simulate several scenarios to collect state-action pairs to form a dataset. This dataset is then employed to train neural networks for policy representation. For each state in the dataset, we determine an estimated optimal action by reevaluating the state under multiple external scenarios. The action with the lowest estimated expected costs over a trajectory is selected. This chosen action acts as the label for the corresponding state in the classification task, guiding the neural network to associate that state with this specific action. Through this process, DCL iteratively develops datasets and improves policies.

## 3 Results and Conclusion

We introduce three practical benchmark policies derived from the literature to test the performance of DCL policy. Cheapest Insertion (CI) policy inserts new requests into the current route-plan in by cheapest insertino directly as the customers requests arrive. The expected arrival time of the new request is used to set the communicated time window to the customer. The second policy ia a Multiple-Scenario Approach (MSA), based on [3]. When a new request arrives, we sample scenarios and future requests and construct possible route plans to decide whether to assign a time window or not. The final policy is Stochastic Lookahead Rollout Policy (LRP) [5]. We also present a Perfect Information (PI) solution as a reference lower bound for routing costs. We consider a problem where all the requests are realized. We carefully balance the cost coefficient of time window exceedances with the waiting cost of customers. This ensures that when employing both the CI and PI policies in the MDP, the total costs are approximately within the same range.

We present some exemplary findings of the different policies using instances with a time window width of 45 minutes in Table 1. We train the neural agent with instances with stochastic travel times from the CVRP benchmark set [2]. Total costs are comprised of waiting and routing costs. While the LRP policy results in the lowest cost, it also requires prohibitively high running times. The DCL policy can find a only slightly more expensive solution (with more dissatisfied customers but lower routing costs) in a fraction of the time, making it applicable for next-day delivery.

Table 1: Average costs of DCL, MSA, LRP, CI, and PI policies

| **Policy** | **Waiting Cost** | **Routing Cost** | **Total Cost** |
|---|---|---|---|
| DCL | 80.95 | 1306.57 | 1387.52 |
| MSA | 106.69 | 1406.13 | 1402.99 |
| LRP | 8.32 | 1351.65 | 1351.09 |
| CI | 0.00 | 1459.38 | 1459.38 |
| PI | 1016.21 | 928.21 | 1944.42 |

The current results of DCL are obtained by limited parameter tuning only. At the time of the conference, we will have investigated several neural network structures, and provide a full suite of computational results on the performance of the method.

## References


[1] M. W. Ulmer and B. W. Thomas, "Enough Waiting for the Cable Guy—Estimating Arrival Times for Service Vehicle Routing", *Transportation Science* 53(3), 897–916, 2019.

[2] P. Augerat, VRP problem instances, http://www. branchandcut. org/VRP/data/, 1995.

[3] A. M. Campbell and M. W. P. Savelsbergh, "Decision support for consumer direct grocery initiatives",*Transportation Science* 39(3), 313–327, 2005.

[4] W. Van Jaarsveld, "Model-based controlled learning of MDP policies with an application to lost-sales inventory control", CoRR, abs/2011.15122, 2020

[5] D.P. Bertsekas, D.P. Castanon. "Rollout Algorithms for Stochastic Scheduling Problems". *Journal of Heuristics* 5, 89–108 (1999). https://doi.org/10.1023/A:1009634810396

# Solving the Two-Echelon Inventory-Routing Problem: A Matheuristic Approach

**Sara Charaf**
Department of Industrial Engineering,
Eindhoven University of Technology, Eindhoven, The Netherlands
Email: s.charaf@tue.nl

**Duygu Taş**
Smart Mobility and Logistics Lab,
Faculty of Engineering and Natural Sciences,
Sabancı University, Istanbul, Turkey

**Simme Douwe P. Flapper & Tom Van Woensel**
Department of Management Science and Technology
Eindhoven University of Technology, Eindhoven, The Netherlands

## 1 Introduction

Efficiently optimizing modern supply chains demands a substantial level of coordination and synchronization in operational decision-making, especially in the context of large-scale distribution networks engaging multiple stakeholders. For instance, a two-echelon inventory-routing problem (2E-IRP) is an optimization problem that arises in two-echelon distribution networks where one or more suppliers fulfill customer demands via a network of Distribution Centers (DCs). DCs serve as storage and consolidation points for freight received from upstream sources, ensuring efficient delivery downstream to the end customers. Companies often rely on Vendor-Managed Inventory (VMI) systems to achieve efficient coordination within supply chains. VMI is a collaborative practice between suppliers and customers that offers mutual benefits. It alleviates customers from inventory management tasks and enhances suppliers' logistics efficiency, thus reducing costs and improving vendor-customer relationships (Yao and Dresner, 2008).

Introducing a VMI strategy in a two-echelon distribution network involves solving the 2E-IRP. Current research on two-echelon distribution networks primarily addresses basic problem variants, overlooking real-world challenges like combining routing problems with inventory decisions (Savelsbergh and Van Woensel, 2016). Indeed, a few papers have studied diverse 2E-IRP applications in diverse domains (Rohmer et al., 2019; Guimarães et al., 2019; Farias et al., 2021; Charaf et al., 2022). One efficient algorithm developed for the one-echelon inventory-routing problem is tabu search (TS) (Archetti et al., 2012, 2017).

In this paper, we address the 2E-IRP, where one or more suppliers fulfill the demand of a set of geographically dispersed customers via DCs over a finite planning horizon. The replenishment of inventories at the DCs and customers follows the Maximum Level policy, where deliveries are constrained by their inventory holding capacity. Two capacitated fleets of vehicles, located at the suppliers and the DCs, can visit the DCs and customers; however, direct deliveries from a supplier to customers are prohibited. DCs and customers can be delivered at most once per period by a designated vehicle. The objective of the 2E-IRP is to minimize the total travel and inventory holding costs over the planning horizon while respecting capacity constraints.

We propose a two-phase matheuristic that integrates TS and mathematical programming formulations. The performance of this solution method is tested on 400 small-sized benchmark instances and 400 newly generated large-sized instances. We also provide insights into its components' effectiveness and impact on solution quality and computational performance.

## 2 Methodology

The first phase of the two-phase matheuristic approach is an initialization phase, aiming at finding an initial feasible solution within a reasonable computational time. For this purpose, we design a heuristic based on the branch-and-price proposed by Charaf et al. (2022). A TS column generator solves the pricing subproblems, and the heuristic terminates after solving the root node. An integer master problem is then solved as a mixed integer program (MIP), resulting in an initial solution. The initial solution may be infeasible but is accepted in the second phase, namely the improvement phase.

The second phase aims to refine the initial solution and consists of a TS-based matheuristic. The neighborhood of an incumbent solution is constructed using seven operators that either remove, insert, swap, or move customer visits, replace the route source, or optimize the route sources simultaneously. The resulting solutions may be feasible or infeasible; however, a subset of infeasible solutions is discarded to prevent a significant degradation in solution quality. To handle infeasibilities, the objective function includes inventory and travel costs and penalty costs to penalize the infeasibility of solutions that do not satisfy the constraints. Moreover, penalty coefficients are dynamically updated to guide the TS solution space exploration. Additionally, two procedures are devised to recover feasibility and optimize upstream deliveries. These procedures entail solving a linear program and a mixed integer program.

In subsequent iterations, the solution chosen is the non-tabu solution with the best score in the neighborhood unless a tabu solution improves the best solution found so far. This selected solution is further refined by implementing improvement procedures. A solution is designated as tabu based on two tabu lists: one records the operators used to generate the selected solutions, and the other records the inventory and travel costs of the chosen solutions. Two diversification methods are implemented to escape local optima: a jump procedure and a perturbation to the penalty coefficients. Finally, the TS is run until a maximum number of iterations without improvement is reached, a maximum number of overall iterations is reached, or the overall run time limit has elapsed.

## 3 Results and Discussion

The matheuristic is implemented in Python and Cython using CPLEX v.22.1. We test the performance of our matheuristic on 400 small-sized benchmark instances introduced by Charaf et al. (2022) involving five to 25 customers, in addition to 400 large-sized instances involving 30 to 50 customers derived from the well-known instances introduced by Archetti et al. (2007). These instances consist of two classes with low and high inventory costs, one supplier and two satellites ($1s2$), or two suppliers and three satellites ($2s3$), and one to two first-echelon vehicles and two to five second-echelon vehicles. The overall time limit of the algorithm is set to one hour, and the maximum number of TS iterations is 600 for small instances and 1000 for large instances.

For small instances, we compare the results obtained using our matheuristic with those provided in Charaf et al. (2022) and by solving the arc-based mathematical formulation of the 2E-IRP using CPLEX with a time limit of three hours. When compared over the 165 instances for which an optimal solution is known, the matheuristic finds 99 optimal solutions with an average gap of 0.70%. Moreover, it improves 159 upper bounds out of 235 known best-upper bounds, with an average gap of (-2.32%). The average solution time is 109 seconds, with a maximum of 516 seconds. The average time spent by the MIP of the initialization phase accounts for 86.36% of the average time spent in the initialization phase and 25.77% of the average total time. The initialization phase generated a feasible solution for all small instances; however, solving the MIP of the initialization phase becomes more challenging for instances with two vehicles as it consumes six times more time than instances with five vehicles.

For large-sized instances, no optimal solution is known. All instances were solved within half an hour, with an average solving time of 495 seconds. During the initialization phase, an infeasible initial solution is generated for 54 instances, 41 involving two vehicles. These infeasible solutions were subsequently repaired in the second phase. For the instances with a feasible initial solution, the average gap improvement is approximately

9.17%, slightly higher for instances with low inventory costs and a low number of vehicles.

Finally, we assess the effectiveness of various components of the TS through a series of 17 performance tests: 7 tests involve disabling one of the operators at a time, four tests evaluate the impact of the tabu list sizes, another 4 analyze the effect of the maximum number of iterations, and two tests evaluate the impact of the improvement procedures. Key findings highlight the crucial role of the operator related to removing customer visits in reducing both the computational times and the average gaps. The importance of the operator related to inserting customer visits in terms of average gaps is evident despite its low number of calls. The feasibility recovery procedure also has a positive impact, while the operator related to optimizing the route sources shows no significant effect.

## References


Archetti, C., Bertazzi, L., Hertz, A., and Speranza, M. G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1):101–116.

Archetti, C., Bertazzi, L., Laporte, G., and Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391.

Archetti, C., Boland, N., and Grazia Speranza, M. (2017). A matheuristic for the multi-vehicle inventory routing problem. *INFORMS Journal on Computing*, 29(3):377–387.

Charaf, S., Taş, D., Flapper, S. D., and van Woensel, T. (2022). A branch-and-price algorithm for the two-echelon inventory-routing problem. *https://arxiv.org/abs/2206.12316*.

Farias, K., Hadj-Hamou, K., and Yugma, C. (2021). Model and exact solution for a two-echelon inventory routing problem. *International Journal of Production Research*, 59(10):3109–3132.

Guimarães, T. A., Coelho, L. C., Schenekemberg, C. M., and Scarpin, C. T. (2019). The two-echelon multi-depot inventory-routing problem. *Computers & Operations Research*, 101:220–233.

Rohmer, S., Claassen, G., and Laporte, G. (2019). A two-echelon inventory routing problem for perishable products. *Computers & Operations Research*, 107:156–172.

Savelsbergh, M. and Van Woensel, T. (2016). 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590.

Yao, Y. and Dresner, M. (2008). The inventory value of information sharing, continuous replenishment, and vendor-managed inventory. *Transportation Research Part E: Logistics and Transportation Review*, 44(3):361–378.

# A multi – commodity location – network design problem with vehicle selection in City Logistics

**Francesco Contu**
Department of Mathematics and Computer Science
University of Cagliari, Cagliari, Italy
Email: francesco.contu97@unica.it

**Teodor Gabriel Crainic**
Department of Management and Technology
University of Quebec, Montréal, Canada
Email: crainic.teodor@uqam.ca

**Massimo Di Francesco**
Department of Mathematics and Computer Science
University of Cagliari, Cagliari, Italy
Email: mdifrance@unica.it

**Enrico Gorgone**
Department of Mathematics and Computer Science
University of Cagliari, Cagliari, Italy
Email: egorgone@unica.it

## 1 Introduction

This work aims to study and develop a decision-support system for planning the distribution of highly customized freight packed in containers, arriving from the sea in cities built around a port. The problem is faced from the point of view of an urban mobility manager who aims to build a tactical plan for managing existing transportation resources, represented by firms located in the outskirts of the city, to distribute the freight from the port to locations in the city at a minimum cost. Altough relevant research exists on multi - commodity location - routing for city logistics ([1], [2]), this work investigates multi - commodity location - network design.

## 2 Problem statement

Consider a maritime city, where containers arrive at the port by ship. Each container carries pallets for various city destinations. Each final customer is associated with a known demand of pallets. The origin of these demands is the port, where pallets are packed in containers. Since the demand of pallets is highly customized, different costs are paid for their transportation at each destination. Each demand represents a commodity, which can also be identified by its destination. Containers cannot be opened at the port and they cannot join destinations within the city due to local regulations; they must be unpacked in intermodal facilities, and the pallets must be delivered to customers by means of a set of city-freighters.

A two-tiered distribution system is considered: in the $1^{st}$ tier, Container-Compatible Vehicles (*CCVs* for short) will operate in order to transport containers from the port to the intermodal facilities located in the outskirts, that are warehouses in which containers can be opened and pallets can be handled (*satellites* for short). In the satellites, the containers will be opened and their content will be transferred to Pallet-Compatible Vehicles (*PCVs* for short). Then, in the $2^{nd}$ tier PCVs will transport pallets from the satellites to customers in the city. The routes are open, i.e. vehicles are not required to return to any satellite or the port after servicing the last customer in the route. Moreover, split delivery is allowed for all destinations in the $2^{nd}$ tier, PCVs have limited capacity and satellites can manage a limited number of containers pallets and PCVs.

A working day is planned, based on a daily expected regular demand over a medium-term planning period and the problem is approached from the point of view of an *urban mobility manager*: *shippers* request to provide transportation of the pallets at the port (stored in containers), specifying the destination for each pallet; *carriers* offer contracts for transporting containers and pallets with CCV and PCV respectively; *intermodal terminal operators* offer contracts for opening the containers and handling the pallets in their satellites. The decisions that need to be made by the urban mobility manager are the selection of satellites required to handle all the incoming containers from the port, the assignment of each container to a satellite, the selection of CCVs required to transport the containers to the corresponding satellite, the selection of PCVs required to deliver all the pallets, the assignment of each selected PCV to a satellite, the routes followed by each PCV starting from satellites, to deliver pallets to customers and number of pallets transported by each PCV for each customer. The goal of the problem is to determine a plan in order to deliver all the demands to the related customer at minimum cost; the cost includes the CCVs utilization costs, the PCVs utilization cost and the satellites utilization costs.

## 3 Problem modelling

The problem is formulated on a network $G = (N, A)$, where nodes $N := n_0 \cup S \cup \Gamma$ represent the port ($n_0$), satellites (set $S$) and customers (set $\Gamma$). The set $A$ of arcs consists of the union of two subsets $A_1$ and $A_2$, which are associated with the $1^{st}$ and the $2^{nd}$ tier. The set of containers $C$ arrives at the port transported by ship and each container $c \in C$, carries $P_c^\gamma$ pallets with destination $\gamma$, for each $\gamma \in \Gamma$. Figure 1 shows a possible network.

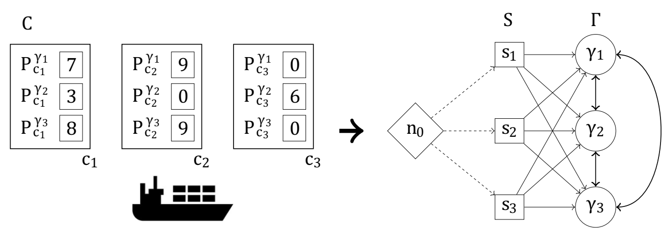



Figure 1: A possible network

An optimization model is proposed, and the variables representing the decisions are defined as follows: $y_s$ is a *satellite-selection* variable that takes the value 1 if satellite $s \in S$ is selected, 0 otherwise; $x^1_{ksc}$ is a *container-transportation* variable that takes the value 1 if container $c \in C$ is moved by the CCV $k$ from the port to satellite $s \in S$, 0 otherwise; $w^2_{k(i,j)}$ is a *routing* variable that takes the value 1 if the PCV $k$ traverses arc $(i, j) \in A_2$, 0 otherwise; $x^2_{k\gamma(i,j)}$ is a *pallet-transportation* variable representing the number of pallets shipped along arc $(i, j) \in A_2$ to customer $\gamma \in \Gamma$ by the PCV $k$, 0 otherwise. The objective function minimizes the costs of selection of satellites, those of selection of vehicles and their assignment to satellites, the costs of operations for each pallet handled at satellites, the transportation costs of containers and pallets, and vehicle routing costs in the $2^{nd}$ tier. As for constraints, each container must be picked up from the port and moved to a satellite by a CCV, each customer must receive the requested amount of freight moved by pallets, all pallets must cross-dock one satellite only, CCVs and a PCVs must be assigned to one satellite at most and the satellite's capacity constraints must be met in terms of number of containers, pallets and PCVs. Yet, the capacity of PCVs holds in the $2^{nd}$ tier.

## 4 Solution Method

We present an Adaptive Large Neighborhood Search (ALNS) meta-heuristc algorithm. The search space is the pair of vectors on *satellite-selection* variables and *container-transportation* variables. At each iteration a *current solution* is modified by a *destroy operator* to *de-assign* (or *remove*) a number of containers from their corresponding satellites. These containers are then *re-assigned* (or *inserted*) back to any open satellite by a *repair operator*. After a pair of destroy-repair operations, the paths of PCVs and pallets are obtained by solving a Network Design problem to obtain a new *incumbent solution*. Possible infeasible solutions are considered by penalization in the violation of satellite capacity constraints.

Destroy operators are divided into two categories: ***large-impact destroy operators*** and ***small-impact destroy operators***, borrowing some ideas from the Adaptive Large Neighborhood Search metaheuristic for the a two-echelon vehicle routing problem [3]. The first operators remove containers as a consequence of an explicit closing or opening of a satellite. They are used less frequently (after a certain number of iterations without improvement). The incumbent solutions obtained from these operators are always accepted as new *current solution*. The latter operators remove containers, without changing the configuration of the satellites. These operators are used at every iteration (except in the case cited above) and the resulting incumbent solutions could be accepted according to several acceptance criteria.

All destroy and repair operators are based on heuristic procedures and are selected by a roulette wheel mechanism with scores associated to the operators based on their past success.

## 5 Conclusion

A meta-heuristic method is proposed and compared to a heuristic algorithm which is based on decomposition by satellite: first a location-allocation problem with PCVs assignment determines the assignment of containers to satellites and the assignment of PCVs to satellites. Then, for each selected satellite, a network-design problem is solved to determine the paths of PCVs and pallets in the $2^{nd}$ tier. All the solutions are integrated together to find a feasible solution for the complete problem. Estimations on the cost impact in the $2^{nd}$ tier are taken into account in the first subproblem.

## References


[1] P. Giannessi, L. Alfandari, L. L´etocart, R. Wolfler-Calvo. The multicommodity-ring location routing problem. "Transportation Science", 50 (2) (2015), pp. 541-558.

[2] M. Boccia, T.G. Crainic, A. Sforza, C. Sterle. Multi-commodity location routing: Flow intercepting formulation and branch-and-cut algorithm. "Computers & Operations Researc", 89, (2018), pp. 94-112.

[3] Hemmelmayr, V. C., Cordeau, J. F., Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. "Computers & operations research", 39(12), 3215-3228.

# The Min Max Multi–Trip drone Location Arc Routing Problem

**Teresa Corberán**
Departament d'Estadística i Investigació Operativa,
Universitat de València, València, Spain
Email: tecorfa@alumni.uv.es

**Isaac Plana**
Departament de Matemàtiques per a l'Economia i l'Empresa,
Universitat de València, València, Spain

**José María Sanchis**
Departament de Matemàtica Aplicada,
Universitat Politècnica de València, València, Spain

## 1 Introduction

Drones have limited flight autonomy, therefore for some applications it is necessary to use other types of ground support vehicles that can serve as a point for launching and retrieving the drones, as well as recharging or exchanging their batteries. The coordination of ground vehicles with drones has been widely discussed in the literature for node routing problems. However, there are few works that have addressed this problem from the point of view of arc routing problems (e.g., Amorosi et al.[1]).

Unlike ground routing problems, drone routing problems are characterized by the fact that drones can enter and leave the edges at any point and serve only part of them, making an already difficult problem much harder.

In the Min Max Multi-Trip drone Location Arc Routing Problem (MM–MT–dLARP) there is a set of lines that have to be traversed (to perform a service) and a depot from which a set of $P$ trucks, each one carrying a drone, must travel to $P$ out of $D$ available points ($D \geq P$), where the drone is launched. Each drone has a limited autonomy which allows it to fly a maximum time, or distance, $L$ before having to get back to the launching point to change its battery so that it can start another route. Once the drone has completed all its routes, the truck goes back to the depot. The goal of the MM-MT-dLARP is to determine the launching point for each truck and find a set of drone routes for each truck, each one starting and ending at its launching point and with flight time not greater than

$L$, in such a way that all the drones' routes jointly traverse all the given lines and the largest total time of all the trucks (time of traveling to the launching point, flight time of the drone and time of traveling back to the depot) is minimized.

As in previous works, we approach the problem by digitizing each line as a polygonal chain with a sequence of points. In this way we obtain a classical ARP where a set of given edges (the segments of the polygonal chains) must be traversed. We call this problem MM–MT–LARP.

In this talk, we present a matheuristic for the MM-MT-dLARP and a formulation and a branch and cut for the MM-MT-LARP. We also present computational results on instances with different characteristics.

## 2 Problem definition

The MM–MT–LARP is defined on an undirected graph $G = (V, E)$, where $E_R \subset E$ is the set of required edges (the segments), $V_R \subset V$ is the set of vertices incident with $E_R$, $\mathcal{D} \subset V$ is the set of isolated vertices corresponding to the launching points, and a vertex $0 \in V$ that represents the depot (and can also be a launching point). Moreover, there is an edge between each pair of vertices in $V$. Note that this implies that there is a parallel edge to each $e \in E_R$. These non–required edges are represented by $E_{NR} \subset E$.

In what follows, and for the sake of simplicity, we will use "cost" to refer to the time or distance associated with the traversal/service of the edges. Each $e \in E_R$ has a service cost $c_e^s > 0$ and each $e \in E_{NR}$ has a deadheading cost $c_e > 0$, given by the Euclidean distance, of traveling directly between its endpoints. For each launching point $d \in \mathcal{D}$, $c_{0d} > 0$ is the cost for a truck to travel from the depot to $d$. The launching points could be considered as facilities to be opened if a truck goes to this point to launch its drone. The cost $c_{0d}$ could represent the fixed cost of opening point $d$.

The goal of the MM-MT-LARP is to determine which $P$ launching points are used, and find, for each one, a set of at most $Q$ drone tours, with cost no greater than $L$, in such a way that the tours jointly traverse the required edges and the maximum cost associated with the trucks is minimum.

To formulate the problem, we define a variable $x_e^{dk}$, $\forall e \in E$ and each flight $dk \in \mathcal{D} \times \mathcal{K}$ ($\mathcal{K} = \{1, \ldots, Q\}$), that takes the value 1 if $e$ is traversed by flight $k$ starting at the launching point $d$, and 0 otherwise, a variable $y_e^{dk}$, $\forall e \in E_{NR}$ and each flight $dk$, with value 1 if $e$ is traversed twice by flight $k$ starting at $d$, and 0 otherwise, and a variable $z^d$, $\forall d \in \mathcal{D}$, with value 1 if $d$ is used by a truck, and 0 otherwise.

$$\text{Minimize} \qquad t$$

$$\sum_{k\in\mathcal{K}}\sum_{e\in E_R} c^s_e x^{dk}_e + \sum_{k\in\mathcal{K}}\sum_{e\in E_{NR}} c_e\big(x^{dk}_e + y^{dk}_e\big) \;+\; 2c_{0d}z^d \le t, \quad \forall d\in\mathcal{D}$$

$$\Big(x^{dk}+y^{dk}\Big)\big(\delta(v)\big) \equiv 0 \pmod 2\ , \qquad \forall dk\in\mathcal{D}\times\mathcal{K},\ \forall v\in V$$

$$\Big(x^{dk}+y^{dk}\Big)\big(\delta(S)\big) \ge 2x^{dk}_f, \qquad \forall dk\in\mathcal{D}\times\mathcal{K}, \quad \forall S\subseteq V\backslash\{d\},\ \forall f\in E(S)$$

$$\sum_{k\in\mathcal{K}}\sum_{d\in\mathcal{D}} x^{dk}_e \ge 1, \qquad \forall e\in E_R$$

$$x^{dk}_e \ge y^{dk}_e, \qquad \forall e\in E,\ \forall dk\in\mathcal{D}\times\mathcal{K}$$

$$\sum_{e\in E_R} c^s_e x^{dk}_e + \sum_{e\in E_{NR}} c_e(x^{dk}_e + y^{dk}_e) \le Lz^d, \qquad \forall dk\in\mathcal{D}\times\mathcal{K}$$

$$\sum_{d\in\mathcal{D}} z^d \le P$$

$$x^{dk}_e \in\{0,1\}, \qquad \forall e\in E,\ \forall dk\in\mathcal{D}\times\mathcal{K}$$

$$y^{dk}_e \in\{0,1\}, \qquad \forall e\in E_{NR},\ \forall dk\in\mathcal{D}\times\mathcal{K}$$

$$z^d \in\{0,1\}, \qquad \forall d\in\mathcal{D}$$

## 3 A branch and cut for the MM–MT–LARP

We have designed and implemented a branch and cut for the problem based on the integer programming formulation proposed and on some families of valid inequalities related to the parity and connectivity conditions of the routes and their maximum cost. Moreover, symmetry–breaking inequalities have been used as well as other inequalities that are satisfied by at least an optimal solution. Separation procedures for each family of inequalities have been proposed.

## 4 A matheuristic for the MM–MT–dLARP

In this section we present the matheuristic proposed for the approximate solution of the MM–MT–dLARP. It is based on the construction of an initial set of solutions. This is done by using two different procedures.

The first one starts by selecting randomly, in each iteration, the launching points to use. We associate an assignment cost to each required edge for every chosen launching point, introducing some randomness to obtain different solutions in each iteration. Then, the edges are assigned to the launching points, trying to balance the number of edges assigned to each point. The problem of finding for each launching point a tour starting and ending at it and traversing its associated edges is a General Routing Problem (Orloff [4]) that is solved to optimality by using the algorithm described in Corberán et al. [3]. From these initial giant tours for each launching point we build a set of tours of cost no greater than $L$ for each drone.

The second one also selects randomly the launching points to use in each iteration, but now edges are assigned depending on a probability that is computed based on their distances to the launching points. For each launching point, a set of tours of cost no greater than $L$ is built with a random selection procedure.

A VND procedure is applied to each solution constructed with the above methods. Four local search procedures based on interchanges of edges inside the routes and between routes are embedded in the VND scheme. Finally, an optimization procedure is applied to each drone tour using the algorithm proposed in [3]. From the best $n$ solutions obtained so far, and in order to consider the drones' characteristics, we try to obtain better solutions by adding intermediate vertices to each required edge and applying again the VND procedure.

The whole procedure is run for a maximum computing time.

## 5 Computational results

Three sets of instances of different sizes have been generated in the same way as in Campbell et al. [2] for the DARP, but adding launching points. The instances have between 13 and 92 original nodes, 9 and 88 original lines, and 3 and 6 launching points (including the depot) from which we have to select between 2 and 5 to use.

The computational results obtained so far show that the branch and cut is only able to optimally solve instances of small and medium size in two hours of CPU. The matheuristic finds good solutions in reasonable times and the addition of intermediate vertices makes it possible to get better solutions in many instances.

## References

[1] L. Amorosi, J. Puerto, and C. Valverde, "A multiple-drone arc routing and mothership coordination problem", *Computers & Operations Research* 159, 106322, 2023.

[2] J.F. Campbell, Á. Corberán, I. Plana, and J.M. Sanchis, "Drone Arc Routing Problems", *Networks* 72, 543–559, 2018.

[3] Á. Corberán, I. Plana, and J.M. Sanchis, "A Branch & Cut Algorithm for the Windy General Routing Problem and special cases", *Networks* 49, 245-257, 2007.

[4] C.S. Orloff, "A Fundamental Problem in Vehicle Routing", *Networks* 4, 35-64, 1974.

# Integrated network equilibrium model for private cars and urban logistic systems

**Cristian E. Cortés**

Department of Civil Engineering, Universidad de Chile and Instituto Sistemas Complejos de Ingeniería (ISCI), Blanco Encalada 2002, Santiago, Chile.

E-mail: ccortes@ing.uchile.cl

**Pablo A. Rey**

Departamento de Industria, Facultad de Ingeniería and Instituto Universitario de Investigación y Desarrollo Tecnológico (IDT), Universidad Tecnológica Metropolitana, , Santiago, Chile.

Email: prey@utem.cl

**Vanesa Vansteenkiste**

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Universidad Nacional de Rosario, Rosario, Argentina.

E-mail: vane61284@hotmail.com

## 1. Introduction

In an effort to model more realistic network settings, we study the relationship between traffic equilibrium and urban freight/service providers. In fact, urban freight and service delivery may be analyzed from two opposite points of view. For logistic agencies, the most relevant aspect is the minimization of costs. From an OR standpoint, challenges and perspectives in city logistics can be found in Crainic et al. (2021, 2023). Social costs generated by freight and service mobility within the urban must be quantified, recognizing a connection with the impact on traffic congestion of vehicles circulating and stopping for delivery tours and last-mile delivery. The quantification of freight demand is essential for evaluating the effects of any city logistic policy. Freight is passive and requires infrastructure, facilities, and vehicles to be loaded/unloaded and transported; customer requirements will also affect freight operations (Holguín-Veras & Thorson, 2000). Commercial vehicle movement within an urban area also has distinguishing features (Holguín-Veras & Patil, 2005; Ruan et al., 2012). Tour or trip chaining is a particularly important element of commercial vehicle movement in an urban setting (Wang & Holguín-Veras, 2008). Freight transport involves activities and decisions made at different dimensions (value, commodities, vehicle trips), and each economic agent pursues profit maximization (Holguín-Veras & Patil, 2008). For freight movement analysis, the tour is appropriate as the analysis unit because freight tours are a result of economic decisions subject to the minimization of the logistic costs associated with daily activities (Khan and Machemel, 2017). Holguín-Veras et al. (2015) propose a novel spatial price equilibrium formulation that explicitly considers delivery tours instead of point-to-point deliveries. The need to explicitly consider the multiple dimensions of the problem has led to works based on game theory, spatial price equilibrium, and other related concepts (Nagurney & Dong, 2002; Friesz & Holguín-Veras, 2005; Holguín-Veras et al., 2015).

## 2. Problem Statement

In this research, we have developed multilevel and integrated urban transport models that combine transport equilibrium phenomena with the optimization behavior of commercial vehicles, considering tours as unit analysis. Equilibrium models provide a reasonable approach to of modeling the macroscopic features of these systems. One initial problem to study corresponds to the classic traffic user equilibrium but with demand specified not as single origin-destination trips but as pairs origin-set of destinations. Each user should visit all its stops and, to optimize its travel time, it must choose both the order of the stops in the tour and the specific route between stops. Two approaches can be followed to formulate an equilibrium model in this context. First, we can use the usual formulation of Wardrop equilibrium, considering only tours visiting all the stops. An alternative formulation is based on separating the decisions of the users into two steps: definition of the sequence of stops and selection of the route between two consecutive stops in the tour. The equivalent optimization formulation in this case results:

$$\min \sum_{a \in A} \int_0^{x_a} t_a(w)\,dw$$

$$\sum_{s \in S_d} n_{ds} = q_d \qquad \forall d$$

$$\sum_{\ell \in R_k} f_\ell = \sum_{d \in D} \sum_{s \in S_d} \delta_{sk} n_{ds} \qquad \forall k$$

$$f_\ell, n_{ps} \geq 0 \quad \text{and} \quad x_a = \sum_{x \in K} \sum_{\ell \in R_k} f_\ell \delta_{a\ell}$$

Here, a nonnegative variable $n_{ds}$ represents the amount of flow serving demand $d$, which visits the stops following the sequence $s$; a nonnegative variable $f_\ell$ represents the flow along path $\ell$. $\delta_{a\ell}$ and $\delta_{sk}$ represent binary parameters equal to one if arc *a* belongs to path $\ell$ , and one if sequence s serves demand *k*, respectively. The link performance function $t_a(w)$ represents the average travel time for crossing arc *a* when the flow is *w* (in vehs/time unit). To make the approach more realistic, the function could depend on both types of flows, $w_1$ for cars, and $w_2$ for delivery vehicles, and the calibration could be conducted differentiating these two types of vehicles. The time spent by the trucks at the delivery points could be trespassed to the arc cost (delay) affecting all types of vehicles, if the stopping infrastructure is the right lane of the road, for example; these details could be easily added to the approach to provide more realisism to the operation.

The first set of constraints ensures that the flow corresponding to each demand is totally allocated to the possible sequences for visiting the required destinations. The second set of equations makes, for each origin-destination pair $k$, the balance between the flows assigned to routes serving the OD pair and the flow allocated to the OD pair obtained from decisions on sequences. The two proposed approaches can be used to adapt the well-known Frank-Wolfe algorithm to find the equilibrium in the new conditions. In the first approach, the subproblem of finding a descent direction consists of a Steiner traveling salesman problem (Letchford et al. 2013) for each demand instead of a set of shortest path problems. Using the second formulation, the problem is divided into two subproblems: the usual all OD-pair shortest path problem and then a set of traveling salesman problems in a smaller contracted network to find the sequences of stops in the tours.

**3. Implementation and preliminary results**

We have implemented this approach considering a combined demand of private cars with a set of tours of demands for last-mile deliveries, wherein the proposed formulation, the sequence of stops, is defined for each tour, modeling an a-priori assignment of sequences performed by each dispatcher. A further effort will be the inclusion of the decision of the sequence of stops as part of the integrated equilibrium. At this stage, we are applying the integrated equilibrium approach in the previous formulation in the context of a combined flow of particular cars and delivery vehicles on the same network, and where our premise is that using the previous approaches, we obtain a Wardrop equilibrium situation in travel costs applied to both the particular vehicles as well as the delivery ones.

One important feature to consider in a practical implementation of these algorithms is a previous transformation (extension) of the network to specify the OD demand for private cars as delivery tours. For each pair OD $(i,j)$, we add to the original network and additional node $ij^*$ and arcs $(j,ij^*)$ and $(ij^*,i)$ and define a demand with origin $i$ and set of destinations $\{ij^*\}$. These additional arcs have cost zero. This way, to satisfy this demand, the flow should be assigned to a tour in the network visiting both $i$ and $ij^*$. From the definition of the extended network, this flow should go from node $i$ to node $j$ as desired.

Our test network for private cars is the well-known Sioux Fall one, used for many transportation studies, from which we took the topology (digraph), the demand, and the BPR performance functions for the travel time on arcs as a function of the flow on them (https://github.com/bstabler/TransportationNetworks/tree/master/SiouxFalls). On this network, in the preliminary tests, we have added three delivery tours: 3000 vehicles starting at depot in node 1, and stopping at nodes 2, 3, 7 and 12; 1500 vehicles starting at the depot in node 1 and stopping in nodes 2, 4 and 15; 5000 vehicles starting at the depot in node 13 and stopping in nodes 3, 4 and 18.

We designed a benchmark to compare our equilibrium results with what we believe could be a typical behavior of dispatchers in practice. They usually plan their routes using offline information of the travel times in the system, for example, under free flow conditions. Then, based on a fixed assignment of such vehicles, we run a standard Frank-Wolfe algorithm considering the assigned fixed flow of delivery vehicles as known.

To analyze the results, we implemented a simple modification to the Frank-Wolfe algorithm to calculate, in each iteration, flow assignments to routes that are consistent with the flow assigment to arcs obtained. Preliminary results show that our approach can find a simultaneous equilibrium for both types of vehicles. The assigment obtained with the benchmark procedure is an equilibrium for private cars. However, travel times for delivery vehicles can be reduced, in some cases, by changing some assignments to other routes. In the equilibrium found with our model, some delivery demands are split in more than one delivery route. This is not possible with the benchmark procedure. Consequently, travel times for some OD car demands in the benchmark assignment overestimate the travel times in equilibrium, and others underestimate them.

## 4. Conclusions

In this research, we are proposing a novel integrated approach to build equilibrium models based on the decisions made by all the vehicles sharing the infrastructure (roads) within an urban context. Nowadays, the decisions made by dispatchers of last-mile deliveries, for example, do not explicitly consider the externalities caused by them to the rest of the traffic, as the road capacity is limited, mainly during periods of high traffic congestion. This proposal is a first step in understating all the factors that arise when the different vehicles interact for different purposes, but all of them are under a common equilibrium given by an adaptation of the Wardrop principle.

## References

Crainic, T. G., Feliu, J. G., Ricciardi, N., Semet, F., & Van Woensel, T. (2023). Chapter 10. Operations research for planning and managing city logistics systems. *Handbook on City Logistics and Urban Freight: 0*, 190.

Crainic T. Perboli G., Ricciardi N. (2021). Network Design and Applications to Transportation and Logistics, *City Logistics, chapter 6*, 507-537, Springer.

Friesz T. and J. Holguín-Veras (2005). Dynamic game–theoretic models of urban freight: formulation and solution approach. In: A. Reggiani and L. Schintler (eds.) *Methods and models in transport and telecommunications: cross Atlantic perspectives*. Springer, 143–161.

Holguín-Veras, J., N. Xu, M. Jaller, and J. Mitchell (2015). A dynamic spatial price equilibrium model of integrated urban production-transportation operations considering freight delivery tours. *Transp. Sci.* 50, 489–519.

Holguin-Veras J. and G.R. Patil (2008). A multicommodity integrated freight origin-destination synthesis model. *TRR* 1906, 74–80.

Holguin-Veras J. and G.R. Patil (2005). Observed trip chain behavior of commercial vehicles. *Network & Spatial Economics* 8, 309–326.

Holguín-Veras J. and E. Thorson (2000) An investigation of the relation-ship between the trip length distributions in commodity-based and trip-based freight demand modeling. *TRR* 1707, 37–48.

Khan M. and R. Machemehl (2017). Analyzing tour chaining patterns of urban commercial vehicles. *Transp. Res. Part A* 102, 84–97.

Nagurney, A., & Dong, J. (2002). A multiclass, multicriteria traffic network equilibrium model with elastic demand. *Transportation Research Part B: Methodological*, *36*(5), 445-469

Ruan, M., J.J. Lin, and K. Kawamura (2012). Modeling urban commercial vehicle daily tour chaining. *Transp. Res. Part E* 48, 1169–1184.

Wang Q. and J. Holguín-Veras (2008) Investigation of attributes determining trip chaining behavior in hybrid microsimulation urban freight models. *TRR* 2066, 1–8.

# The disaggregated integer L-shaped method for the stochastic vehicle routing problem

Lucas Parada[(1)], Robin Legault[(2)], Jean-François Côté[(3)], and Michel Gendreau[(1)]

[(1)]CIRRELT, Polytechnique Montréal,
[(2)]Operations Research Center, Massachusetts Institute of Technology,
[(3)]CIRRELT, Université Laval,
jean-francois.cote@fsa.ulaval.ca

## 1 Introduction

This talk presents a new integer L-shaped method, named the disaggregated integer L-shaped method (DLM). This method is tailored for a specific class of two-stage stochastic programs in which, given a first-stage solution, the recourse cost can be expressed as a sum of independent recourse functions involving disjoint sets of first-stage variables. The DLM also requires each of the resulting recourse functions to be monotonic. In a minimization problem, this property means that adding first-stage variables to a component of the recourse function cannot decrease its value.

We propose an implementation of the DLM for the vehicle routing problem with stochastic demands (VRPSD). In this problem, each customer has to be visited exactly once by a fleet of capacitated vehicles. Customer demands are modeled by independent random variables (RVs) with known distributions and are only observed when a vehicle arrives at the customer's location. A failure occurs when a vehicle arrives at a customer's location with a remaining capacity smaller than the customer's demand. When this happens, a recourse action must be implemented to satisfy the current and next customer demands. This work applies the detour-to-depot (DTD) recourse action, which performs a back-and-forth trip from the customer to the depot to restock the vehicle.

Our contributions are as follows. First, we introduce the DLM in the context of the VRPSD. Second, we demonstrate that, although the DTD recourse function is not monotonic in general, it is for Poisson and some normal distributions when the sum of the expected demands on each route respects the vehicle capacity. Third, we present several new lower bounds on the recourse that take advantage of the monotonicity property. Finally, numerical results show our new method achieves state-of-the-art results on instances of the literature.

## 2 Related literature

For brevity, we only review the approaches that apply the integer L-shaped method to the VRPSD. The integer L-shaped method was proposed by [4] to solve two-stage stochastic programs where both stages have binary variables. It was first used on the VRPSD by [1], and and it was successively improved by [5], [3], and [2] by the use of lower bound functionals (LBFs). These inequalities bound the recourse for a much broader range of solutions than optimality cuts. The main advantages of the method are that it can manage solutions that include long routes, it is compatible with several distributions, and it can handle many different types of recourse functions. Unfortunately, it is less efficient for instances requiring more than three vehicles.

## 3 Mathematical model

The VRPSD is defined on an undirected graph $G = \{N_0, E\}$ where $N_0 = \{0, 1, ..., n\}$ is the set of nodes, with 0 being the depot, $N = \{1, ..., n\}$ the set of customer nodes, and $E = \{(i, j) : i, j \in N, i < j\}$ the set of edges. Traveling on edge $(i, j)$ incurs a travel cost of $c_{ij}$. Each customer $i \in N$ has a non-negative demand given by the RV $\xi_i$, with $\mathbb{E}[\xi_i] = \mu_i$ and $var(\xi_i) = \sigma_i^2$. Demand variables are assumed to be independently distributed. A fleet of identical vehicles is available to satisfy customer requests. Each vehicle has a capacity $Q$ and must follow a route that starts and ends at the depot. Each customer must be visited exactly once. A route is defined as a sequence of customers starting and ending at the depot, and it is feasible if the sum of the expected demands respects the vehicle capacity. Let $M$ be the set specifying the number of vehicles that can be used in a solution. Let $E(S)$ be the edges with both endpoints in $S$ and $\delta(h)$ the set of edges incident to node $h$. The VRPSD can be formulated as follows. The variable $x_{ij}$ indicates the number of times edge $(i, j) \in E$ is traversed. The variable $x_{0i}$ equals 2 if a vehicle serves a single customer $i \in E$, the other $x_{ij}$ variables are binary. A binary variable $z_m$ decides whether $m$ vehicles are used. Let $\mathcal{Q}(x)$ denote the expected recourse cost of the first-stage solution $x = (x_{ij})$. The model is:

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij} + \mathcal{Q}(x) \tag{1}$$

$$\text{s.t.} \sum_{i\in N} x_{0i} = \sum_{m\in M} 2mz_m, \tag{2}$$

$$\sum_{(i,j)\in\delta(h)} x_{ij} = 2 \qquad h \in N, \tag{3}$$

$$\sum_{(i,j)\in\delta(S)} x_{ij} \leq |S| - \left\lceil \frac{\sum_{i\in S}\mu_i}{Q} \right\rceil \qquad S \subseteq N, \tag{4}$$

$$\sum_{m\in M} z_m = 1, \tag{5}$$

$$x_{ij} \in \{0, 1\} \qquad (i, j) \in E(N), \tag{6}$$

$$x_{ij} \in \{0, 1, 2\} \qquad (i, j) \in E(0). \tag{7}$$

The objective (1) is to minimize the sum of travel costs plus the expected recourse cost. Constraint (2) imposes that $m$ routes must be connected to the depot if $m$ vehicles are used. Constraints (3) ensure that customers are visited exactly once. Constraints (4) impose that the expected demand on each route does not exceed the vehicle capacity and that the routes are connected to the depot. Constraint (5) ensures that the number of vehicles that are used in the solution is an element of $M$. Constraints (6) and (7) define the domain of the variables.

For the DTD recourse policy, it is possible to separate the calculation of the recourse by route.

Let $\mathcal{R}^\nu$ be the set of routes in solution $x^\nu$. Then, the expected recourse cost is defined as:

$$\mathcal{Q}(x^\nu) = \sum_{r \in \mathcal{R}^\nu} \mathcal{Q}(r), \tag{8}$$

where $\mathcal{Q}(r) = \min\{\mathcal{Q}^1(r), \mathcal{Q}^2(r)\}$ is the minimum expected recourse cost of both orientations of route $r = (0, i_1, \ldots, i_t, 0)$. The recourse of route $r$ in the first orientation is calculated as follows.

$$\mathcal{Q}^1(r) = 2 \sum_{j=1}^{t} \sum_{l=1}^{\infty} \mathbb{P}\left( \sum_{k=1}^{j-1} \xi_k \leq lQ < \sum_{k=1}^{j} \xi_k \right) c_{0j} \tag{9}$$

The recourse function sums the probability of failure over all customers. A failure happens at customer $j$ if the sum of previous demands is smaller than $lQ$, for $l = 1$ to $\infty$, and is strictly greater than $lQ$ when including the demand of $j$.

# 4 Disaggregated integer L-shaped method

The main idea of the DLM is to replace $\mathcal{Q}(x)$ by a sum of variables $\theta_i$, one for each customer $i \in N$. The purpose of this transformation is to express the contribution of each customer to the cost of the second stage by the gradual addition of a new type of optimality cuts during the resolution. These cuts require a monotic recourse to be valid. They are added for feasible paths that are found, and the cut of each path only involves the variables associated with the edges of the path and the $\theta_i$ variables of the customers of the path. They allow to bound the recourse function more effectively than the traditional optimality cuts by being active for more solution. This gives the following objective function.

$$\min \sum_{(i,j) \in E} c_{ij} x_{ij} + \sum_{i \in N} \theta_i \tag{10}$$

Our method starts by solving a model with the objective (10), with relaxed rounded capacity inequalities and integrality constraints. During the resolution, whenever a feasible integer solution is found, an optimality cut is added for each route $r = (0, p, 0)$ having a positive recourse, where $p = (i_1, \ldots, i_t)$ is the sequence of customers. Let $x(p) = \sum_{j=1}^{t-1} x_{i_j i_{j+1}}$ and $N(p)$ be the set of customers in path $p$. The new optimality cut follows.

$$\sum_{i \in N(p)} \theta_i \geq \mathcal{Q}(r) \left( x(p) - |p| + 1 \right) \tag{11}$$

We propose a new type of LBFs that use simple structures to bound the recourse. The new type of cuts uses the $\theta_i$ variables to bound the recourse for sets of customers $S$. It requires a valid lower bound $L(S)$ on the recourse for any path visiting consecutively the customers of $S$. Let $x(S) = \sum_{(i,j) \in E(S)} x_{ij}$. The new LBFs are as follows:

$$\sum_{i \in S} \theta_i \geq L(S) \left( x(S) - |S| + \left\lceil \frac{\sum_{i \in S} \mu_i}{Q} \right\rceil + 1 \right). \tag{12}$$

We propose three different $L(S)$: $L_1(S)$ for sets $S$ that require only one vehicle to serve its customer, $L_2(S)$ for sets $S$ that require more than one vehicle to serve its customer, and $L_3(S)$ for an initial global lower bound on the recourse.

## 5 Computational results

This section reports results on the instances of [3]. Table 1 presents the results against the integer L-shaped methods of [3] and [2] under the columns JRGL14, HS23. Each row contains 30 instances with the same number of customers $n$ and the same fleet size $\bar{m}$. The number of optimal solutions found (Opt), the average computing time (Time(s)), average optimality gap (Gap%) are reported.

| | | JRGL14 | | | HS23 | | | DL-Shaped | |
|---|---|---|---|---|---|---|---|---|---|
| n | $\bar{m}$ | Opt | Time(s) | Gap% | Opt | Time(s) | Gap% | Opt | Time(s) |
| 40 | 4 | 9 | 1240.7 | 1.5 | 28 | 91.9 | 1.6 | 30 | 2.0 |
| 50 | 3 | 16 | 6918.0 | 0.7 | 29 | 101.5 | 2.9 | 30 | 9.8 |
| 50 | 4 | 5 | 1360.8 | 1.9 | 25 | 224.8 | 1.7 | 30 | 30.9 |
| 60 | 2 | 24 | 1393.0 | 0.4 | 30 | 72.2 | - | 30 | 1.2 |
| 60 | 3 | 6 | 2766.0 | 0.7 | 27 | 191.4 | 1.2 | 30 | 16.7 |
| 60 | 4 | 3 | 4922.0 | 2.0 | 25 | 262.1 | 1.9 | 30 | 25.3 |
| 70 | 2 | 17 | 2577.5 | 0.5 | 30 | 99.2 | - | 30 | 2.0 |
| 70 | 3 | 9 | 1753.3 | 1.5 | 24 | 563.7 | 1.5 | 30 | 46.2 |
| 80 | 2 | 13 | 1809.2 | 0.5 | 28 | 193.7 | 1.0 | 30 | 19.8 |
| Total or avg. | | 102 | 2711.4 | 1.2 | 246 | 185.7 | 1.6 | 270 | 17.1 |

Table 1: Performance comparison on the instances of [3]

Table 1 indicates that the DLM achieves state-of-the-art results by solving to optimality all the 270 instances of the set, while [3] and [2] respectively solve 102 and 246 instances. Furthermore, our average resolution time is 17.1 seconds per instance, compared to 2711.4 and 185.72 seconds for the instances that were solved to optimality by [3] and [2], respectively.

## 6 Conclusion

We presented a new approach for solving a class of stochastic integer programs in which the first-stage solutions can be decomposed into disjoint components. The method is applied to the vehicle routing problem with stochastic demands under the detour-to-depot recourse policy. Our computational experiments show that it achieves state-of-the-art results on instances from the literature.

## References


[1] M. Gendreau, G. Laporte, and R. Séguin. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation science*, 29(2):143–155, 1995.

[2] Y. Hoogendoorn and R. Spliet. An improved integer l-shaped method for the vehicle routing problem with stochastic demands. *INFORMS Journal on Computing*, 35(2):423–439, 2023.

[3] O. Jabali, W. Rei, M. Gendreau, and G. Laporte. Partial-route inequalities for the multi-vehicle routing problem with stochastic demands. *Discrete Applied Mathematics*, 177:121–136, 2014.

[4] G. Laporte and F. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

[5] G. Laporte, F. Louveaux, and L. Van Hamme. An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50(3):415–423, 2002.

# Service Network Design with Uncertainty on Water Levels for Intermodal River Transport

**Bita Payami Shabestari**
School of Management and CIRRELT
Université du Québec à Montréal, Montréal, Canada
Email: payami_shabestari.bita@courrier.uqam.ca

**Ioana C. Bilegan**
Université Polytechnique Hauts-de-France, LAMIH, CNRS, UMR 8201
F-59313 Valenciennes, France

**Teodor Gabriel Crainic** & **Walter Rei**
School of Management and CIRRELT
Université du Québec à Montréal, Montréal, Canada

## 1 Introduction

Barge transportation, a key component of intermodal freight transportation, is known for its cost-effectiveness and environmental sustainability and plays a crucial role in facilitating the exchange of freight between maritime ports and their hinterlands, as well as among river ports. Carriers in this sector often use consolidation strategies to combine smaller shipments, enhancing operational efficiency and cost-effectiveness. However, this requires precise coordination of shipping schedules, freight specifications, and service requirements.

Despite extensive research on barge transportation, there is a notable gap in studies focusing on the tactical level of consolidation-based barge transportation planning, particularly concerning the scheduled service network design with resource and revenue management (SSND-RRM) problem. Our study addresses this gap, focusing on the tactical planning of barge intermodal transportation, explicitly considering the challenges raised by operational constraints such as water-level variability.

Most service network design cases, including SSND models for planning consolidation-based transportation systems, typically operate under the assumption that there are no significant variations in the system's state, encompassing both supply and demand sides, over the planning horizon [2]. SSND models generally presume, for example, that the size of freight, its availability, and delivery times, as well as the departure and arrival times of services and the capacity offered by each resource supporting the service over the designed network, remain known and constant throughout the planning horizon. This is often not the case, however. Hence, our research centers on the uncertainty in infrastructure and resource capacity, particularly relevant for barge transportation where water-level fluctuations can significantly impact vessel capacity. Lower water levels can decrease vessel capacity due to increased grounding risks, while higher water levels might allow for greater freight capacity but could introduce navigational challenges under bridges and through certain canal sections [1]. These fluctuations require a stochastic approach to service network design, where capacity is uncertain due to variations in water level.

We present methodology aiming to establish a tactical operations plan, given predicted water levels, that maximizes the expected carrier's revenue while accounting for future adjustments to the plan when information is revealed and predictions are reliably updated, to fulfill the demands of shippers and optimize the utilization of the carrier's resources.

## 2 Problem Statement

The barge transportation system operates within a network of waterways and ports, each with distinct characteristics that impact vessel operations. Variations in water levels, influenced by climatic conditions such as rainfall and river flows, affect vessel navigation, their freight-carrying capacity, and the possibility to access ports. The berthing capacity of port terminals, measured in length units, is also a critical factor. To simplify the presentation, and because the modelling approach of the impact on navigation are similar regardless of the direction of the change in water levels, our focus is on the decreased water-level situation. This decision is also supported by the observation that, despite occasional increases in water levels due to excessive rainfall, the current predominant issue within the logistics and transportation industry is the trend towards lower water levels.

The transportation system involves several stakeholders, namely, a number of shippers and the carrier. The former, ranging from manufacturers, to traders and intermediaries, to retailers, initiate the demand for transportation services to move freights with specific weights and volumes from their origins to their intended destinations within specified time intervals. We categorize shippers into three groups based on their contracts with the carrier: regular (long-term agreements), partial spot (with demands that could be partially met only), and full spot (with demands that may either be fully serviced or not be serviced at all). Shipper needs differ, some prioritizing express transit for urgent goods, while others seeking cost-effective solutions.

The carrier responds by offering transportation services, performed by vessels that follow waterway routes within the physical network, to meet and consolidate these diverse demands. Vessels of different types are considered in this study, each being characterized by freight-carrying capacities (e.g., in weight and number of containers), draft, and length.

Services are characterized by the physical routes and the terminals where they stop, the schedules indicating arrival and departure times at ports, the speed, and the type of the vessel assigned to perform it, the latter providing the service capacity, possibly impacted by the water levels of the river segments and ports making up the service route. The carrier incurs various costs, which vary with cargo type and vessel size, to set up services and operate vessels, as well as to load, unload, store, and consolidate freight at terminals.

## 3 Methodology

We present a *Scheduled Service Network Design with Resource and Revenue Management* (SSND-RRM) modeling framework for tactical planning in consolidation-based freight transportation, which extends prior research [3], and three stochastic programming formulations, addressing key operational constraints such as water levels variations and terminal berthing capacities. The objective is to maximize carrier revenue over a tactical planning horizon (e.g., a season), the plan detailing the service network, resource utilization, and freight-handling and transport activities for a given schedule length (e.g., a week), executed repeatedly throughout the season.

The deterministic SSND-RRM formulation is defined over a time-space network built for a discretization of the schedule length. The decision variables target the selection of the vessel-characterized services, the selection of the partial (with the demand volume to be serviced) and full-spot demands, the commodity flows on the service legs and handled in ports, and the numbers of vessels used or idling. The objective function maximizes the net revenue of the carrier computed as the difference between the total revenue obtained from regular and selected partial and full-spot shippers, and the operation costs of setting up services, circulating vessels to support them, and moving freight through the service network of the selected services and port terminals. Two sets of flow conservation constraints control the movement of freight through the network and on the service vessels and, thus, the demand satisfaction. Capacity-linking constraints enforce the feasibility of service-vessel utilization in terms of tonnage and number of containers. Water-level constraints ensure that the feasibility of service selection in terms of the respective vessels and routes, while design-balance constraints enforce the circulation of vessels given the selected services. Fleet availability is also enforced for each vessel type.

Water level variability, a critical factor affecting service capacity, is represented through expert-determined probability distributions, for critical waterway segments and ports, which yield variation figures for the vessel capability to load and berth, given its type [4]. Such predictions are made for the next season at a fairly high aggregation level and are later repeatedly updated during operations for short (e.g., the schedule length) horizons. In stochastic-programming terms, this corresponds to the information-revelation process.

When the updated predicted water levels are lower than initially anticipated, the vessel capacities decrease, which forces the carrier to make hard decisions, e.g., either refuse part of the demands to be transported and pay any associated penalties, or decide to adjust activities by re-optimizing the demand itineraries or restructuring the service plan.

1. *Simple recourse* (2-SPSR), using non-negative decision variables in the second stage for unmet demand with corresponding penalties;

2. *Partial recourse* (2-SPPR), re-optimizing demand itineraries in the second stage; high-cost ad-hoc transfers to land-based transportation is also considered;

3. *Full recourse* (2-SPFR), considering the selection of additional services or modes in the second stage, as well as the associated optimization of demand flows.

At the Odysseus conference, we will present the tactical planning problem for consolidation-based freight carriers in the presence of uncertainty on the infrastructure capacity, emphasizing the challenging issue of varying water levels for intermodal barge transportation, together with the deterministic SSND-RRM formulation. We will then focus on the water-level uncertainty and the tactical-plan adjustment strategies (the recourses) for the three stochastic-programming SSND-RRM variants. The results of an extensive experimentation campaign, performed both with a well-known commercial software and our own in-development meta-heuristic, will also be discussed. We will focus on model accuracy, impact of deterministic and random parameters on model and solution-method performance, the comparative behaviour of the recourse strategies, and the insights gained relative to the impact of infrastructure capacity uncertainty on the tactical planning of consolidation-based freight carriers.

## References


[1] M. Prandtstetter, P. Widhalm, H. Leitner, I. Czege, A Novel Visualisation Tool for Reliable Inland Navigation, *Transportation Research Procedia*, 2022.

[2] T.G. Crainic, M. Gendreau, and B. Gendron, *Network Design with Applications to Transportation and Logistics*, Springer, 2021.

[3] I.C. Bilegan, T. G. Crainic, and Y. Wang, Scheduled Service Network Design with Revenue Management Considerations and an Intermodal Barge Transportation Illustration," *European Journal of Operational Research* 300(1), 164–177, 2022.

[4] A. Christodoulou, P. Christidis, and B. Bisselinkb, Forecasting the Impacts of Climate Change on Inland Waterways, *Transportation Research Part D: Transport and Environment* 82, 102159, 2020.

# Capacity Planning with Supplier Selection and Uncertainty on Contract Fulfilment

Teodor Gabriel Crainic[1,3], Guido Perboli[1,2], and Walter Rei[1,3],
[1] CIRRELT, Montreal, Canada [2] DIGEP, Politecnico di Torino, Turin, Italy
Email: guido.perboli@polito.it

[3] ESG, UQAM, Montreal, Canada
Email: teodorgabriel.crainic@cirrelt.net, rei.walter@uqam.ca

## 1 Introduction

We consider a logistics capacity planning problem where a shipper negotiates capacity resources, e.g., containers, vans, ship/train slots, and warehouse space, from logistics-service suppliers to move or store goods. The shipper must decide which suppliers to collaborate with and how much capacity to secure from each to satisfy forecast demand, given uncertainty regarding future demand and availability of the contracted capacity.

We introduce the *Stochastic Variable Cost and Size Bin Packing with Capacity Loss and Supplier Selection* (*SVCSBP[CL][CS]*) problem, extending the Bin Packing (BP) problem family, and propose a two-stage stochastic programming formulation. The first stage targets the selection of suppliers and the contracting of capacity. The second stage models adapting the first-stage plan to revealed information on actual demand and capacity loss, by acquiring extra capacity and assigning items to available bins. The *SVCSBP[CL][CS]* problem and formulation extend the literature by introducing multiple suppliers with reliability-cost trade-offs and the management of the uncertain availability of contracted capacity.

## 2 Problem Setting

We identify the demand as the number of *items* to move or to store, characterized by *size* (or *volume*). The capacity units, named *bins* according to the BP vocabulary, may be of different types, due to the transportation modes or warehousing spaces considered, being characterized by particular *unit costs* and *sizes* (volumes).

A supplier is a logistics-service provider characterized by its offer in terms of bins of various types, an *activation cost* and a *reliability factor*. The activation cost is intended as a one-time fee the supplier charges upon creating a new business with the shipper. Reliability is a crucial factor in choosing a supplier, defined in this work as the ability to provide the contracted capacity dependably and accurately, i.e., provide at the right time the correct type and quantity of bins actually offered, compared with what was agreed in the contract. The reliability factor is intended as the shipper's confidence level regarding the supplier's reliability to fulfill the contract. It is presented as a deterministic parameter estimated based on internal and external information (e.g., historical data) about the supplier's performance. A selected set of suppliers is then considered feasible if its average reliability factor value is higher than a *minimum reliability requirement*.

The tactical planning process involves the selection of a set of supplier contracts, each contract specifying the number of capacity units of each type the supplier agrees to provide each time the shipper calls during operations. This capacity is secured based on a forecast of uncertain parameters, and some time in advance of the "next" medium-term operation period (e.g., season), being intended to be used repeatedly for the duration of that activity horizon. According to the common practice in the industry, the shipper contracts more than one supplier, in order to improve resiliency while alleviating the supply chain volatility that is lowering operational margins.

Considering that contracts are established well in advance of operations, their selection involves uncertainty, e.g., the future demands and available booked capacity units may be subject to random changes [2, 3]. In turn, this entails that the capacity plan might require adjustments to be made when it is repeatedly used over the considered time horizon. We consider several sources of uncertainty, assuming that the information revelation process provides, each time the plan is applied, the actual values of the uncertain parameters.

The first major source of uncertainty is the *demand*. More precisely, the number of items and the size of each item to transport or store at any given occurrence of operations can be different from the expected value. This may result in insufficient booked capacity available on the shipping day, generating additional costs to secure the missing capacity.

Second, we consider the uncertainty on the *availability of the contracted capacity*. The contracted bins may be unavailable on a given day due to mechanical failures, accidents, market fluctuations creating shortages for specific types of bins, etc. These adverse *random-event* situations impact what the supplier had promised to the shipper, yielding a *capacity loss* and, thus, compromising the contract fulfillment, reputation, and profits. Following random-event situations, this capacity loss is also stochastic, varying with time, supplier, and type of capacity unit.

A third major source of uncertainty is the *availability of the future, ad-hoc capacity*, which could be secured on the spot market, at a higher price than the contracted capacity, to react to the variations of demand and contracted bins. The number, type (size), and cost of these ad-hoc capacity units are random parameters.

The shipper can make operational decisions to adjust the plan when actual information is revealed and the variations in stochastic parameter values are observed. These so-called *recourse actions* concern securing ad-hoc capacity through the spot market and re-optimizing the assignment of items to the available bins. We also assume, without explicitly modeling it, that the shipper deploys re-selling strategies of the surplus capacity when the observed overall demand is lower than estimated.

The loss of contracted capacity also implies additional costs. For the carrier, other than the cost of securing ad-hoc capacity, there are the costs to reassign and arrange the items into the available bins. On the other hand, not delivering the booked capacity that the supplier promised to the shipper may compromise the contract fulfillment. Hence, for the supplier, this violation comes at a cost paid to compensate for its lack of reliability and poor service quality affecting the contract fulfillment [4]. We model this *compensation cost* as a non-delivery penalty proportional to the lost capacity.

## 3 Model and Solution Method

We model the *SVCSBP[CL][CS]* problem as a two-stage stochastic programming formulation [1]. The *first stage* concerns the planning decisions, i.e., the selection of the suppliers and the *a priori* booking of the bins of various types, volumes and fixed costs to be used repeatedly over the planning horizon to move or store the estimated demand of items. The *second stage* refers to the decisions taken at operation time when the actual demand and available contracted capacity are revealed. These recourse actions concern the procurement of additional bins on the spot market and the re-assignment of the items to the available bins, either originally contracted or ad-hoc.

The model minimizes the total cost, i.e., the total cost of selecting suppliers, the total fixed cost of the booked bins in the capacity plan, and the expected cost associated with securing ad-hoc capacity and adjusting the packing during operations. The model includes constraints to break the symmetry, which usually characterizes packing problems, to impose a lower bound on the number of suppliers the shipper must select, to guarantee that a minimum percentage of the total volume offered by each selected supplier is bought, and to enforce the minimum requirement on the average reliability (weighted by the booked capacity) of the selected suppliers. Linking constraints connect first-stage supplier and bin-type selection decision variables and the respective capacity values. Second-stage constraints refer to packing, ensuring that each item is packed into a single bin (contracted or ad-hoc), and that the total volume of items packed in a bin does not exceed the actual volume of the bin.

The *SVCSBP[CL][CS]* is a complex problem, computationally challenging particularly when dimensions grow. We thus propose a two-level meta-heuristic, which exploits the problem structure and enhances the Adaptive Large Neighborhood Search [ALNS, 5] with memory structures and intensification and diversification phases.

It is noteworthy that, given a selection of suppliers, *SVCSBP[CL][CS]* becomes a Stochastic Variable Cost and Size Bin Packing with Capacity Loss problem, which can be addressed by state-of-the-art methods [e.g., the Progressive Hedging-based meta-heuristic of 3]. Consequently, the top level of the meta-heuristic explores the space of the supplier subsets, while, at the second-level, the search space defined by a supplier subset is explored by the PH-based meta-heuristic of [3].

The local search of the top level is provided by a combination of destroy and rebuild ALNS operators. The search is enhanced by intensification and diversification operators, which dive around good solutions or project the search in (hopefully) little-explored regions of the search space, respectively. These operators, as well as those of the ALNS local search, fix selection variables based on behaviour markers learned (computed and updated), during the local and global search phases, by analyzing the consists of local and global best solutions.

The *SVCSBP[CL][CS]* problem and formulation, as well as the proposed meta-heuristic, will be detailed at the conference. The results of extensive computational tests and analyzes will also be discussed at the conference, focusing on the relevance of the *SVCSBP[CL][CS]* and the accuracy and effectiveness of the meta-heuristic.

## References

[1] Birge, J.R., Louveaux, F., 1997. Introduction to Stochastic Programming. Springer Series in Operations Research and Financial Engineering, Springer.

[2] Crainic, T.G., Gobbato, L., Perboli, G., Rei, W., 2016. Logistics Capacity Planning: A Stochastic Bin Packing Formulation and a Progressive Hedging Meta-heuristic. European Journal of Operational Research 253, 404–417.

[3] Crainic, T.G., Perboli, G., Rei, W., Rosano, M., Lerma, V., 2023. Capacity Planning with Uncertainty on Contract Fulfillment. European Journal of Operational Research Forthcoming.

[4] Guo, Z., Fang, F., Whinston, A.B.. Managing Supply Uncertainty with an Information Market. In R. Sharman, H. R. Rao, and T. S. Raghu (Eds.), *Exploring the Grand Challenges for Next Generation E-Business*, pages 80–95, 2011. Springer Berlin Heidelberg.

[5] Ropke, S., Pisinger, D., 2006. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. Transportation Science 40, 455–472.

# A Tetris-based Beam Search algorithm for the Distributor's Pallet Loading Problem

**Davide Croci, Ola Jabali, Jacopo Libè, Federico Malucelli**
Politecnico di Milano, Milano, Italy
Email: davide.croci@polimi.it

**Joe Naoum-Sawaya**
Ivey Business School,
University of Western Ontario, London, Canada

## 1 Introduction

This paper focuses on a Distributor's Pallet Loading Problem (DPLP), which consists of orthogonally stacking a set of cuboid-shaped items in a minimal number of pallets. The recent growth in e-commerce has garnered attention to study the DPLP with highly heterogeneous products. Fundamentally, the DPLP is a Three-Dimensional Bin Packing Problem (3DBPP) with additional constraints to obtain pallet layouts that can be used in real-world operations. In this regard, a review on 3DBPPs and their real-world variants is presented in Ali et al. (2022). In particular, we consider (1) orientation constraints, ensuring that items are packed given a set of possible orientations; (2) static stability constraints, guaranteeing that items will not fall due to gravity; (3) load bearing constraints, indicating that each item must be able to sustain the load of boxes placed on top of it; and (4) weight-limit constraints, indicating that a pallet's total weight does not exceed a given threshold. We model static stability by imposing that a minimum percentage of the base area of each item must lie on top of other items, and we model load bearing constraints using a load-distribution graph (Gzara et al., 2020). Assuming that all items must be packed, we consider a hierarchical multi-objective function. As a first objective, we minimize the number of used pallets. As a second objective, we maximize the average pack density of the used bins, which is the ratio between the total volume of packed items and the volume of the minimum cuboid containing them (Gzara et al., 2020).

We propose a competitive state-of-the-art algorithm for DPLPs. Differently from recent DPLP literature, we assemble boxes with a new strategy inspired by the dynamics of the popular game Tetris. Specifically, the main contributions of this paper are as follows: 1) we introduce a new constructive heuristic for the DPLP called Tetris Heuristic (TH), 2) we develop a new beam search algorithm for the DPLP called Tetris Beam Search

(TBS), which uses TH as a core component, 3) we introduce a series of algorithmic enhancements aimed at efficiently evaluating the considered constraints, and 4) considering sets of highly heterogeneous products, we demonstrate the effectiveness and efficiency of our algorithm by benchmarking its performance against state-of-the-art algorithms.

## 2 Tetris Heuristic for the DPLP

Due to the complexity of 3DBPPs, heuristic methods are commonly used for large instances. In particular, real-world variants of the 3DBPP are typically solved with layer building heuristics. These excel in instances with weakly heterogeneous items, as they tend to create compact layers (Ali et al., 2022). However, layer building heuristics are not effective in instances with highly heterogeneous items, since compact layers are difficult to obtain. To address this common case we develop a new constructive heuristic for the DPLP called Tetris Heuristic (TH). This algorithm takes as input the bins' dimensions, $(W, D, H)$, their maximum load $M$, a set of items $I$ with dimensions $(w_i, d_i, h_i)$, weight $p_i$, and maximum load bearing capacity $m_i$. As depicted in Figure 1, TH iteratively places a subset of items in a bin. As items are introduced, they "fall" to the lowest available position to achieve efficient packing. TH maintains a list $P$ of vertical coordinates $z_p$, on

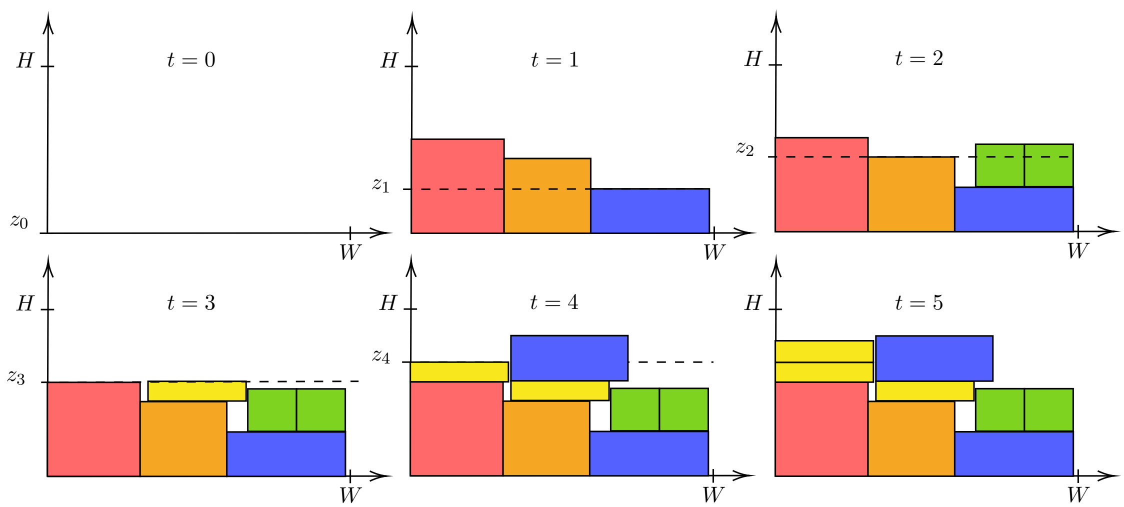



Figure 1: View of a bin built with TH, $z_i$: the lowest available $z$ coordinate at iteration $i$.

which new items could be placed. We refer to the horizontal section of a bin at $z_p$ as a "horizontal plane". For each horizontal plane $p$, TH keeps track of its "supporting items", i.e., the items directly under $z_p$ that can potentially provide support for additional items to be placed above $z_p$. Moreover, TH tracks the "intersecting items" of each plane $p$, which are items whose maximum vertical coordinate is greater than $z_p$. These items may not necessarily be resting on $p$, but they restrict the available space for packing additional items on it. Let $I_t$ be the set of items available for packing at iteration $t$. At each iteration, TH takes the horizontal plane $p$ with the lowest $z_p$ value, and packs a subset of items on it. The items' placement is obtained by solving a Two-Dimensional Orthogonal Knapsack Problem (2DOKP), where the knapsack has the same size $(W, D)$ as the bin's base, and the items are rectangular of size $(w_i, d_i)$ and value $w_i d_i$ for each $i \in I$. The goal is to place the most valuable subset of items on $p$, while accounting for non-overlap constraints with $p$'s "intersecting items", as well as the real-world constraints of our DPLP. Due to the non-linearity of these constraints, we solve the 2DOKP with a heuristic algorithm based on the concept of Corner Points (Martello et al., 2000). Our algorithm places items firstly sorted by height and secondly by volume. This favors the creation of new horizontal planes with substantial support, allowing more items to be placed in successive iterations. Whenever items are placed on a plane $p$, that plane is discarded. TH then generates new planes based on the top surface of the placed items, and updates all the planes' supporting and intersecting item sets.

## 3 A Tetris Beam Search algorithm for the DPLP

We introduce Tetris Beam Search (TBS), a beam search algorithm which exploits TH in solving the DPLP. Beam search is a tree-search algorithm that expands only the top $\beta$ most promising nodes at each level $k$ of the search tree. Let $S_k$ be the set of nodes at level $k$ of the search tree. Each state $s \in S_k$ consists of a list of open bins $B_s$, and a list of items to pack $I_s$. For each open bin $b \in B_s$, TBS tracks the set of packed items $Q_b$ and the set of horizontal planes $P_b$, as described in Section 2. At each iteration $k$, TBS applies a series of rules to select the top $\beta$ most promising states $T_k \subseteq S_k$, and then proceeds to expand each $s \in T_k$ generating a set of successor states. This is done by first partitioning the items in $I_s$ into groups according to their height. Then, for each open bin $b \in B_s$, new states are generated by placing the most voluminous subset of items of each group in $b$ with TH. If no insertion is possible, a new bin is opened, and a successor state is generated by placing item groups in the newly open bin. Whenever items are placed in a bin $b \in B_s$, $P_b$ is updated as described in Section 2. The successors of each $s \in T_k$ are collected in $S_{k+1}$ for the next iteration. The algorithm terminates when $S_{k+1}$ is empty, returning the best complete solution found so far.

## 4 Results and Conclusions

We performed a series of numerical experiments to evaluate the performance of our proposed algorithm. All experiments were conducted on a Linux machine equipped with a four-core Intel Core i7 CPU clocked at 3 GHz, and with 16 GB of RAM. We tested our algorithm on 140 highly heterogeneous instances obtained with the instance generator of Gzara et al. (2020). We compare against their results, as well as against results from Tresca et al. (2022), who solve a more relaxed version of our DPLP. We note that the generator of Gzara et al. (2020) has stochastic elements, therefore our instances might slightly differ from the ones used in other works. We perform experiments with a value of $\alpha = 0.7$ as the minimum percentage of base area to be supported for static stability. We also set the size of the beam search tree to $\beta = 10$. Table 1 outlines the aggregated results obtained by our algorithm (TBS), as well as the results reported by Gzara et al. (2020) (GEY), and Tresca et al. (2022) (TCCCD). Each row corresponds to a group of 20 instances with the same number of items to pack, indicated in column $|I|$. For each group of instances, we report the average number of open bins, the average pack density of open bins and the average execution time obtained by the three solution algorithms. We observe that TBS outperforms recent algorithms in almost every instance group. Considering both benchmarks, our algorithm achieves an average improvement of 26% in number of open bins, and a 24% in pack density. Furthermore, the computational efficiency of TBS is notably higher, with processing times being an order of magnitude faster compared to the results reported by GEY and TCCCD. Building on these results, we further evaluate TBS on instances provided by our industrial partner, with a number of items $|I|$ ranging from 2020 up to 2953. We observe that TBS is able to obtain solutions of high quality also in this case, with a computational time that is acceptable for real-world logistics operations.

| | Avg. Bins (#) | | | Avg. Pack Density (%) | | | Avg. Time (s) | | |
|---|---|---|---|---|---|---|---|---|---|
| $\vert I \vert$ | GEY | TCCCD | TBS | GEY | TCCCD | TBS | GEY | TCCCD | TBS |
| 100 | 1.90 | 1.88 | **1** | 0.45 | **0.78** | 0.69 | 7.05 | 24.00 | **0.58** |
| 150 | 2.00 | 2.28 | **1.65** | 0.49 | 0.65 | **0.66** | 18.06 | 37.48 | **1.59** |
| 200 | 2.6 | 2.83 | **2** | 0.49 | 0.64 | **0.69** | 43.38 | 35.01 | **2.86** |
| 500 | 4.8 | 5.6 | **3.7** | 0.59 | 0.57 | **0.75** | 441.08 | 157.21 | **16.18** |
| 1000 | 8.7 | 9.2 | **7.05** | 0.63 | 0.61 | **0.79** | 1331.79 | 367.02 | **72.97** |
| 1500 | 12.25 | 12.4 | **9.9** | 0.67 | 0.65 | **0.82** | 2337.02 | 1007.75 | **122.40** |
| 2000 | 17.05 | 17.15 | **13.25** | 0.64 | 0.68 | **0.83** | 5723.95 | 1909.28 | **185.52** |

Table 1: Comparison with literature

## References

S. Ali, A. G. Ramos, M. A. Carravilla and J. F. Oliveira, "On-line three-dimensional packing problems: A review of off-line and on-line solution approaches", *Computers & Industrial Engineering*, 168, 108-122, 2022.

F. Gzara, S. Elhedhli and B. C. Yildiz, "The Pallet Loading Problem: Three-dimensional bin packing with practical constraints", *European Journal of Operational Research* 287(3), 1062-1074, 2020.

S. Martello, D. Pisinger and D. Vigo, "The Three-Dimensional Bin Packing Problem", *Operations Research* 48(2), 256–267, 2000.

G. Tresca, G. Cavone, R. Carli, A. Cerviotti and M. Dotoli, "Automating Bin Packing: A Layer Building Matheuristics for Cost Effective Logistics", *IEEE Transactions on Automation Science and Engineering*, 19(3), 1599-1613, 2022.

# A Rollout Algorithm for Truck Scheduling Using Estimated Time of Arrival

**Maximiliano Cubillos*, Ola Jabali, Elena Tappia**
Politecnico di Milano, Milano, Italy
*Email: maximiliano.cubillos@polimi.it

**Justin Goodson**
Saint Louis University, St. Louis, USA

## 1 Introduction

Yard management consists of optimization problems related to the synchronization of truck arrivals, departures, and resource utilization within logistic distribution centers. We consider minimizing the total waiting time of inbound trucks at a yard by optimizing the truck-to-dock assignment decisions. The inherent uncertainty of truck arrival times and their impact on assignment decisions can be mitigated by the use of technologies that provide estimated time of arrivals (ETA) in quasi-real-time. The aim of this paper is to propose a dynamic algorithm that accounts for ETA information to make online truck assignment decisions.

We consider a set $\mathcal{J}$ of $n$ inbound trucks to be assigned to a set $\mathcal{I}$ of $m$ docks in a distribution center. Upon arrival, trucks can be directly assigned to start processing at a dock, or to a waiting yard. Each truck has a known processing time $p_j$, and we assume processing at a dock is non-preemtive. The arrival time of each truck follows a distribution which is never observed by the decision maker. Instead, the decision maker has a *belief distribution* for each truck, which is periodically updated based on ETA information using a Bayesian filter. We formulate the resulting problem as a Markov Decision Process (MDP) with the objective of finding a policy that minimizes the total waiting time of all trucks. This is defined as the difference between the moment the truck starts being processed and the time it arrives to the yard. The main contributions of this study are as follows: (1) we propose an MDP model with tailored decision epochs and states to address the integration of the ETAs into the scheduling decisions; (2) to overcome the explosion in the dimension of the state and action spaces, we propose a heuristic solution method based on a rollout algorithm with an embedded Iterated Local Search (ILS) heuristic; (3) aside from a perfect information bound, we propose a penalized lower bound based on an information relaxation scheme; and (4) we demonstrate the effectiveness of our algorithm compared to the proposed lower bounds and benchmark scheduling policies, based on generated test instances.

## 2 MDP formulation

We assume a decision epoch $k$ is triggered by the first event among: (1) the arrival of a truck, (2) a truck finishes its process, i.e., a dock becomes available, and (3) *schedule verification* epochs. A schedule verification epoch is triggered when reaching an expected next arrival of a truck, before an actual truck arrival. An action $x_k$ is a feasible assignment of trucks to docks from the set of trucks that are either at the waiting yard or just arrived. Based on the action and the possible decision epoch triggers, a state $s_k$ transits from a post-decision $s_k^x$ to a pre-decision state $s_{k+1}$, updating belief distributions based on the ETA information. We define a waiting time function $W_k$ as the expected difference between the current and the next decision epoch times:

$$W_k(s_k, x_k) = \sum_{j \in \mathcal{J}_k^w} \mathbb{E}[t_{k+1} - t_k \mid s_k, x_k],$$

where $t_k$ is the time and $\mathcal{J}_k^w$ is the set of trucks in the waiting yard in decision epoch $k$. Since the next decision epoch time $t_{k+1}$ is unknown at $t_k$, the expectation is taken with respect to the belief arrival distribution of the incoming trucks. The objective is to find a policy $\pi$ in the set of feasible policies $\Pi$, defined as a sequence of truck-to-dock assignments, that minimizes the expected value of the waiting times from the initial state until a terminal state $K$:

$$V(\pi^*) = \min_{\pi \in \Pi} \mathbb{E}\left[\sum_{k=0}^{K} W_k(s_k, x_k^\pi(s_k)) \Big| s_0\right].$$

## 3 Rollout algorithm

A rollout algorithm is an adaptive online forward dynamic programming mechanism that makes real-time decisions for realized states by employing a lookahead decision rule. Following the early work of Bertsekas et al. [1], this methodology has been successfully applied in different dynamic applications [2]. The algorithm operates within a loop, which continues until the current state reaches a terminal state $K$. Within each iteration, a decision rule is evaluated based on the current states to determine the best action. We propose a decision rule that looks one decision epoch ahead in the future for all possible post-decision states $s_k^x$, generates heuristic policies $\pi(s_{k+1})$ based on the heuristic algorithm $\mathcal{H}$ for the set of all possible future states, and selects the action $x_k$ that minimizes the expected waiting time based on $\pi(s_{k+1})$. The set of possible future states is based on a competition between the arrival of all incoming trucks, where each event represents the case of one of the incoming trucks arriving first, or a dock becoming available first. A subset of the heuristic policy input and the probability of reaching the state $s_{k+1}$ are calculated based on the belief distributions. Both the probabilities and the expected arrival times used in $\mathcal{H}$ require conditional distributions on the minimum of the arrival distributions, which are intractable. To overcome this, we estimate their values using simulation.

The heuristic $\mathcal{H}$ solves the underlying deterministic scheduling problem, which is equivalent to the Parallel Machine Scheduling Problem with release dates (PMSPR). To solve the PMSPR, we propose an ILS heuristic that uses as input the expected arrival times of the incoming trucks given a reachable state $s_{k+1}$. The ILS uses the sequence of trucks based on the start service times of a solution, iterates feasible solutions applying a local search to produce local optima, and diversifies the solution. The algorithm starts with an initial solution, given by a First Come First Serve (FCFS) rule, which is improved through repeated iterations of a Variable Neighbourhood Descent (VND) search and perturbation phases. In the VND, we use four neighborhood structures based on swap and reinsertion moves in the sequence of trucks.

## 4 Lower bounds

In order to assess the performance of our algorithm, we propose two lower bounds based on information relaxations of the problem. First, we propose a bound in which we assume actual arrival times to be known at the beginning of the planning horizon for all trucks. We refer to this problem as the perfect information bound (PIB). The resulting problem is equivalent to the PMSPR. We obtain the PIB by proposing a sequence-based MILP. Second, similar to Brown et al. [3], we propose a tighter bound in which we assume the *distribution* of arrival times are known, while the actual arrivals are unknown. We refer to this problem as the penalized bound (PB). In the PB, we select a penalty function that compensates for the distribution information and derive a MILP formulation based on the original MDP formulation.

## 5 Results and discussion

We compare the expected objective value of the rollout policy with the PIB, PB, and the FCFS and the Shortest Processing Time First (SPTF) rules as a benchmark policies. For this, we compute the average over 10 realizations of arrival times for generated test instances, where each instance corresponds to one realization of processing times from a uniform distribution $p_j \sim \mathcal{U}(1, 20)$. We assume the actual arrival times are normally distributed, with mean parameters generated from a uniform distribution in the range $(0, \mathcal{P})$, such that $\mathcal{P} = \sum_{j \in \mathcal{J}} p_j / m$, and the variance parameters are set to 2. For each instance, we compute the gap between the objective function obtained by the tested policies using gap(%) $= 100 \frac{V_{\text{PB}} - V_x}{V_{\text{PB}}}$, where $V_{\text{PB}}$ and $V_x$ are the objective values of the PB and the given policy, respectively. The experiments were conducted on a computer with a 3.1

GHz Dual-Core Intel Core i5 with 16 GB of RAM.

In Table 1 we summarize preliminary results for test instances using $m = 2$ and $n = 10$, setting the belief distributions equal to the actual distributions. First, we observe that the PB provides a tighter lower bound compared to the PIB, increasing the average objective value from 39.6 to 42.2. Second, we observe that the rollout algorithm obtains lower or equal gaps compared to the benchmark policies in all cases, with an average gap of 2.5% compared to 23.9% and 9.4% for FCFS and SPTF, respectively. We will further evaluate the effect of the dynamics on generating realistic ETA update scenarios.

| | PIB | | PB | | FCFS | | SPTF | | Rollout | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst. | Obj. | Runtime (s) | Obj. | Time (s) | Obj. | Gap(%) | Obj. | Gap(%) | Obj. | Gap(%) | Time (s) |
| 1 | 49.1 | 2.1 | 52.7 | 26.1 | 61.5 | 16.8 | 54.4 | 3.3 | 53.7 | 2.0 | 70.5 |
| 2 | 83.5 | 4.4 | 89.4 | 76.1 | 113.6 | 27.1 | 89.4 | 0.0 | 89.4 | 0.0 | 70.2 |
| 3 | 28.3 | 0.4 | 29.4 | 2.3 | 41.4 | 40.9 | 38.3 | 30.2 | 30.9 | 5.2 | 55.0 |
| 4 | 8.8 | 0.5 | 9.1 | 2.0 | 10.1 | 10.2 | 10.1 | 10.3 | 9.5 | 4.3 | 66.4 |
| 5 | 28.3 | 0.5 | 30.6 | 4.0 | 38.0 | 24.1 | 31.6 | 3.3 | 30.9 | 1.0 | 64.0 |

Table 1: Comparison of the rollout algorithm with the PIB, PB, and benchmark policies.

## References


[1] D. Bertsekas, J. Tsitsiklis and C. Wu, “Rollout algorithms for combinatorial optimization”, *Journal of Heuristics* 3 (3), 245–262, 1997.

[2] J. Goodson, B. Thomas, and J. Ohlmann, “A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs”, *European Journal of Operational Research* 258(1), 216–229, 2017.

[3] D. Brown, J. Smith and P. Sun, “Information relaxations and duality in stochastic dynamic programs”, *Operations Research* 58(4), 785–801, 2010.

# Equitable Workload Allocation in Vehicle Routing Problem with Heterogeneous Drivers

**Iman Dayarian**
Culverhouse College of Business, University of Alabama, USA
Email: idayarian@ua.edu

## 1 Introduction

Efficient solutions are not necessarily equitable. Their acceptance and implementation may be contingent on a sufficiently fair distribution of resources, responsibilities, and benefits among different stakeholders. In fact, a solution that maximizes the sum of utilities of all the players might not be implementable, because some of the parties might consider it "unfair" as such a solution may be achieved at the expense of some players. In the conventional transportation planning and distribution contexts, the focus has been devoted to either efficiency or equity but rarely to both. In routing problems in the private sector, the most common equity considerations concern internal stakeholders, i.e., the drivers or other personnel providing the service [1]. The aim is to balance the workload allocation to ensure acceptance of operational plans in order to maintain employee satisfaction and morale, reduce overtime, and reduce bottlenecks in resource utilization. Practical examples include balancing the workload of service technicians [2], home healthcare professionals [3], and volunteers [4]. We, therefore, see a new fundamental research gap, which must be addressed and this research seeks to fill it. By focusing on the context of crowdsourced last-mile delivery, this research seeks to develop a novel solution approach for balancing efficiency and equity among crowdsourced drivers. In this context, a variant of the vehicle routing problem (VRP) in which drivers are independent contractors (crowdsourced) will be explored, where the equity indicator is measured based on the workloads assigned to different drivers and consequently benefits earned by them.

## 2 Problem Statement

The problem is defined on a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ and $\mathcal{A}$ represent the set of locations and the arcs within graph $\mathcal{G}$, respectively. A group of customers place orders online and expect deliveries at their home locations. Let $\mathcal{D} \subset \mathcal{N}$ be the set of delivery locations of the available orders, where $q_j$ denotes the quantity/size of the delivery at the location of customer $j \in \mathcal{D}$. The travel time between each pair of locations $(j, j'),\ j, j' \in \mathcal{N}$

for which an arc exists in $\mathcal{A}$ is given by $t_{jj'}$ while the transportation cost along an arc $(j, j')$, denoted by $c_{jj'}$, is proportional to $t_{jj'}$. Online orders are all fulfilled from a single depot $o$. The deliveries are performed by a set of available occasional drivers, denoted by $\mathcal{V}$. Each driver $v \in \mathcal{V}$ is characterized by their vehicle capacity, $Q_v$, their origin, $s_v$, and their destination, $e_v$, where they are headed after finishing the delivery task. A driver may be willing to dedicate a maximum of $\gamma t_{s_v,e_v}$ minutes to make deliveries, where $\gamma$ is an indication of the time flexibility of drivers. Drivers are compensated based on the deliveries they make (the prize received by serving a customer), and the mileage traveled.

This problem is a variant of the open vehicle routing problem (OVRP) [5] with a single depot in which the drivers have prefixed ending points. The set of feasible tasks and routing assignment $\mathcal{X}$ incorporates the assignment of delivery tasks to the drivers while specifying the sequence of visits to the customers (routing decision) by each driver. Suppose that $g(\boldsymbol{x})$ returns the mileage cost of a given routing solution $\boldsymbol{x}$. The company aims to maximize workload assignment equity while maintaining a certain level of efficiency. In our definition, a workload assignment is said to be equitable when *profit ratios* of the drivers employed are as close as possible to each other. The profit ratio of driver $v$ is defined as $\rho_v(\boldsymbol{x}) = p_v(\boldsymbol{x})/\overline{p}_v$, where $p_v(\boldsymbol{x})$ is the profit of employed driver $v$ under the workload assignment $\boldsymbol{x}$, and $\overline{p}_v$ is the maximum profit that the employed driver $v$ could possibly earn. In essence, $\overline{p}_v$ is representative of the potentials of driver $v$, given their personal characteristics including time flexibility, vehicle capacity, and origin and destination locations. The extra cost paid to improve workload assignment equity in a solution is called the *cost of equity*. We assume that the company is willing to accept a set of routes that involve up to $\alpha\%$ increase in the total mileage compensation compared to the least-cost solution if a more equitable profit distribution can be achieved. Therefore, any solution with a mileage cost within the interval $[z^*, (1+\alpha)z^*]$ is considered *efficient* for the company, where $z^*$ corresponds to the least possible mileage compensation given the set of customers $\mathcal{D}$ and the available fleet of drivers $\mathcal{V}$. The routing solution associated with $z^*$ does not necessarily employ all available drivers in $\mathcal{V}$ and may suggest using only a subset $\mathcal{V}^* \subseteq \mathcal{V}$. Once the subset of drivers $\mathcal{V}^*$ associated with $z^*$ is identified, we then aim to improve workload assignment equity in a solution that only involves drivers in $\mathcal{V}^*$. Notice that if a driver is not assigned any delivery tasks, his compensation is assumed to be zero.

Balancing profit ratios has been studied in the context of Nash Social Welfare (`NSW`), which has roots in game theory and pertains, in particular, to bargaining problems. In the solution of a bargaining problem the players, who compete for a higher gain, agree to form a grand coalition [6]. In this study, we aim at maximizing the profit ratio of drivers as the players of a bargaining game using Nash's method. Inspired by the expanded version of `NSW` [7], to equitably assign a set of delivery jobs among drivers in $\mathcal{V}^*$ (i.e., employed drivers), the problem can be formulated as $\max_{\boldsymbol{x}\in\mathcal{X}}\{\prod_{v\in\mathcal{V}^*}\rho_v(\boldsymbol{x}) : g(\boldsymbol{x}) \leq (1+\alpha)z^*\}$.

## 3 Solution Approach

To compute the minimum total mileage cost (MTMC), i.e., $z^*$, we propose to employ a branch-and-price approach. Let $\mathcal{R}_v$ be the set of all *feasible* routes for driver $v \in \mathcal{V}$. If a driver is selected by the company, he must first go to the depot to pick up the orders assigned to him and deliver them on the way to his destination. Let binary variable $x_{vr} := 1$ if driver $v \in \mathcal{V}$ is selected and is assigned route $r \in \mathcal{R}_v$; 0 otherwise. Binary parameter $\delta_{jr}$ equals 1 iff delivery of order $j \in \mathcal{D}$ is performed on route $r \in \mathcal{R}_v$ of driver $v \in \mathcal{V}$. Let $c_{r_v}$ be the mileage compensation that the company pays to driver $v \in \mathcal{V}$, and $p_{r_v}$ be the profit of driver $v$ from operating *feasible* route $r \in \mathcal{R}_v$. Using these notations, the path-based formulation of the VRP problem can be stated as Model (1).

$$
\begin{aligned}
(\mathtt{MTMC}) \quad z^* = \min & \sum_{v\in\mathcal{V}} \sum_{r\in\mathcal{R}_v} c_{r_v} x_{vr} && && \text{(1a)}\\
\text{s.t.} & \sum_{v\in\mathcal{V}} \sum_{r\in\mathcal{R}_v} \delta_{jr} x_{vr} = 1 && j \in \mathcal{D}, && \text{(1b)}\\
& \sum_{r\in\mathcal{R}_v} x_{vr} = 1 && v \in \mathcal{V} && \text{(1c)}\\
& x_{vr} \in \{0,1\} && v \in \mathcal{V}, r \in \mathcal{R}_v, && \text{(1d)}
\end{aligned}
$$

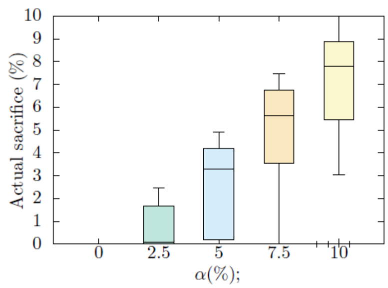
$$
\begin{aligned}
(\mathtt{NSW}) \quad \max & \prod_{v\in\mathcal{V}^*} \Big(\sum_{r\in\mathcal{R}_v} p_{r_v} x_{vr}\Big) && && \text{(2a)}\\
\text{s.t.} & \sum_{v\in\mathcal{V}^*} \sum_{r\in\mathcal{R}_v} \delta_{jr} x_{vr} = 1 && j \in \mathcal{D}, && \text{(2b)}\\
& \sum_{r\in\mathcal{R}_v} x_{vr} = 1 && v \in \mathcal{V}^*, && \text{(2c)}\\
& \sum_{v\in\mathcal{V}^*} \sum_{r\in\mathcal{R}_v} c_{r_v} x_{vr} \le (1+\alpha) z^*, && && \text{(2d)}\\
& x_{vr} \in \{0,1\} && v \in \mathcal{V}^*, r \in \mathcal{R}_v, && \text{(2e)}
\end{aligned}
$$

Constraints (1b)-(1d) guarantee that each customer is visited exactly once and each vehicle is assigned to exactly one route. Recall that $\mathcal{V}^* \subseteq \mathcal{V}$ is the subset of vehicles selected by the company as the result of minimizing its total cost through solving the MTMC problem. In order to compute the NSW solution for improving the workload equity for drivers selected through solving the MTMC problem (referred to as *drivers in the coalition*), we employ a second branch-and-price approach. The path-based formulation of NSW can be stated as Model (2). Observe that NSW is nonlinear. So, in order to employ a branch-and-price approach, it should be linearized. To solve the linearized version of NSW, once again, we can employ column generation. However, this column generation approach is substantially different from the one proposed for the MTMC problem because its subproblems are significantly more challenging than those available in the literature on classical VRPs.

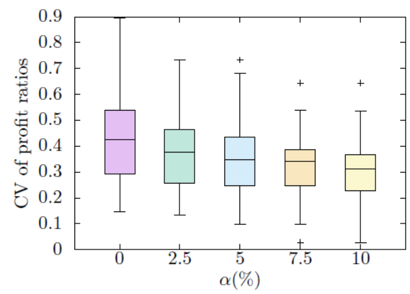
## 4 Preliminary Results and Conclusion

We generated random instances by adapting instance “R101” of Solomon’s VRP benchmark [8] with time windows, with different numbers of customers and drivers with different origins and destinations. We used the demand information in “R101” to set the quantity of deliveries, $q_j$. Figure 1a shows the actual sacrifice of the company in terms of cost for different values of $\alpha$. Observe that although the company has the opportunity to sacrifice $\alpha$% in terms of cost, the NSW may not be able to use all of it due to the discrete nature of the problem. The medians of the boxplots show that in 50% of instances about 75% of $\alpha$ is used. The fact that more equitable solutions are obtained can be seen from Figure

1b where it shows the medians of the Coefficient of Variance (CV) of profit ratios are dropped by about 30% when comparing $\alpha = 0\%$ and $\alpha = 10\%$. Our results show that the proposed method can effectively improve equity in workload allocation and consequently profit distribution among a fleet of heterogeneous drivers.




(a) Actual sacrifice of the company




(b) Equity measure - Coefficient of Variance

Figure 1: Sensitivity analysis for different values of $\alpha$.

# References

[1] Thibaut Vidal, Gilbert Laporte, and Piotr Matl. A concise guide to existing and emerging vehicle routing problem variants. *European Journal of Operational Research*, 2019.

[2] Fred Blakeley, Burçin Argüello, Buyang Cao, Wolfgang Hall, and Joseph Knolmajer. Optimizing periodic maintenance operations for schindler elevator corporation. *Interfaces*, 33(1):67–79, 2003.

[3] Shuguang Liu. A hybrid population heuristic for the heterogeneous vehicle routing problems. *Transportation Research Part E: Logistics and Transportation Review*, 54:67–78, 2013.

[4] Justin C Goodson. Election day routing of rapid response attorneys. *INFOR: Information Systems and Operational Research*, 52(1):1–9, 2014.

[5] Feiyue Li, Bruce Golden, and Edward Wasil. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & operations research*, 34(10):2918–2930, 2007.

[6] Roberto Serrano. Fifty years of the Nash program, 1953-2003. *Investigaciones Economicas*, 29(2):219–258, May 2005.

[7] Ehud Kalai. Nonsymmetric nash solutions and replications of 2-person bargaining. *International Journal of Game Theory*, 6(3):129–133, 1977.

[8] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

# Optimal Carsharing Zonification and Pricing

**Jiali Deng**
Department of Mathematical Sciences,
University of Copenhagen, Copenhagen, Denmark
Email: jd@math.ku.dk

**Giovanni Pantuso**
Department of Mathematical Sciences,
University of Copenhagen, Copenhagen, Denmark

## 1 Introduction

Carsharing pricing decisions have attracted significant attention in the research literature [1]–[6]. They have been identified as a promising instrument to resolve fleet imbalances [7], and improve profits and service rates. Among other things, prices are commonly differentiated geographically [4], [5], that is, dependent on the origin and/or destination of the rental. This typically implies that the business area is partitioned into distinct pricing zones that are independent of pricing decisions [1], [4] and provided a priori [5]. The decision of how to optimally divide a business area into pricing zones has not been investigated in detail.

In order to bridge this gap, in this talk, we focus on the problem of partitioning a set of carsharing stations into distinct pricing zones. The problem, which is motivated by an underlying industrial case, is considered within a one-way station-based carsharing system with a given fixed set of stations. For the service provider, the goal is thus to adjust prices and pricing zones periodically during the day, and for small intervals of time (e.g., every hour), in order to adapt to changes in demand patterns. The prices are differentiated by the origin and/or destination of the trip. Each pricing zone is a subset of the stations.

Motivated by the necessity to communicate the pricing mechanism in an easy and intuitive manner via mobile applications, the resulting partition must be such that the zones created form individual "islands" or, in other words, they are "visually disjoint". This particular requirement gives rise to a rich set partitioning problem. It shares similarities with tessellation [8] and districting problems [9], inspired by which, we formalize the zonification problem mathematically in Section 2. This, in turn, gives rise to a set of mixed integer linear constraints to define this type of partition. By including pricing decisions, we show that the resulting problem can be formulated as a (possibly nonlinear) MIP problem. A tailored integer Benders decomposition approach is developed to solve this problem exactly for which several problem-specific improvements are devised. In the experimental study, we demonstrate both the effectiveness of the integer Benders decomposition method and the co-optimized pricing and zonification decisions.

## 2 Mathematical Formulation

Given a discrete metric space $(\mathcal{I}, d)$ we are concerned with the problem of finding a special partition of $\mathcal{I}$ whose characteristics can satisfy the requirements of carsharing zonification sketched in Section 1 and are described in Definition 2.1.

**Definition 2.1 (Discrete tessellation)** *Let $\mathcal{G} \subseteq \mathcal{I}$. A collection $\mathcal{V}(\mathcal{G}) \subset 2^{\mathcal{I}}$ of subsets of $\mathcal{I}$ is called the discrete tessellation of $\mathcal{I}$ induced by $\mathcal{G}$ iff the following properties hold:*

1. (Disjunction) For every two sets $\mathcal{V}, \mathcal{U} \in \mathcal{V}(\mathcal{G})$ we have $\mathcal{V} \cap \mathcal{U} = \emptyset$
2. (Cover) $\bigcup \mathcal{V}(\mathcal{G}) = \mathcal{I}$
3. (One generator) For every $\mathcal{V} \in \mathcal{V}(\mathcal{G})$ we have $|\mathcal{V} \cap \mathcal{G}| = 1$
4. (Closest to generator) For every set $\mathcal{V} \in \mathcal{V}(\mathcal{G})$ let $c \in \mathcal{V}$ be the element such that $\{c\} = \mathcal{V} \cap \mathcal{G}$. Then, for every element $v \in \mathcal{V}$ of the set we have that $d(v, c) \leq d(v, k)$ for all $k \in \mathcal{G}$.

Properties 1 and 2 define a partition of $\mathcal{I}$ (i.e., a disjoint cover). Property 3 ensures that each set in the partition contains exactly one element of the set $\mathcal{G}$. The elements of $\mathcal{G}$ are thus understood as the *generators* of the tessellation. Finally, property 4 characterizes the partition as a tessellation. It states that each point in $\mathcal{I}$ is assigned to the subset that contains its closest generator from $\mathcal{G}$ in the sense of the metric $d$. Thus, each set of the partition $\mathcal{V}(\mathcal{G})$ contains the points of $\mathcal{I}$ that are closest to the single element of $\mathcal{G}$ in $\mathcal{V}$ than to any other element of $\mathcal{G}$ in the sense of the metric.

We are particularly concerned with the problem of finding an optimal tessellation according to some measure of performance. That is

$$\max_{\mathcal{G}} \left\{ R\big(\mathcal{V}(\mathcal{G})\big) \mid Properties(1) - (4) hold \right\} \tag{1}$$

where, $R$ is a mapping from the set of all partitions of $\mathcal{I}$ to the real numbers.

This type of discrete tessellation can be enforced by a mixed-integer linear formulation. We introduce binary variables $a \in \{0,1\}^{|\mathcal{I}| \times |\mathcal{I}|}$. Variable $a_{ii}$ takes value 1 if $i \in \mathcal{I}$ is designated as a generator, while variable $a_{ij}$ takes value 1 if element $j$ is a generator and element $i$ belongs to same subset as $i$. The set of all feasible discrete tessellations made of exactly $S$ subsets can be expressed using $\mathcal{O}(|\mathcal{I}|^3)$ mixed-integer linear constraints as follows:

$$\mathcal{T} := \left\{ a \in \{0,1\}^{|\mathcal{I}| \times |\mathcal{I}|} \;\middle|\; \begin{array}{cr} \sum_{j \in \mathcal{I}} a_{ii} = S & \\ \sum_{j \in \mathcal{I}} a_{ij} = 1 & \forall i \in \mathcal{I} \\ a_{ij} \leq a_{jj}, & \forall i, j \in \mathcal{I} \\ d(i, j_1) a_{i,j_1} \leq d(i, j_2) a_{j_2,j_2} + d(i, j_1)(1 - a_{j_2,j_2}) & \forall i, j_1, j_2 \in \mathcal{I} \end{array} \right\}$$

By using the above set of feasible discrete tessellations, we further include pricing decisions into the zonification problem. We denote the set of price levels by $\mathcal{L}$. Binary variables $\lambda_{ijl}$ take value 1 if price $l \in \mathcal{L}$ is applied between the zones generated by $i \in \mathcal{I}$ and $j \in \mathcal{I}$, 0 otherwise. Let $\alpha := (\alpha_{ijl})_{i,j \in \mathcal{I}, l \in \mathcal{L}}$ and $\lambda := (\lambda_{ijl})_{i,j \in \mathcal{I}, l \in \mathcal{L}}$. The pricing problem can be expressed as follows:

$$\max \; Q(a,\lambda,\alpha) \tag{2a}$$

$$\text{s.t.} \; \sum_{l\in\mathcal{L}} \lambda_{ijl} \geq a_{ii} + a_{jj} - 1, \qquad \forall i,j \in \mathcal{I} \tag{2b}$$

$$\sum_{l\in\mathcal{L}} \lambda_{ijl} \leq a_{ii}, \qquad \forall i,j \in \mathcal{I} \tag{2c}$$

$$\sum_{l\in\mathcal{L}} \lambda_{ijl} \leq a_{jj}, \qquad \forall i,j \in \mathcal{I} \tag{2d}$$

$$a_{i_1,j_1} + a_{i_2,j_2} + \lambda_{j_1,j_2,l} \leq \alpha_{i_1,i_2,l} + 2, \qquad \forall i_1,i_2,j_1,j_2 \in \mathcal{I}, \forall l \in \mathcal{L} \tag{2e}$$

$$\sum_{l\in\mathcal{L}} \alpha_{ijl} = 1, \qquad \forall i,j \in \mathcal{I} \tag{2f}$$

$$a \in \mathcal{T} \tag{2g}$$

$$\lambda, \alpha \in \{0,1\}^{|\mathcal{I}|\times|\mathcal{I}|\times|\mathcal{L}|} \tag{2h}$$

The function $Q(a,\lambda,\alpha)$ represents the performance (e.g. profits, service rates) obtained by the rentals occurred as a consequence of the prices. Constraints (2b)-(2d) ensure that a price level will be assigned to pair $(i,j)$ iff both $i$ and $j$ are designated as generators of a zone. Constraints (2e) ensure that, if $i_1$ is assigned to zone $j_1$ and $i_2$ is assigned to zone $j_2$, then the price level between zones $j_1$ and $j_2$ applies to stations $i_1$ and $i_2$. Constraints (2f) ensure that exactly one price level is applied to each station pair. The resulting problem (2) is, in general, nonlinear MIP problem. The linearity of the problem is dependent on the specification of function $Q(a,\lambda,\alpha)$.

# 3 Approach

We propose a tailored integer Benders decomposition (BD in what follows) to obtain exact solutions to the pricing zonification problem. The method is build upon the integer optimality cuts of the type introduced in [10]. A number of efficiency measures aimed to improve the BD method developed are devised. These are, namely, reformulation of feasible region $\mathcal{T}$ and addition of two types of valid inequalities.

# 4 Results

We perform experiments on instances based on a real carsharing service in the city of Copenhagen, Denmark, and find empirical evidence in terms of the performance of the decomposition method and the effect of joint zonification and pricing decisions.

We observe that on large instances BD significantly outperforms the solver as shown in Table 1. The average optimality gap of BD is 7.47% while that of the solver is 29.07%. The optimality gap reduction becomes more pronounced when $S$ decreases, thus when the feasible region is smaller. The decrease in the optimality gap is consistent across different numbers of customers. Furthermore, BD successfully closes the optimality gap (within the target tolerance) in 6 out of the 27 instances, while none of these instances is solved to optimality by the solver.

We compare the performance of the carsharing system when using the jointly optimized pricing and zonification decisions with two benchmarks. The *zip-code partition* benchmark is obtained by partitioning the carsharing system according to zip codes and keep the partition fixed when optimizing prices. The *no-partition* benchmark, however, is obtained by setting an uniformly optimized price for all stations and no partition of the business area is applied. The

Table 1: Average optimality gaps after 1800 seconds.

| | | | Gap (%) | | | Instances solved | |
|---|---|---|---|---|---|---|---|
| $\lvert\mathcal{K}\rvert$ | $\lvert\mathcal{I}\rvert$ | $S$ | Solver | BD | Reduction (%) | Solver | BD |
| 400 | 20 | 3 | 29.28 | 1.37 | 95.32 | 0/3 | 2/3 |
| 600 | 20 | 3 | 38.03 | 3.25 | 91.45 | 0/3 | 2/3 |
| 800 | 20 | 3 | 38.56 | 4.30 | 88.84 | 0/3 | 1/3 |
| 400 | 20 | 4 | 21.81 | 6.81 | 68.78 | 0/3 | 1/3 |
| 600 | 20 | 4 | 30.70 | 10.22 | 66.71 | 0/3 | 0/3 |
| 800 | 20 | 4 | 32.31 | 7.54 | 76.66 | 0/3 | 0/3 |
| 400 | 20 | 5 | 17.36 | 8.05 | 53.63 | 0/3 | 0/3 |
| 600 | 20 | 5 | 25.38 | 12.36 | 51.30 | 0/3 | 0/3 |
| 800 | 20 | 5 | 28.24 | 13.32 | 52.83 | 0/3 | 0/3 |

results in Table 2 show that by ensuring co-optimized zones and prices, the profit increases substantially compared to zip-code based partition and no-partition benchmarks, while the service rate is also significantly high, which turns to be 76.43% on average.

Table 2: Service rates and profits for the optimal partition compared to benchmarks.

| Case size | Optimal partition | | Zip-code partition | | No-partition | |
|---|---|---|---|---|---|---|
| | Service rate (%) | Profit (Euro) | Service rate (%) | Profit (Euro) | Service rate (%) | Profit (Euro) |
| $\lvert\mathcal{K}\rvert = 400$ | 78.13 | 201.82 | 81.25 | 182.46 | 56.25 | 152.34 |
| $\lvert\mathcal{K}\rvert = 600$ | 78.63 | 322.58 | 79.49 | 302.24 | 49.57 | 254.98 |
| $\lvert\mathcal{K}\rvert = 800$ | 72.53 | 417.56 | 81.69 | 395.64 | 55.63 | 342.60 |

# References


[1] B. Boyacı and K. G. Zografos, "Investigating the effect of temporal and spatial flexibility on the performance of one-way electric carsharing systems," *Transportation Research Part B: Methodological*, vol. 129, pp. 244–272, 2019.

[2] S. Zhang, H. Sun, X. Wang, Y. Lv, and J. Wu, "Optimization of personalized price discounting scheme for one-way station-based carsharing systems," *European Journal of Operational Research*, vol. 303, no. 1, pp. 220–238, 2022.

[3] M. Soppert, C. Steinhardt, C. Müller, and J. Gönsch, "Differentiated pricing of shared mobility systems considering network effects," *Transportation Science*, vol. 56, no. 5, pp. 1279–1303, 2022.

[4] D. Jorge, G. Molnar, and G. H. de Almeida Correia, "Trip pricing of one-way station-based carsharing networks with zone and time of day price variations," *Transportation Research Part B: Methodological*, vol. 81, pp. 461–482, Nov. 2015, ISSN: 01912615.

[5] G. Pantuso, "Exact solutions to a carsharing pricing and relocation problem under uncertainty," *Computers & Operations Research*, vol. 144, p. 105 802, 2022.

[6] C. Müller, J. Gönsch, M. Soppert, and C. Steinhardt, "Customer-centric dynamic pricing for free-floating vehicle sharing systems," *Transportation Science*, vol. 0, no. 0, null, 2023.

[7] S. Illgen and M. Höck, "Literature review of the vehicle relocation problem in one-way car sharing networks," *Transportation Research Part B: Methodological*, vol. 120, pp. 193–204, 2019.

[8] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, "Spatial tessellations: Concepts and applications of voronoi diagrams," 2009.

[9] J. Kalcsics and R. Z. Ríos-Mercado, "Districting problems," *Location science*, pp. 705–743, 2019.

[10] G. Laporte and F. V. Louveaux, "The integer l-shaped method for stochastic integer programs with complete recourse," *Operations research letters*, vol. 13, no. 3, pp. 133–142, 1993.

# The electric vehicle routing and overnight charging scheduling problem on a multigraph

**Guy Desaulniers**
Department of Mathematics and Industrial Engineering and GERAD
Polytechnique Montréal, Montréal, Canada
Email: Guy.Desaulniers@gerad.ca

**Daniel Yamín**
Departamento de Ingeniería Industrial and COPA
Universidad de los Andes, Bogotá, Colombia

**Jorge E. Mendoza**
Department of Logistics and Operations Management
HEC Montréal, Montreal, Quebec, Canada

## 1 Introduction

In recent years, logistics companies that deal with goods distribution in urban city centers have progressively transitioned from internal combustion engine vehicles (ICEVs) to much more sustainable electric vehicles (EVs). Despite a vast amount of capital investment from automakers and battery manufacturers, EVs still have significant disadvantages over ICEVs, such as a high acquisition cost, limited driving range, slow recharging times, and lack of comprehensive recharging infrastructure. While the acquisition cost is expected to decrease over the following years, the restrictive autonomy and recharging hassles need to be addressed before the wide adoption of EVs in urban logistics.

To deal with EVs' limitations, researchers have investigated two broad approaches. The first one is to plan recharging stops along a route [1] and assumes that a widespread recharging infrastructure is in place. However, that is not the case in most urban centers, and due to the growing lack of urban spaces, an adequate charging infrastructure might not be possible soon. In addition, deliveries are typically carried out during the day when time-dependent energy costs are more expensive, en route charging can yield undesirable driver's idle times, and the autonomy of newly available EVs is typically sufficient to perform delivery routes in urban areas. For these reasons, operators usually prefer charging their vehicles at their facilities and overnight [2]. Thus, the second approach, which consists in designing routes that can be completed with a full battery load, can be more realistic in the context of inner-city logistics. This approach gives birth to a new breed of coupled routing and charging scheduling problems in which the decision maker must simultaneously plan the routes and the charging operations. These problems are especially challenging to solve because the two components are intimately intertwined. For instance, designing long routes may be ideal from a routing cost point of view, but those routes will likely require long charging operations that may be impossible to schedule due to the number of chargers available at the depot. Also, good coordination between the routing and the scheduling components is capital to preserve the life of the EVs' batteries. As pointed out in [3], to prevent *calendar aging*, a charging operation should finish as close as possible to the departure time of the associated vehicle.

In light of the discussion above, we introduce the multigraph-based electric vehicle routing and overnight charging scheduling problem with time windows (mE-VRSPTW), in which a fleet of EVs must be routed to serve a set of customers during a given day, and their overnight charging operations scheduled such that the total traveled distance is minimized. Besides the usual load and time window constraints, routes must comply with energy requirements. Furthermore, the EVs are recharged prior to performing their routes, using a limited number of identical chargers located at the depot. The recharging process of each EV occurs non-preemptively on a single charger and according to a piecewise-linear recharging function. The problem is defined on a multigraph, meaning that there are alternative arcs (i.e., paths on the road network) to travel between two locations. This representation captures the trade-off between consumed energy and traveled distance.

## 2 Mathematical model and solution method

To formulate the mE-VRSPTW, we use the following notation. Let $\mathcal{C}$ be the set of customers to service and $B$ the number of chargers available at the depot. For the charging operations, we discretize the planning horizon into a set of timesteps $\mathcal{T}$. Each charging operation must start (resp. end) at the beginning (resp. end) of a timestep. Furthermore, let $\mathcal{R}$ be the set of all feasible routes, where a route includes its overnight charging operation. For each route $r \in \mathcal{R}$, we define the following parameters: its traveled distance $c_r$; for each customer $i \in \mathcal{C}$, a binary indicator $a_i^r$ equal to 1 if customer $i$ is visited in route $r$; for each timestep $t \in \mathcal{T}$, a binary indicator $b_t^r$ equal to 1 if the vehicle assigned to route $r$ charges during timestep $t$. We also introduce a binary variable $\lambda_r$ that takes value 1 if route $r$ is selected in the solution and 0 otherwise.

With this notation, the mE-VRSPTW can be formulated as the following route-based set-partitioning model:

$$\min \quad \sum_{r \in \mathcal{R}} c_r \lambda_r \tag{1}$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}} a_i^r \lambda_r = 1 \qquad \forall i \in \mathcal{C}, \tag{2}$$

$$\sum_{r \in \mathcal{R}} b_t^r \lambda_r \leq B \qquad \forall t \in \mathcal{T}, \tag{3}$$

$$\lambda_r \in \{0, 1\} \qquad \forall r \in \mathcal{R}. \tag{4}$$

Objective function (1) minimizes the total traveled distance. Constraints (2) ensure that each customer is visited exactly once, whereas constraints (3) limit the number of vehicles recharging in each timestep by the number of available chargers.

To solve the mE-VRSPTW based on this formulation, we devise a branch-price-and-cut (BPC) algorithm (see [4]), i.e., a branch-and-cut algorithm where column generation (CG) is applied to solve the linear relaxations. In our case, the CG master problem, namely, the linear relaxation of (1)–(4), is reinforced with rounded capacity and subset-row inequalities. The CG pricing problem is an elementary shortest path problem with resource constraints which is defined on a network containing two subnetworks. The charging scheduling subnetwork is acyclic and has one node per timestep, whereas the routing subnetwork has one node per customer and may contain cycles. A path representing a feasible route visits a sequence of charging nodes before visiting some customer nodes. To find integer solutions, we derive specific-purpose branching rules.

To ease the solution of the pricing problem, we employ the well-known *ng*-path relaxation [5]. To solve this relaxed pricing problem, we develop a specialized backward labeling algorithm that relies on two dominance rules accounting for energy consumption and recharging time and that differ depending in which subnetwork it is applied. We also devise a heuristic labeling algorithm that works on a reduced network and applies simpler dominance criteria.

## 3 Computational results

Through extensive computational experiments on mE-VRSPTW instances derived from the Solomon's benchmark instances, we demonstrate that the BPC algorithm can efficiently solve instances with up to 50 customers (in at most 53 minutes). The good performance of the algorithm can be attributed to the fact that it uses some of the state-of-the-art techniques for vehicle routing BPC-based algorithms. Furthermore, we analyze the impact of charging scheduling on computational time. The problem without charging scheduling is solved about three times faster, corroborating that the mE-VRSPTW is a very challenging problem from a theoretical and computational perspective due to its combined routing and scheduling structure. In addition, EV charging scheduling at companies' facilities with limited chargers can be difficult. Thus, it must be addressed when planning EV routing. Moreover, we study the impact of multigraph representation on the optimal value. Using this representation, routing models can find better solutions by having alternatives to balance conflicting resources. Yet, this improvement in solution quality is at the expense of larger computational times.

The central role of EVs in the urgent task of replacing fossil fuels with renewable energy sources is transforming the nature of routing and scheduling problems, exposing several fascinating research opportunities. From an application standpoint, our results show that the BPC algorithm could be used as an analytical tool in real-world goods distribution schemes.

## References


[1] N.D. Kullman J.C. Goodson and J.E. Mendoza, “Electric vehicle routing with public charging stations”, *Transportation Science* 55(3), 637–659, 2021.

[2] E. Morganti and M. Browne, “Technical and operational obstacles to the adoption of electric vans in France and the UK: An operator perspective”, *Transport Policy* 63, 90–97, 2018.

[3] S. Pelletier, O. Jabali, G. Laporte, “Charge scheduling for electric freight vehicles”, *Transportation Research Part B* 115, 246–269, 2018.

[4] L. Costa, C. Contardo, G. Desaulniers, “Exact branch-price-and-cut algorithms for vehicle routing”, *Transportation Science* 53(4), 946–985, 2019.

[5] R. Baldacci, A. Mingozzi, R. Roberti, “New route relaxation and pricing strategies for the vehicle routing problem”, *Operations Research* 59(5), 1269–1283, 2011.

# Submodular Dispatching with Multiple Vehicles

**Ignacio Erazo** **Alejandro Toriello**
Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia, USA 30332

## Motivation

The rise of e-commerce has put a spotlight on the last mile of the order fulfilment process, which can represent up to 50% of total logistics costs. E-commerce has also played a major role in the significant increase in the number and size of fulfillment centers and warehouses; order picking is of similar importance within these facilities, accounting for up to 55% of operating costs. A common element in many processes within an e-commerce supply chain, including same-day delivery (SDD), order picking and shelf re-stocking, is the need to dispatch (i.e. deliver, process, pick or re-stock) orders or items that become available at different times, but where batching yields economies of scale in dispatching time; recently, [ET23] proposed the Submodular Dispatching Problem (SMD) and its generalizations to model the tension between these two elements, focusing on the case in which one vehicle (or picker, or server) dispatches or processes orders. In this work, we expand the model to the case of many identical vehicles.

## Problem Statement

The problem is characterized by (i) a set of orders $N := \{1, 2, \ldots, n\}$ that must be served by a fleet of $m$ identical, uncapacitated delivery vehicles, (ii) a release time $r_i \in \mathbb{R}_+$ for each order $i \in N$, where we assume without loss of generality that $0 = r_1 \leq r_2 \leq \ldots \leq r_n$, and (iii) a function $f : 2^N \to \mathbb{R}_+$ indicating the time it takes a vehicle to dispatch a batch of orders, where

$$0 \leq f(S) \leq f(S \cup i),\ S \subseteq N \setminus i,\ i \in N \text{ (non-negative, non-decreasing)},$$

$$f(S) + f(S \cup \{i, j\}) \leq f(S \cup i) + f(S \cup j),\ S \subseteq N \setminus \{i, j\},\ i, j \in N \text{ (submodular)}.$$

A solution consists of a partition of $N$ into batches and an assignment of each batch in the partition to one of the $m$ vehicles. A batch cannot be dispatched before all of its orders are available, and a vehicle can be dispatched with at most one batch at a time. The objective is to minimize the makespan, the time at which all vehicles have completed their dispatches. This problem is strongly NP-hard; it generalizes, among others, parallel machine scheduling with a makespan objective [Mut20], tactical design for SDD systems with a makespan objective [SET22], multiple-picker order-picking in a warehouse with a tree topology, and machine scheduling with family setup times and a makespan objective [KIL21].

## MILP Formulation

SMD with multiple vehicles can be modeled as a mixed-integer linear program (MILP); we use $x_{S,k} \in \{0,1\}$ to indicate if batch $S \subseteq N$ is dispatched by vehicle $k \leq m$. There can be at most $n$ dispatches, and using $N_i := \{1, 2, \ldots, i\}$, if batch $S \subseteq N_i$ with $S \ni i$ is dispatched, we call this the $i$-th dispatch; that is, we index a dispatched batch by its latest order. We use $t_{i,k} \in \mathbb{R}$ to denote the departure time of the $i$-th dispatch if it is executed by vehicle $k$; we also let $z$ be the makespan.

**Proposition 1.** *The MILP* (1) *solves SMD with multiple vehicles:*

$$
\begin{aligned}
\min \quad & z \\
\text{s.t. } & t_{i,k} \geq r_i && \forall i \in N,\ \forall k = 1, \ldots, m && \text{(1a)} \\
& t_{i+1,k} \geq t_{i,k} + \sum_{S \subseteq N_i, S \ni i} x_{S,k} f(S) && \forall i \in \{1, \ldots, n-1\},\ \forall k = 1, \ldots, m && \text{(1b)} \\
& z \geq t_{n,k} + \sum_{S \subseteq N_n, S \ni n} x_{S,k} f(S) && \forall k = 1, \ldots, m && \text{(1c)} \\
& \sum_{k=1}^{m} \sum_{j \geq i} \sum_{\substack{S \subseteq N_j \\ S \ni \{i,j\}}} x_{S,k} = 1 && \forall i \in N && \text{(1d)} \\
& z \geq 0, t \geq 0, x \in \{0,1\}
\end{aligned}
$$

*Furthermore, the linear relaxation can be solved in polynomial time.*

The linear relaxation must be solved using column generation. In particular, for each $i, k$, the separation problem for the $x_{S,k}$ variables with $S \subseteq N_i$ and $S \ni i$ is

$$
\min_S \left\{ \beta_{i,k} f_k(S) - \sum_{j \in S} \gamma_j : i \in S \subseteq N_i \right\} = \min_S \left\{ \beta_{i,k} f_k(S \cup i) - \gamma_i - \sum_{j \in S} \gamma_j : S \subseteq N_{i-1} \right\},
$$

where $\beta \geq 0$ and $\gamma$ are dual variables. This is a submodular minimization problem, which can be solved in polynomial time.

## Interval-Solvable Functions

A solution is of interval type if all dispatched batches have a minimum index $i$, a maximum index $j \geq i$, and the batch dispatches all orders in the interval $[i, j]$. We define a function

$f$ to be interval-solvable if any instance with dispatch time function $f$ has an optimal solution of interval type.

**Proposition 2.** *Let $\tau_i > 0$ for $i \in N$, $\tau_0 \geq 0$, and let $g$ be a concave, non-decreasing function with $g(0) = 0$. The following submodular functions are interval-solvable:*

$$1.\ f(S) = \sum_{i \in S} \tau_i, \qquad 2.\ f(S) = \max_{i \in S}\{\tau_i\} + \tau_0, \qquad 3.\ f(S) = g(|S|) + \tau_0.$$

Proposition 2 identifies classes of problems for which the MILP (1) only requires a polynomial number of variables $x_{S,k}$.

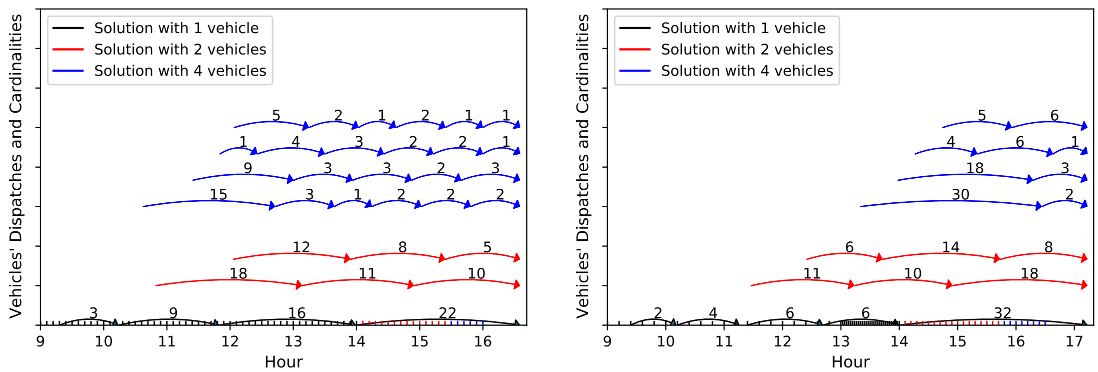
## Application: Tactical Design for SDD

We study the effect of fleet size on an SDD system, with the goal of obtaining insights for tactical design. We use a case study from [SET22], where the dispatch time approximation is given by $f(S) = 10 + 1.5|S| + 24\sqrt{|S|}$ minutes, and 50 orders are expected to arrive between 9 AM and 2 PM. This function is interval-solvable, which allows us to solve (1) to optimality by explicitly including all interval batch variables in the formulation.

Assume the delivery fleet has $m_1$ vehicles, resulting in makespan $\phi^*$. Suppose we increase the fleet to $m_2 > m_1$. We seek to serve as many orders as possible with a makespan $\phi \leq \phi^*$. How does the structure of the solution change? By how much can we increase the order window beyond 2 PM, and how many extra orders do we serve? We study these questions by studying two different scenarios: (a) a constant order arrival rate during the initial ordering period (9 AM to 2 PM), and (b) order arrivals exhibit a peak at the end of the order window, between 1 PM and 2 PM. For both scenarios (a) and (b) we set the arrival rate of orders after 2 PM to be constant and equal to the rate of the initial ordering period (one every six minutes).

Figure 1 illustrates the experiment results. For both scenarios, as we increase the fleet size, the number of orders dispatched increases and the ordering period can be extended; however the structure of the new optimal solutions strongly depends on the arrival process before the original deadline of 2 PM.

For scenario (a), when the fleet increases from one to two vehicles, the average batch size is not affected significantly, but the starting time of the first dispatch is delayed by almost 90 minutes. This indicates that the system is able to reduce the window of time in which delivery vehicles are expected to operate, while increasing the number of served orders from 50 to 64. On the other hand, increasing from two to four vehicles significantly reduces the average batch size and thus the dispatch efficiency; the first dispatch with four vehicles also leaves earlier than the corresponding first dispatch for two vehicles in the uniform arrival case, and the total number of orders dispatched only increases from 64 to 70.




(a) Uniformly distributed arrivals.

(b) Arrival peak between 1-2 PM.

Figure 1: Effect of adding vehicles to the delivery fleet, depending on arrival rates.

For scenario (b), when going from one to two vehicles the dispatches lose monotonicity on their cardinality, but the balancing of loads between dispatches is improved and the start of the first dispatch is delayed by 2 hours; furthermore the number of orders dispatched increases by 17. When going from two to four vehicles, the efficiency over dispatches becomes similar to the one-vehicle scenario; however, the earliest departure time is now delayed by 4 hours, and the total number of orders dispatched is 75.

These results suggest that some arrival patterns are more likely to benefit from additional vehicles; the system may significantly increase the number of orders dispatched, maintain a roughly equal workload and serve batches of reasonable size, all while delaying the start of delivery operations. The latter effect may be important in allowing the company to share the fleet across multiple services, e.g. by operating next-day delivery in the morning and SDD in the afternoon.

# References

[ET23] Ignacio Erazo and Alejandro Toriello. Optimizing the trade-off between batching and waiting: Subadditive dispatching. Working paper, 2023.

[KIL21] Arthur Kramer, Manuel Iori, and Philippe Lacomme. Mathematical formulations for scheduling jobs on identical parallel machines with family setup times and total weighted completion time minimization. *European Journal of Operational Research*, 289(3):825–840, 2021.

[Mut20] İbrahim Muter. Exact algorithms to minimize makespan on single and parallel batch processing machines. *European Journal of Operational Research*, 285(2):470–483, 2020.

[SET22] Alexander M. Stroh, Alan L. Erera, and Alejandro Toriello. Tactical design of same-day delivery systems. *Management Science*, 68(5):3444–3463, 2022.

# Learning to Optimize Load Plans with Volume Splitting

**Alan Erera**, Ritesh Ojha, Wenbo Chen, Hanyu Zhang, Pascal Van Hentenryck
H. Milton Stewart School of Industrial and Systems Engineering,
Georgia Institute of Technology, Atlanta, Georgia, USA
Email: alan.erera@isye.gatech.edu

Reem Khir
School of Industrial Engineering,
Purdue University, West Lafayette, Indiana, USA

## 1 Tactical Load Planning with Volume Splitting

E-commerce continues to show robust growth; Morgan Stanley projects the global market to grow from \$3.3 trillion in 2022 to \$5.4 trillion in 2026. Home delivery of small packages and parcels is critical for the success of e-commerce, and efficient transportation of small shipments over long distances is enabled by effective *freight consolidation networks*. Consolidation networks operated by today's largest package and LTL carriers move volume from hundreds of possible origin terminals to hundreds of possible destination terminals. Each shipment in these networks is transported either by a direct *load* or by a sequence of loads forming a *transfer path* from origin to destination with intermediate transfers (unload-sort-reload). For background on consolidation network planning, see the review in [1].

In this work, we consider a service network design problem called *tactical load planning with volume splitting*. Load planning is to determine how many trailer or container loads (perhaps of different equipment types) to plan for dispatch over time between pairs of terminals. Here, we suppose that so-called flow planning decisions are known. A flow plan specifies a transfer path for each shipment, where a transfer path consists of one or more dispatches (and intermediate sorts) through a time-expanded network and ensures on-time arrival. Determining a cost-effective flow plan is a challenging stochastic optimization problem for large carriers like UPS or FedEx with hundreds of terminals, given daily variation in inducted shipment volume; we refer the reader to [3, 4, 2] for optimization methods.

Flow plans typically specify a *unique* transfer path for volume at a terminal with the same destination and service class; thus, it is possible to project total volumes moving between pairs of terminals and to containerize into loads. However, large carriers have an important cost-saving advantage simply because they operate so many terminals: shipments need not follow this unique *primary* flow path since many *alternate* transfer paths exist that also ensure on-time arrival. Carriers take advantage of alternates by building load plans that assume volume splitting at a terminal, where package volume is split into loads following the primary flow path as well as loads following feasible alternates.

We work with one of the largest package carriers in the US which operates a network with over 1,200 interconnected package sorting terminals moving roughly 10 million packages per day. We focus on building a load plan for a week (or more) given pre-specified flow decisions (primary and alternate flow paths). Each terminal in this network operates between 2 to 4 sorting periods (or *sorts*) each day, and several load types (for example, short, long, and extra-long trailers and containers) are used to containerize sorted volume for outbound dispatch during each sort. Packages are grouped into four service classes (next-day, 2-day, 3-day, and ground). Given a volume projection and the option to split volume across primary and alternate flow paths, it is a challenging optimization problem to determine an optimal load plan covering all terminals and sorts for a week. In practice, load planners focus on small portions of the network at a time and build up a load plan terminal by terminal. The first goal of this research effort is to automate this manual planning process using optimization technology. Since *single-terminal* load planning problems with volume splitting can still be difficult to optimize exactly for medium or large terminals, the second goal is to develop an approach that uses machine learning to train *optimization proxies* for this task.

## 2 Optimization Models and Trained Optimization Proxies

Consider a single-terminal load planning problem with volume splitting over a set of consecutive sorting periods (for example, a single operating day). Given a point forecast of newly-arriving (inducted) volume and upstream terminal transfer volume, two decisions are modeled: how many loads of each type should be built on outbound lanes during each sort, and how should shipment volume be split across primary and alternate lanes such that the built loads feasibly containerize all volume. Note that a *lane* is a single arc in a transfer path for a shipment. Total inbound volume is grouped into *commodities*, where each commodity $k$ represents all shipments arriving during a specific sort with a common final destination $d$ and service class $s$; all shipments of commodity $k$ share the same primary and alternate flow paths.

A generic mixed-integer programming model for single-terminal outbound load plan optimization is given by (1):

$$\begin{aligned}
\underset{x,y}{\text{Minimize}} \quad & \sum_{a \in A} \sum_{v \in V_s} c_{a,v} y_{a,v} && && \text{(1a)}\\
\text{subject to} \quad & \sum_{a \in A^k} \sum_{v \in V_a} x^k_{a,v} = q^k, && \forall k \in K, && \text{(1b)}\\
& \sum_{k \in K : a \in A^k} x^k_{a,v} \leq Q_v \, y_{a,v}, && \forall a \in A, v \in V_a, && \text{(1c)}\\
& x^k_{a,v} \geq 0 && \forall k \in K, a \in A^k, v \in V_a, && \text{(1d)}\\
& y_{a,v} \in Z_{\geq 0} && \forall a \in A, v \in V. && \text{(1e)}
\end{aligned}$$

Continuous variables $x^k_{av}$ split commodity $k$ volume across the compatible load lanes $A^k$ (primary and alternates), and integer variables $y_{a,v}$ specify the number of loads to plan on lane $a$ of type $v$. The example objective function in (1a) minimizes the total cost of all planned loads. Finally, constraints (1b) ensure that all commodity volume $q^k$ is assigned to a compatible load lane while (1c) ensures that the total capacity planned on a lane is sufficient to containerize all assigned capacity, where $Q_v$ is the capacity of one load of type $v$. In this research, we also consider variants of (1) that seek to modify a pre-specified *base load plan* by adding or cutting some loads of type $v$ from lanes $a$.

Solving variants of (1) is challenging for practically-sized instances. Consider three example test instances each representing a single operating day of four sorts for terminals of different sizes. Note that sorts are frequently not independent because loads created during a sort can sometimes be held for continued loading in the subsequent sort. Table 1 summarizes the number of outbound load lane arcs, the number of commodities, and the number of planned loads from the base plan for these instances. The total volume processed by the large terminal is about 11 times that of the medium terminal, and the extra-large terminal processes 40 times more than the medium. Solving

Table 1: Characteristics of Test Instances

| Category | #Arcs | #Commodities | #Loads in Base Plan |
|---|---|---|---|
| M | 92 | 9,000 | 150 |
| L | 399 | 15,000 | 550 |
| XL | 1,602 | 20,000 | 2,000 |

(1) directly with a modern solver like Gurobi leads to optimality gaps of up to 2% after 60s of solve time, and gaps often persist due to poor lower bounds. A more major issue is that models variants that seek to find solutions that do not deviate far from a base load plan are even more difficult to optimize; such variants are critical because it is important in practice that optimized load plans do not vary dramatically given minor variations in the input data. We thus also explore using *trained optimization proxies* to learn near-optimal load planning solutions.

When building an optimization proxy for models like (1), the goal is to train a machine learning model to produce a near-optimal solution $(\hat{x}, \hat{y})$ solely from the input vector $(q)$. Note we train separate models for each terminal; thus, other inputs like $c$, $A^k$, and $Q^v$ are fixed and known. An important challenge is that the optimal load plan must be *feasible*; the loads $\hat{y}$ must be sufficient to allow a feasible volume split to containerize the input volumes $q^k$. We thus first predict near-optimal loads $\hat{y}$, and then solve subsequent but simpler optimization models to ensure feasibility. Note that given $\hat{y}$, model (1) is a linear program. We formulate this LP with auxiliary variables to guarantee feasibility by adding capacity to each lane; if the solution has positive-valued auxiliary variables (and is thus infeasible), we solve a simpler binary integer program that minimizes the cost of adding extra capacity to a subset of capacity-infeasible lanes.

We train each of the optimization proxy models using 8,000 instances built by varying the input volumes to represent real-world volume variations. To build the training data set, each of these instances is solved with Gurobi until optimality or a 3,600s time limit. The proxy models are multiple layer perceptron neural networks and are hyperparameter-tuned using a grid search with learning rate in $\{10^{-1}, 10^{-2}\}$, number of layers in $\{3, 4, 5\}$, and hidden dimension in $\{128, 256\}$. They are implemented using PyTorch and trained using the Adam optimizer to minimize an L1 loss function between the predicted solution and the Gurobi solution.

## 3 Results Preview

A comprehensive computational study examines both the effectiveness of our load planning methodology and the promise of optimization proxies for enabling practical use of optimization technology in real-world application. First, our results demonstrate that volume splitting can lead to substantial practical benefits. When all volume is restricted to its primary lanes, roughly 30% more capacity must be planned than we find in our optimized solutions with volume splitting. Our partner provided test instances where manual load planners already adjusted a base plan by moving some volume to alternates. Nevertheless, our optimization technology finds an additional 10-15% of capacity reduction by shifting more volume to alternates. This represents significant cost savings.

Table 2: Optimality Gap (%)

| Instance | Model (1) | | | | | | GH | | Proxies | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1s | 5s | 10s | 30s | 60s | 1800s | Gap | Time (s) | Gap | Time (s) |
| M | 2.59 | 0.55 | 0.48 | 0.48 | 0.48 | 0.48 | 3.84 | 3.12 | 1.14 | 0.33 |
| L | 51.15 | 5.22 | 2.18 | 1.71 | 1.41 | 1.39 | 12.85 | 13.28 | 3.80 | 1.10 |
| XL | 77.35 | 14.02 | 10.41 | 2.93 | 2.07 | 0.93 | 17.01 | 121.55 | 5.21 | 2.49 |

Table 2 demonstrates the effectiveness of the trained optimization proxies on these problems. When solving (1) directly, gaps are reported for each instance size up to 1,800s of solve time. The Proxies columns provide the provable gaps from the optimization proxies after feasibility recovery; the solve times here include the recovery step. While the costs of solutions found by the proxy are slightly higher, the average time to find these solutions given a trained model is quite fast.

## 4 Conclusions and Ongoing Work

Single-terminal load planning with volume splitting optimization problems can be effectively learned and represented by a machine-learning optimization proxy. Such trained optimization proxies can be very useful in practice, returning near-optimal solutions to network load planners in seconds given trained models. Implementation of such a system requires significant investment, however, since proxies must be trained and re-trained when fundamental system parameters begin to deviate substantially from those underlying the initial training data. Detection of such changes is also important. Ongoing work is examining load planning decisions for a small cluster of interacting terminals simultaneously and includes some additional decisions for shifting volumes between the terminals. Such problems are more challenging because outbound changes for one terminal affect inbound volumes at the interacting terminals. If time permits, these results will also be presented.

## References


[1] Ilke Bakir, Alan Erera, and Martin Savelsbergh. Motor carrier service network design. In *Network Design with Applications to Transportation and Logistics*, pages 427–467. Springer, 2021.

[2] Natashia Boland, Mike Hewitt, Luke Marshall, and Martin Savelsbergh. The continuous-time service network design problem. *Operations research*, 65(5):1303–1321, 2017.

[3] Alan Erera, Michael Hewitt, Martin Savelsbergh, and Yang Zhang. Improved load plan design through integer programming based local search. *Transportation Science*, 47(3):412–427, 2013.

[4] Kathleen Lindsey, Alan Erera, and Martin Savelsbergh. Improved integer programming-based neighborhood search for less-than-truckload load plan design. *Transportation science*, 50(4):1360–1379, 2016.

# A Deep Reinforcement Learning Algorithm for the Vehicle Routing Problem with Stochastic Demands and Outsourcing

**Mohsen Dastpak**[1], **Fausto Errico** [1], **Ola Jabali**[2]
[1] École de technologie supérieure, Canada, [2]Politecnico di Milano, Italy

## 1 Introduction

We consider a daily distribution problem faced by a logistics service provider (LSP) operating a fixed fleet to service customers whose demand is stochastic. In particular, the set of customers is variable from one day to another. At the beginning of each day a set of customers along with their demand distributions are revealed. Given this set, the LSP *commits* to servicing a subset of customers, and *outsources* the remaining subset at a given cost. The LSP operates its fleet under daily work shift duration constraints, which may be violated at an overtime cost. We denote the resulting problem as the VRP-DO. This problem is relevant for last-mile applications where daily outsourcing decisions are pivotal and customer demand is uncertain.

We model the VRP-DO as a two-level problem. The first level determines the committed and outsourced customers. Given the set of committed customers, the second-level problem, denoted by VRP-D, establishes the LSP's expected routing costs. To minimize such costs, considering the stochastic customer demands, we assume that the LSP operates its fleet in a dynamic fashion. Thus, we formalize the VRP-D as a Markov Decision Process (MDP).

With the aim of providing an efficient daily tool to optimize the VRP-DO, we propose a heuristic algorithm denoted by I-DQNCO. Following the two-level structure of the problem, this algorithm features an Iterated Local Search (ILS) algorithm that evaluates different committed and outsourced customer partitions. The evaluation of such partitions is fairly complex, as it entails solving a dynamic vehicle routing problem with stochastic demands. Since this complex problem is cast in a local search setting and must be evaluated numerous times, we propose a tailored Deep Q Network-based algorithm (DQNCO) for it. The DQNCO enables evaluating the expected costs of the committed customers instantaneously, and thus can be deployed within a local search algorithm. However, the DQNCO requires an intensive training phase, which may entail several days of computing time. This is in conflict with the need to solve the VRP-DO on a daily basis. To overcome this limitation, we modify the MDP formulation of the VRP-D in such a way to enable the

DQNCO to handle random customer sets drawn from a suitable probability distribution. By doing so, our DQNCO is trained off-line once, allowing the I-DQNCO to be readily used on a daily basis.

We observe that the two-level decision structure displayed by VRP-DO is a rather common feature among combinatorial problems. In many of these, the second-level decision problem is itself a very complex and often stochastic dynamic combinatorial problem. We believe that one of the major contributions of this paper is to put forward the idea that a Q Network can successfully be used to obtain an almost instantaneous and accurate estimation of the second-level problem. This estimation may then be embedded in traditional heuristic schemes, an ILS in our case, that explores the first-level decision space. The other contributions of this paper are the introduction of the VRP-DO, and providing an efficient solution algorithm for it. This is demonstrated by benchmarking the I-DQNCO's performance against three methods.

## 2 Problem Definition

The LSP operates a set of $m$ vehicles denoted by $\mathcal{V} = \{v_1, v_2, ..., v_m\}$. Each vehicle is dispatched from and must return to a depot $l_0$. Vehicles are homogeneous with capacity $Q$. An overtime cost $\phi$ is incurred for each time unit exceeding a given limit $L$. At the beginning of a day, a set of customers $\mathcal{C} = \{c_1, \ldots, c_n\}$, which is drawn from a given distribution function $\Gamma^C$, requests service. Each customer $c \in \mathcal{C}$ is characterized by a location $l_c$, an expected demand $\bar{d}_c$, and a demand distribution function $\Gamma_c^D$. The actual demand $d_c$ of customer $c$ is revealed at the first time customer $c$ is visited. Let $G = (\mathcal{N}, \mathcal{E})$ be a complete graph with the set of nodes $\mathcal{N} = \{l_0, c_1, ..., c_n\}$, and the set of edges $\mathcal{E} = \{(i,j)|i,j \in \mathcal{N}\}$. Let $\tau_{ij}$ be the travel time between node $i$ and $j$.

The LSP determines the subset of outsourced customers $\mathcal{C}^o \subset \mathcal{C}$ incurring a cost of $\Psi(\mathcal{C}^o)$, which, similar to [1], is assumed to be a piece-wise linear function of $\sum_{c\in\mathcal{C}^o} \bar{d}_c$. The set of committed customers $\mathcal{C}^p = \mathcal{C} \setminus \mathcal{C}^o$ should be served by the LSP's fleet. The demand of customer $c \in \mathcal{C}^p$ is revealed upon visiting it, with $w = \{w^c\}_{c\in\mathcal{C}^p}$ being the vector of demand realizations. A vehicle serves the demand as much as possible, and any remaining demand is met by the same or other vehicles at a later time. Let $x_c = 1$ if customer $c \in \mathcal{C}$ is committed, and $x_c = 0$ if outsourced. Let also $\pi_r$ be the *routing policy* for the VRP-D. The duration of the trip of vehicle $v \in \mathcal{V}$ when operating policy $\pi_r$ to serve customers in $\mathcal{C}^p$ with realized demands $w$ is $T_{\pi_r}(\mathcal{C}^p, v, w)$. The VRP-DO is as follows:

$$\min_x \quad \Psi(\mathcal{C}^o) + R(\mathcal{C}^p), \tag{1}$$

$$\mathcal{C}^o = \{c \in \mathcal{C}|x_c = 0\},\ \mathcal{C}^p = \{c \in \mathcal{C}|x_c = 1\}, \tag{2}$$

$$R(\mathcal{C}^p) = \min_{\pi_r} \mathbb{E}_w \left[ \sum_{v\in\mathcal{V}} \Big[ T_{\pi_r}(\mathcal{C}^p, v, w) + (\phi - 1)\Big( (T_{\pi_r}(\mathcal{C}^p, v, w) - L)^+ \Big) \Big] \right] \tag{3}$$

where $R(\mathcal{C}^p)$ is the optimal expected *routing cost*, which consists of travel and overtime costs, for visiting the customer set $\mathcal{C}^p$, and the operator $(.)^+$ returns the overtime value.

Our first level problem consists in determining the set $\mathcal{C}^p$. Given this set, our second-level problem (i.e., VRP-D) entails solving the minimization problem in (3). We formalize the VRP-D as an MDP. We define a *decision epoch* as a point in time where a vehicle is ready to depart from its current location (i.e., the depot or a customer). At each decision epoch $k$, the state of the system is $s_k = (F^{\mathcal{C}}, F^{\mathcal{V}}, t_k)$, where $F^{\mathcal{C}} = [(l_c, h_c, \bar{d}_c, \hat{d}_c)]_{c\in\mathcal{C}^p}$ and $F^{\mathcal{V}} = [(l_v, a_v, q_v, g_v)]_{v\in\mathcal{V}}$ indicate the state of customers (location, availability, expected demand, and unserved demand) and vehicles (current destination, arrival time, available capacity, and not-terminated indicator), respectively. The time at decision epoch $k$ is $t_k$. We define the action set by an $m$-dimensional vector of $y_k = (y_k^1, ..., y_k^m)$, with $y_k^v$ indicating the action of vehicle $v$ at decision epoch $k$. The transition function $S^M()$ describes the evolution of the system in time. Let $w_{k+1}$ denote the customers' demand revealed between decision epoch $k$ and $k+1$. The evolution of the system is described by the relation $s_{k+1} = S^M(s_k, y_k, w_{k+1})$. We define the cost function as $C(s_k, y_k) = \sum_{v\in\mathcal{V}} C_v(s_k, y_k^v)$, where $C_v(s_k, y_k^v) = \min(a_v, L) - \min(t_k, L) + \phi(\max(a_v, L) - \max(t_k, L))$ is the cost function of vehicle $v$, and $a_v$ is the arrival time of vehicle $v$ at its destination when taking action $y_k^v$. We define the value function as $V^{\pi_r}(s_k) = \mathbb{E}_w[C(s_k, y_k) + \gamma V^{\pi_r}(s_{k+1})]$, $\forall s_k \in S_{\mathcal{C}^p}$.

# 3 Proposed Solution Method

We introduce an ILS algorithm for the VRP-DO. This algorithm explores combinations of committed and outsourced customer sets as a first-level problem with two neighborhood structures, *add* and *swap*. The evaluation of each partition requires solving the resulting VRP-D as a second-level problem. The previously described VRP-D value function is specific to a given set of committed customers $\mathcal{C}^p$. However, the set $\mathcal{C}^p$ follows a distribution function $\Gamma^p$, which depends not only on the daily customer distribution $\Gamma^c$, but also on decision variables $x$, as explored by the ILS. To tackle this challenge we first introduce a suitable distribution $\bar{\Gamma}^p$, which does not depend on $x$. We then develop a generalization of the MDP formulation by treating the customer set $\mathcal{C}^p$ as a stochastic variable and modify the value function as $V^{\pi_r}(s_k) = \mathbb{E}_w[C(s_k, y_k) + \gamma V^{\pi_r}(s_{k+1})]$, $\forall s_k \in S$ with $S = \bigcup_{\mathcal{C}^p\sim\bar{\Gamma}^p} S_{\mathcal{C}^p}$. The generalized MDP enables the off-line estimation of the value function of the second-level problem, which can then be deployed in the ILS.

Solving the generalized MDP is challenging. Therefore, we instead solve a related formulation, the MDP-CO, which approximates the generalized MDP in two ways. First, a consecutive action selection procedure is implemented, restricting each decision epoch to the *active* vehicle, denoted $\bar{v}$. This reduces the action space to that of a single vehicle, aligning with the problem's characteristic where simultaneous arrivals of multiple vehicles at their destinations are unlikely. Second, we adopt a fixed-size vector, called the *observation* of the active vehicle $o_{k,\bar{v}}$, instead of the state $s_k$. This vector is derived from an observation function $O(s_k, \bar{v})$, which is implement as the Graph Attention Network (GAN) depicted in Figure 1. Specifically, this GAN consists of three attention-based blocks. The

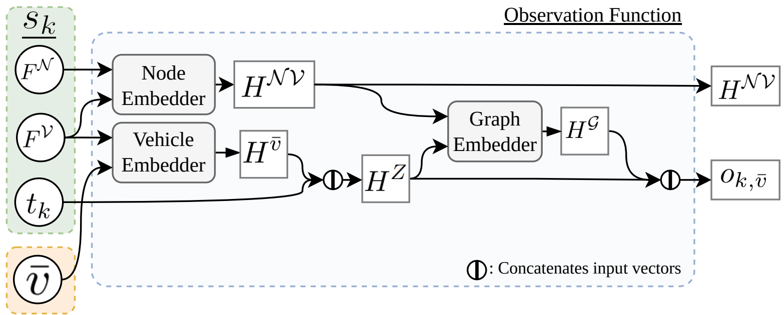



Figure 1: Structure of the GAN-based Observation Function

node and the vehicle embedders are responsible for capturing the graph structure of customer and vehicle positions, respectively. The graph embedder aggregates all customers and vehicles into one vector by considering the relevance of their information to the active vehicle. Finally, we develop a deep Q-learning algorithm implementing the MDP-CO.

# 4 Preliminary Computational Results

We benchmark I-DQNCO against three methods deploying the same ILS structure, but with different second-level evaluation methods. In particular we consider: Random Routing Policy (IRP), Greedy Routing Policy (IGP), and Hyper-greedy Routing Policy (IHP). A given first-level solution $\mathcal{C}_p$ is then evaluated by simulating the second-level corresponding policy. In IRP the active vehicle $\bar{v}$ chooses the next customer $c$ randomly among the unserved ones, IGP chooses $c = \arg\min_{c \in \mathcal{C}_p} \tau_{\bar{v},c}$ (the closest), while for IHP $c = \arg\max_{c \in \mathcal{C}_p} \frac{\min(\hat{d}_c, q_{\bar{v}})}{\tau_{\bar{v},c}}$ (the one with the most favorable ratio of expected demand / distance). Instances are generated similarly to [2]. Here we consider three distributions, each implying a customer density $D \in \mathcal{D} = \{\text{Low, Moderate, High}\}$. Table 1 reports preliminary results, where we can see that our method significantly outperforms the benchmarks.

| $D$ | IRP | IGP | IHP | I-DQNCO | %IRP | %IGP | %IHP |
|---|---|---|---|---|---|---|---|
| Low | 923.3 | 848.8 | 848.0 | 708.0 | -23.31% | -16.58% | -16.50% |
| Moderate | 2281.5 | 1909.1 | 1929.5 | 1477.7 | -35.23% | -22.60% | -23.42% |
| High | 3513.0 | 2902.0 | 2932.2 | 2669.6 | -24.01% | -8.01% | -8.96% |
| | | | | | **-27.52%** | **-15.73%** | **-16.29%** |

Table 1: Block 1: density. Block 2: average costs. Block 3: cost reduction by I-DQNCO

# References

[1] Dabia, S., Lai, D., and Vigo, D. (2019). *An exact algorithm for a rich vehicle routing problem with private fleet and common carrier.* Transportation Science, 53(4):986–1000.

[2] Dastpak, M., Errico, F., and Jabali, O. (2023). *Off-line approximate dynamic programming for the vehicle routing problem with a highly variable customer basis and stochastic demands.* Computers & Operations Research, 159:106338.

# Strategic Decision-Making in Biorefinery Siting: A Stochastic Optimization Approach Considering Price and Biomass Uncertainties in Navarre, Spain

**Adrian Serrano-Hernandez**
Institute of Smart Cities, GILT-OR Research Group,
Public University of Navarre, Pamplona, Spain
Email: adrian.serrano@unavarra.es

**Luis Cadarso**
European Institute for Aviation Training and Accreditation (EIATA),
Universidad Rey Juan Carlos, Fuenlabrada, Spain

**Javier Faulin**
Institute of Smart Cities, GILT-OR Research Group,
Public University of Navarre, Pamplona, Spain
Email: javier.faulin@unavarra.es

## 1 Introduction

This paper addresses the problem of locating a lignocellulosic biorefinery in Northern Spain for the production of bioethanol, a renewable and environmentally friendly alternative [1]. Supply chain decisions related to infrastructure location are also taken into account, considering factors such as crop and biomass selection from surrounding fields [2]. The uncertainties in biorefinery location primarily comes from two sources: the potential shortage of biomass availability and the volatility of prices. On the one hand, the availability of resources for supplying a biorefinery is critical, given the weather-dependent and highly seasonal nature of biomass. On the other hand, biomass prices play a critical role in determining the type of biomass selected. Finally, we also consider the warehousing policy, allowing for renting during the project duration. Thus, the problem involves strategic decisions regarding plant location, tactical decisions about the location and timing of warehouse infrastructures, and operational decisions for purchases management and crop selection during different periods. To address uncertainty, a stochastic multi-stage biorefinery location model is proposed, employing a multistage scenario tree for strategic and tactical uncertainties, and two-stage scenario trees rooted with strategic and tactical nodes for operational uncertainties.

## 2 Methodology

To address uncertainties in prices and biomass availability, a scenario analysis approach is employed, visualized in a tree structure. Strategic scenarios, representing specific realizations of uncertain parameters over the temporal horizon, follow a root-to-leaf path in the tree. Let $\mathcal{N}$ be the set of nodes in the strategic multistage scenario tree based on biomass prices. Note that these nodes, represent both strategic decisions and operational periods, uphold the nonanticipativity principle (consider $\sigma^n$ be the immediate ancestor node to node $n$). Let also consider $\Pi$ to be the set of operational scenarios ($\pi \in \Pi$). Operational uncertainties are represented by operational scenarios in a two-stage tree rooted with a strategic node based on biomass availavilities. Thus, the strategic multistage scenario tree incorporates economic evolution scenarios, featuring pessimistic, neutral, and optimistic perspectives on price evolution, whereas the operational scenarios address for low, normal, and high biomass disposals. Additionally, $\mathcal{I}$ is the set of crop fields, $i \in \mathcal{I}$; $\mathcal{J}$ is the set of potential biorefineries, $j \in \mathcal{J}$; $\mathcal{W}$ is the set of warehouses, $w \in \mathcal{W}$; $\mathcal{K}$ is the set of vehicles, $k \in \mathcal{K}$; $\mathcal{P}$ is the set of products, $p \in \mathcal{P}$; $\mathcal{T}$ is the time set, $t \in \mathcal{T}$. Decision variables are based on $X_{jn}$, a binary variable indicating whether the biorefinery is built in potential location $j$ at strategic node $n$ (1 if true, 0 otherwise); $Y_{wn}$, binary variable indicating whether the warehouse $w$ is set up at strategic node $n$ (1 if true, 0 otherwise); $Q_{ijpt\pi}$, $Q_{iwpt\pi}$, and $Q_{wjpt\pi}$, tons of product $p$ bought in crop $i$ or located in warehouse $w$ and transported to warehouse $w$ or to biorefinery $j$ at time $t$ at operational node $\pi$; $V_{iwjkt\pi}$, the number of vehicles of type $k$ going from crop $i$ or warehouse $w$ to biorefinery $j$ at time $t$ at operational node $\pi$; $B_{ipt\pi}$, tons of product $p$ bought in crop $i$ at time $t$ at operational node $\pi$; $C_{pijt\pi}$, biorefinery $j$ consumption of product $p$ at time $t$ at operational node $\pi$; $BS_{jpt\pi}$, stock in potential location $j$ of product $p$ at time $t$ at operational node $\pi$; $WS_{wpt\pi}$, stock in warehouse $w$ of product $p$ at time $t$ at operational node $\pi$. Finally, the parameters used are $h_p$ (humidity of product $p$); $\eta$ (biorefinery monthly consumption); $\beta$ (proportion of consumption that can be stocked at the biorefinery); $\xi_{pt}$ (binary parameter equals to 1 if product $p$ is available at time $t$, and 0 otherwise); $d_{ijw}$ (distance from crop $i$ to potential location $w/j$ and between them); $cap_k$ (capacity of a vehicle type $k$); $\phi_p$ (season duration of product $p$); $\varphi_p^n$ (price of product $p$ at strategic node $n$); $\psi_{ip\pi}$ (total production of product $p$ in crop $i$ at operational node $\pi$); $\alpha_{pi}$ (exploitation factor of product $p$ in crop $i$ at operational node $\pi$); $FC_k$ (transportation fixed cost of vehicle type $k$); $VC_k$ (transportation variable cost of vehicle type $k$); $\varsigma$ (stock cost at the biorefinery); $\omega_n$ (cost of opening a warehouse at strategic node $n$); $\rho$ (capacity of a warehouse); $\kappa$ (stock cost at the warehouse); $\delta_t$ (losses on stock from time $t$ to time $t+1$); $\gamma$ (losses on transportation); $pn_n$ (weight of strategic node $n$); and $ppi_\pi$ (weight of operational node $\pi$).

The following mixed integer linear programming model is proposed to solve the biorefinery stochastic model.

$$\begin{aligned}\text{Min } \sum_{n\in\mathcal{N}} pn_n \Bigg[ &\sum_{\substack{i\in\mathcal{I}\\ p\in\mathcal{P}\\ t\in\mathcal{T}\\ \pi\in\Pi}} p\pi_\pi B_{ipt\pi}\varphi_p^n + \sum_{\substack{i\in\mathcal{I}\\ j\in\mathcal{J}\\ w\in\mathcal{W}\\ t\in\mathcal{T}\\ k\in\mathcal{K}\\ \pi\in\Pi}} p\pi_\pi (FC_k V_{ijwtk\pi} + 2VC_k V_{ijwtk\pi} d_{ijw}) \\ &+ \sum_{\substack{j\in\mathcal{J}\\ p\in\mathcal{P}\\ t\in\mathcal{T}\\ \pi\in\Pi}} p\pi_\pi BS_{jpt\pi}\varsigma + \sum_{\substack{w\in\mathcal{W}\\ p\in\mathcal{P}\\ t\in\mathcal{T}\\ \pi\in\Pi}} p\pi_\pi WS_{wpt\pi}\kappa \Bigg] + \sum_{\substack{n\in\mathcal{N}\\ w\in\mathcal{W}}} pn_n Y_{wn}\omega_n \end{aligned} \tag{1}$$

Such that,

$$\sum_{j\in\mathcal{J}} X_{jn} = 1, \quad \forall n \in \mathcal{N} \tag{2}$$

$$X_{j\sigma^n} \le X_{jn}, \quad \forall j \in \mathcal{J}, \forall n \in \mathcal{N} \tag{3}$$

$$B_{ipt\pi} \le \frac{\psi_{ip\pi}\alpha_\pi\xi_{pt}}{\phi_p}, \quad \forall i \in \mathcal{I}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T}, \forall \pi \in \Pi \tag{4}$$

$$B_{ipt\pi} = \sum_{w\in\mathcal{W}} Q_{iwpt\pi} + \sum_{j\in\mathcal{J}} Q_{ijpt\pi}, \quad \forall i \in \mathcal{I}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T}, \forall \pi \in \Pi \tag{5}$$

$$\begin{aligned}&\sum_{i\in\mathcal{I}} Q_{iwpt\pi}(1-\gamma) + CS_{wp(t-1)\pi}(1-\delta_t) \\ &= \sum_{j\in\mathcal{J}} Q_{wjpt\pi} + CS_{wpt\pi}, \quad \forall w \in \mathcal{W}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T}, \forall \pi \in \Pi\end{aligned} \tag{6}$$

$$\begin{aligned}&\sum_{i\in\mathcal{I}} Q_{ijpt\pi}(1-\gamma) + \sum_{w\in\mathcal{W}} Q_{wjpt\pi}(1-\gamma) + BS_{j(p(t-1))\pi}(1-\delta_t) = \frac{C_{pjt\pi}}{1-h_p} + BS_{jpt\pi}, \\ &\forall j \in \mathcal{J}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T}, \forall \pi \in \Pi\end{aligned} \tag{7}$$

$$\sum_{p\in\mathcal{P}} C_{pijt\pi} = X_{jn}\eta, \quad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}, \forall n \in \mathcal{N}, \forall \pi \in \Pi \tag{8}$$

$$\sum_{p\in\mathcal{P}} BS_{jpt\pi} \le X_{jn}\beta\eta, \quad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}, \forall n \in \mathcal{N}, \forall \pi \in \Pi \tag{9}$$

$$\sum_{p\in\mathcal{P}} WS_{wpt\pi} \le Y_{wn}\rho, \quad \forall w \in \mathcal{W}, \forall t \in \mathcal{T}, \forall n \in \mathcal{N}, \forall \pi \in \Pi \tag{10}$$

$$V_{ijwtk\pi} \ge \frac{\sum_p Q_{ijwpt\pi}}{\text{cap}_k}, \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall t \in \mathcal{T}, \forall k \in \mathcal{K}, \forall \pi \in \Pi \tag{11}$$

$$X_{jn}, Y_{wn} \in [0,1], \quad V_{ijwtk\pi} \in \mathbb{N} \tag{12}$$

The objective function (Equation 1) aims to minimize buying, transporting, and storing biomass costs. Constraints 2 ensure only one biorefinery per strategic node, while Constraints 3 refer to the nonanticipativity principle. Constraints 4 outline limits on biomass acquisition, tied to crop production. After purchasing biomass, Constraints 5 determine whether it goes to a warehouse or the biorefinery. Constraints 6 and 7 maintain a consistent flow of biomass over time, and Constraints 8 specify necessary biomass consumption. Stocks at the biorefinery and intermediate warehouses are regulated in Constraints 9 and 10, respectively. Constraints 11 address the heterogeneous fleet, and Constraint 12 defines the variables nature.

## 3 Results and Conclusions

The computational results present findings from both deterministic and stochastic versions of the biorefinery location problem, considering tactical and operational conditions of product logistics. In the deterministic model, site #4 is recommended for the biorefinery location. However, the stochastic optimization model suggests shifting the biorefinery to site #21, revealing notable differences in plant location decisions between deterministic and stochastic approaches. Additionally, this shift leads to a huge change in the supply chain decisions. Thus, the stochastic approach shows a slight increase in costs on average (approximately 0.68%), driven by factors such as biomass and storage expenses.

These findings offer valuable insights into the impact of uncertainty on decision-making processes in biorefinery logistics. In our application, stochastic optimization enhances warehouse planning flexibility, allowing for adjustments in response to proposed scenarios. In contrast, the deterministic version employs a fixed warehouse selection for the entire project time horizon, limiting adaptability to changing conditions. Therefore, the importance of considering stochastic factors when optimizing supply chain operations is underrated, as well as the benefits of flexible warehouse planning. By understanding these dynamics, practitioners can make more informed and adaptable decisions to enhance overall supply chain performance in similar contexts.

## References

[1] N. Singh, R. Singhania, P. Nigam, C. Dong, A. Patel, M. Puri. Global status of Lignocellulosic Biorefinery: Challenges and Perspectives. *Bioresource Technology*, 344, 126415, 2022.

[2] A. Serrano-Hernandez, J. Faulin. Locating a Biorefinery in Northern Spain: Decision Making and Economic Consequences. *Socio-Economic Planning Sciences*, 66, 82-91, 2018.

# Service Design with Outsourcing Decisions

**Elena Fernández**
Departamento de Estadística e Investigación Operativa,
Universidad de Cádiz, Cádiz, Spain
Email: elena.fernandez@uca.es

**Ivana Ljubić**
Department of Information Systems, Decision Sciences and Statistics,
ESSEC Business School of Paris

**Nicolás Zerega**
Departamento de Estadística e Investigación Operativa
Universidad de Cádiz, Cádiz, Spain

In recent years, there has been a significant increase in the outsourcing of various practices, which has been particularly prominent in the logistics sector [3]. The airline industry has also experienced outsourcing in processes such as check-in, luggage management, and cabin crew [2]. Notably, there are cases in which flights are outsourced to third-party airlines [1]. Given this increasing trend, it is crucial to study and model this type of situations from a network design perspective to gain a better understanding of how outsourcing decisions contribute to the overall revenue of a major firm when it faces such a process. It is equally important to consider the carriers' viewpoint as they too aim to optimize their revenues.

In this work we introduce the Service Design Problem with Outsourcing Decisions (SDOD), which addresses the transportation of demand between different origins and destinations (*commodities*) when a major firm (*the leader*) already possesses a hub network and chooses to outsource the service for the demand coming from and going to non-hub locations using third-party companies (*carriers*) so as to maximize its overall profit. The process can be seen as a Stackelberg game with a single-leader and multiple independent followers. The leader makes decisions regarding the outsourcing of services: *i*) how to allocate non-hub nodes to carriers, and *ii*) the outsourcing fees for routing commodities through hub nodes.

Once they receive the offer and allocation provided by the leader, carriers decide which commodities they accept to serve, taking into account their reservation prices. We assume that the leader has perfect information regarding the carriers' reservation prices. Hence, by anticipating the optimal response of the followers, the leader also decides which commodities will ultimately be served and how served commodities will be routed.

We initially model the SDOD as a bi-level Mixed Integer Non-Linear Programming model. Then, we discuss some properties and show that SDOD is NP-hard. Leveraging the properties associated with the independency of allocations and costs of carriers, we discretize the outsourcing fee decisions, enabling us to express the SDOD as a single-level MILP. The fees and solutions derived from this model are shown to be bi-level optimal.

## 1 Problem definition

Let $G = (V, A)$ be a given directed graph. Let $H \subset V$ be the set of *hub nodes* and $V \setminus H$ the set of *non-hub nodes*. The backbone network is given by the complete network $A_H = \{(i, j) : i, j \in H\}$ including *loops*. The arc set is $A = A_H \cup \{(i, j) : i \in V \setminus H, j \in H\} \cup \{(i, j) : i \in H, j \in V \setminus H\}$. Service demand is given as set of commodities, indexed in a set $R$. The set $K$ of external carriers is also given. In addition:

- Each commodity $r \in R$ is associated with a triplet $(o(r), d(r), w^r)$, where $o(r), d(r) \in V$ denote its origin and destination nodes, respectively, and $w^r > 0$ the amount of flow that must be sent from $o(r)$ to $d(r)$. Commodities do not necessarily have to be served; in case commodity $r \in R$ is served, it produces a revenue $b^r > 0$.
- We are given $\overline{c}_{ij}^{rk} > 0$, the *reservation price* of carrier $k \in K$ for routing commodity $r \in R$ through access arc $(i, j)$, $i = o(r)$, $j \in H$, or distribution arc $(i, j)$, $i \in H$, $j = d(r)$. $\overline{c}_{ij}^{rk} > 0$ accounts for the cost of $k$ for routing $r$ through $(i, j)$ plus an additional profit fee that the carrier charges the leader to accept the outsourcing fee.

**The decisions of the leader are the following:**

- Allocate each non-hub $i \in V \setminus H$ to at most one carrier. Let $a : V \setminus H \mapsto K \cup \{0\}$ be a mapping such that $a(i) = k \in K$ if $i$ is allocated to $k$, and $a(i) = 0$ if $i$ is not allocated to any carrier.
- For each $r \in R$ and each $i \in H$, determine outsourcing fees for access arcs $p_i^r$ (if $o(r) \notin H$) and distribution arcs $q_i^r$ (if $d(r) \notin H$).
- Identify the set of commodities to be served, $R^* \subseteq R$. Service routes are of the form $o(r) - i(r) - j(r) - d(r)$, where $i(r), j(r) \in H$ are hubs (possibly $i(r) = j(r)$) decided by the leader. The *intermediate leg* of each served commodity $r \in R$, $(i(r), j(r))$, will be handled by the leader, incurring his own routing costs $c_{ij}^r$, whereas service of the *first* and *third legs*, $(o(r), i(r))$ and $(j(r), d(r))$, respectively, will be handled by the carrier allocated to $o(r)$ and $d(r)$, respectively. No commodity can be routed unless some outsourcing fee for its first, respectively third leg has been accepted by the involved carriers. Hence, the commodities entailing both a first and a third leg, cannot be routed unless a first and a third leg have been accepted. We assume

a multiple allocation strategy so commodities with the same non-hub origin but different destinations can be connected to the backbone network through different hubs, and flows may arrive to a given non-hub destination from different hubs.

**The decisions of the followers are the following:** For each $r \in R$ such that $o(r) \in V \setminus H$ is allocated to carrier $k \in K$, the carrier observes the offered outsourcing fees $p_i^r$ for all $i \in H$ and chooses the most profitable among them, or refuses to serve commodity $r$, if the resulting profit is negative. Similarly, the carrier allocated to $d(r)$ observes the outsourcing fees $q_i^r$ for all $i \in H$ and chooses the most profitable one, or refuses to serve commodity $r$, if the resulting profit is negative. The profit functions for routing the first, respectively third leg, for $r \in R$, $k \in K$ and outsourcing fees $p$ and $q$ are defined as:

$$F_k(r,p) = \max\left\{0, \max_{i \in H}\left\{p_i^r - \bar{c}_{o(r)i}^{rk}\right\}\right\} \qquad \text{and} \qquad T_k(r,q) = \max\left\{0, \max_{i \in H}\left\{q_i^r - \bar{c}_{id(r)}^{rk}\right\}\right\}.$$

**Definition 1.1 (Service Design Problem with Outsourcing Decisions)** *The SDOD can be expressed as the following bi-level optimization problem:*

$$(SDOD) \qquad \max_{\substack{R^* \subseteq R,\, a: V\setminus H \mapsto K \\ p \geq 0,\, q \geq 0}} \sum_{r \in R^*} [b_r - C_r(a,p,q)]$$

*where, for each commodity $r \in R$ the routing costs $C_r(a,p,q)$ are calculated as:*

$$C_r(a,p,q) = \begin{cases} \min\left\{p_{i^*}^r + c_{i^*j^*}^r + q_{j^*}^r : i^* \in \arg\max F_{a(o(r))}(r,p), \right. \\ \qquad\qquad\qquad \left. j^* \in \arg\max T_{a(d(r))}(r,q) \ \right\} & o(r) \in V \setminus H,\ d(r) \in V \setminus H \\ \min\left\{p_{i^*}^r + c_{i^*j}^r : i^* \in \arg\max F_{a(o(r))}(r)\right\} & o(r) \in V \setminus H, j = d(r) \in H \\ \min\left\{c_{ij^*}^r + q_{j^*}^r : j^* \in \arg\max T_{a(d(r))}(r,q)\right\} & d(r) \in V \setminus H, i = o(r) \in H \\ \infty, & \textit{othwerwise.} \end{cases}$$

The definition of $i^*$ and $j^*$ above reveals the bi-level nature of the problem. In the definition of $C_r(a,p,q)$, if $F_{a(o(r))}(r,p) < 0$, or $T_{a(d(r))}(r,q) < 0$, then the allocated carrier refuses to serve the first, respectively third leg of commodity $r$, and hence the overall routing cost must be set to $\infty$. W.l.o.g. we assume that all commodities $r \in R$ such that $o(r), d(r) \in H$ are preprocessed, given that $i$) service to these commodities will not be outsourced, and $ii$) the leader's profit for each of these commodities is simply $b_r - c_{o(r)d(r)}^r$. Hence the commodity will be included in $R^*$ if and only if the resulting profit is positive.

We consider an optimistic bi-level optimization setting in which the leader chooses the most profitable routing path in case there are multiple optimal responses of the followers. This is embedded in the definition of function $C_r(.)$: if there are multiple hubs $i^*$ or $j^*$ through which the respective carriers can route commodity $r$ while receiving the same profit, the leader will choose to route $r$ through hubs $i(r), j(r)$ such that the overall routing cost and outsourcing fees is minimized, namely $(i(r), j(r)) \in \arg\min\{p^r_{i^*} + c^r_{i^*j^*} + q^r_{j^*}\}$.

The above definition demonstrates the trade-off faced by the leader. On the one hand it can be convenient to increase the outsourcing fees so as to get more positive responses, which may result on a higher overall revenue for the served commodities. On the other hand, high outsourcing fees reduce the net profit of the leader, possibly resulting in fewer served commodities.

## 2 Our results and conclusions

- We show that the SDOD is NP-hard, even when the backbone network consists of a single hub node.
- We present and compare several approaches to solve the SDOD. The first one distinguishes each outsourcing fee based on the carrier, commodity, and hub used for the connection. The second one aggregates the first and third legs by carriers. The third one explicitly focuses on the commodities, while the fourth one determines the routing of commodities as an implicit path based on carrier allocations.
- We have run extensive computational tests to compare the alternative approaches on benchmark instances generated from the well-known AP data set [4] . Computational results demonstrate the superiority of the implicit paths formulation which is able to optimally solve instances of 200 nodes and 6 carriers within one hour.
- Our findings provide motivation for studying more complex systems in the future.

## References

[1] European Commission, Directorate-General for Mobility and Transport, Landru, M., Marquis, C., Lieshout, R. et al., "Study on the economic developments of the EU air transport market: final report", Publications Office, 2021.

[2] European Commission, Directorate-General for Mobility and Transport, C. Molina, S. Levin, L. Castillo, "Study on employment and working conditions of aircrews in the EU internal aviation market: final report", Publications Office, 2019.

[3] Dimitrios J. Dimitriou. D. J. Dimitriou, "The Evolution in Transport Operator's Corporate Structure: Ownership and Governance", in *Outsourcing and Offshoring.* IntechOpen, 2021.

[4] A. T. Ernst, M Krishnamoorthy, "Efficient algorithms for the uncapacitated single allocation $p$-hub median problem", *Location Science* 4(3), 139–154, 1996.

# Incorporating time-dependent demand patterns in the optimal location of capacitated charging stations

**Gianfranco Guastaroba**
Department of Economics and Management,
University of Brescia, Brescia, Italy
Email: gianfranco.guastaroba@unibs.it

**Carlo Filippi**
Department of Economics and Management,
University of Brescia, Brescia, Italy

**Lorenzo Peirano**
Department of Economics and Management,
University of Brescia, Brescia, Italy

**M. Grazia Speranza**
Department of Economics and Management,
University of Brescia, Brescia, Italy

## 1 Content of the talk and contributions to the literature

Sustainable transportation is one of the major challenges that modern countries are facing. Several sources indicate that the freight transportation sector generates the largest share of GreenHouse Gas (GHG) emissions. According to the United States Environmental Protection Agency, in 2020 the transportation sector produced 27% of the total GHG emissions in the US, mostly generated from burning fossil fuels by cars, trucks, ships, trains, and planes. Furthermore, data provided by the European Environment Agency highlight that in the EU more than 22% of the GHG emissions came from the transportation sector.

Despite technical advances have made available a range of options for sustainable mobility, there are still important obstacles that must be overcome for their mass adoption. Among such options, Electric Vehicles (EVs) are considered one of the major directions to reduce the environmental impact of mobility and make urban areas more sustainable. In the 2021 edition of the Global EV Outlook 2021, the International Energy Agency pointed out that at the end of 2020 the global EVs stock hit 10 millions units, with 3 millions newly registered EVs. Europe was the fastest growing market, with a sales share equal to 10% and some leading countries, such as Norway, which registered a record high sales share of 75%. This trend was accelerated by many countries of the European Union through substantial financial incentives. However, the decision of potential EV buyers is still strongly affected by two major issues. First, the purchase cost of an EV is still higher than that of a traditional internal combustion engine vehicle. Additionally, the limited travel range of an EV and the still long charging times are well-known to generate anxiety in the potential buyers [5]. In fact, the willingness of drivers to purchase an EV strongly depends on the availability of charging stations nearby their points of interests (e.g., home and work). As the number of charging stations is growing, thanks to public and private investments, the problem of determining an optimal location and size of charging stations for EVs has recently attracted an increasing academic attention. A recent overview of the main modeling and algorithmic approaches employed in this research area is available in [4].

There are a number of factors that make the location of charging stations substantially different from other, more classical, location problems, in particular the choice of the charger to install (e.g., slow, quick, fast), and the characteristics of the charging demand. The type of charger is a key factor to be taken into account, as it impacts the charging time. As of the end of 2021, there exist three main types of charger. Level 1 chargers, also referred to as *slow* chargers, that can take up to 40 hours to raise the level of a standard battery EV from 10% to 80% of the capacity. These chargers are most suitable for private usage. Level 2 chargers, sometimes called *quick* chargers, can charge up to 10 times faster than a level 1 charger, and are the most commonly used types for daily EV charging. Given the same battery characteristics mentioned above, the charging time is about 4.5 hours. The level 3 or *fast* chargers can reduce the charging time to 40 minutes or even less. The type of charger demanded by EVs is affected by the urban layout. For example, slow chargers will be demanded in residential areas so that EVs can be recharged over the night at low cost. An interesting study of the factors influencing the charging demand is provided in [6].

In this talk, we study the problem of determining an optimal deployment of charging stations for EVs within an urban environment. Our modeling approach assumes that the demand is node-based -that is, drivers demanding to charge their EVs are associated with one/few fixed locations, which represent, for instance, their workplace, residence or specific service facilities (such as commercial activities). This modeling approach is the best suited for urban settings, and it is often used in the literature in such setting as it allows, on the one hand, to neglect the limited driving range of EVs (cf. [1]) and, on the other hand, to extend classic discrete location models (e.g., set covering as in [3], and maximum coverage problems as in [2]) to incorporate technical constraints specific to EVs.

Nevertheless, in the classical location models a customer is characterized by the distance from any potential location and by a single quantity -a measure of the demand. The models do not consider a temporal dimension of the problem, which basically corresponds to assuming that the demand is uniformly distributed over the time period of interest of the location decision. On the contrary, the charging demand of EVs fluctuates over time, with peaks of demand in periods of time where the traffic volume is high. Neglecting the demand dynamics may lead to solutions where the charging capacity deployed is not sufficient to satisfy the demand during the peak times.

In the present research, different types of chargers have to be located in pre-defined potential locations, modeled as nodes of a network. The urban area is partitioned into sections. A customer is associated with each section of the urban area. Its demand in a certain time interval is the number of EVs in that section that need to be recharged. The customer is located in the center of gravity of the section and is modeled as a node of the network. The urban area is also partitioned in zones (e.g., commercial, industrial, or residential) that have different needs in terms of minimum number of each type of charger deployed in the zone.

We present, over a discretized time horizon, an optimization model that introduces a temporal dimension which, to the best of our knowledge, has never been introduced in the literature on location problems and captures the dynamics of the charging demand. The model takes into account several characteristics of the real problem: multiple types of chargers (each with its own charging speed and installation cost), the capacitated nature of the charging stations (in terms of maximum number of chargers that can be installed), and a minimum number of chargers to be installed in different zones (e.g., commercial, residential, industrial). Further, assuming that a charger can take more than one period to fully recharge an EV, the proposed multi-period formulation includes constraints to keep track of the usage of chargers across consecutive time periods and to ensure that no other vehicles are assigned to any occupied charger. This novel approach guarantees a correct sizing of the solution, in terms of number of stations opened and number of chargers installed, and ensures that the demand is completely satisfied in all time periods. In order to assess the value of introducing the temporal dimension in the location problem, which makes the optimization model more complex, we present also a single-period optimization model that captures the same specificities of the problem but ignores the temporal aspect. In both models, the objective is the minimization of a convex combination of two terms: the total cost of deploying the charging stations and installing the chargers, and the average distance traveled by the customers to reach the assigned charging station. The two optimization problems are cast as Mixed Integer linear Programs (MIP). We compare the two MIPs through a theoretical and a computational analysis. We show, through a worst-case analysis, that a solution to the single-period model may fail to satisfy a large portion of the charging demand. Extensive computational experiments are run on different classes of randomly generated instances. The results confirm the importance of explicitly considering the dependence on time of the demand. In fact, the single-period model is based on the common assumption that the charging demand is uniformly distributed across the planning horizon. In an application context such as the one at hand, where the demand fluctuates significantly during the day and across different zones of the same urban area, the single-period model produces solutions that are not capable of serving a large portion of the charging demand, especially in those time periods where the demand is prominently concentrated. The computational experiments also include a parametric analysis of the relative weight assigned to the objective function components.

## References


[1] M.F. Anjos, B. Gendron and M. Joyce-Moniz, "Increasing electric vehicle adoption through the optimal deployment of fast-charging stations for local and long-distance travel", *European Journal of Operational Research* 285(1), 263–278, 2020.

[2] G. Dong, J. Ma, R. Wei and J. Haycox, "Electric vehicle charging point placement optimisation by exploiting spatial statistics and maximal coverage location models", *Transportation Research Part D: Transport and Environment* 67, 77–88, 2019.

[3] K. Huang, P. Kanaroglou and X. Zhang, "The design of electric vehicle charging network", *Transportation Research Part D: Transport and Environment* 49, 1–17, 2016.

[4] M. Kchaou-Boujelben, "Charging station location problem: A comprehensive review on models and solution approaches", *Transportation Research Part C: Emerging Technologies* 132, 103376, 2021.

[5] D. Pevec, J. Babic, A. Carvalho, Y. Ghiassi-Farrokhfal, W. Ketter and V. Podobnik, "A survey-based assessment of how existing and potential electric vehicle owners perceive range anxiety", *Journal of Cleaner Production* 276, 122779, 2020.

[6] R. Wolbertus, M. Kroesen, R. Van Den Hoed and C. Chorus, "Fully charged: An empirical study into the factors that influence connection times at EV-charging stations", *Energy Policy* 123, 1–7, 2018.

# The Stochastic 3D Bin Selection Problem: Branch-and-Repair for Multi-stage Stochastic Programs

**Pirmin Fontaine**
School of Management,
Catholic University of Eichstätt-Ingolstadt, Ingolstadt, Germany,
Email: pirmin.fontaine@ku.de

## 1 Introduction

E-commerce is continuously growing, resulting in an increasing number of parcels every year that is shipped around the globe. This increase is further intensified by the ongoing urbanization. Therefore, research on urban logistics is continuously growing and tries to improve the efficiency in both parcel and freight transportation. [1] One major challenge is the poor utilization of resources resulting from not well-packed trucks, containers, and wrong-sized parcels. Decreasing the unused space in parcels or containers would allow to pack goods more efficiently and reduce the number of needed vehicles. Therefore, also the portfolio of such parcels is important for e-commerce retailers.

While in the deterministic case, the question on the number and size of the used parcel types has been recently answered, the dynamic nature of the problem requires tools to address the stochasticity in demand over longer planning horizons. To improve efficiency and sustainability, we analyze the trade-off between inventory holding, costs of unused space, and the costs of variety in a multi-period setting. For e-commerce retailers, less variety imply that fewer parcel types need to be kept in stock. Additionally, dynamic order policies allow the shift of inventories between different parcel types.

To address this gap, we introduce the stochastic three-dimensional bin selection problem (S3D-BSP) and formulate it as a multi-stage stochastic program. Then, we develop an exact solution method building on branch-and-repair [2] that shows how to efficiently solve multi-stage stochastic programs.

## 2 Problem setting

For the S3D-BSP, we consider a planning horizon defined by a set of periods $T$, a set $J$ of rectangular bin types (parcels), and a set of orders $O$. Each of these orders $o \in O$ consists of a set of rectangular items $I_o$ that have to be packed into a selected bin. For packing those items, we assume the standard assumptions of three-dimensional packing with rotation: (1) Items are allowed to be rotated along the three axes, (2) the bin has to enclose all packed items, and (3) items are not allowed to overlap.

Both, the bin types $j \in J$ and the items $i \in I_o$ of the orders $o \in O$ are specified by a length $L_j$, a width $W_j$, and a height $H_j$ and $l_{io}$, $w_{io}$, and $h_{io}$, respectively. To include the stochastic nature of the demand over the planning horizon, we assume a set of scenarios $S$ with probability $\pi_s$ resulting in a demand $d_{ots}$ for each order $o \in O$, stage $t \in T$, and scenario $s \in S$.

The main decision is to determine the optimal set of bin types that is used over the full planning horizon. Therefore, the binary decision $\hat{n}_j$ indicates if a bin type is part of the portfolio $(= 1)$ or not $(= 0)$. Further, $\hat{q}_{jts} \in \{0, 1\}$ defines if an order of bin type $j$ is placed in period $t$ of scenario $s$. And, $q_{jts}$ is the number of bin types that are ordered. The available inventory is tracked through decision variable $p_{jts}$. Finally, the integer decision variable $n_{ojts}$ defines the number of orders $o$ that are assigned to bin type $j$ in period $t$ in scenario $s$. To avoid extensive notation, let $3DBPP(o, t, s)$ define the constraints and decision variables of a three-dimensional container loading problem (CLP) (see, [3]), which has to be satisfied for each order since we are interested in one bin type portfolio for all orders and each order is packed into one bin.

The objective function includes costs of unused space $c^S$, costs of variety $c^V$, inventory holding costs $c^I$, and ordering costs $c^O$. Then, we can define the S3D-BSP as follows:

$$
\begin{aligned}
\min \sum_{s \in S} \sum_{t \in T} \pi_s \Bigg( c^S \Bigg( \sum_{o \in O} \sum_{j \in J} L_j W_j H_j n_{ojts} - \sum_{o \in O} \sum_{i \in I_o} l_{io} w_{io} h_{io} \Bigg) \\
+ c^I \sum_{j \in J} p_{jts} + c^O \sum_{j \in J} \hat{q}_{jts} \Bigg) + c^V |O| \sum_{j \in J} \hat{n}_j \qquad (1)
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.} \quad & 3DBPP(o, t, s) && \forall o \in O, s \in S, t \in T && (2) \\
& \sum_{j \in J} n_{ojts} = d_{ots} && \forall o \in O, s \in S, t \in T && (3) \\
& \hat{q}_{jts} \leq \hat{n}_j && \forall j \in J, s \in S, t \in T && (4) \\
& q_{jts} \leq M \hat{q}_{jts} && \forall j \in J, s \in S, t \in T && (5) \\
& p_{j,t-1,s} + q_{jts} - \sum_{o \in O} n_{ojts} = p_{jts} && \forall j \in J, s \in S, t \in T && (6) \\
& && \text{(non-anticipativity constraints)} && (7)
\end{aligned}
$$

$$\hat{n}_j, \hat{q}_{jts} \in \{0,1\} \qquad \forall o \in O, j \in J \tag{8}$$

$$n_{ojts}, q_{jts}, p_{jts} \in \mathcal{Z} \tag{9}$$

We minimize the trade-off between the costs for unused space, inventory holding costs, ordering costs, and the costs of variety. Constraints (2) ensure that the CLP constraints are satisfied. Constraints (3) assign orders to bin types and ensure demand satisfaction. A bin replenishment order can only be placed if the bin is part of the portfolio (Constraints (4)), and only if an order is placed, bins can be reordered (Constraints (5)). Inventory balance is ensured through constraints (6). Finally, classical non-anticipativity constraints are included.

## 3 Solution method

Since already the deterministic version is very difficult to solve and the S3D-BSP includes many CLPs in each scenario and stage, which is $\mathcal{NP}$-hard, we develop an efficient decomposition.

We use branch-and-repair [2] that builds on the idea of branch-and-check and decomposes the problem into a master and a subproblem. The master problem consists of the bin selection, the order assignment, the replenishment policy, and the inventory balance that is solved within a branch-and-cut tree. At each integer node, the subproblem checks the feasibility of the packing problem and adds combinatorial cuts in case of violation to forbid infeasible assignments. Additionally, we try to repair the infeasible solution and thus generate upper bounds. The chosen decomposition further allows to relax several binary variables in the master problem and improves the performance.

Besides known acceleration techniques for the deterministic case, we additionally develop stochastic methods that reduce the number of subproblem evaluations and to strengthen the generated cuts.

## 4 Results

For preliminary results on the computational performance, we generate datasets based on the deterministic version of the problem. We use 100 orders with random demand between 90 and 110 per order in each stage and scenario. Branch-and-repair was implemented in C++ using the Gurobi API as solver. The time limit was set to 7,200 seconds and we report both the run time and the number of instances solved to optimality. Additionally, we used a gap of 2% as stopping criterion that is often used in two-stage stochastic programming. Table 1 reports the aggregated results for different sizes of the multi-stage stochastic program depending on the number of stages $T$ and the number of scenarios $S$. The results show that branch-and-repair is able to solve large multi-stage stochastic programs efficiently to optimality. Keeping in mind that each stage and each scenario consists out of 100 three-dimensional bin packing problems (3D-BPPs), the decomposition allows to avoid extensive evaluations of these packing problems.

Table 1: Branch-and-repair performance

| | | gap 0% | | gap 2% | |
|---|---|---|---|---|---|
| $T$ | $S$ | run time (sec) | opt | run time (sec) | opt |
| 4 | 16 | 179 | 5 / 5 | 147 | 5 / 5 |
| 4 | 81 | 1777 | 5 / 5 | 1177 | 5 / 5 |
| 5 | 32 | 886 | 5 / 5 | 276 | 5 / 5 |
| 5 | 243 | 6538 | 2 / 5 | 3067 | 2 / 5 |
| 6 | 64 | 3411 | 5 / 5 | 3166 | 5 / 5 |
| 7 | 128 | 5489 | 2 / 5 | 3662 | 3 / 5 |

## 5 Conclusions

In summary, the contributions are: We have introduced a new problem setting for determining the optimal bin type portfolio and reordering policy in a stochastic three-dimensional packing problem that we define as the S3D-BSP. We have developed a solution method building on branch-and-repair. The numerical study shows that the methodology is able to solve large multi-stage stochastic programs to optimality even if, in each scenario and stage, several NP-hard subproblems need to be solved. At the conference, we will give further details on the methodology, extensive numerical results, and managerial implications.

## References

[1] T. G. Crainic, N. Ricciardi, and G. Storchi. Models for evaluating and planning city logistics systems. *Transportation Science*, 43(4), 432–454, 2009.

[2] P. Fontaine and S. Minner. A branch-and-repair method for three-dimensional bin selection and packing in e-commerce. *Operations Research*, 71(1), 273–288, 2023.

[3] C. Paquay, M. Schyns, and S. Limbourg. A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application. *International Transactions in Operational Research*, 23(1-2), 187–213, 2016.

# Vehicle Inventory Models for Direct Delivery Scheduling Problems

**Michael Forbes**
School of Mathematics and Physics
The University of Queensland
Email: m.forbes@uq.edu.au

**Lucas Sippel**
School of Mathematics and Physics
The University of Queensland
Email: l.sippel@uq.edu.au

**Peter Gibson**
School of Mathematics and Physics
The University of Queensland

## 1 Introduction

We consider the direct delivery scheduling problem in a network (DDSP-N), a problem first introduced by Emde and Zehtabian [1]. In this problem, customers' requests need to be served by direct delivery from suppliers.

More formally, following the notation of Gschwind et al. [2], we have a set of suppliers $S$, customers $N$ and requests or trips $I$. For each request we know: $s_i$, the supplier from which goods are shipped; $n_i$, the customer who receives the goods; $r_i$, the ready time, or earliest time of departure of the request; $d_i$, the deadline by which the request much be completed at customer $n_i$; $p_i$, the processing time of the request, which includes loading time at the supplier, movement time to the customer, unloading time at the customer and any other activities required to complete the request; $w_i$, the weight of the request to be used in the objective function. Finally, we define the travel time $\tau_{ns}$ as the travel time between customer $n$ and supplier $s$.

Emde and Zehtabian [1] proposed a compact formulation for the DDSP-N and also presented a heuristic algorithm. They consider an objective function which first minimises the number of vehicles needed to service all requests and then minimises the total weighted delay of requests. Gschwind et al. [2] proposed an exact branch-cut-and-price algorithm for the same problem, extended to multiple suppliers. Emde et al. [3] proposed an exact approach for the DDSP-N with an extension to milk run delivery scheduling based on Benders decomposition. They further consider replacing the total weighted delay with the maximum weighted delay.

Here we propose a compact formulation based on tracking the inventory of vehicles at each supplier. All data sets proposed for this problem discretise time to 10 minute intervals and we utilise this to our advantage. In section 2 we give the formulation of the problem, followed by an outline of the overall algorithm in section 3. We conclude with results and discussion in section 4.

## 2 Formulation

Let $T$ be the set of discretised time points, $Depot$ be the location of the depot and $V$ be a set of time-space vertices, $V = \{(s,t)|s \in S, t \in T\}$. We define arcs of the following form: starting arcs $(Depot, v), v \in V$; ending arcs $(v, Depot), v \in V$; waiting or inventory arcs $((s,t),(s,t+1)), s \in S, t \in T$; and service arcs $((s_i,t),(s,t+p_i+\tau_{n_i,s})), i \in I, s \in S, r_i \leq t \leq d_i - p_i$.

Denote the full set of arcs by $A$ where each arc $a \in A$ flows from vertex $a^-$ to vertex $a^+$, has cost $c_a$ (variously the vehicle cost, the weighted delay or zero) and covers the set of requests $I_a$ (containing either one request or no requests). Finally, integer variables $x_a$ denote the number of vehicles using arc $a$.

DDSP-N can be formulated as an integer programming problem as follows.

$$\text{minimise } z = \sum_{a \in A} c_a x_a \tag{1}$$

subject to

$$\sum_{a \in A | i \in I_a} x_a = 1, \qquad \forall i \in I \tag{2}$$

$$\sum_{a \in A | a^+ = v} x_a = \sum_{a \in A | a^- = v} x_a \qquad \forall v \in V \tag{3}$$

$$x_a \geq 0, \text{integer} \qquad \forall a \in A \tag{4}$$

The objective function (1) minimises the total routing cost. Constraints (2) ensure that each request is served exactly once. Finally, constraints (3) conserve the flow of vehicles.

## 3 Algorithm

Our overall algorithmic approach takes advantage of the following well known proposition.

**Proposition 1** *Let $z^*$ be an upper bound on the optimal objective for a minimisation Mixed Integer Program (MIP). Suppose the solution to the linear programming (LP) relaxation gives a lower bound $z^{LP}$ and assigns reduced cost $\rho_a > 0$ to binary variable $x_a$. If $z^{LP} + \rho_a > z^*$ then $x_a = 0$ in every optimal solution to the MIP.*

We tentatively fix to zero variables which will cause the number of vehicles to increase and those variables which will increase the MIP objective $z^{MIP}$ such that $z^{MIP} > (1 +$

$\delta)z^{LP}$ where $\delta$ is a parameter chosen to be 0.005 in our computational testing. This parameter is our "guess" for the size of the MIP/LP gap.

In the overall solution approach we first minimise the number of vehicles and then minimise the weighted delay cost, with the number of vehicles constrained and the vehicle cost set to 0. In what follows DDSP-N denotes the formulation of the model, as modified by different objective functions and additional constraints during the execution of the algorithm.

1. Solve the LP relaxation of DDSP-N, with the objective set to the number of vehicles used. Denote the optimal objective as $z^v$.
2. Set $n^v = \lceil z^v \rceil$ and $\Phi^v = \{a : a \in A, \rho_a > n^v - z^v\}$. Add the constraint $\sum_{a \in \Phi^v} x_a = 0$.
3. Add a constraint to fix the number of vehicles to $n^v$.
4. Solve the LP relaxation of DDSP-N with the objective set to minimizing the weighted delay. Denote the optimal objective as $z^{LP}$
5. Set $\Phi^0 = \{a : a \in A, \rho_a > \delta \cdot z^{LP}\}$. Add the constraint $\sum_{a \in \Phi^0} x_a = 0$.
6. Solve DDSP-N. If feasible, the resulting objective is $z^{MIP}$.
7. If DDSP-N is feasible and $z^{MIP} \leq (1 + \delta)z^{LP}$, stop with the optimal solution.
8. Otherwise, replace the constraint $\sum_{a \in \Phi^0} x_a = 0$ with $\sum_{a \in \Phi^0} x_a \geq 1$.
9. If DDSP-N was feasible in step 6:
    - Set $\Phi^0 = \{a : a \in \Phi, \rho_a > z^{MIP} - z^{LP}\}$. Add the constraint $\sum_{a \in \Phi^0} x_a = 0$.
    - Solve DDSP-N and either improve on the solution from step 6 or show no such improved solution exists. In any event, stop with the optimal solution.
10. Otherwise, solve the DDSP-N as modified in step 8. If this is feasible, stop with the optimal solution. Otherwise, increment $n^v$ remove all extra constraints and return to step 3.

## 4 Results

We ran our algorithm on the 120 test instances from Gschwind et al [2]. Run time was limited to one hour. The key points to note are:

- We solved 117 instances to optimality, which includes 29 instances that had not previously been solved to optimality.

- Our algorithm was approximately 70 time faster than Gschwind et al. on the 88 instances they solved to optimality. In all these cases our optimal solutions are identical to those of Gschwind et al.

- In only one instance did we need to increase the target number of vehicles in step 10 of the overall algorithm.

Emde et al [3] introduce a number of variations. These include: only one supplier, multiple vehicle types with limited availability of each vehicle type, a modified objective function (minimise maximum weighted cycle time) and the possibility of milk run deliveries (a single trip serves multiple customers). They propose a Constraint Programming solver and a MIP solver using Benders Decomposition. They limit the CP solver to 1 hour per instance and the MIP solver to 30 minutes.

We modified our algorithm appropriately and ran it on their 84 instances. The key points to note are:

- We solved all instances to optimality, which includes 15 instances that had not previously been solved to optimality.

- Our total run time for all instances was 135 seconds, with a maximum run time of 10 seconds. Our optimal objective is consistent with the results of Emde et al in all cases.

- Our approach is orders of magnitude faster than Emde et al on these instances.

We will give detailed results in our presentation. We will also discuss other problems where similar approaches may be particularly useful and an extension to the approach when the assumption of time discretisation is relaxed.

## References


[1] S. Emde, S. Zehtabian, "Scheduling direct deliveries with time windows to minimize truck fleet size and customer waiting times", *International Journal of Production Research* 57(5), 1315–1330, 2019.

[2] T. Gschwind, S. Irnich, C. Tilk, S. Emde, "Branch-cut-and-price for scheduling deliveries with time windows in a direct shipping network", *Journal of Scheduling* 23, 363-377, 2020.

[3] S. Emde, S. Zehtabian, Y. Disser, "Point-to-point and milk run delivery scheduling: models, complexity results, and algorithms based on Benders decomposition", *Annals of Operations Research* 322, 467-496, 2023.

# A two-stage stochastic model for dual cycling under uncertain RoRo cargo arrival

**Asefe Forghani**
Department of People and Technology, Roskilde University, Denmark
Email: asefef@ruc.dk

**Line Reinhardt**
Department of People and Technology, Roskilde University, Denmark

**Dario Pacino**
DTU Management, Technical University of Denmark, Denmark

## 1 Introduction and background

Minimising turnaround times for RoRo (Roll-on/Roll-off) ships is crucial for enhancing cost efficiency and environmental sustainability. It enables the adoption of slow steaming practices, which, according to the literature, effectively reduce GHG emissions in RoRo shipping [1]. For instance, a speed reduction of approximately 10% to 40% can decrease $CO_2$ emissions by 27.05% to 78.39% in the RoRo shipping sector [2].

Classical discharge-then-load policies can be unproductive as tugs (terminal vehicles) spend half of their time driving without cargo. The use of dual cycling, as loading and discharging operations occur simultaneously, is known to reduce handling time. This strategy is effective in decreasing turnaround time and increasing the efficiency of quay cranes or tugs [3].The primary advantage of this strategy is that it can be implemented immediately without requiring new technology or infrastructure and no extra investment.

Research in this domain has focused mainly on container shipping and assumes deterministic cargo arrivals. In RoRo shipping, however, the deterministic modelling approach is usually unrealistic and cannot reflect the inherent uncertainty in RoRo cargo arrivals. Jia et al. [4] were the only researchers to examine dual cycling operations in RoRo terminals. While their work provides foundational insights, it is limited by the unrealistic assumption that all cargo is available at the terminal when planning is initiated. To bridge this gap, our study introduces a two-stage mixed-integer stochastic model to minimise the expected time to complete all discharging and loading tasks, considering cargo arrival scenarios.

## 2 Problem description

Figure 1 provides a schematic representation of the problem under investigation. The planning problem arises when a RORO vessel arrives at a maritime terminal. The vessel has a number of cargoes that need to be discharged (red), while in the terminal yard loading cargo (green) stands ready. Also, in the case of multi-port operations, some cargoes may remain on the vessel for later ports (blue), but this is not the case in shuttle services. As the figure stows, tugs are used for cargo handling operations. Not all cargo booked to be loaded is available in the yard, and its arrival time is not known. Given that a stowage plan (assignment of cargo to positions in the vessel) is available, the planning problem is the sequencing of the load and discharge operations of the tugs.

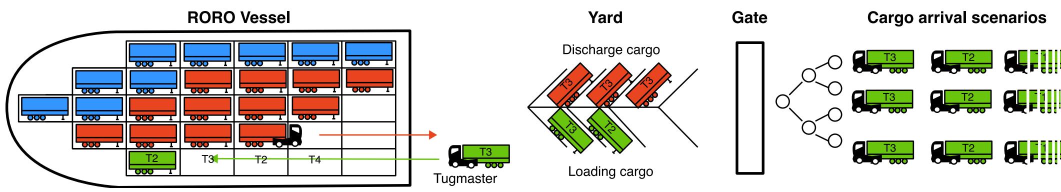



Figure 1: Schematic overview of the investigated problem.

To further improve handling operations, the flexibility of the stowage plan can be exploited. In particular, cargo can be grouped in types (e.g., by weight and/or owner). Cargoes of the same type, can then be assigned to any vessel position of the same type (indicated by T1, T2, T3, and T4 in Figure1). Since cargo handling operations commence as soon as the vessel arrives in the terminal, it is important that the planned sequence of tug operations takes into account the uncertain arrival times of some of the cargo. To study the impact that the uncertain cargo arrivals have on the sequencing problem, we propose the implementation of a two-stage stochastic program. The first stage (here-and-now) decisions, are the sequencing and scheduling of tugs operations for the cargo that is readily available in the yard. The recourse decisions (second stage) are based on a set of scenarios, each describing a possible realisation of the arrival times.

## 3 Two-stage stochastic model

The abbreviated form of the proposed two-stage stochastic model is presented below, where the objective is to minimise the expected turnaround time across all cargo arrival scenarios, and the constraints model the immediate operational decisions.

$$\min \mathbf{E}_{\xi} U(x, y, \xi) \tag{1}$$

s.t.

$$y_\iota^H \geq x_\iota^H \qquad \forall \iota \in (\mathcal{D} \cap \mathcal{L}) \tag{2}$$

$$\sum_{j \in \mathcal{L}} B_{j\nu} y_j^H \leq I_\nu^0 \qquad \forall \nu \in \mathcal{C} \tag{3}$$

$$\sum_{i \in \mathcal{D}} x_i^H + wsp^H t + \sum_{j \in \mathcal{L}} y_j^H + wps^H + wpp^H = K \tag{4}$$

$$x_i^H, y_j^H \in \{0, 1\} \qquad \forall i \in \mathcal{D}, j \in \mathcal{L} \tag{5}$$

$$wsp^H, wps^H, wps^H \in \mathbb{Z} \tag{6}$$

In the model's abbreviated form, *wait-and-see* decisions are captured in the objective function (1) through $\mathbf{E}\xi U(x, y, \xi)$, which calculates the expected makespan for all discharging and loading operations across scenarios ($\xi$). The constraints address only *here-and-now* decisions: Constraint (2) mandates that slots at the intersection of sets $\mathcal{D}$ and $\mathcal{L}$, representing discharging and loading respectively, must be emptied before being filled, where $x_i^H$ is a binary variable indicating whether slot $i$ is discharged (assigned a value of one) in the current period, and $y_j^H$ is a binary variable for loading, also assigned a value of one if slot $j$ is loaded. Constraint (3) ensures that loading does not exceed the initial availability of cargo ($I_\nu^0$) for each category $\nu$ within the cargo set $\mathcal{C}$; here, $B_{j\nu}$ is a binary parameter indicating if the predetermined slot type for $\nu$ is assigned to slot $j$. Constraint (4) ensures that the number of tugs used for loading and discharging operations, including those in transit from the ship to the port ($wsp^H$) and from the port to the ship ($wps^H$), as well as those idling at the port ($wpp_{ts}$), does not exceed the total number available ($K$) during the current decision horizon.

## 4 Preliminary results

Table 1 presents the preliminary results, outlining four small instances differentiated by scenario variability and desirability scores. Scenario variability measures the differences in cargo arrival patterns; more variability means greater differences among scenarios. Desirability scores are calculated based on cargo types and their arrival times, with higher scores for cargoes arriving earlier at the port. Instances LVHD and LVLD have lower variability, whereas HVHD and HVLD have higher variability. They are also ranked by desirability scores, with LVHD and HVHD recording higher scores, and LVLD and HVLD with lower desirability scores, i.e., relatively later arrival times.

The study assesses dual cycling's effect on makespan—the time for cargo discharging and loading—under uncertain arrival times. In three instances, dual cycling led to a shorter makespan, while in one, no improvement was observed. The smaller improvements in instances with lower desirability scores are justifiable due to cargo arriving later when most discharging is already done, limiting the benefits of dual cycling.

The study also examines two additional indicators: EVPI indicates how much perfect information could improve decisions, and VSS shows the benefits of using a stochastic model. These limited preliminary tests prevent definitive conclusions; however, initial findings suggest that higher variability may lead to greater benefits from perfect information and stochastic models. This makes sense as higher variability means greater potential rewards from precise knowledge (perfect information) and uncertainty-adaptive models (stochastic solutions) over an average nominal deterministic counterpart.

Table 1: Impact of dual cycling strategy, perfect information, and stochastic solutions.

| Instance | Discharge-then-load | | Dual cycling | | EVPI | VSS |
|---|---|---|---|---|---|---|
| | Time (s) | Makespan | Time (s) | Makespan | | |
| LVHD | 4.74 | 8.00 | 4.47 | 7.00 | 0.00 | 0.00 |
| LVLD | 7.42 | 9.30 | 3.55 | 9.30 | 0.00 | 0.70 |
| HVHD | 7.25 | 8.00 | 6.85 | 7.00 | 1.00 | 0.00 |
| HVLD | 7.19 | 9.00 | 6.74 | 8.70 | 2.30 | 1.30 |

## 5 Future work

To tackle the impracticality of rerunning the proposed model for real-world scale each period, we have developed a framework using a rolling horizon, scenario reduction strategies, and a lemma to cut down on reruns. Future comparative tests will assess its effectiveness.

## References


[1] H. Lindstad, R. Verbeek, M. Blok, S. Van Zyl, A. Hübscher, H. Kramer, J. Purwanto, O. Ivanova, and H. Boonman, "GHG emission reduction potential of EU-related maritime transport and on its impacts", *TNO report*, 2015.

[2] N. R. Ammar, "Energy- and cost-efficiency analysis of greenhouse gas emission reduction using slow steaming of ships: case study RORO cargo vessel," *Ships and Offshore Structures*, vol. 13, no. 8, pp. 868–876, Nov. 2018. DOI: 10.1080/17445302.2018.1470920.

[3] A. V. Goodchild and C. F. Daganzo, "Double-cycling strategies for container ships and their effect on ship loading and unloading operations," *Transportation Science*, vol. 40, no. 4, pp. 473–483, 2006.

[4] B. Jia, K. Tierney, L. B. Reinhardt, and J. Pahl, "Optimal dual cycling operations in roll-on roll-off terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 159, Mar. 2022, Art. no. 102646. DOI: 10.1016/j.tre.2022.102646.

# Integrated Optimization of Train Path Assignment and Rolling Stock Planning in Rail Freight Transportation

Louis Fourcade[1,2,*] Stéphane Dauzère-Pérès[1] Juliette Pouzet[2]

[1]Mines Saint-Etienne, Univ Clermont Auvergne
CNRS, UMR 6158 LIMOS
Gardanne, France
E-mail: {louis.fourcade, dauzere-peres}@emse.fr

[2]SNCF
Direction Technologies, Innovation et Projets Groupe
La Plaine Saint-Denis, France
Email: juliette.pouzet@sncf.fr

## 1 Problem description

Rail is one of the most energy-efficient and sustainable means of transporting goods. In the last decade, freight transportation has been one of the few economic sectors in Europe to experience a drastic increase in greenhouse gas emissions. To be competitive and increase its market share, rail freight transportation must be able to offer resource-efficient transportation plans.

Our work tackles the integrated problem of simultaneously optimizing train path selection and rolling stock assignment in rail freight transportation. More precisely, given a set of client requests, a list of available train paths and the characteristics of the available rolling stock units, railroad freight operators must take at least three key types of decisions in their planning process. The first problem is the **train path selection problem**, where a set of train paths to transport the client requests (also called demands in this abstract) in a timely manner over the physical network must be determined. The second problem, strongly related to the first one, is the **line planning problem**, where a detailed plan of convoys is determined. This plan specifies the trains and train paths assigned to each demand. The third problem is the **rolling stock planning problem**, where the rolling

*Corresponding author.

stock units must be assigned to the selected train paths and planned to ensure that the convoys can actually be performed. Although many structural and economical decisions, such as customer contracts and train path availability, are taken on a yearly basis, the transportation plans are optimized on a weekly basis.

By leveraging integer linear programming (ILP) and mathematical decomposition methods, we present two mathematical models and a Column Generation (CG) approach to solve the above three problems simultaneously. Real-world instances derived from 2022 and 2023 freight transportation data in the French railway network are used to analyze and compare the compact integrated ILP model and the CG approach.

## 2 Short literature review

Our work extends to four of the planning steps described in [1] for railway planning: Line Planning, Timetable Generation, Train Routing and Rolling Stock Schedules. As we are considering integrated planning decisions, our approach is relevant both at the strategic level and the tactical level. Rail freight transportation differs from rail passenger transportation in several key aspects. In passenger transportation and by extension in most articles optimizing integrated planning problems in this context, most of the lines, timetables, and routes are constructed as cyclical plans. Typical examples are recurring trips between cities many times per day or passenger-driven demands between a workplace and a home [2]. Rail freight transportation requires a much more demand-oriented approach [3].

The integrated resolution of several planning problems usually considered sequentially is not a field of research specific to rail transport. In fact, since the early 2000s, several problems have been studied in both the rail and air transport industries. A column generation model for the integrated line planning and rolling stock planning problem is introduced in [3], and an extension of the concept is proposed in [4]. Recently, column generation for an urban transit network was also considered with great results in [5]. Lagrangian decomposition is also considered in [6] and [7]. Heuristics and other solution approaches also exist, but they often rely on very specific operational constraints to be effective [1].

## 3 Method and models

Leveraging mathematical models already used at Fret SNCF for the different planning stages, we defined a compact MILP model that solves our integrated problem. This model is based on a double graph modeling for the selection of the trains paths and routes and for the definition of the rolling stock chained duties. Various coupling constraints focusing on weight assignment, rolling stock compatibility and train path coverage ensure the feasibility of the proposed integrated plan. Solving this model with a standard solver, although feasible in reasonable computational times for small instances, becomes impossible for large instances on the entire French network. This is why we considered decomposition methods. First we defined a Lagrangian decomposition scheme on two sub-problems. Then a column decomposition approach is proposed after rewriting the compact MILP model. We break down our integrated problem into two "natural" sub-problems: (1) A routing problem that determines a set of convoys for each customer and (2) A rolling stock planning problem that determines the individual rolling stock routes.

### 3.1 Column subsets

An originality of our approach is that, rather than only considering a single set of columns, a set of columns is associated to each problem. The master problem is used to couple both sets of columns. More precisely, the set of routing decisions includes available alternative routes for each customer. Enumerating these alternatives can be done in a relatively short computational time using a path enumeration algorithm. The set of rolling stock decisions is composed of plans for a single rolling stock unit. Each plan specifies the duties of a rolling stock unit. Since rolling stock duties are less timely constrained than the routes of customers and there are a very large number of possible rolling stock duties, the set of rolling stock plans cannot be enumerated in an exhaustive way. The pricing problem in the column generation approach is reformulated as a longest path problem in an acyclical graph and solved by a label propagation algorithm derived from the Bellman Ford approach.

### 3.2 Advantages of the Column Generation approach

Our column generation approach has several major advantages over the compact MILP model. First, the inclusion of discrete costs provides a streamlined way of introducing non-linear routing and rolling stock costs. Some rolling stock duties may have scaling costs when considering maintenance operations or network disruptions. Incidentally, railway operators sometimes need to deal with spot missions or unforeseen constraints that require the manual definition of a rolling stock plan. This is why we have implemented an interface for the managers, so that they can interact and manually add columns in the model, and even force columns to be used.

## 4 First numerical results and conclusions

Our models were tested on instances derived from the actual 2022 and 2023 annual freight plans of Fret SNCF. On small instances, the compact MILP model reaches very encouraging results, and confirms a gain of 3 to 4 rolling stock units for a one-week plan with

160 demands. However, the computational time increases exponentially with the original catalog of train paths and the number of demands to consider. The column generation approach is not fast on small instances, but larger instances can be solved more easily. The gains observed with the compact MILP model are confirmed with the column generation approach. However, the pricing problem in the column generation approach is currently quite slow to solve on large instances. We are currently investigating promising algorithms that are more efficient to generate a new subset of improving columns.

Finally, the global optimization model can probably not be used as such in a freight railway industrial context given the operational constraints in the planning process. However, an interactive approach giving leeway for managers to influence and tweak the optimal solution is being explored as it appears to be relevant for decision support.

## References

[1] Y.Yue, J. Han, S. Wang and X. Liu, "Integrated Train Timetabling and Rolling Stock Scheduling Model Based on Time-Dependent Demand for Urban Rail Transit: Integrated train timetabling and rolling stock scheduling model based on time-dependent demand for urban rail transit", *Computer-Aided Civil and Infrastructure Engineering*, 32(10), 856–873, 2017

[2] A. De-Los-Santos, G. Laporte, J. A. Mesa and F. Perea, "Evaluating passenger robustness in a rail transit network", *Transportation Research*, 20(1), 34–46, 2012.

[3] L. Bach, M. Gendreau and S. Wøhlk, "Freight railway operator timetabling and engine scheduling", *European Journal of Operational Research*, 241(2),309–319, 2015.

[4] L. Bach, T. Dollevoet and D. Huisman, "Integrating Timetabling and Crew Scheduling at a Freight Railway Operator", *Transportation Science*, 50(3):878–891, 2016.

[5] H. Pan, L. Yang and Z. Liang, "Demand-oriented integration optimization of train timetabling and rolling stock circulation planning with flexible train compositions: A column-generation-based approach", *European Journal of Operational Research*, 305(1),184–206, 2023.

[6] S. Dauzère-Pérès, D. De Almeida, O. Guyon and F. Benhizia, "A Lagrangian heuristic framework for a real-life integrated planning problem of railway transportation resources", *Transportation Research Part B: Methodological*, 74:138–150, 2015.

[7] X. Xu, C. Li and Z. Xu, "Integrated train timetabling and locomotive assignment", *Transportation Research Part B: Methodological*, 117,573–593, 2018.

# An optimal control model for determining freight rail transport access costs

**Ricardo Garcia-Rodenas**
Departamento de Matemáticas, Escuela Superior de Informática,
Universidad de Castilla-La Mancha, Spain. Email: Ricardo.Garcia@uclm.es

**Esteve Codina**
Department of Statistics and Operations Research,
Universitat Politècnica de Catalunya, Barcelona, Spain.

**Luis Cadarso**
Aerospace Systems and Transport Research Group,
European Institute for Aviation Training, Fuenlabrada, Madrid, Spain.

**Maria Luz Lopez-Garcia**
Departamento de Matemáticas, Escuela Superior de Informática,
Universidad de Castilla-La Mancha, Spain.

**José Ángel Martín-Baos**
Departamento de Matemáticas, Escuela Superior de Informática,
Universidad de Castilla-La Mancha, Spain.

## 1 Objectives

The liberalization of European railway markets, as outlined in European Directive EU 91/440/EEC, mandates a vertical separation governance structure within the railway industry (e.g. see [2] and [3] for an analysis). This vertical separation involves the establishment of two distinct entities: the Infrastructure Manager (IM), responsible for providing railway resources, and the Freight Operating Company (FOC), which operates freight services utilizing the infrastructure provided by the IM. Determining track access pricing between these entities is of utmost interest for European countries. Extensive monitorings are carried out (see e.g. [1]) emerging as a significant and urgent challenge, with profound implications not only for the profitability of IMs and FOCs but also for the efficiency and quality of the entire railway system.

Furthermore, this issue is pivotal in achieving the target objectives set by the European Union (EU), which aims to achieve a modal shift away from road freight transport towards more sustainable modes of transportation. Specifically, the EU has set a goal of achieving a modal shift of 30% by 2030 and at least 50% by 2050 for shipments exceeding 300 km. In this context, rail freight transport is widely recognized as a potentially cost-effective and environmentally sustainable alternative due to its ability to realize economies of scale, reduce pollutant emissions, and mitigate other externalities.

In this paper, we propose an optimal control method for pricing train paths so that the IM maximizes the utilization of public funding while promoting competition in the rail freight market and taking into account both environmental impact and road safety.

The methodology presents several challenges. First, the model needs to account for elastic demand to capture the effect of costs and travel times on modal split. For this reason, road freight transportation is simplified, and a modal split is performed using a logit model. Second, it involves replacing a basic toll per kilometer scheme with one that incorporates the capacity of the railway network into the pricing process. This means that congested routes should be subject to higher charges than less-demand routes. Hence, a dynamic cargo flow model with capacities on arcs has been chosen. The temporal aspect of the model allows for consideration of dynamic demand patterns and temporally variable capacities (day/night). The third element of the model deals with a non-additive cost structure and interdependence of train paths when sharing railway segments. This has been achieved by pricing the total revenue received for the use of train paths over the entire planning period.

## 2 Dynamic network modeling and demand

The dynamic network model captures the propagation of flows within the network. This model in turn consists of two elements. The first is the basic arc model and the second is the flow propagation model within the network by means of a point-queuing model, as those applied in dynamic traffic assignment problems (e.g. in [4]). The essential difference with the one presented here is that it allows flows in both directions for the same arc in order to model two-way circulations in some arcs of the network. The equations of arc dynamics are:

$$f(t) = f^s(t) + f^{\overline{s}}(t); \qquad q(t) = q^s(t) + q^{\overline{s}}(t)$$

$$d(t) = d^s(t) + d^{\overline{s}}(t); \qquad q^j(t) = \int_0^t \left[d^j(\xi) - f^j(\xi)\right] \mathrm{d}\xi, \qquad j \in \{s, \overline{s}\}$$

$$f(t+v) = \begin{cases} \min(d(t), k), & \text{if } q(t) = 0 \\ k, & \text{if } q(t) > 0 \end{cases}$$

There is a vertical queue at the arc's entrance $q^s(t)$. Congestion only affects the queue, not the travel time $v$ on the arc. It always takes the same amount of time to traverse the arc; the issue is that entry to the arc may be delayed.

*Network-based model.* The freight railway network is modeled by a graph $G = (V, A)$, where $A$ is a set of edges, and $V$ is a set of vertices. Each edge $a$ represents a portion of rail infrastructure with homogeneous characteristics (loading gauge, maximum slope, maximum permissible train weight, and maximum permissible train length). Space imitations do not allow to show the remaining equations of the model, but from them it will possible to evaluate the trip time $v_r(t)$ on path $r$.

We consider a cost structure as a service rather than a product. A transportation service is provided at each moment, and you must pay for the time of usage. The infrastructural performance of a path $r$ is determined by the corresponding worst performance of its arcs. This will determine that train composition will have a maximum length, determining the maximum number of freight railcars and, ultimately, the characteristics of the path, such as its maximum slope, will determine the maximum weight for a train of a given type to operate on $r$ with a single locomotive. Then, a prototype train that operating on path $r$ can be established. The cost structures we have considered are: (i) $A_r(t)$ gathers the cost of train driver(s), cost of locomotives, cost of rolling stock per unit of time. (ii) $\lambda_r(t)$ involves energy consumption and toll payments to the railway network operator. (iii) $C_r$ is a fixed cost that remains constant regardless of the train's characteristics, such as shunting costs at the origin and/or destination, or other fixed administrative costs. Then, equation (1) states that total demand $D_\omega(t)$, divides into the demand by road plus that served by rail. Equation (2) defines the queue $q^s_{a_r}$ stored in the first arc $a_r$ of path $r$, being $\kappa$ a constant transforming freight flows into railway flows. The modal split is determined by a logit model following (4).

$$D_\omega(t) = L_\omega(t) + \sum_{r \in R_\omega} d_{ra_r}(t) \tag{1}$$

$$q^s_{a_r}(t) = \int_0^t \left[\kappa \cdot d^s_{ra_r}(\xi) - f^s_{a_r}(\xi)\right] \mathrm{d}\xi, \tag{2}$$

$$U_r(t) = (A_r(t) + \lambda_r(t)) \cdot v_r(t) + C_r \tag{3}$$

$$d_{ra_r}(t) = \frac{\exp\left(\alpha^t - \beta U_r(t)\right)}{\exp\left(\alpha^L - \beta U_\omega\right) + \sum_{r' \in R_\omega} \exp\left(\alpha^t - \beta U_{r'}(t)\right)} \cdot D_\omega(t) \tag{4}$$

## 3 Objective function

The aim is maximizing IM access fee revenues while considering environmental and road safety costs.

$$\text{Max } z = \sum_\omega \sum_{r \in R_\omega} \int_0^{T_{\max}} \lambda_r(\xi) \cdot v_r(\xi) \cdot f_{ra_r}(\xi) \mathrm{d}\xi - \eta \sum_\omega \int_0^{T_{\max}} L_\omega(\xi) \mathrm{d}\xi$$

The first term represents all the revenues that the IM will receive for the use of train path $r$. The constraint imposes dynamic pricing, where different amounts are paid at different time intervals. The second term of the objective function is the number of tons transported by road and parameter $\eta$ economically values its environmental and road safety costs (see, for instance [5] for prizing methodologies of the environmental impact of carbon emissions).

## 4 Solution approach and numerical experiments

The resolution of the preceding model relies on its discretization. Discretization can be done at regular time intervals, or as we have done in this study, using irregular time intervals, while ensuring that the entry into the railway network via any train path corresponds to a unit of freight tonnage. These units can be linked to a prototype train, and occasionally, we will refer to it as a *package* or simply a *train*.

The approach for solving the problem will be a discrete event simulation model with events associated with packets entering from incident arcs or from external demand. The objective function is also discretized, and functions $\lambda_r(t)$ give rise to a finite set of optimization variables $\lambda_r(t_j) = \lambda_{jr}$. The resulting optimization model is the optimization of a function that depends on these variables and others obtained by the simulation model.

During the conference, the application of the previous scheme will be shown on a case study of the Mediterranean Corridor in Spain.

## References


[1] Ninth Annual Market Monitoring Working Document. IRG-rail. Independent Regulators' Group-rail. April 2021.

[2] Ait-Ali, A. and Eliasson, J. "European railway deregulation: an overview of market organization and capacity allocation". *Transportmetrica A: Transport Science* Vol 18 (3), pp 594-618. 2022.

[3] Nash C. "Rail infrastructure charges in Europe". *Journal of Transport Economics and Policy* . Vol 39 (3) p. 259-278. 2005.

[4] Li, J., Fujiwara, O. and Kawakami, S. "A reactive dynamic user equilibrium model in network with queues". Transportation Research Part B. 34 (2000). 605-624

[5] "Estimating the Value of Carbon: Two Approaches" (October 2020). https://media.rff.org/documents/RFF_NYSERDA_Valuing_Carbon_Synthesis_Memo.pdf

# A two-stage stochastic programming model with recourse for a Production Routing Problem with uncertain availability of vehicles


**Michel Gendreau**
CIRRELT and Département de mathématiques et de génie industriel
Polytechnique Montréal, Montreal, Canada
Email: michel.gendreau@polymtl.ca

**Alline Zanette**
CIRRELT and Département de mathématiques et de génie industriel
Polytechnique Montréal, Montreal, Canada

**Jorge E. Mendoza**
CIRRELT and Département de gestion des opérations et de la logistique
HEC Montréal, Montreal, Canada

**Walter Rei**
CIRRELT and Département d'analytique, opérations et techno. de l'information
École des Sciences de la Gestion, UQÀM, Montreal, Canada


## 1 Motivation

In the last decades, supply chain planning has become a major concern of many companies. It is now well known that optimizing one activity of the supply chain often prevents the achievement of better solutions in others [5]. Therefore, one should consider integrated problems, such as the Production Routing Problem (PRP), in which one performs the joint and simultaneous optimization of production, inventory, distribution, and routing decisions [1]. While many variants of the general PRP can be found in the literature, most of them consider only deterministic data. This is a significant concern, as uncertainty is a major issue in supply-chain management. Up to now, studies have considered uncertainty only in demand, quantity of products returned and quantity of components in each product. In this presentation, we consider a PRP over a finite horizon, with a single capacitated production facility, a single product, a fleet of homogeneous vehicles, and uncertainty in the availability of vehicles. This is a problem setting that has seldom been considered in previous studies (see, however, [9]), but that is commonly found in industrial environments.

## 2 Problem description

The PRP under study is defined on a complete graph $\mathcal{G} = \{\mathcal{N}, \mathcal{A}\}$, where $\mathcal{N} = \{0, \ldots, n\}$ is the set of nodes and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$ is the set of arcs. The plant is located at node 0 and $\mathcal{N}_c = \mathcal{N} \setminus \{0\}$ is the set of customers. A single product is produced by the plant with production capacity $C_t$ along a finite discrete time horizon $\mathcal{T} = \{1, \ldots, l\}$. When production takes place, a fixed setup cost $f$ and a variable production cost $g$ per unit are incurred. The products can be stored by the plant and by the customers up to an inventory limit of $L_i$, incurring an inventory holding cost per unit of $h_i$, for $i \in \mathcal{N}$. We observe that the holding costs may vary depending on each location.

The distribution is performed using a limited fleet of homogeneous vehicles, each with capacity $Q$. Each customer $i$ has a known demand $d_{it}$ in each period $t$, which needs to be fulfilled. While performing the deliveries, a transportation cost $c_{ij}$ is incurred when a vehicle travels directly from $i$ to $j$. Additionally, each tour starts and ends at the production plant, each vehicle can perform at most one trip per period, and split deliveries are not allowed. In this context, a set of routes $R_t$ is defined for each period $t \in \mathcal{T}$.

The number of available vehicles is an independently distributed random parameter $\mathcal{K}$, with respect to periods, and its value depends on the general random event $\omega \in \Omega$. Once the latter is realized, $\mathcal{K}$ becomes known, and it assumes the specific values $\mathcal{K}(\omega)$. In order to obtain a manageable model, one can sample $s \in \mathcal{S}$ scenarios from the random space to empirically define the possible realizations of the random parameter. In this case, $\mathcal{K}$ can assume independent values, with respect to period $t$ and scenario $s$, being represented by $\mathcal{K}(s) = \{K_1^s, \ldots, K_l^s\}$. We let $\mathcal{S} = \{1, \ldots, S\}$ be the finite set of all possible scenarios, and let $\rho^s > 0, \forall s \in \mathcal{S}$, be the probability that scenario $s$ occurs. For a given period $t$, if the minimum number of vehicles required to serve all the customers is not available, some routes cannot be performed and a recourse policy should be adopted.

The recourse policies involve redirecting the customers who could not be served due to the lack of vehicles. The first option is to reinsert those clients into existing routes, if the corresponding vehicle capacity allows it; the second is to outsource the distribution to a third party, which will individually deliver the products to each customer. The recourse policy chosen should be the one with the lowest cost.

The goal of the problem is to simultaneously minimize production, inventory holding, routing and recourse costs, while respecting inventory limits, and production and vehicle capacities.

## 3 Model

We propose a classical stochastic programming two-stage formulation with recourse: production, inventory and routing plans are decided in the first stage, while the recourse decisions belong to the second stage, after the realization of the random event $\omega \in \Omega$. An important feature of our modeling approach for the stochastic problem is that it relies on extending an efficient formulation for the deterministic PRP without vehicle indices (a so-called "two-index formulation"). In the first-stage model, the main decision variables are binary variables that control the production of each product in each period, whether a customer is visited in a given period or not, and routing decisions in each period; continuous variables are used to track quantities produced and delivered, as well as inventories of the product considered at the end of each period. The objective function minimizes the sum of production, transportation, inventory holding, and expected recourse costs.

In the first-stage problem, constraints are defined to represent the inventory balance at the production plant and clients. Other constraints enforce the inventory limit at the production facility while the quantity of products at the clients cannot exceed their inventory capacity after the deliveries. Some constraints force a setup if there is production in a specific period, also limiting the production quantity to the plant's capacity in each period. A positive delivery to client $i$ in period $t$ is allowed only if this node is visited in the same period; the quantity of products delivered is also limited by the minimum between the inventory limit at the retailer and the vehicle's capacity. Moreover, each client can be visited at most one time per period, and the degree constraints at clients and plant nodes are imposed. A customer can only be visited in a period $t$ if at least one vehicle leaves the depot in the same period, and the number of routes are limited by the maximum number of vehicles that can be available in a period. The compliance with the vehicle capacity is ensured by constraints that also prevent subtours from occurring. Another subtour elimination constraint (SEC) is included in the model in order to speed up the solution, as it is stronger.

The objective function of the second-stage model minimizes the recourse costs. The number of routes than can be maintained in a period is limited by the number of vehicles available. A constraint guarantees that if a route is kept, its clients cannot be reassigned to either another route or to receive an outsourced delivery; otherwise, the reassignment is performed. A client who was previously assigned to a route in period $t$ can only be transferred to a new route if the latter is kept in the solution. For each route that is not discarded, the amount of products delivered to the newly inserted customers has to respect the slack capacity of the vehicle.

## 4 Solution approach and computational results

To solve efficiently the proposed model, one must exploit the inherent structure of the problem at hand. An obvious approach is to apply Benders decomposition (BD) [2, 6] by separating the two stages of the problem. Direct application of BD is non-trivial and not really effective. This has forced us to explore a wide range of more advanced methods that include as Partial BD [3], Logic-based BD [4], Benders dual decomposition [7], and Local Branching-based heuristics [8].

Computational experiments are now under way on instances derived from the benchmark instances described in [1]. Detailed computational results will be reported at the conference.

## References

[1] Y. Adulyasak, J.-F. Cordeau and R. Jans, "The production routing problem: A review of formulations and solution algorithms", *Computers and Operations Research* 55, 141-152, 2015.

[2] J.F. Benders, "Partitioning procedures for solving mixed-variables programming problems, *Numerische Mathematik* 4, 238-252, 1962.

[3] T.G. Crainic, M. Hewitt, F. Maggioni and W. Rei, "Partial Benders decomposition: General methodology and application to stochastic network design", *Transportation Science* 55(2), 414-435, 2021.

[4] J. Hooker, "Logic-Based Benders Decomposition Tutorial", CPAIOR Master Class 2016, Banff, May 2016.

[5] F. Neves-Moreira, B. Almada-Lobo, J.-F. Cordeau, L. Guimarães and R. Jans, "Solving a large multi-product production-routing problem with delivery time windows", *Omega* 86, 154-172, 2019.

[6] R. Rahmaniani, T.G. Crainic, M. Gendreau and W. Rei, "The Benders decomposition algorithm: A literature review", *European Journal of Operational Research* 259(3), 801-817, 2017.

[7] R. Rahmaniani, S. Ahmed, T.G. Crainic, M. Gendreau and W. Rei, "The Benders dual decomposition method", *Operations Research* 68(3), 878-895, 2020.

[8] W. Rei, M. Gendreau and P. Soriano, "A hybrid Monte Carlo Local Branching algorithm for the single vehicle routing problem with stochastic demands", *Transportation Science* 44, 136-146, 2010.

[9] A. Zanette, W. Rei and M. Gendreau, "A chance-constrained model for a Production Routing Problem with uncertain availability of vehicles", Odysseus 2021, Tangiers, May 2022.

# A gas pipeline surveillance problem solved with a two-phase iterative approach using machine learning techniques

**Nadia Ghernaout**[a,b,*], **Martin Cousineau**[c], **Christelle Guéret**[a], **David Rivreau**[d]

[a] *Univ Angers, LARIS, SFR MATHSTIC, F-49000 Angers, France*

[b] *Research and Innovation Center for Energy (RICE), GRTgaz*

[c] *Department of Logistics and Operations Management, HEC Montréal, Montréal, Canada*

[d] *UCO, LARIS, SFR MATHSTIC, F-49000 Angers, France*

[*] Email: nadia.ghernaout@grtgaz.com

## 1 Introduction

GRTgaz, the French natural gas transmission system operator, conducts daily surveillance of the pipeline network. This annually organised inspection aims to ensure the safety and the availability of the 32,000 km pipeline network by identifying potential incidents that may appear. Planning and conducting these operations are costly regarding time, human and financial resources as each segment of the pipeline is associated with a specific inspection frequency, spacing of visits and surveillance modes (e.g., car, plane). Considering a year horizon, our aim is to plan and determine on a weekly basis the set of feasible tours and modes used that minimize the overall cost of the solution. Our problem is formulated as a Periodic Capacitated Arc Routing Problem (PCARP) with a multimodal fleet. Our problem is multi-depot and involves a heterogeneous fleet of vehicles where planes can use different depots for their departure and arrival. As such, it generalizes the Periodic Capacitated Arc Routing Problem (PCARP) and the Capacitated Arc Routing Problem with Intermediate Facilities (CARPIF) with multimodal fleet. There is a significant amount of literature on Arc Routing Problems (for instance, refer to [2]), but to the best of our knowledge, our problem has never been studied. In order to solve such complex problems, methods decomposing the problem, especially according to decision levels, have proven to be effective. For instance, [3] proposed a two-phase iterative heuristic to alternatively solve a lot-sizing and scheduling problem. In this context, over the past few years, there has been a growing interest in combining optimization and machine learning (ML) methods to decrease computational time especially for real-world problems. For instance, in [1] the authors investigate regression models to approximate the total distance of a VRP in order to plan waste collection. In this work, our contributions are two-fold. First, we tackle a multimodal fleet PCARP. Second, we propose a two-phase approach which iterates between tactical and operational decisions and includes machine learning techniques to provide fast cost approximations.

## 2 Problem Description

The aim is to inspect GRTgaz's territory which is divided into 75 sectors. The pipeline network is divided into portions (required edges) to be monitored either by motor vehicles (hereon, vehicles) or planes. The vehicles are allocated by sector, while planes are deployed over the full territory. We consider a year horizon $W$ of 52 periods (weeks) and we denote as $S$ the set of sectors.

Vehicle inspection is conducted in each sector $s \in S$ where a technician unit can perform up to 5 rounds a week $w \in W$. For each sector $s \in S$, we introduce a set of nodes $X^v(s)$ and a set of edges $E^v(s)$ accessible to vehicles. Each edge $(i, j) \in E^v(s)$ is associated with a traversal time $t^v_{ij}$. A subset $R^v(s) \subset E^v(s)$ of required edges must be inspected and is associated with a surveillance time $t^v_{ij}{}' > t^v_{ij}$. Furthermore each sector has a set of depots $D^v(s)$ where each tour must begin and end. A vehicle tour can not exceed $\delta^v$ minutes and its hourly cost is $c^v$ euros.

Plane inspection covers all the sectors for each week $w \in W$. Aerial technicians plan up to 7 rounds a week. Thus similar to vehicle inspection, we denote as $X^p$ the set of nodes, $E^p$ the set of edges associated with their traversal time $t^p_{ij}$, and $R^p \subset E^p$ the subset of required edges associated with their surveillance time $t^p_{ij}{}'$. Plane tour can't exceed $\delta^p$ minutes and their hourly cost is $c^p$ euros. Unlike motor vehicle tours, planes do not necessary end their tour where they started. We denote as $D^p$ the set of aerodromes where plane can take off and land for each tour. We denote as $d^m \in D^p$ the main depot where the plane must be at the beginning and end of each week.

Finally, we consider an undirected connected graph $G = (X, E)$ with nodes set $X = \{\bigcup_{s \in S} X^v(s)\} \cup X^p$ and edges set $E = \{\bigcup_{s \in S} E^v(s)\} \cup E^p$. Each required edge $(i, j) \in R = \{\bigcup_{s \in S} R^v(s)\} \cup R^p$ must be monitored $f_{ij}$ times across the time horizon and consecutive visits should be spaced at least $\Delta_{\min}(i, j)$ and at most $\Delta_{\max}(i, j)$ weeks apart. The objective is to minimize the total routing and surveillance cost.

## 3 Solution Approach

Our solution consists in an iterative two-phase approach that decomposes the multimodal fleet PCARP into two sub-problems. First, the scheduling phase aims to allocate each required edge to weeks of the year horizon and surveillance modes based on frequency constraints and an estimation of overall cost provided by regression models. Then, in the routing phase, the objective is to build tours for each week, sector and mode based on the planning provided by the scheduling phase. Once tours are computed and associated to their corresponding costs, this additional data is used to retrain regression models that will enable to refine predictions. Then, a new iteration begins and the scheduling phase uses updated regression coefficients. This method iterates until a stopping criteria is met, e.g., a time limit, a number of iterations or a non-improvement of the solution.

### 3.1 Scheduling Phase

The scheduling step is formulated as a Mixed Integer Linear Programming (MILP) model. The objective function is the minimization of the total estimated cost of the tours, i.e., $\sum_{s \in S} \sum_{w \in W} \hat{c}^{v,sw} + \sum_{w \in W} \hat{c}^{p,w}$ where the $\hat{c}$ values are estimated with a pretrained linear regression model for each surveillance mode. In particular, these values are estimated as $\hat{c} = \theta^{\top} \alpha$ within the contraints of the scheduling model, where $\theta$ and $\alpha$ are $k$-dimensional vectors denoting respectively the parameters and features of a linear regression model. The first $q$ features are fixed and given, while the remaining $k - q$ features are constructed from the decision variables of the scheduling model (see Section 3.3 for more details). Finally, our scheduling model is subject to frequency constraints, i.e., minimum frequency of visits for each required edge, and minimum and maximum time spacing for consecutive surveillance on the same edge. We also ensure that a required edge can only be serviced by a suitable mode.

### 3.2 Routing Phase

Regarding vehicle routing, the objective is to solve a CARP for each sector $s \in S$ and each week $w \in W$. This problem is also formulated as a MILP model and is subject to routing constraints: the tour must start and end at the same depot and a vehicule tour can not exceed a certain time limit. Concerning plane routing, the aim is to solve a CARPIF for each week $w \in W$. Indeed, contrary to vehicle routing, a plane can use several intermediate depots between tours. We first solve a Rural Postman Problem (RPP) that creates a giant tour starting and ending at the main depot and serving required edges. Then, to ensure that the tour time limit is respected, we add intermediate depots. The output for this phase is a set of tours for each week, surveillance mode and sector (if applicable).

### 3.3 Cost Approximation Model

Given a set of required edges and a mode, the cost approximation model estimates a tour's cost. For this purpose, several spatial features are extracted in order to describe problem instances. A first set of $q$ features in $\alpha$ (see Section 3.1) is specific to each sector, without considering the selected required edges. They are computed once at the pre-training of the model and given at each iteration. They provide overall information about the sector, e.g., area and perimeter of the convex hull, total length of all required edges, position of depots regarding those edges. Remaining $k - q$ features in $\alpha$ globally characterize the distances between selected required edges and their distance to the depot(s). Moreover, some features are specific to vehicle and plane tours. We select the most relevant ones using regularization methods (e.g., lasso regression). Finally those variables are used to train a multiple linear regression model for routing cost approximation. In order to embed these models into the MILP scheduling phase we only keep linear variables. Model's parameters are passed to the scheduling phase and are updated at each iteration.

## 4 Conclusion

GRTgaz's gas pipeline surveillance problem is very challenging from a computational point of view as the real world instance contains close to 20,000 required edges to serve periodically using different modes. Regression models have shown to be effective for providing very fast cost approximation for plane and vehicule tours. Those models were validated on GRTgaz real world instance. Computational results obtained with the whole approach will be presented at the conference.

## References

[1] Fabian Akkerman, Martijn Mes, and Wouter Heijnen. Distance approximation for dynamic waste collection planning. In *Computational Logistics: 11th International Conference, ICCL 2020, Enschede, The Netherlands, September 28–30, 2020, Proceedings 11*, pages 356–370. Springer, 2020.

[2] Ángel Corberán, Richard Eglese, Geir Hasle, Isaac Plana, and José María Sanchis. Arc routing problems: A review of the past, present, and future. *Networks*, 77(1):88–115, 2021.

[3] Stephane Dauzère-Péres and Jean-Bernard Lasserre. Integration of lotsizing and scheduling decisions in a job-shop. *European Journal of Operational Research*, 75(2):413–426, 1994.

# Combining ground drones, public transportation and traditional vehicles in last-mile distribution

**Gianpaolo Ghiani, Emanuela Guerriero, Emanuele Manni, Deborah Pareo**
Department of Engineering for Innovation, University of Salento, Lecce, Italy
Email: emanuele.manni@unisalento.it

## 1 Introduction

In the past few years, the rise of e-commerce and the push towards a more sustainable last-mile logistics have encouraged the development of new distribution paradigms with vehicle less sensitive to traffic congestion, such as aerial and ground-based drones. In this article we deal with a routing problem in which last mile-distribution is carried out by traditional vans along with a fleet of ground drones (GD) starting their trips from stations of the public transportation (PT) network, where we assume that the freight arrive by using the PT lines. The problem entails building both van and drone routes in such a way to minimize the sum of fixed and variable costs while satisfying the constraints imposed by the limited vehicle range and capacity. We develop two variants of a matheuristic scheme based on a generalization of [1] in which van and drone routes are jointly designed.

Our problem is related to a number of drone-assisted *Vehicle Routing Problems* (VRPs, see [3] for a comprehensive overview of the role played by GDs in city logistics). Among all the most relevant contributions, [4] propose a two-echelon van-based robot routing problem with pickup and delivery that is solved with an *Adaptive Large Neighborhood Search* (ALNS) algorithm. [5] introduce the VRP with time windows and delivery robots for which a two-stage matheuristic based on clustering approaches is designed to tackle medium-size instances. Our problem is also related to three more subfields of the VRP literature. First of all, it is linked to fleet size and mix problems (see [6] for a literature survey). Secondly, it can be cast as a multi-depot VRP ([7]). Finally, it is a location-routing problem since it includes the selection of the PT stations where the GDs are based ([8]). In this light, a contribution close to our paper is [9], in which formulations and a heuristic are presented for the fleet size and mix location-routing problem with time windows. However, our paper differs for two main aspects: only ground drones may be based at the stations of the PT system; range constraints affect the design of the GD routes.

## 2 Matheuristic

The basic idea of our matheuristic approach is to solve a Location-Routing Problem with Vans and autonomous delivery Robots (LRPVR) in two steps. In step 1, we solve a hybrid set-partitioning and location-based model in which the ADR routes are determined using a set-partitioning approach, whereas for van routes we use a formulation inspired by the Location Based Heuristic (LBH) of [1], which is proven to be asymptotically optimal for any distribution of customer demands and locations. LBH is based on formulating the routing problem as a location problem known as the Capacitated Concentrator Location Problem (CCLP). Then, in step 2, the solution to the CCLP is transformed into a solution to the routing problem.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a complete graph. Here, $\mathcal{V} = \{0\} \cup \mathcal{V}^c \cup \mathcal{V}^p$ is the set of vertices, for which 0, $\mathcal{V}^c$, and $\mathcal{V}^p$ represent the vans depot, the set of customers, and the set of PT stops, respectively and $\mathcal{A} = \mathcal{A}^v \cup \mathcal{A}^d$ is the arc set, where $\mathcal{A}^v$ contains the arcs that can be traversed by vans, whereas $\mathcal{A}^d$ is the set of arcs that can be traversed by delivery robots.

The cost of each van route is approximated as the cost of the simple tour that starts at the depot, goes to the concentrator and then back to the depot, plus the costs of inserting customers assigned to the concentrator into such a simple tour. In the CCLP, each customer is a potential concentrator, and the cost of selecting $j \in \mathcal{V}^c$ as a concentrator is approximated as $2c_{0j}$. Then, the approximated cost of associating customer $i \in \mathcal{V}^c$ with concentrator $j \in \mathcal{V}^c$ is determined as in the Seed Tours Heuristic [1], i.e., $\bar{c}_{ij} = c_{0j} + c_{ji} + c_{i0} - c_{0j} - c_{j0} = c_{ji} + c_{i0} - c_{j0}$.

Moreover, we denote as $d_i$ the demand of customer $i \in \mathcal{V}^c$; $q^v$ and $q^d$ the capacities of a van and a drone, respectively; $l_{ij}$ the distance between $i \in \mathcal{V}$ and $j \in \mathcal{V}$; $L^v$ and $L^d$ the maximum distance that can be covered by a van and a drone, respectively. In addition, let $\mathcal{R}^d$ be the set of drone routes satisfying the capacity and maximum traveled distance constraints, and $b_r$ the cost of route $r \in \mathcal{R}^d$. Finally, let $a_{ir}$ be a binary constant equal to 1 if and only if customer $i \in \mathcal{V}^c$ is included in route $r \in \mathcal{R}^d$.

Decision variables are: $z_i$, equal to 1 if $i \in \mathcal{V}^c$ is served by a drone and 0 if it is served by a van; $y_j$, equal to 1 if $j \in \mathcal{V}^c$ is a concentrator, and 0 otherwise; $s_r$, equal to 1 if drone route $r \in \mathcal{R}^d$ is selected and 0 otherwise; $x_{ij}$, $i, j \in \mathcal{V}^c$, are decision variables equal to 1 if customer $i$ is linked with concentrator $j$, and 0 otherwise; we assume $x_{jj} = 1$ if $j \in \mathcal{V}^c$ is a concentrator.

Thus, our mathematical formulation is:

$$\text{Minimize} \sum_{r \in \mathcal{R}^d} b_r s_r + 2 \sum_{j \in \mathcal{V}^c} c^v_{0j} y_j + \sum_{i,j \in \mathcal{V}^c, i \neq j} \bar{c}_{ij} x_{ij} \tag{1}$$

s.t.

$$\sum_{r\in\mathcal{R}^d} a_{ir}s_r \geq z_i \qquad \forall i \in \mathcal{V}^c \tag{2}$$

$$\sum_{j\in\mathcal{V}^c} x_{ij} = 1 - z_i \qquad \forall i \in \mathcal{V}^c \tag{3}$$

$$x_{ij} \leq y_j \qquad \forall i, j \in \mathcal{V}^c. \tag{4}$$

$$\sum_{i\in\mathcal{V}^c} d_i x_{ij} \leq q^v \qquad \forall j \in \mathcal{V}^c \tag{5}$$

$$x_{ij} \in \{0,1\} \qquad \forall i, j \in \mathcal{V}^c \tag{6}$$

$$y_j \in \{0,1\} \qquad \forall j \in \mathcal{V}^c \tag{7}$$

$$s_r \in \{0,1\} \qquad \forall r \in \mathcal{R}^d \tag{8}$$

$$z_i \in \{0,1\} \qquad \forall i \in \mathcal{V}^c. \tag{9}$$

The aim of objective function (1) is to minimize the sum of the costs of ADR routes and the approximated costs of van routes. Constraints (2) are the set-partitioning constraints. Constraints (3) impose that, if variable $z_i$ is equal to 0, then customer $i \in \mathcal{V}^c$ is assigned to exactly one concentrator. Constraints (4) ensure that, when a customer $i \in \mathcal{V}^c$ is assigned to a concentrator $j \in \mathcal{V}^c$, then the corresponding $y_j$ is equal to 1. Moreover, (5) are capacity constraints, whereas constraints (6)–(9) enforce the domain of the variables.

## 3 Discussion section

We have compared our approach (MATH-mTSP and MATH-mVRP) with both a mathematical formulation for the whole problem solved by an off-the-shelf solver (SOLV) and a standard ALNS procedure with destroy and repair operators inspired by [4]. We have considered instances with up to 500 customers related to a drug distribution problem to pharmacies in the urban area of Rome, Italy (see [10] for a description of the instances and the cost structure). The experiments have shown that the proposed approach can provide consistent solution improvements, in particular for the larger instances. Indeed for instances with 50 and 100 customers SOLV outperform the other approaches while when the instance size increases to 200, MATH-mTSP is slightly better. Moreover for the largest instances (300, 400 and 500 customers) MATH-mTSP always finds the best solution.

## Acknowledgements

This research was partly supported by the Ministero dell'Università e della Ricerca (MIUR) of Italy, PRIN project "Optimizing sustainable multi-modal and multi-tasking last-mile distribution system with carbon-free autonomous vehicles, ground robots, drones, and public transport", CUP Master H53D23002000006, CUP F53D23002720006. This support is gratefully acknowledged.

## References

[1] J. Bramel and D. Simchi-Levi, "A location based heuristic for general routing problems", *Operations Research* 43(4), 649-660, 1995.

[2] J. Bramel and D. Simchi-Levi, "On the effectiveness of set covering formulations for the vehicle routing problem with time windows", *Operations Research* 45(2), 295-301, 1997.

[3] O. Kunze, "Replicators, Ground Drones and Crowd Logistics A Vision of Urban Logistics in the Year 2030", *Transportation Research Procedia* 19, 286-299, 2016.

[4] S. Yu, J. Puchinger and S. Sun, "Van-based robot hybrid pickup and delivery routing problem", *European Journal of Operational Research* 298(3), 894-914, 2022.

[5] C. Chen, E. Demir, Y. Huang and R. Qiu, "The adoption of self-driving delivery robots in last mile logistics", *Transportation Research Part E: Logistics and Transportation Review* 146, 102214, 2021.

[6] A. Hoff, H. Andersson, M. Christiansen, G. Hasle and A. Løkketangen, "Industrial aspects and literature survey: Fleet composition and routing", *Computers & Operations Research*, 37(12), 2041-2061, 2010.

[7] T.-R.-P. Ramos, M.-I. Gomes and A.-P.-B. Póvoa, "Multi-depot vehicle routing problem: a comparative study of alternative formulations" *International Journal of Logistics Research and Applications*, 23(2), 103-120, 2020.

[8] S.-T.-W. Mara, R.-J. Kuo and A.-M.-S. Asih, "Locationrouting problem: a classification of recent research", *International Transactions in Operational Research* 28(6), 2941-2983, 2021.

[9] C. Koç, T. Bektaş, O. Jabali and G. Laporte, "The fleet size and mix location-routing problem with time windows: Formulations and a heuristic algorithm" *European Journal of Operational Research*, 248(1), 33-51, 2016.

[10] A. De Maio, G. Ghiani, D. Laganà and E. Manni. "Sustainable last-mile distribution with ground drones and public transportation". 2023. Submitted.

# Efficient Inter Terminal Container Transport using Amphibious Vehicles - A Simulation Approach

**Vijay Sathya Ghiridharan**
Department of Maritime and Transport Technology,
Delft University of Technology, Delft, The Netherlands
Email: vijaysathya1998@gmail.com

**Dr. Lavanya Meherishi**
Department of Engineering,
University of Exeter, Exeter, The United Kingdom

**Prof.Dr.Ir. Dingena Schott**
Department of Maritime and Transport Technology,
Delft University of Technology, Delft, The Netherlands

**Dr. Jovana Jovanova**
Department of Maritime and Transport Technology,
Delft University of Technology, Delft, The Netherlands

## 1 Introduction

Globalization has led to a substantial surge in international trade over the past century. Due to this increase the demand for efficient maritime trade operations has never been more crucial. This phenomenon has driven the need for innovations in container handling equipment and transshipment strategies within container terminals. In this context, a novel concept of Amphibious Autonomous Guided Vehicles (AAGVs) emerges as a potential game-changer. AAGVs, capable of seamlessly transitioning between land and water, present an innovative solution to streamline container transshipment processes. By integrating these vehicles into container terminal operations, we aim to reduce re-handling points, alleviate congestion, and ultimately enhance overall port efficiency. This research paper investigates the potential impact of implementing AAGVs in container terminal operations, with a specific focus on the Port of Rotterdam. By employing simulation-based analyses, we aim to quantitatively compare the efficiency gains achieved through this innovative approach. Through this study, we seek to provide valuable insights into the transformative potential of AAGVs in optimizing container terminal logistics. AAGVs possess the unique ability to operate on both land and water, potentially replacing conventional AGVs and barges. This innovation stems from the adaptability of existing Amphibious vehicle technology. The design of the AAGV closely mirrors conventional AGVs in terms of size and power. Notably, it seamlessly transitions between land and water autonomously, utilizing conceptual transfer systems. This enables it to navigate calm port waters efficiently. The design emphasizes compatibility with existing port machinery. The AAGV's versatility allows it to interact with all sides of the port, potentially replacing a number of handling equipment required for container transshipment [1].

# 2 Methodology

This section analyzes and optimizes specific scenarios using modeling and simulation. It starts with a review of the Benchmark scenario, detailing current container logistics for Inter-Terminal Transport (ITT). Four distinct cases, tailored to different needs, are analyzed and compared against the Benchmark, incorporating both current and proposed chains with AAGVs. Using AnyLogic: Personal Learning Edition 8.8.3 for Agent-Based Modeling, the study dissects container handling dynamics between terminals across four primary cases: Storage to Storage(SS), Vessel to Vessel(VV), Storage to Vessel(SV), and Vessel to Storage(VS). These cases offer insights into complex container re-handling points, with subsections exploring specific sub-cases based on the inter-terminal handling equipment used. This structured breakdown enables a comprehensive analysis of logistic chain design intricacies for terminal-to-terminal container transport.

## 2.1 State Chart Model

The model is used to initiate a process. The state first enters to normal work where the terminals are at a base state where there is no requirement for inter terminal transport. Then at a defined rate, the state changes from normal work to wanting details.

$$TransitionRate = \frac{ContainerDemand}{CapacityofMaterialHandlingEquipment} \tag{1}$$

The wanting details state is a state which triggers the requirement for Inter Terminal Transport. In this state the process model is initiated. Upon the receipt of a message "Delivered!", the state changes back to normal work.

## 2.2 Process Model

Process modelling Library is used to simulate container delivery logistics between terminals. The model encompasses demand order generation, resource allocation, container loading with equipment variations, navigation to destinations, delivery processing, and task completion notifications. Output metrics include container output and average transport time per container, providing a robust framework for logistic analysis and optimization.

## 2.3 Optimization Experiment

An optimization experiment is required to ensure efficient usage of the resource units. To ensure this efficient usage, utilization of the resource unit has to be optimized.

$$UtilizationRate = \frac{NumberofResourceUnitsBeingUsed}{TotalNumberOfResourceUnits} * 100 = \frac{Demand(D)}{FleetSize(V)} * 100[\%] \tag{2}$$

Therefore a Genetic Algorithm optimization is used to set agent utilization to a maximum of 85%. This percentage is assumed to be the upper limit of the operating window to ensure resource redundancy in unexpected situations. The objective is to maximize the truck utilization while ensuring it does not exceed 85%.

Table 1: Indices, Sets, Parameters and Decision Variable

| | |
|---|---|
| **Indices and Sets** | |
| i | Index of cases (i) $\forall i \in I = \{$SS-T, SS-B, SS-A, VV-T, VV-B, VV-A, SV-T, SV-B, SV-A, VS-T, VS-B, VS-A$\}$ |
| **Parameters** | |
| $D_i$ | Number of resource units used in case i $\forall i \in I$ |
| $V_i$ | Total number of resource units in case i $\forall i \in I$ |
| **Decision Variable** | |
| $U_i$ | Utilization rate of resource unit in case i $\forall i \in I$ and $U_i = \frac{D_i}{V_i}$ |

Objective Function:

The objective function is to maximize the utilization of the total number of resource units in the system.

$$Maximize \quad U_i \quad \forall i \in I \tag{3}$$

Main Constraint:

This constraint ensures that the utilization rate is maintained lesser than or equal to the chosen optimality of 85%.

$$U_i \leq 85\% \quad \forall i \in I \tag{4}$$

Additional Constraint 1:

This constraint ensures that the demand parameter is lesser than the fleet size parameter. This is because the number of resource units used has to be lesser than the total number of resource units.

$$D_i \leq V_i \quad \forall i \in I \tag{5}$$

Additional Constraint 2:

This constraint ensures that the decision variable lies between 0 and 100%.

$$U_i \in [0, 100][\%] \quad \forall i \in I \tag{6}$$

Additional Constraint 3:

This constraint ensures that the demand parameter is always a natural number.

$$D_i \in N \quad \forall i \in I \tag{7}$$

Additional Constraint 4:

This constraint ensures that the fleet size parameter is always a natural number.

$$V_i \in N \quad \forall i \in I \tag{8}$$

Where,

- N is the set of natural numbers; N$\rightarrow [1, \infty)$
- Percentage is a positive real number

The outputs of the simulation are post processed to obtain KPIs such as throughput and fulfillment rate. Throughput is the rate at which a material moves through a system per unit time and fulfillment rate is the ratio between containers that passed through the system with respect to the number of containers that were supposed to pass.

## 3 Case Study - The Port of Rotterdam

The Port of Rotterdam stands as a pivotal global port, crucial for European trade, and embarked on the expansive Maasvlakte 2 project to address the escalating demands of international shipping and trade. This initiative involved the establishment of cutting-edge container handling terminals like APM Terminals Rotterdam and Rotterdam World Gateway, equipped to manage the world's largest container vessels. This development solidifies Rotterdam's position as a central hub for containerized cargo in Europe. Furthermore, the project involved extensive enhancements, including the deepening and widening of navigation channels, facilitating access for larger vessels with drafts of up to 23 meters. Such deepwater access becomes pivotal in accommodating the escalating sizes of container ships [3]. In 2014, the Deep Sea Terminals of Maasvlakte handled 1803 containers [4], and projections for 2030 estimate a daily demand ranging from 5890 to 9151 TEU/day [2]. This research aims to assess the viability of employing Amphibious Automated Guided Vehicles (AAGVs) over Truck and Barge for Container Inter-Terminal Transport (ITT) in ports like Rotterdam.

### 3.1 Results and Conclusion

This sections sums up all the inferences made from various results for the case study across all experiments. Findings reveal that AAGVs, while causing maximum fleet size due to uni-modality, exhibit longer transfer times compared to trucks, with barges taking the longest. AAGVs significantly enhance throughput and fulfillment rates, outperforming trucks in most cases where trucks display slightly lower rates. Barges, despite their high capacity, yield notably lower throughput and fulfillment rates due to additional process requirements. Overall, AAGVs and trucks emerge as more efficient than barges for Maasvlakte's inter-terminal transport. Graphs in figures like 1, 2, 3 & 4 demonstrate these trends, prompting consideration of averaging effects, especially as AAGVs and trucks show closely aligned performance, demanding further analytical attention.

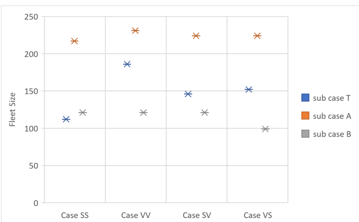



Figure 1: FleetSize

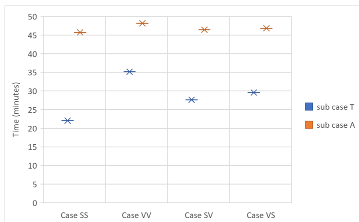



Figure 2: Time

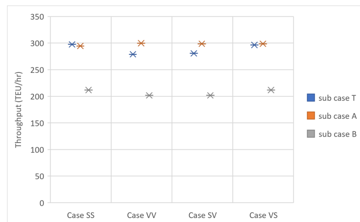



Figure 3: Throughput

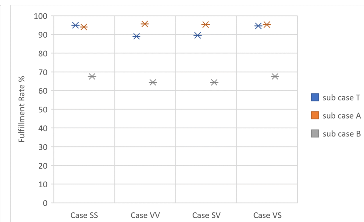



Figure 4: Fulfillment Rate

To understand the performance edge of AAGVs and trucks over each other, a route analysis is performed. From this it can be seen that performance is affected not just route wise, but also case wise. The result inferences of employing AAGVs as the inter terminal handling equipment route wise are discussed below and correlated with respect to distance.

The analysis of multiple routes between Hutchinson Ports ECT Euromax and various destinations reveals consistent trends favoring AAGVs over trucks in terms of throughput and fulfillment rates. Across distinct cases, AAGVs consistently exhibit superiority. For instance, in the route to Rotterdam World Gateway, AAGVs in Case VV demonstrate a 9% higher throughput and a 6% higher fulfillment rate compared to trucks. Similarly, to APM Terminals Maasvlakte 2, AAGVs consistently outperform trucks with advantages in throughput and fulfillment rates ranging from 8% to 9% across different cases. Routes to Hutchinson Ports Delta 2 also reflect this trend, displaying AAGVs achieving significantly higher throughput (up to 28%) and fulfillment rates (up to 21%) in Case VV, compared to trucks. Additionally, in sub-cases SV and VS, AAGVs maintain higher rates, ranging from 3% to 4%, compared to trucks' lower performance. These findings underscore the overall advantage of AAGVs over trucks across multiple routes, consistently contributing to higher throughput and fulfillment rates in various transport scenarios.

It's apparent that AAGVs consistently cover shorter distances compared to trucks across various routes due to geographical factors. This reduced travel distance from Hutchinson Ports ECT Euromax to Rotterdam World Gateway, APM Terminals Maasvlakte 2, Hutchinson Ports Delta 2, and Hutchinson Ports ECT Delta hints at potential logistical efficiency for AAGVs. This efficiency contributes to their superior performance in throughput and fulfillment rates observed in the analyzed scenarios. This observation suggests a specific profile for routes suitable for employing AAGVs. AAGVs could be considered as the inter-terminal handling equipment when their travel distance is at least 50% less than that covered by trucks. Additionally, a sensitivity analysis is performed to see how change in demand affects these KPIs. Fleet Size increase with a constant slope while time taken for a handling equipment to move from origin to destination remains constant. Throughput increases with a constant slope and fulfillment rate remains constant. The slope of throughput is greater than that of the fleet size. Hence, employing AAGVs is suitable when demand is greater for short range inter terminal transport. Employing AAGVs also contribute towards freeing barge berths and the work schedule of multiple re-handing equipment within a terminal.

# References

[1] Vijay Sathya Ghiridharan. "Efficient Inter Terminal Container Transport using Amphibious Vehicles - A Simulation Approach". In: *repository.tudelft.nl* (2023). URL: `http://resolver.tudelft.nl/uuid:2b90cd21-a572-46b8-bcc9-d6b976bbcb60` (visited on 03/03/2024).

[2] Lars de Lange. *Inter-terminal transport on Maasvlakte 1 and 2 in 2030 Towards a multidisciplinary and innovative approach on future inter-terminal transport options Deliverable 1.2 Generator for future inter terminal container transport demand scenarios on the Maasvlakte 1 and 2*. 2014. URL: `http://inter-terminal.net/itt-reports/ITT%20-%20D1.2%20-%20Scenarios%20-%20Demand%20scenario%20generation.pdf` (visited on 09/18/2023).

[3] *New port area at sea*. Port of Rotterdam. URL: `https://www.portofrotterdam.com/en/building-port/ongoing-projects/new-port-area-sea`.

[4] Adrie Spruijt, Ron van Duin, and Frank Rieck. "Intralog Towards an autonomous system for handling inter-terminal container transport". In: *Proceedings of EVS30 Symposium* (2017). URL: `http://resolver.tudelft.nl/uuid:9026d12a-0f9c-4a6b-a3fa-6c04f8668102`.

# Vehicle Routing Problem with Divisible Deliveries and Pickups under Demand Uncertainty

Alessandro Gobbi, Daniele Manerba
Department of Information Engineering, Università degli Studi di Brescia, Brescia, Italy
Email: alessandro.gobbi@unibs.it

Francesca Vocaturo
Department of Economics, Statistics and Finance "Giovanni Anania"
Università della Calabria, Rende (CS), Italy

## 1 Introduction

Nowadays, e-commerce companies have to efficiently manage both the delivery of ordered goods and the return requests, i.e., they also deal with the so-called Reverse Logistics. More precisely, along with delivery operations, a company must accommodate non-delayable (or *mandatory*) return requests. At the same time, to increase customer satisfaction and speed up the return of goods to the market, delayable (or *optional*) return requests can be fulfilled as well. For this reason, in this work, we study a stochastic version of the Vehicle Routing Problem with Divisible Deliveries and Pickups [1], where the optional return demand is subject to uncertain positive variations as in realistic scenarios [2]. With the aim of minimizing the total cost, the problem seeks to route a homogeneous fleet of vehicles so as to satisfy the mandatory deliveries and pickups and to ensure that at least a minimum percentage of optional pickups is fulfilled as well, to avoid that the company accumulates too many pickup requests for the upcoming days. To deal with the uncertain environment, we propose a new problem formulation based on two-stage Stochastic Programming (SP) with recourse. In particular, our second-stage problem involves the possibility of redirecting vehicles to the depot (*detour*) to unload already picked-up goods and then fulfill the remaining customers' requests or activate *spot-market* pickup services.

## 2 Problem Statement and Mathematical Formulation

Let $M = 1, 2, \ldots, m$ be the set of customers. Let be $N = 1, 2, \ldots, 2m$ the set of customer requests, in which the item $i$ and $i+m$ are the pickup and delivery requests of customer $i$, respectively. Therefore, we consider $N$ split in two disjunctive subsets $N_D$ and $N_P$ where the former includes all the delivery requests and the latter includes all the pickup requests. Let us define a graph $G = (V, A)$, with node set $V = N \cup \{0\}$, where 0 is the depot, and arc set $A = \{(i,j) : i, j \in V, i \neq j\}$. Each node $i \in N_D$ and $j \in N_P$ has a mandatory amount of goods $q_i, q_j \geq 0$ to deliver or to pick up, respectively. Moreover, each node $i \in N_P$ has an optional pickup demand composed of a deterministic term $\bar{o}_i \geq 0$, and a stochastic oscillation $\tilde{o}_i(\xi) \geq 0$ that depends from a random variable $\xi$. A travel cost $c_{ij} \geq 0$ is assigned to each arc $(i,j) \in A$, with $c_{ij} = 0$ if $|i - j| = m$. Finally, $\alpha$ is the minimal percentage of optional demand to fulfill by vehicles, each of them has the same capacity $C$. The problem aims at determining a set of routes to visit all the customers to fulfill their requests, minimizing the overall travel cost. Note that a customer who needs both pickup and delivery may be served through separate visits. However, each individual pickup or delivery request must be fulfilled within a single visit.

Let us define the following variables: $x_{ij} \in \{0,1\}$ taking value 1 if a vehicle travels on arc $(i,j) \in A$, 0 otherwise (first stage route); $y_i \in \{0,1\}$ taking value 1 if a vehicle visits node $i \in N$, 0 otherwise; $z_i \in \{0,1\}$ taking value 1 if a vehicle fulfills optional demand of customer $i \in M$, 0 otherwise; $D_{ij}, P_{ij} \geq 0$ measuring the amount of delivery and pickup goods carried on arc $(i,j) \in A$, respectively. The first-stage problem is then:

$$\min \quad \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} + \mathbb{E}[\delta(\mathbf{x}, \mathbf{y}, \mathbf{D}, \mathbf{P}, \mathbf{z}, \xi)] \tag{1}$$

subject to

$$\sum_{i \in V} x_{ji} = \sum_{i \in V} x_{ij} = y_j \quad j \in N \tag{2}$$

$$y_{j+m} \leq y_j \quad j \in N_P \tag{3}$$

$$q_j \leq C y_j \quad j \in N \tag{4}$$

$$z_j \leq y_j \quad j \in N_P \tag{5}$$

$$\sum_{i \in V} D_{ij} - q_j = \sum_{i \in V} D_{ji} \quad j \in N_D \tag{6}$$

$$\sum_{i \in V} P_{ij} + q_j + \bar{o}_j z_j = \sum_{i \in V} P_{ji} \quad j \in N_P \tag{7}$$

$$\sum_{i \in V} D_{ij} = \sum_{i \in V} D_{ji} \quad j \in N_P, \quad \sum_{i \in V} P_{ij} = \sum_{i \in V} P_{ji} \quad j \in N_D \tag{8}$$

$$\sum_{i \in N} D_{i0} = 0, \quad \sum_{i \in N} P_{0i} = 0 \tag{9}$$

$$D_{ij} + P_{ij} \leq C x_{ij} \quad (i,j) \in A \tag{10}$$

$$\sum_{i \in N_P} \bar{o}_i z_i \geq \alpha \sum_{i \in N_P} \bar{o}_i \tag{11}$$

Eqs. (2) are classical pairing constraints. Eqs. (3) impose that, for a customer having a non-null mandatory delivery request, also its pickup node is visited, thus allowing a pickup possibility there once the actual demands are revealed. Eqs. (4) force the visit to node $j$ with a demand $q_j > 0$. Eqs. (5) ensure that a vehicle can fulfill the request of node $j$ if and only if $j$ is visited. Eqs. (6)–(9) set the flow of goods to deliver and pickup. In particular, Eqs. (7) manage the possibility to fulfill or not optional demands. Eqs. (10) ensure that each vehicle capacity $C$ is not exceeded. Finally, Eq. (11) guarantees that at least $\alpha$ percent of total optional demand is fulfilled.

The objective function (1) minimizes the total travel cost including the expected cost of a second-stage problem, which depends on the uncertainty. Let us define the following variables: $r_i \in \{0,1\}$ taking value 1 if a customer $i \in M$ is *critical*, i.e., if it would be necessary to collect its optional pickup in order to achieve $\alpha$; $\tilde{x}_{ij}(\xi) \in \{0,1\}$ taking value 1 if no vehicles travel from node $i$ to node $j$ (second-stage route); $\tilde{z}_i(\xi) \in \{0,1\}$ taking value 1 if a vehicle fulfills optional demand of customer $i \in M$, 0 otherwise; $\tilde{P}_{ij}(\xi) \in \mathbb{R}$ measuring the variation of pickup goods carried on arc $(i,j) \in A$; $\lambda(\xi) \in \mathbb{R}$ measuring the optional demand not satisfied. Let $\kappa_i$ be the unitary spot market cost for serving node $i$, and $\chi_{ij}$ be the detour cost from a node $i$. Then, the second-stage problem is:

$$\delta(\mathbf{x},\mathbf{y},\mathbf{D},\mathbf{P},\xi) = \min_{\mathbf{r},\tilde{\mathbf{x}},\tilde{\mathbf{P}},\tilde{\mathbf{z}},\lambda} \sum_{i \in N_P} \kappa_i r_i(\xi) + \sum_{(i,j)\in A} \chi_{ij}(x_{ij} - \tilde{x}_{ij}(\xi)) \tag{12}$$

subject to

$$\tilde{x}_{ij}(\xi) \leq x_{ij} \quad i \in N, j \in N \tag{13}$$

$$\sum_{k \in V} \tilde{x}_{ik}(\xi) = y_i \quad i \in V \tag{14}$$

$$\sum_{i \in V} \tilde{x}_{ij}(\xi) = \sum_{i \in V} \tilde{x}_{ji}(\xi) \quad j \in N \tag{15}$$

$$\tilde{z}_j(\xi) \leq y_j \quad j \in N_P \tag{16}$$

$$\sum_{i \in V}(P_{ij} + \tilde{P}_{ij}(\xi)) + q_j + (\bar{o}_j + \tilde{o}_j(\xi))\tilde{z}_j = \sum_{i \in V}(P_{ij} + \tilde{P}_{ji}(\xi)) \quad j \in N_P \tag{17}$$

$$\sum_{i \in V}(P_{ij} + \tilde{P}_{ij}(\xi)) = \sum_{i \in V}(P_{ij} + \tilde{P}_{ji}(\xi)) \quad j \in N_D \tag{18}$$

$$\sum_{i \in N}(P_{0i} + \tilde{P}_{0i}(\xi)) = 0 \tag{19}$$

$$\sum_{k \in V} D_{ik} + (P_{i0} + \tilde{P}_{i0}(\xi)) \leq C \quad i \in V \tag{20}$$

$$D_{ij} + (P_{ij} + \tilde{P}_{ij}(\xi)) \leq C \quad i \in V, j \in N \tag{21}$$

$$P_{ij} + \tilde{P}_{ij}(\xi) \leq C\tilde{x}_{ij}(\xi) \quad i \in N, j \in N \tag{22}$$

$$\tilde{P}_{i0}(\xi) \leq C\tilde{x}_{i0}(\xi) \quad i \in N \tag{23}$$

$$\tilde{P}_{ij}(\xi) \geq -P_{ij} \quad i \in V, j \in V \tag{24}$$

$$\sum_{i \in N_P} (\bar{o}_i + \tilde{o}_i(\xi))\tilde{z}_i(\xi) + \lambda(\xi) \geq \alpha \sum_{i \in N_P} (\bar{o}_i + \tilde{o}_i(\xi)) \quad i \in V \tag{25}$$

$$\sum_{i \in N_P} (\bar{o}_i + \tilde{o}_i) r_i(\xi) \geq \lambda(\xi) \tag{26}$$

$$r_i(\xi) \leq 1 - \tilde{z}_i(\xi) \quad i \in N_P \tag{27}$$

The objective function (12) minimizes the total *spot market* cost due to critical nodes plus the total cost of *detour arcs*, i.e., the new arcs between the depot and the subset of nodes visited in the first-stage solution. Eqs. (13) allow to no longer travel a first stage arc $(i, j) \in A$. Eqs. (14)–(15) guarantee that new routes are feasible and force to visit only the customers already visited in the first-stage routes. Eqs. (16) allow satisfying optional demand only of visited customers. Eqs. (17)–(19) correctly set the flow of goods to pick up in new routes, even on detour arcs. Eqs. (20)–(23) impose the upper bound $C$ to the exploit capacity of each vehicle. Eqs. (24) ensure that the pickup flow does not become negative. Eqs. (25) set the value of $\lambda(\xi)$ equal to the surplus between the total optional demand fulfilled and the minimum percentage required. Finally, Eqs. (26)–(27) ensure that a node $i$ is considered critical only if its optional demand is not fulfilled and is necessary to reach $\alpha$.

## 3 Methodological Approach

We approximate the proposed model through a scenario-based deterministic equivalent problem, where the optional pickup demand oscillation is discretized by 50 scenarios from different Normal probability distributions (namely, with mean $\mu = \{\frac{1}{15}C, \frac{1}{10}C, \frac{2}{15}C\}$, and standard deviation $\sigma = \mu/2$). We optimally solve 440 instances, i.e., 10 repetitions of each combination of 11 different numbers of customers (from 5 to 15) and $\alpha = \{0.5, 0.6, 0.7, 0.8\}$, using Gurobi with a timelimit set to 4h. The results show that the average Value of Stochastic Solution (i.e., the value of solving our SP model instead of using expected values) varies from 3.8 to 8.2%, and increases as $\alpha$ increases. Furthermore, the average contribution of *detour* operations to the total cost of recourse strategies (if any) is remarkable, namely, 76%, 86%, and 90% for the three distributions, respectively.

## References


[1] G. Nagy, N.A. Wassan, M.G. Speranza, C. Archetti, "The vehicle routing problem with divisible deliveries and pickups", *Transportation Science* 49(2), 271-294, 2015.

[2] M. Salavati-Khoshghalb, M. Gendreau, O. Jabali, W. Rei, "A Rule-Based Recourse for the Vehicle Routing Problem with Stochastic Demands", *Transportation Science* 53(5), 1334-1353, 2019.

# Tactical helicopter offshore personnel transportation planning

**Irina Gribkovskaia**

Department of Logistics, Molde University College -

Specialized University in Logistics, Molde, Norway

Email: irina.gribkovskaia@himolde.no

**Gaute Øiestad Slettemark**

Department of Logistics, Molde University College -

Specialized University in Logistics, Molde, Norway,

Logistics and Emergency Response Department (EPN JOS LE), Equinor ASA,

Email: gaos@equinor.com

## 1 Introduction

In offshore energy logistics, contracted helicopters transport personnel to and from offshore installations. Reliable, cost-efficient, and on-time personnel transportation is vitally important to maintain planned activities at installations. The objective of our research is on the development of optimization decision support tools for helicopter planning at the tactical level from a single heliport to a group of offshore installations for a stable transport demand period.

The tactical helicopter planning in offshore personnel transportation focuses on the selection of helicopter resources for a period of stable demand and the construction of a recurring weekly flight table where flights are assigned to the days of the week and the daily time slots to ensure coverage of installations weekly flight demands while minimizing costs. This is the first study that formalizes the tactical helicopter offshore personnel transportation planning problem from a single heliport on the Equinor example, where selection of optimal helicopter resources is integrated with weekly flight table planning.

## 2 Literature

The majority of research on helicopter offshore personnel transportation planning in the last decade is related to the daily helicopter transportation planning problems encountered in different offshore energy production areas such as the Brazilian basins ([1], [2], [3], [4]), the Norwegian Continental Shelf ([5], [6]), the Gulf of Mexico ([7]), and the Persian Gulf ([8]). These works present optimization models and solution algorithms for routing of daily flights, assignment of passengers to flights, and flight rescheduling caused by uncertainty factors. There are two papers [9] and [10] on offshore helicopter fleet size and mix planning at the

Santos basin in Brazil. The first is dedicated to logistics network planning for the 20-years horizon, mapping new airfield needs and finding best locations, and pre-sizing fleets for long-term hiring. The other work [10] focuses on finding trajectories and distances travelled between heliports and installations, and allocation of installations and helicopters of different sizes to airports to satisfy seats demand for weekly departures. However, this study assumes a given fleet, predefined helicopter daily operating windows, and does not provide tools for construction of weekly flight tables.

## 3 Methodology

We introduce the novel approach adopted by Equinor on the Norwegian Continental Shelf (NCS) to tactical helicopter resource planning where selection of optimal helicopter fleet to be contracted for a period of several weeks or months and weekly flight table construction are decided together with the assignment of operating window for each helicopter for efficient matching with the installations weekly flight demand. The advantages of the formulated tactical planning flight based ILP model are in integrating decisions on the selection of the helicopter fleet, the assignment of operating windows to helicopters, the assignment of flights to the days of the week, and to the daily time slots. The modelled flight scheduling policies include no-gap stacking of flights to the start of the helicopter operating windows, even spread of each helicopter's flights and spread of flights to each installation throughout the week, while respecting the installations' opening windows.

We also develop an iterative algorithm based on the flight-based model decomposition, where at the first stage the decisions on helicopter resources and flights for each day are made and at the second stage the flights are scheduled separately for each day with respect to the installations' opening windows. The decomposition-based method guarantees optimal solutions for the real-size problem instances with a limited number of available helicopter operating window options, exploiting the restrictive enumeration procedure.

## 4 Experiments

Preliminary tests on the Equinor examples illustrate that the developed model realistically represents the goals and the planning requirements of the integrated helicopter resource selection and weekly flight planning. The experiments conducted on the real-like instances of various sizes show the advantages of the decomposition-based method compared with the flight-based model, both in the solution quality and computation time, enabling to solve larger real-life problems.

The developed flight-based model and the decomposition-based algorithm provide a major improvement for the planners of energy companies operating on the NCS that today have no professional decision support tools for tactical helicopter planning and are important for every offshore energy operator that requires helicopter crew change services. The problem formulation and the decision support tools may be expanded by adding uncertainties in flight demands and in weather conditions.

## References


[1] Bastos, Y. B., Fleck, J. L., & Martinelli, R. (2020). A stochastic programming approach for offshore flight scheduling. *IFAC-PapersOnLine*, 53(4), 478-484.

[2] De La Vega, J., Santana, M., Morabito., R., Pureza, V., Bastos, Y., & Ribas., P. (2021). Model-based solution approaches for short-term flight rescheduling problem in aerial passenger transportation to maritime units. *International Transactions in Operational Research*, 29(6), 3400-3434.

[3] De La Vega, J., Vieira, T., Santana, M., Pureza, V., Morabito, R., Tavares, R., Bastos, Y., & Ribas, P. C. (2022). Helicopter recovery in an oil and gas industry: Model and solution approaches. *EURO Journal on Transportation and Logistics*, 11, 100084.

[4] Vieira, T., De La Vega, J., Tavares, R., Munari, P., Morabito, R., Bastos, Y., & Ribas, P. C. (2021). Exact and heuristic approaches to reschedule helicopter flights for personnel transportation in the oil industry. T*ransportation Research Part E: Logistics and Transportation Review*, 151, 102322.

[5] Gribkovskaia, I., Kovalev, M., Halskau Ø. (2015). Minimizing takeoff and landing risk in helicopter pickup and delivery operations, *Omega*, 55, 73-80.

[6] Nafstad, G. M., Haugseth, A., Høyland, V., & Stålhane, M. (2021). An exact solution method for a rich helicopter flight scheduling problem arising in offshore oil and gas logistics. *Computers & Operations Research*, 128, 105-158.

[7] Navarre, J.T. (2021) Randomized greedy algorithm for helicopter optimization in the energy industry: a practical approach to model development and solution deliverance. *International Journal of Energy Sector Management*, 16(1), 149-170.

[8] Abbasi-Pooya, A., & Kashan, A.H. (2017). New mathematical models and a hybrid grouping evelution strategy algorithm for optimal helicopter routing and crew pickup and delivery. *Computers & Industrial Engineering*, 112, 35-56.

[9] Hermeto, N. D. S. S., Ferreira Filho, V. J. M., & Bahiense, L. (2014). Logistics network planning for offshore air transport of oil rig crews. *Computers & Industrial Engineering*, 75, 41-54.

[10] Pereira, H. L., Ferreira Filho, V. J. M., & Vianna, I. G. (2023). Optimal allocation of fleet of helicopters and airports to transport passengers in offshore maritime units. *Pesquisa Operacional*, 43, 1-32.

# The Fulfillment Regionalization Problem

**Nidhima Grover, Alejandro Toriello**
H. Milton Stewart School of Industrial and Systems Engineering,
Georgia Institute of Technology, Atlanta, U.S.A
Email: ngrover9@gatech.edu

**Xiaoyan Si**
Amazon, Bellevue, U.S.A.

## 1 Introduction

In today's retail industry, sales to customers occur through a spectrum of channels, such as traditional customer walk-in to store, buy-online-pickup-in-store [1], buy-online-deliver-to-customer, walk-in-to-store-get-product-delivered. In many of these formats, the retailer can choose which inventory location or fulfillment center (FC) to fulfill items from, bringing opportunities of inventory pooling and providing a much broader product selection [2]. The fulfillment decision, although complex, can bring financial gains when optimized for resources and operating costs. With the unprecedented growth of the retail industry, companies now have the opportunity to strategically divide their fulfillment networks into regional networks. Such a method, called regionalization, simplifies the fulfillment decision, is scalable, and has recently provided retailers with significant gains [3].

When inventory is distributed across multiple FCs, retailers want to make forward looking decisions to fulfill orders from nearby FCs to ship efficiently at low cost [4]. The complexity of these decisions comes from inventory imbalance due to uncertainty in demand of each SKU, supply perturbations, and demand rerouting due to limited capacity. However, methods used by companies for optimizing the location from where to fulfill each order often result in myopic solutions. Orders are fulfilled from nearby FCs which results in capacity limitations or inventory getting depleted for future orders. Future orders are then assigned to far-away FCs leading to high shipping costs and slower speeds.

Regionalization is one strategy to overcome the shortcomings mentioned above. We partition the network such that each partition has sufficient fulfillment capacity to satisfy its demand. For an order originating in the region, we prioritize its fulfillment from a set of FCs assigned to the region. This fulfillment decision depends on availability of inventory, speed, and shipping cost. Regionalization saves inventory and fulfillment capacity of the FCs for the orders that arise within the region, while simplifying the network.

Traditional districting problems in literature involve balancing demand across districts while imposing contiguity [5]. To the best of our knowledge, such problems have not dealt with demand and supply balance in each district, nor with the objective of minimizing the distance between demand and supply nodes. In this study, we propose methodologies to define the regions, as well as balancing FC capacities with the demand of the regions.

## 2 Problem Statement

Let *tiles* denote the smallest geographic areas that we are clustering into regions, which can be 3 or 5-digit zip codes, for example. The number of tiles is $n$ and number of regions is $m$. Let $l$ denote the number of FCs. Let $x_{ij} = 1$ if tile $i \in [n]$ is assigned to region $j \in [m]$, 0 otherwise. Let $y_{kj} = 1$ if FC $k \in [l]$ is assigned to region $j \in [m]$, 0 otherwise. Let parameters $q_i$ denote the average daily demand of tile $i \in [n]$, $C_k$ denote the average daily ship capacity of FC $k \in [l]$, $d_{ki}$ denote the miles by road from FC $k$ to tile $i$, $Q_{\min}, Q_{\max}$ denote the minimum and maximum demand of a region respectively, and $P_{ij}$ denote the penultimate tile in the shortest path from centroid of region $j$ to tile $i$.

$$\min \sum_j \sum_{i,k} q_i * d_{ki} * x_{ij} * \left( \frac{C_k y_{kj}}{\sum_k C_k y_{kj}} \right) \tag{1}$$

$$\text{s.t.,} \sum_j x_{ij} = 1 \quad \forall i \in [n] \tag{2}$$

$$\sum_j y_{kj} = 1 \quad \forall k \in [l] \tag{3}$$

$$\sum_i q_i x_{ij} \leq \sum_k C_k y_{kj} \quad \forall j \in [m] \tag{4}$$

$$Q_{\min} \leq \sum_i q_i x_{ij} \leq Q_{\max} \quad \forall j \in [m] \tag{5}$$

$$x_{ij} \leq x_{P_{ij}j} \quad \forall i \in [n], j \in [m] \tag{6}$$

$$x_{ij}, y_{jk} \in \{0, 1\} \quad \forall i \in [n], j \in [m], k \in [l] \tag{7}$$

Constraints 2 and 3 respectively ensure that each tile and FC are assigned to exactly one region. Constraint 4 ensures that every region has sufficient fulfillment capacity to serve the demand of the region. We want to ensure a minimum level of storage capacity for risk pooling to capture variability in demand, for which we currently impose a lower bound on the demand of a region as a proxy (constraint 5). We also impose an upper bound on the demand of a region to ensure that each FC assigned to a region ships a high volume to the last mile delivery nodes in the region, thus leading to fewer vehicles and lower cost. Contiguity and compactness of the regions ensure ease of operations and administration, and we impose contiguity in constraint 6 based on shortest paths, ensuring that if a tile is assigned to a region, then all tiles lying on the shortest path from the tile to the centroid are also assigned to the same region. The objective 1 assumes that the demand of a tile is fulfilled from each regional FC in proportion to its capacity, and minimizes the transit distance by road from each regional FC to the tile, weighted by this demand. Note that the objective as defined above is non-linear hence, we use the methodology described below.

## 3 Preliminary Results

We have used the following methodology consisting of three integer programs solved sequentially to obtain a feasible solution. We first generate centroids with the Hess formulation [6] to obtain centroids for the regions. This model does not impose contiguity explicitly but uses a moment-of-inertia objective to ensure compactness. Then, we use an IP to assign tiles and FCs to regions such that we minimize the sum of demand weighted transit distance from region centroid to tile and the capacity weighted transit distance from FC to region centroid: $\sum_{i,j} q_i d_{ij} x_{ij} + \sum_{k,j} C_k d_{kj} y_{kj}$ over constraints 2-7, where $d_{ij}$ is the transit distance by road from centroid of region $j$ to tile $i$, and $d_{kj}$ is the transit distance by road from FC $k$ to centroid of region $j$. Fixing the FC to region assignments $y_{kj}$, we then use the model defined by equations 1-7.

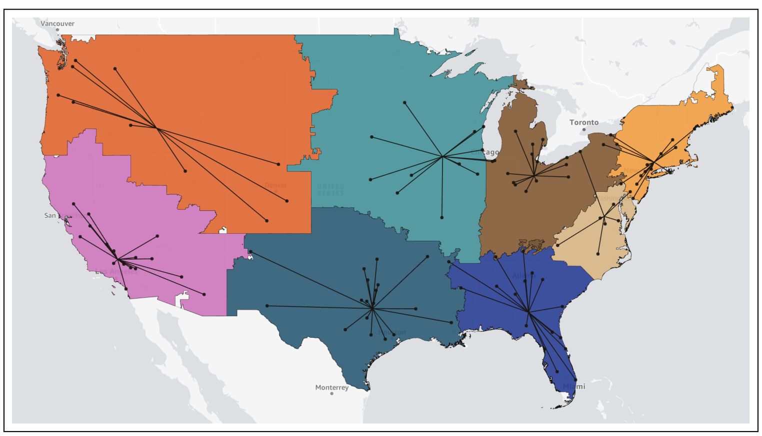



Figure 1: 8-region design from model with 8% and 17% as lower and upper bounds on regional demand. Black dots indicate FCs, and arcs denote the mapping of FCs to regions. The map is representative and does not show actual locations of FCs or region boundaries.

To evaluate a solution, our industry partner also measures inventory availability, and percentage of orders that can be delivered within one day. We partition the network into 8 regions for this experiment. Figure 1 illustrates the solution obtained using 8% and 17% as lower and upper bounds on regional demand. Table 1 illustrates the impact of the bounds on the evaluation metrics. We observe that as we tighten the bounds on demand from 5-20% to 11.5-13.5%, the objective value increases by 9%, implying higher cost. The difference between highest and lowest inventory availability across all regions reduces by 10 absolute percentage, indicating more fairness in inventory availability across regions.

| Bounds on regional demand | 11.5-13.5% | 10-15% | 8-17% | 5-20% |
|---|---|---|---|---|
| Objective function value | x | ↓ 1.4% | ↓ 4.49% | ↓ 8.34% |
| % of orders with 1-day speed | x | ↑ 0.24% | ↑ 1.96% | ↓ 1.34% |
| Average demand-weighted inventory availability (%) | x | ↓ 0.02 | ↑ 0.02 | ↓ 0.26 |
| Highest inventory availability (%) out of all regions | x | x | x | ↑ 0.57 |
| Lowest inventory availability (%) out of all regions | x | ↓ 0.92 | ↓ 3.68 | ↓ 9.56 |

Table 1: Comparison of overall metrics as we vary the bounds on regional demand. The numbers shown are relative to the baseline represented as 'x'.

## 4 Ongoing Research

Some of the open research questions that we are currently addressing are the following: what is a good lower bound model for the problem; what conditions on the clustering of FCs can ensure that the region boundaries would be contiguous without imposing contiguity constraints; and what is the performance of different algorithms for the problem.

## References


[1] K. Saha and S. Bhattacharya, "'buy online and pick up in-store': Implications for the store inventory," *European Journal of Operational Research*, vol. 294, no. 3, pp. 906–921, 2021.

[2] A. J. Schmitt, S. A. Sun, L. V. Snyder, and Z.-J. M. Shen, "Centralization versus decentralization: Risk pooling, risk diversification, and supply chain disruptions," *Omega*, vol. 52, pp. 201–212, 2015.

[3] S. O'Neill, "Sizing down to scale up: How Amazon reworked its fulfillment network to meet customer demand." `https://www.amazon.science/news-and-features/how-amazon-reworked-its-fulfillment-network-to-meet-customer-demand`, 2023.

[4] J. Acimovic and V. F. Farias, "The fulfillment-optimization problem," in *Operations Research & Management Science in the age of analytics*, pp. 218–237, INFORMS, 2019.

[5] A. Buchanan, "Political Districting." `https://optimization-online.org/wp-content/uploads/2023/06/political_districting.pdf`, 2023.

[6] S. W. Hess, J. Weaver, H. Siegfeldt, J. Whelan, and P. Zitlau, "Nonpartisan political redistricting by computer," *Operations Research*, vol. 13, no. 6, pp. 998–1006, 1965.

# A Unified Branch-Price-and-Cut Algorithm for Multi-Compartment Pickup and Delivery Problems

**Timo Gschwind**
Chair of Logistics, Faculty of Business Studies and Economics,
RPTU Kaiserslautern-Landau, Kaiserslautern, Germany
Email: gschwind@rptu.de

**Marjolein Aerts-Veenstra**
Department of Operations, Faculty of Economics and Business,
University of Groningen, Groningen, The Netherlands

**Marilène Cherkesly**
Département d'analytique, opérations et technologies de l'information,
Université du Québec à Montréal and GERAD, Montréal, Canada

## 1 Introduction

In this talk, we introduce, model, and solve the pickup and delivery problem with time windows and multiple compartments (PDPTWMC), which generalizes the well-studied pickup and delivery problem with time windows (PDPTW) to vehicles with multiple compartments. The PDPTWMC is closely related to the pickup and delivery problem with multiple stacks ([1]) and to vehicle routing problems with multiple compartments (VRPMCs, [2]). In the former, stacks can be seen as a special case of compartments that have to be operated using specific loading rules which allows for less flexibility. For the latter, the recent survey [2] highlights three compartment-related attributes that have been considered in the literature.

We adapt these attributes to the pickup-and-delivery context and study variants of the PDPTWMC according to them. Namely, we consider *compartment capacity flexibility*, *item-to-compartment flexibility*, and *item-to-item compatibility*. Furthermore, we develop a unified branch-price-and-cut (BPC) algorithm that can tackle all combinations of these attributes. The core of the approach is a newly derived bidirectional labeling algorithm to solve the pricing problems. To the best of our knowledge, this is the first time that the PDPTWMC is studied. It is also the first time that a unified algorithm is proposed to tackle three important compartment-related attributes. Finally, we conduct extensive computational experiments to understand the impact of considering (or not) some compartment-related attributes on the performance of the algorithm and to derive related managerial insights.

## 2 Problem Description

In the PDPTWMC, we are given a set of customer requests that have to be serviced by an unlimited fleet of homogeneous vehicles located at a common depot. To complete a request, a vehicle must transport an item with given demand from a pickup location to its corresponding delivery location. Each pickup and delivery location has a a time window during which the service must start. Each item has known characteristics (e.g., frozen or ambient good), which can be compatible or incompatible with other items or vehicle compartments. The set of items is $P$. All vehicles have an overall capacity and several compartments. The set of compartments is denoted $M$. Furthermore, compartments have characteristics (e.g., frozen compartment), which can be compatible or incompatible with some items (e.g., frozen goods can only be loaded in frozen compartments).

We study variants of the PDPTWMC according to the three considered compartment-related attributes. *Compartment capacity flexibility* allows the capacities (i.e., the sizes) of the compartments to be flexible within given minimum and maximum capacities. The capacities remain the same throughout the routes. The *item-to-compartment flexibility* specifies which items $i \in P$ are compatible ($b_{im} = 1$) or incompatible ($b_{im=0}$) with which compartments $m \in M$. For the *item-to-item compatibility*, items $i \in P$ can be compatible ($u_{ij} = 1$) or incompatible ($u_{ij} = 0$) with other items $j \in P$, and incompatible items cannot be simultaneously in the same compartment of the vehicle.

The PDPTWMC consists of determining a set of routes with minimum cost such that all requests are completed exactly once and all routes are feasible. A route is feasible if it satisfies the following constraints: 1) pairing and precedence for customer requests, 2) time windows, 3) maximum vehicle capacity, 4) minimum and maximum compartment capacities, 5) item-to-compartment compatibility, and 6) item-to-item compatibility.

## 3 Branch-Price-and-Cut Algorithm

We formulate the PDPTWMC as a standard set-partitioning model, where variables correspond with feasible routes and that we solve with a BPC algorithm. As typical for vehicle routing problems, the most time-consuming part of the BPC is the solution of the pricing problem, which corresponds to an elementary shortest path with resource constraints (ESPPRC) in the underlying network. To solve the ESPPRC pricing problems of the PDPTWMC, we propose an ad-hoc unified labeling algorithm that is able to handle different settings for the three compartment-related attributes. The algorithm builds upon [3] which has demonstrated how to effectively implement bidirectional labeling for pickup and delivery problems using different cost matrices in the forward and backward labelings.

**Pricing Problem** The main components of our forward labeling are as follows:

*Resources.* For each label $F$ and in addition to the standard resources for i) the last visited vertex of the partial path $\eta(F)$, ii) the earliest feasible start of service $t(F)$, iii) the accumulated reduced cost $c(F)$, and iv) the set of completed requests $S(F)$, our algorithm keeps track of v) the current item-to-compartment assignment in the vehicle $O^m(F)$ (i.e., the open requests) and vi) the required capacity of each compartment $\psi^m(F)$.

*Propagation.* The propagation of label $F$ to a vertex $j$ may result in multiple extensions if $j$ is a pickup location and a single extension otherwise. Each extension is characterized by the compartment it relates to, i.e., the compartment the corresponding item is loaded on or unloaded from. We define the set of potential extensions respecting pairing and precedence, item-to-compartment flexibility, and item-to-item compatibility as

$$\mathcal{H}^F(j) = \begin{cases} \left\{m \in M | j \notin S(F) \cup \bigcup_{s \in M} O^s(F), b_{jm} = 1, u_{ij} = 1 \ \forall i \in O^m(F)\right\} & \text{if } j \text{ is pickup,} \\ \{m \in M | j^+ \in O^m(F)\} & \text{if } j \text{ is delivery,} \\ \{m_1 \in M | O^m(F) = \emptyset \ \forall m \in M\} & \text{if } j \text{ is depot.} \end{cases}$$

Here, $j^+$ denotes the request corresponding to delivery location $j$ and $m_1$ denotes the first compartment. For each label and potential extension, a new label is created according to the standard resource extension functions (REFs) from the PDPTW and newly derived REFs for the PDPTWMC-specific resources.

*Dominance.* We show that a label $F_1$ dominates another label $F_2$, if

$$\begin{gathered} \eta(F_1) = \eta(F_2) \qquad t(F_1) \leq t(F_2) \qquad c(F_1) \leq c(F_2) \qquad S(F_1) \subseteq S(F_2) \\ O^m(F_1) \subseteq O^m(F_2), \ \forall m \in M \qquad \psi^m(F_1) \leq \psi^m(F_2), \ \forall m \in M. \end{gathered} \tag{1}$$

Our backward labeling algorithm is analog to the forward case, with the roles of pickup and delivery swapped in the extensions. The bidirectional labeling algorithm then performs both forward and backward labeling up to a half-way point and on different reduced-cost matrices. A tailored merge procedure ensures PDPTWC-feasibility of the created routes and determines their correct reduced costs given the different reduced-cost matrices.

We accelerate the solution of the pricing problems using well-established techniques (e.g., partial pricing or unreachable requests) and the following tailored techniques:

*Symmetry Reduction.* First, we consider label extensions to empty compartments. If several compartments are empty in label $F$, some of the resulting new labels may be identical except for symmetry and we perform a single one of these extensions. Second, we consider dominance with comparable compartments. To strengthen the dominance rule, the strict one-by-one comparison of the compartments in (1) can be relaxed by also checking for symmetric assignments of items to compartments in label $F_2$. Both symmetry-reduction techniques can be applied simultaneously. For bidirectional labeling, the merge has to be adapted accordingly.

*Fixed Compartment Capacities.* If the capacities of the compartments are fixed, the labeling algorithm can be simplified by removing the label components $\psi^m(F)$ and adapting the REFs and the conditions for label feasibility, dominance, and merge feasibility.

**Cutting and Branching** To strengthen the formulation, we use two well-established families of valid inequalities, namely rounded capacity inequalities and subset-row inequalities. The following standard hierarchical branching scheme is applied. We first branch on the number of vehicles. We then branch on the outflow of a subset of vertices of size two.

## 4 Computational Results

Computational experiments are performed on instances adapted from the C2 instance set proposed in [1]. For each of the compartment-related attributes, we examine different scenarios that combine to a total of 90 variants of attribute settings and a total of 28 710 instances. Our BPC with symmetry reduction solves 26 665 instances to optimality; around 3% more optima and an average speedup of more than 40% (and up to factor four for instances with high item-to-compartment flexibility) compared to the variant without symmetry reduction. Overall, we find that the solution quality improves with increasing compartment capacity flexibility (up to 19% less vehicles and 4% less distance traveled on average), increasing item-to-compartment flexibility (up to 22% and 9%), and increasing item-to-item compatibility (up to 26% and 7%). A more detailed analysis provides interesting insights regarding the interaction of the three attributes in different scenarios.

## References

[1] M. Cherkesly, G. Desaulniers, S. Irnich, and G. Laporte, "Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and multiple stacks", *European Journal of Operational Research*, 250(3), 782-793, 2016.

[2] M. Ostermeier, T. Henke, A. Hübner, and G. Wäscher, "Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions", *European Journal of Operational Research*, 292(3), 799-817, 2021.

[3] T. Gschwind, S. Irnich, A.K. Rothenbächer, and C. Tilk, "Bidirectional labeling in column-generation algorithms for pickup-and-delivery problems", *European Journal of Operational Research*, 266(2), 521-530, 2018.

# Assessing the Impact of Driver Overtime in the Transportation of Flowers through a Retail Network

**Mario Guajardo**
Department of Business and Management Science,
NHH Norwegian School of Economics, Norway
Email: mario.guajardo@nhh.no

**Christian Braathen**
Department of Business and Management Science,
NHH Norwegian School of Economics, Norway

## 1 Introduction

While the vehicle routing problem (VRP) has been thoroughly studied, a large share of studies ignore social constraints (Lahyani et al., 2015). These constraints, which relate to the drivers, are important for many reasons, such as their cost implications and their importance in coping with labor regulations. Motivated by an application at the largest wholly-owner flower chain in Norway, this article addresses how social constraints affect the routes and the quality of the solution. We are particularly interested in studying the effect of allowing drivers to work overtime, as this is a usual practice but is commonly not included in the formulation of the routing problem.

## 2 Background

While driver wages are a substantial component of total distribution costs—the most common objective in VRP—drivers receive little focus in VRP studies. Specifically, in the taxonomy of Tan and Yeh (2021), only two components explicitly mention drivers, and only a minority of VRP papers include working hour considerations and legislation protecting the drivers—such as required breaks when driving throughout the day (Lahyani et al., 2015).

Furthermore, the concept of working duration might be ambiguous and subject to specific stipulations from local legislation. Hence, Rincon-Garcia et al. (2020) proposed the following four general definitions not involving regional legislation. First, “route duration” refers to the period a driver and their vehicle spend after departing the depot until returning to the depot after the last service request. Second, “accumulated working time” is the accumulated time during which the driver performs tasks, such as driving, loading, and unloading, and predicted or unpredicted waiting times—for example, due to delays or arrival prior to time windows. Third, “accumulated driving time” describes the accumulated time driven by the vehicle before a required break from driving. Accumulated driving time is reset after each such break. Fourth, “total accumulated driving time” is the driver’s aggregated driving time throughout the working day. Each of these terms may have its own set of constraints in VRP.

Although many of these concepts are acknowledged by the practitioners in charge of routing, it can be challenging to grasp all the details of the routes, the trade-offs between them, and their implications for the total costs. Moreover, the route planners in the flower chain have routing software available, but the ability to incorporate these social constraints is limited. In practice, planners tend to manually tweak the software solutions to cope with these social constraints or manually do the whole routes from scratch. With access to some real-data instances and both the practitioners’ and the software’s solutions, we have studied in detail the impact of social constraints in this VRP, as described below.

## 3 Mathematical Model

To address the problem, we formulate a mixed integer linear programming model characterized by a heterogeneous fleet of capacitated trucks, multi-trips, time windows, deliveries and split pickups, asymmetric distances, real-life speed limits, and driver availability. Due to space limitations, we do not outline the mathematical formulation of the model here. It is important to remark, though, that the decision variables include several aspects about the drivers, such as the amount of overtime that drivers are planned to face during the workday, the amount of time the drivers are planned to work during the workday—driving between locations, delivering containers, and picking up containers—and the amount of time the drivers must rest given the amount of driving during the workday.

Our solution approach consists of two main steps. First, we generate many candidate routes, where we consider most of the problem’s conditions, except for pickups and multi-trips. Secondly, we formulate a route-based model, where the pickups and multi-trips are incorporated. The mathematical programming model’s aim is to allocate drivers and vehicles to predefined candidate routes, decide pickup amounts on each allocated route, and apply the various social constraints.

To explore the effect of social constraints on different metrics, we implement different model variants, including traditional objective functions with and without considering overtime. We find solutions to these different variants by implementing the model in a mathematical programming language and using the solver CPLEX.

## 4 Results

Table 1 summarizes results for seven real-world data instances obtained from the flower chain company. While we could have performed a computational study with many randomly generated instances and outlined average results across instances, our study deliberately focuses on a few real-world instances. By doing so, we can compare our solution with the manual scheduling alternative and the commercial alternative instance by instance and in detail. Furthermore, to understand the impact of social constraints, the test instances are assessed using multiple objective functions, which include minimizing total costs, minimizing the number of drivers used, and minimizing the total distance driven within each delivery day.

As can be seen in Table 1, our solution consistently outperforms manual planning and the commercial (routing software) solution regarding total day costs across all test instances. Compared to the manually produced schedule, our solution outperforms it by 17.4%–36.4%, with more considerable relative cost savings achieved for the smallest test instances. However, nominal cost savings are the greatest for the largest instances. Similarly, our solution outperforms the commercial solution, with total day cost savings being at least 9.7%–25.5%. These cost savings are substantial for total driving and salary costs, indicating that our solution chooses fewer or shorter routes, less expensive vehicles, and less expensive drivers in terms of the average hourly wage rate or total hours worked.

Table 1: Test instance costs

| Instance | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Total day costs (NOK) | | | | | | | |
| Overtime model (our solution) | 41 556 | 40 077 | 30 931 | 31 790 | 41 953 | 66 822 | 80 074 |
| Manual planning | 55 234 | 51 904 | 47 924 | 50 019 | 57 622 | 80 893 | 97 379 |
| Commercial solution | | 44 394 | 38 176 | 37 480 | 56 276 | 75 091 | 93 361 |
| Total driving costs (NOK) | | | | | | | |
| Overtime model (our solution) | 12 223 | 12 656 | 12 433 | 14 050 | 16 188 | 25 529 | 30 156 |
| Manual planning | 19 741 | 19 775 | 19 775 | 19 100 | 23 149 | 29 890 | 31 413 |
| Commercial solution | | 17 742 | 17 742 | 16 093 | 23 751 | 27 680 | 30 724 |
| Total salary costs (NOK) | | | | | | | |
| Overtime model (our solution) | 27 962 | 27 421 | 18 498 | 16 527 | 24 000 | 38 457 | 46 845 |
| Manual planning | 32 334 | 32 128 | 28 147 | 28 547 | 31 152 | 46 633 | 61 157 |
| Commercial solution | | 26 651 | 20 433 | 19 932 | 28 556 | 43 836 | 58 520 |
| Total toll station costs (NOK) | | | | | | | |
| Overtime model (our solution) | 1371 | 0 | 0 | 1212 | 1765 | 2836 | 3073 |
| Manual planning | 3159 | 0 | 0 | 2370 | 3323 | 4372 | 4813 |
| Commercial solution | | 0 | 0 | 1457 | 3970 | 3572 | 4118 |

## 5 Concluding Remarks

Our work has studied the impact of social constraints in a real-world case of a VRP, in which route planners cannot cope in detail with these constraints using routing software and often tend to address these by manually tweaking the software solution. We have proposed an optimization framework to implicitly incorporate these social constraints using a route generation procedure and a mixed integer linear programming model. Our results outperform both the manual and the software solutions, and from a detailed study of all these solutions, we can derive the following insights.

- The cost savings of our solution are caused mainly by assigning fewer routes and reducing the average distance between each location within a route when excluding the driving to/from the warehouse.
- While our solution plans for overtime to create a more efficient solution in all test instances, the overtime usage largely outperforms that of the manual schedule and the commercial solution's usage when there are no practical limits to overtime usage.
- Limiting overtime to the smallest allowable overtime usage to undertake all deliveries leads to longer total working hours and longer total driving distance because the most valuable usage of overtime occurs for locations far away from the headquarters when these relatively remote locations are constrained by working day duration more than vehicle capacity.

- A more flexible policy for overtime usage positively affects the total day costs and the total working hours of the drivers. While overtime fatigues the drivers, all else being equal, this reduced number of working hours ensures that the drivers can work less on other days during the working week because the same amount of goods can be delivered with fewer resources.

Since this article seeks to enhance understanding of social constraints in real-life contexts, thoroughly studying a handful of test instances has been more favorable than studying a vast set of test instances. A natural extension for future work is to conduct a computational study in a more extensive set of instances and develop specialized solution methods for the different combinations of features that might appear in these instances.

## References


Lahyani, R., Khemakhem, M., & Semet, F. (2015). Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research*, *241*(1), 1–14.

Rincon-Garcia, N., Waterson, B., Cherrett, T. J., & Salazar-Arrieta, F. (2020). A metaheuristic for the time-dependent vehicle routing problem considering driving hours regulations–an application in city logistics. *Transportation Research Part A: Policy and Practice*, *137*, 429–446.

Tan, S.-Y., & Yeh, W.-C. (2021). The vehicle routing problem: State-of-the-art classification and review. *Applied Sciences*, *11*(21), 10295.

# An exact algorithm for a new variant of the Team Orienteering Problem

**Alberto Guastalla**, **Roberto Aringhieri**
Computer Science Department, University of Turin, Torino, Italy
Emails: alberto.guastalla@unito.it, roberto.aringhieri@unito.it

## 1 Introduction

The Team Orienteering Problem (TOP) can be described through a complete direct graph $G = (V, A)$ in which each node $k$ has a profit $p_k$ and each arc $(i,\ j)$ has a cost $t_{ij}$. $V$ represents the set of customers and $A$ the set of traversable arcs. The objective is to find a set of $m$ different routes collecting the maximal sum of profits from the visited nodes without exceeding a specific time budget $T_{\max}$ for each of them. Each route must start from the source node and must end to the destination one (1 and $n$, respectively). From the analysis of two real TOP applications (namely [1, 2]) we identified three peculiar but general features: (i) the service time $s_k$ at each node $k$, (ii) a set $M$ of mandatory and optional nodes and (iii) a set $I_p$ of physical incompatibilities between nodes: two nodes are not directly connected. In accordance with the literature, we also consider a set $I_l$ of logical incompatibilities: two nodes cannot be visited by the same route. The Team Orienteering Problem with Service Time and Mandatory and Incompatible Nodes (TOP-ST-MIN) is a variant of the TOP that considers also these three features. In this paper, we report a compact two-index mathematical formulation for the TOP-ST-MIN and describe a new Branch & Cut exact algorithm for its solution.

## 2 Mathematical formulation

Starting from the two-index formulation for the TOP in [3], the TOP-ST-MIN can be modeled as follows.

$$\max \quad \sum_{k=2}^{n-1} p_k\, y_k \tag{1a}$$

$$\text{s.t.} \quad \sum_{j=2}^{n} x_{1j} = \sum_{i=1}^{n-1} x_{in} = m, \tag{1b}$$

$$\sum_{i=1}^{n-1} x_{ik} = \sum_{j=2}^{n} x_{kj} = y_k, \quad \forall\, k \in V, \tag{1c}$$

$$\sum_{j=2}^{n} z_{kj} - \sum_{i=1}^{n-1} z_{ik} = \sum_{j=2}^{n} (t_{kj} + s_k)\, x_{kj}, \quad \forall\, k \in V, \tag{1d}$$

$$z_{ij} \leq (T_{\max} - s_j - t_{jn})\, x_{ij}, \quad \forall\, (i,j) \in A, \tag{1e}$$

$$z_{ij} \geq (t_{1i} + s_i + t_{ij})\, x_{ij}, \quad \forall\, (i,j) \in A, \tag{1f}$$

$$z_{1k} = t_{1k}\, x_{1k}, \quad \forall\, k \in V, \tag{1g}$$

$$y_k = 1, \quad \forall\, k \in M, \tag{1h}$$

$$x_{ij} = 0, \quad \forall\, (i,j) \in I_p, \tag{1i}$$

$$v_k \geq j \cdot x_{1k}, \quad \forall\, k \in V, \tag{1j}$$

$$v_k \leq j \cdot x_{1k} - (n-2)\,(x_{1k} - 1), \quad \forall\, k \in V, \tag{1k}$$

$$v_j \geq v_i + (n-2)\,(x_{ij} - 1), \quad \forall\, i,j \in V, \tag{1l}$$

$$v_j \leq v_i + (n-2)\,(1 - x_{ij}), \quad \forall\, i,j \in V, \tag{1m}$$

$$v_i \geq v_j + 1 - (n-2)\,(1 - u_{ij}), \quad \forall\, (i,j) \in I_l, \tag{1n}$$

$$v_i \leq v_j - 1 + (n-2)\, u_{ij}, \quad \forall\, (i,j) \in I_l, \tag{1o}$$

$$0 \leq x_{ij} \leq 1, \quad \forall\, (i,j) \in A, \tag{1p}$$

$$0 \leq x_{1n} \leq m, \tag{1q}$$

The integer variable $x_{ij}$ is 1 if and only if the arc $(i,j)$ is traversed, 0 otherwise (the empty routes are counted by the variable $x_{1n}$). The binary variable $y_k$ is 1 if and only if the node $k$ is visited, 0 otherwise. The continuous variable $z_{ij}$ defines the arrival time at node $j$ coming from the node $i$ and can be thought as the amount of flow that passes through the arc $(i,j)$. The continuous variable $v_k$ represents the index of the first node in the route that visits node $k$ (the route index). This idea has been introduced by [4]. The binary variable $u_{ij}$ is 1 if and only if $v_i - v_j \geq 1$, 0 otherwise.

The objective function (1a) maximizes the overall score collected by the routes. The constraint (1b) guarantees that each route starts from the source node and ends to the destination one. The (1c) constraints guarantee the connectivity of each route. The (1d) and (1e) are the classical Gavish-Graves (GG) subtours elimination constraints adapted for the TOP. The (1f) set the lower bound on the duration of each route. The (1g) bound the flow originating from the initial depot. The (1h) constraints ensure that all the mandatory nodes will be visited. The (1i) constraints guarantee that all the physical incompatibilities between nodes are satisfied. The (1j) and (1k) constraints guarantee that the route identifier for the node $k$ assumes the same index of the first visited node of that route. The (1l) and (1m) constraints guarantee that the index of the first visited node is forwarded to the next nodes in the route. The (1n) and (1o) constraints guarantee that all the logical incompatibilities are satisfied (we linearized the not equals relation $v_i \neq v_j$).

# 3 A Branch & Cut algorithm

In order to optimally solve the TOP-ST-MIN, we developed a Branch & Cut exact algorithm based on the following four different types of valid inequalities.

**Infeasible set inequalities.** For each infeasible route $r$, we build a directed subgraph $\mathcal{G} = (V(r), A(r))$. $V(r)$ represents the set of nodes composing the route $r$ and $A(r)$ the set of arcs across the nodes in $V(r)$. Thus, we calculate the Held-Karp lower bound for the TSP on $\mathcal{G}$ to find the tightest lower bound (LB) on the cost of a route visiting all the nodes in $V(r)$. Then, if LB is greater than $T_{\max}$, we can state that there is no feasible route visiting all the nodes in $V(r)$. In this case, we can consider this valid inequality:

$$\sum_{i}^{V(r)} \sum_{j \neq i}^{V(r)} x_{ij} \leq |V(r)| - 2 \tag{2}$$

**Infeasible route inequalities.** These inequalities can be seen as a simpler version of the *Infeasible set* inequalities. In fact, given an infeasible route $r$ of cardinality $l$, we are able to forbid it with this valid inequality:

$$\sum_{i=1}^{l-1} x_{\overline{r}_i \overline{r}_{i+1}} \leq \sum_{k=2}^{l-1} y_{\overline{r}_k}, \tag{3}$$

**Subpath inequalities.** We also consider a slightly modified version of the Path inequalities in [5]. These cuts are able to forbid infeasible routes leveraging feasible subpaths. Taking a feasible subpath $p$ of cardinality $l$ (contained in an infeasible route $r$), we can consider two similar types of valid inequalities:

$$\sum_{i=1}^{l-1} x_{p_i p_{i+1}} - \sum_{k=2}^{l-1} y_{p_k} - \sum_{v}^{L(p)} x_{v p_1} \leq 0 \quad \text{and} \quad \sum_{i=1}^{l-1} x_{p_i p_{i+1}} - \sum_{k=2}^{l-1} y_{p_k} - \sum_{v}^{R(p)} x_{p_l v} \leq 0 \tag{4}$$

$L(p)$ and $R(p)$ represent the sets of nodes that is possible to add at the beginning and at the end of the subpath $p$ without making $p$ infeasible. To be clear, this pair of cuts imposes that the subpath $p$ must be left-connected (right-connected) with a node in $L(p)$ ($R(p)$).

**Subtours elimination constraints.** We decided to strengthen the formulation adding a different type of subtours elimination constraints (SECs) during the Branch & Cut computation:

$$\sum_{(i,j) \in U \times U} x_{ij} \leq \sum_{i \in U} y_i - y_j, \quad \forall\, U \subseteq \{2, \ldots, n-1\},\ j \in U. \tag{5}$$

This type of SECs have been adapted from the Dantzig-Fulkerson-Johnson (DFJ) SECs and therefore are exponential in the number of nodes thus, we separate them along the Branch & Bound tree. Every time we get an optimal fractional solution ($\overline{x}$, $\overline{y}$, $\overline{z}$), we check for the presence of violated SECs building an induced graph $\overline{G} = (V, \overline{A})$ in which $V = \{1, \ldots, n\}$ and $\overline{A} = \{(i,j) : \overline{x}_{ij} > 0\}$. We formulated the Lemma 3.1 using the definition below:

**Definition 3.1.** *For any set $U$ associated with no cycles in the induced graph $\overline{G}$, we have that the sum of flow traversing the arcs of the nodes in $U$ must be less than or equals to the minimal sum of flow generated by a set of nodes with cardinality $|U|$ - 1. More precisely:*

$$\sum_{(i,j)\in U\times U} \overline{x}_{ij} \leq \min_{S\subset U:|S|=|U|-1} \sum_{s}^{S} \overline{y}_s$$

**Lemma 3.1.** *Any violated SEC* (5) *can only be identified in those sets associated with cycles in the induced graph $\overline{G}$. A set $U$ is associated with cycles in the induced graph $\overline{G}$ if and only if the arcs involved with the nodes included in $U$ form one or more cycles in $\overline{G}$.*

*Proof.* To prove it, assume that the set $U$ is not associated with cycles in the graph $\overline{G}$. Then, we have that $\sum_{(i,j)\in U\times U} \overline{x}_{ij} \leq \min_{S\subset U:|S|=|U|-1} \sum_{s}^{S} \overline{y}_s$ by the Definition 3.1. Now, we can easily observe that $\min_{S\subset U:|S|=|U|-1} \sum_{s}^{S} \overline{y}_s \leq \sum_{i\in U} \overline{y}_i - \overline{y}_j$ for each $j \in U$. For transitivity, we obtain that $\sum_{(i,j)\in U\times U} \overline{x}_{ij} \leq \sum_{i\in U} \overline{y}_i - \overline{y}_j$ for each $j \in U$ that are exactly the SECs (5) for a fixed set $U$. Thus, no SECs (5) are violated for any chosen set $U$ as assumption. $\square$

The proposed Branch & Cut algorithm uses all the valid inequalities introduced before. Every time we find a relaxed solution, we calculate the entire set $R$ of routes contained in the induce graph $\overline{G}$ with a modified version of the classical Deep First Search (DFS) algorithm. First of all, we calculate the cost of each route $r$ in $R$. If the cost of $r$ is greater than $T_{\max}$, we calculate the Held-Karp lower bound (LB) on the subgraph $\mathcal{G}$. If this LB results to be greater than $T_{\max}$, then, the associated Infeasible set inequality (2) is checked for violation. Otherwise, we check if the corresponding Infeasible route inequality (3) is violated. To separate the Subpath inequalities (4) we first calculate all the subpaths for $r$. For each subpath $p$ we generate the sets $L(p)$ and $R(p)$ and we check the corresponding cut for violation. Finally, we consider all the elementary cycles inside the graph $\overline{G}$ and we check if the associated Subtour elimination constraint (5) is violated.

## References

[1] R. Aringhieri, S. Bigharaz, D. Duma, A. Druetto, A. Grosso, Alberto Guastalla, "The daily swab test collection problem", *Annals of Operations Research* 5(3), 211-230, 2022.

[2] R. Aringhieri, S. Bigharaz, D. Duma, A. Guastalla, "Fairness in ambulance routing for post disaster management", *Central European Journal of Operational Research* 6(2), 315-332, 2021.

[3] N. Bianchessi, R. Mansini, M. G. Speranza, "A branch-and-cut algorithm for the Team Orienteering Problem", *International Transactions in Operational Research* 25(2), 627-635, 2018.

[4] M. G. S. Furtado, P. Munari, R. Morabito, "Pickup and delivery problem with time windows: A new compact two-index formulation.", *Operations Research Letters* 45 (4), 334–341, 2017.

[5] M. Fischetti, J. J. Salazar González, P. Toth, "Solving the Orienteering Problem through Branch-and-Cut", *INFORMS Journal on Computing* 10(2), 133-148, 1998.

# A large neighborhood search for tactical planning in cooperative two-tier city logistics systems

**Johannes Gückel**
Ingolstadt School of Management,
Catholic University Eichstätt-Ingolstadt, Ingolstadt, Germany
Email: Jgueckel@ku.de

**Teodor Gabriel Crainic**
Interuniversity Research Centre on Enterprise Networks,
Logistics and Transportation (CIRRELT)
University of Quebec in Montreal, Canada

**Pirmin Fontaine**
Ingolstadt School of Management,
Catholic University Eichstätt-Ingolstadt, Ingolstadt, Germany

## 1 Introduction

Urbanization is expected to rise significantly, with a projected increase from 55% in 2018 to 68% in 2050, as highlighted in a United Nations (2018) forecast [1]. The resulting increase in transportation volume for freight in urban areas leads to more traffic congestion, noise, pollution, and environmental emissions, presenting substantial challenges for Logistics Service Providers (LSPs) and municipalities.

In this context, a Two-Tier City Logistics (2T-CL) system with City Distribution Centers (CDCs) on the outskirts of the city and satellite depots within the city can potentially reduce monetary and environmental impact costs. In these systems, several LSPs operate simultaneously and use the same infrastructure. We contribute to the literature by extending the cooperation representation in 2T-CL to explicitly include various LSPs who share both their customers and their resources. For this, we propose a mathematical formulation and a solution approach based on a metaheuristic for tactical planning 2T-CL systems with cooperating LSPs.

## 2 Problem description

We consider a 2T-CL system. This physical system consists of CDCs located on the outskirts of the city and satellites, transdock-type facilities, located within the city. The first-tier delivery takes place by urban vehicle services starting on a specific time period at CDCs and delivering freight to a sequence of satellites. Starting from the satellites, the final delivery to the customer's location takes place using smaller, environmentally friendly vehicles called city freighters.

We explicitly consider that each LSP has its own customer demands and resources. This includes the urban vehicles to provide first-tier services, the fleet of city freighters for the second-tier delivery, and capacities at satellites. Further, we consider that each demand has a release date at CDCs and a due date on which it must be delivered to its final location. The cooperation of LSPs takes place by sharing their resources (capacities at satellites, urban vehicles, city freighters) and sharing their demands in the sense that one LSP can satisfy the customer demands of another LSP. To address the issue of LSPs being hesitant to fully share their customer demands or cooperate on just one tier, we consider rules that prevent the full sharing of demands. The overall goal is to minimize the total system costs.

## 3 Mathematical formulation

We formulate the problem as a mixed-integer program over a discrete planning horizon using a service network design formulation on the first tier. Unlike previous publications for tactical planning (e.g., [2]), the second tier is integrated using a vehicle routing problem with release and due dates instead of relying on approximations. Therefore, our mathematical model decides on the following: the services that are selected, the assignment of demands to services and satellites, and the second-tier routing decisions for city freighters.

Our objective function aims to minimize the overall system costs consisting of 1) operating costs for first-tier services, 2) costs associated with assigning demands to services and consequently to the CDC from which the service departs, and 3) vehicle routing costs for second-tier city freighters.

We take the following constraints into account: On the first tier, we consider that each demand must be assigned to exactly one service and satellite. Different capacity constraints regarding the number of urban vehicles and the maximum demand volume per urban vehicle are taken into account. Additionally, we consider the maximum demand volume and maximum number of urban vehicles that can operate at a satellite at the same time. On the second tier, we consider vehicle routing constraints. Each demand must be assigned to a city freighter, the capacity of city freighters must not be exceeded, the correct flow of the city freighters must be ensured, and each demand must arrive at its final location with respect to its due date. For the synchronization of the two tiers, we consider two connection constraints. The first one states that the assignment to a satellite on the first tier must match the departure of the city freighter the demand is assigned to on the second tier. The second one is regarding the timing; the arrival at the satellite must occur before the departure. Further, we limit the demand sharing through the parameters $\alpha_1$ and $\alpha_2$, which indicate lower bounds for the percentage of own demand volume each service provider has to fulfill by its own urban vehicles on the first- ($\alpha_1$) and by own city freighters on the second ($\alpha_2$) tier.

## 4 Solution approach

As both the service network design problem and the vehicle routing problem are NP-hard problems, we propose a novel Integrated Two-step Large Neighborhood Search with problem-specific operators for the service design and for the assignment of demands to services and satellites as well as for the second-tier routing decisions. We start by constructing an initial feasible solution with a greedy construction heuristic for the first and second tier. After we have an initial solution, we iteratively apply a two-step procedure.

In Step 1, we destroy our solution by removing services. As the solution gets infeasible when removing services because previously assigned demands are getting unassigned, we try to reassign the unassigned demands to other services in the existing service design. If not all demands can be assigned to existing services, we insert other services. Thereby, problem-specific operators conduct the selection of services for removal and insertion.

In Step 2, we iteratively remove demands from the solution and reinsert them, given the current service design. Thereby, we use problem-specific operators regarding the demand assignment and routing to remove demands. We reinsert them on their cheapest possible insertion position regarding the assignment to service and satellite and second-tier routing.

After that, we start again with Step 1. Our algorithm terminates after a maximum number of iterations without improvement.

## 5 Numerical experiments

We assess our solution approach by benchmarking our heuristic against the commercial solver Gurobi using various test instances based on a major German city's network. The instances vary in size (number of demands, $|D|$) from 5 to 40, with three instances generated for each demand count. The comparison involves executing five runs of our heuristic and running Gurobi with a one-hour time limit. Table 1 presents the average performance over all three instances per instance size of both Gurobi (columns 2-5) and our heuristic (columns 6-9). Our heuristic attains exact optimal solutions for small instances, on par with Gurobi, and outperforms it by delivering superior solutions in less computing time for larger instances.

| | Gurobi | | | | Heuristic (5 runs) | | | | $\Delta$ |
|---|---|---|---|---|---|---|---|---|---|
| $\lvert D\rvert$ | Obj | Optimal | GAP [%] | Time [s] | Best | Avg. | $\sigma$ [%] | $\overline{Time}[s]$ | [%] |
| 5 | 91.80 | 3/3 | 0 | 1 | 91.80 | 91.80 | 0 | 3 | 0 |
| 10 | 120.60 | 3/3 | 0 | 124 | 120.60 | 120.60 | 0 | 9 | 0 |
| 15 | 165.32 | 0/3 | 13.58 | 3600 | 163.00 | 163.70 | 0.22 | 18 | -0.98 |
| 20 | 206.50 | 0/3 | 21.69 | 3600 | 197.95 | 199.29 | 0.63 | 23 | -3.49 |
| 30 | 314.12 | 0/3 | 31.99 | 3600 | 281.02 | 284.30 | 1.17 | 46 | -9.49 |
| 40 | - | 0/3 | - | 3600 | 354.36 | 357.14 | 0.74 | 87 | - |

Table 1: Average performance benchmark vs. Gurobi

# 6 Conclusion

In the presentation, we will illustrate the problem and our mathematical formulation. Following this, we will provide a detailed discussion of the metaheuristic. We plan to enhance the metaheuristic prior to the presentation by incorporating adaptive components and additional operators. Additionally, we will present more detailed results on the heuristic's performance, along with numerical experiments and managerial insights into the cost-saving potentials of (partial) cooperation and the sharing of total system costs.

# References


[1] United Nations , “World Urbanization Prospects,” 2018, United Nations, Department of Economic and Social Affairs, Population Division, URL: https://population.un.org/wup/publications/

[2] P. Fontaine , T. G. Crainic, O. Jabali, W. Rei, “Scheduled service network design with resource management for two-tier multimodal city logistics“, *European Journal of Operational Research*, 294(2), 558-570, 2021

# Dynamic shipment-to-service matching for interurban transportation systems with multimodal networks


**Wenjing Guo**
School of Transportation and Logistics Engineering,
Wuhan University of Technology, Wuhan, China
Email: wenjingguo@whut.edu.cn

**Teodor Gabriel Crainic**
Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation,
Montreal, Canada

**Michel Gendreau**
Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation,
Montreal, Canada

**Wenfeng Li**
School of Transportation and Logistics Engineering,
Wuhan University of Technology, Wuhan, China

**Walter Rei**
Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation,
Montreal, Canada


Mar 04, 2024

## 1 Introduction

Interurban transportation systems that provide movements of freights over multimodal networks between different cities are becoming more and more popular in response to the trend towards sustainability [1, 2, 3]. Traditionally, most of interurban freight transportation are organized by truck companies independently, which causes high transportation costs, high empty drives, and heavy carbon emissions. With the challenges of global warming, green transportation models, such as high-speed railways and inland waterways, have been increasingly used in long-haul transportation. The integration of network connectivity is revolutionizing interurban transportation systems and providing diverse mobility options, including trains, barges, trucks, metros, buses, drones, and small vans. These advancements promise a more reliable, efficient, seamless, and sustainable experience for freight transportation. In the same time, it makes the transportation planning problem more complex with the consideration of scheduled services and flexible services, line services and shuttle services, contracted services (i.e., the services with known schedules and capacities) and spot services (i.e., the services that are unknown before their announcements), and the consideration of transshipments between different modes.

This paper investigates the operational planning problem of an interurban transportation system in which an intelligent decision support platform (IDSP) aims to provide optimal decisions on the selection of shipment requests received from shippers and multimodal service offers received from carriers, and decisions on shipment-to-service assignments and time schedules for accepted requests and offers. On the one side of the system, many shippers (e.g., producers, wholesalers, and distributors) make shipment requests for cost and time-efficient transportation of their product loads. Each shipment needs to be transported from a given shipper location to a consignee location within given time windows. On the other side, many carriers (e.g., transportation service providers), of diverse modes and types, make service offers for urban and long-haul transportation and request profitable loads. Each service provides a limited transport capacity on a specific route with or without time schedules, served by one or multiple vehicles with the same or different modes. In the middle, the logistics service operator using the IDSP for 'automated' planning and optimizing operations - aims to profitably and simultaneously satisfy the needs of both categories of stakeholders. The IDSP receives requests and offers continuously over time and optimizes in time and space the selection of shipment requests and service offers, shipment-to-service assignments, shipment itineraries, and service schedules through consolidation of shipments of different shippers into the same vehicles and synchronization of activities in an interconnected interurban transportation network. The recent developments in information technologies such as cloud computing and Internet of Things allow real-time monitoring of shipments' and vehicles' status and information sharing among stakeholders, which facilitates the adoption of such a platform in practice.

In the literature, most of the studies investigates either dynamic demand-to-supply matching on road networks [4, 5, 6] or shipment routing problems on multimodal networks [7, 8, 9, 10]. None of the existing studies consider the dynamic shipment-to-service matching on multimodal networks. To the best of our knowledge, this is the first paper that considers the selection of shipment requests and service offers simultaneously in interurban transportation systems with multimodal networks at the operational level. To bridge this gap in the literature, the first contribution of this paper is that we develop a mathematical model that integrates the decisions on the acceptance or rejection of shipment requests and service offers, and decisions on shipment-to-service assignments, shipment itineraries, and time schedules for accepted requests and offers while taking into account the time and capacity limitations on multimodal services and terminals. Besides, we consider that both requests and offers arrive at the platform dynamically. The decisions made at each time are not all to be put into practice. In this paper, we develop a rolling horizon framework to control the implementation and re-optimization of decisions when new requests and offers are received. To produce good-quality solutions rapidly, an adaptive large neighborhood search (ALNS) heuristic algorithm with dynamic path generation is designed to solve the optimization problems at each decision time. Finally, we conduct extensive numerical experiments to evaluate the performance of the rolling horizon approach in comparison to a first-come-first-out approach that does not consider re-optimization, and assess the efficiency of the ALNS heuristic in terms of computation time and solution quality in comparison to CPLEX solver.

## References

[1] R. Giusti, D. Manerba, G. Bruno, and R. Tadei, "Synchromodal logistics: An overview of critical success factors, enabling technologies, and open research issues," *Transportation Research Part E: Logistics and Transportation Review*, vol. 129, pp. 92–110, 2019.

[2] G. Taherkhani, I. C. Bilegan, T. G. Crainic, M. Gendreau, and W. Rei, "Tactical capacity planning in an integrated multi-stakeholder freight transportation system," *Omega*, vol. 110, p. 102628, Jul. 2022.

[3] I. C. Bilegan, T. G. Crainic, and Y. Wang, "Scheduled service network design with revenue management considerations and an intermodal barge transportation illustration," *European Journal of Operational Research*, vol. 300, no. 1, pp. 164–177, Jul. 2022.

[4] R. Tian, C. Wang, Z. Ma, Y. Liu, and S. Gao, "Research on vehicle-cargo matching algorithm based on improved dynamic bayesian network," *Computers & Industrial Engineering*, vol. 168, p. 108039, Jun. 2022.

[5] R. Ausseil, J. A. Pazour, and M. W. Ulmer, "Supplier menus for dynamic matching in peer-to-peer transportation platforms," *Transportation Science*, vol. 56, no. 5, pp. 1304–1326, Sep. 2022.

[6] A. Park, R. Chen, S. Cho, and Y. Zhao, "The determinants of online matching platforms for freight services," *Transportation Research Part E: Logistics and Transportation Review*, vol. 179, p. 103284, Nov. 2023.

[7] W. Guo, B. Atasoy, W. B. van Blokland, and R. R. Negenborn, "Global synchromodal transport with dynamic and stochastic shipment matching," *Transportation Research Part E: Logistics and Transportation Review*, vol. 152, p. 102404, Aug. 2021.

[8] J. P. Müller, R. Elbert, and S. Emde, "Integrating vehicle routing into intermodal service network design with stochastic transit times," *EURO Journal on Transportation and Logistics*, vol. 10, p. 100046, 2021.

[9] Y. Zhang, W. Guo, R. R. Negenborn, and B. Atasoy, "Synchromodal transport planning with flexible services: Mathematical model and heuristic algorithm," *Transportation Research Part C: Emerging Technologies*, vol. 140, p. 103711, Jul. 2022.

[10] R. B. Larsen, W. Guo, and B. Atasoy, "A real-time synchromodal framework with co-planning for routing of containers and vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 157, p. 104412, Dec. 2023.

# Solving practical single- and multi-depot electric bus scheduling problems

**Xenia Haslinger**
Institute of Production and Logistics Management,
Johannes Kepler University Linz, Austria
Email: xenia.haslinger@jku.at

**Elisabeth Gaar**
Department of Mathematics,
University of Augsburg, Germany

**Sophie N. Parragh**
Institute of Production and Logistics Management,
Johannes Kepler University Linz, Austria

## 1 Introduction

In the European Union, around 90% of the fuel used for transportation is accountable for nearly 25% of all greenhouse gas emissions [1]. Climate goals can only be met with the development and use of 'cleaner' ways of transportation. Besides other technology options, the interest of bus operators is especially high in electric alternatives.

Since the deployment of electric buses is increasing, research focuses on the electric vehicle scheduling problem (EVSP) and its variants, where the utilization of electric bus fleets and charging infrastructure are considered. In the EVSP, an extension of the vehicle scheduling problem, a number of scheduled trips are to be assigned to a number of electric buses with limited battery capacity, so that distance and charging restrictions apply. An overview of the existing literature is given by Perumal et al. [6]. In the last years, several researchers have investigated the multi-depot EVSP (MDEVSP), but only a few rely on exact methods for solving it. Liu and Ceder [4] proposed a model based on deficit function theory and a mathematical program for determining schedules for electric fleets with an optimal number of fast chargers and partial charging. Jiang et al. [3] presented an integer programming model and developed a branch-and-price algorithm for large-scale MDEVSP under a partial recharging policy. Likewise, Wu et al. [8] developed a branch-and-price algorithm for the MDEVSP under full recharging with power grid characteristics. Gksiokalitis et al. [2] studied the MDEVSP with time windows under full charging policy.

They formulated a mixed-integer program and introduced valid inequalities to tighten the search space. Most of the proposed models in existing literature rely on 3 indices.

Our work is motivated by a real-world problem. We evaluate 2- and 3-index mixed-integer linear programs for the single-depot EVSP (SDEVSP) and the MDEVSP considering partial recharge, where buses are located at various depots and different electric technology options are available. We develop a branch-and-cut algorithm for the 2-index formulation and compare it to solving the compact models with an off-the-shelf solver directly. Benchmark data enriched with electric vehicle requirements as well as real-world data of rural bus lines in Austria are used for computational evaluation.

## 2 Problem description

In this study, we address the SDEVSP and MDEVSP, where the MDEVSP generalizes the SDEVSP. In the MDEVSP: (1) every service trip is assigned to exactly one bus, (2) every bus is associated with a single depot, (3) each bus starts at its respective depot and returns to it only once, e.g. at the end of its daily schedule, (4) the set of timetabled bus trips start and end at particular locations and times, and (5) are carried out by a homogeneous fleet of electric buses, considering partial recharge. A solution for the SDEVSP or MDEVSP is a set of bus schedules, where each bus starts and ends at its respective depot, each service trip is covered by exactly one bus and the buses do not exceed their battery capacity.

We define a set of service trip nodes $V^I$ as well as a set of charging nodes $V^C$ and a set of depots $V^D$, $V = V^I \cup V^C \cup V^D$. Each bus starts from its respective depot and returns there after its schedule. We determine an energy usage $q_i$ for each service trip $i$ and an energy usage $u_{i,j}$ for traveling between each pair of nodes $i, j$, respectively. We specify the battery capacity $s^{max}$ and ensure that the energy level of a bus does not fall below a minimum state of charge at node $i \in V$ called $s_i^{min}$. We calculate the time period between two trips and, consequently, allow a maximum amount $h_c$ to be charged in a charging node $c \in V^C$. Finally, we refer to the maximum number of buses per depot $k$ as $b_k$. In the objective function, in order to support the decision on bus type, we minimize lexicographically the number of buses to cover all timetabled trips, followed by the number of charging events during a day, and finally the energy spent on trips without passengers, so called deadhead trips.

For our practical bus scheduling problem, we introduce different technology options. For pure battery electric buses (BEB) we consider two options: overnight charging (ONC) and opportunity charging (OPC). In the ONC case, charging is only possible at the depot, mainly over night but if necessary also during the day. In the OPC case, the idea is to increase the state of charge during idle times at bus stops. Therefore, buses using OPC have the ONC recharging options and additionally, they can recharge at bus stops at the end of each of their respective service trips. Another type of emission-free buses are fuel-cell electric buses (FCEB), which can only refuel at their respective refueling station. As hydrogen tanks are filled within relatively short periods of time and provide sufficient energy for the schedule of a day, they perform similarly to diesel buses in the operational process. In all settings, we consider the recharging time to be a linear function of the amount of charged battery.

## 3 Methodology

We propose a commodity flow formulation based on different graph representations for modeling bus schedules that consider the energy demand of different technologies. The base network contains all trips that need to be served. It is used as a basis for the design of technology-specific networks. In order to avoid decision variables concerning the time spent on charging, we incorporate this information into the graph construction directly, such that any solution to our model will automatically be time feasible. Therefore, service trips are only connected with a charging node $c \in V^C$ if there is sufficient time available to travel from service trip $i$ to the charging station $c$ and then to service trip $j$. For the SDEVSP and MDEVSP we present a 3-index mixed-integer linear program. Then we reformulate the 3-index formulations into 2-index formulations and separate constraints of exponential size in a cutting plane fashion.

## 4 Preliminary results

We apply our SDEVSP model to a real-world problem with 12 bus lines in Austria, where the respective lines are subcontracted to different operators. We solve the problem using Julia and CPLEX with a time limit of six hours for each of the three technology options ONC, OPC and FCEB. To test the robustness of the solution and gain managerial insights, several different settings have been evaluated: (1) The base setting, where we assume the tank/battery capacity to be 100%. (2) The alternative parameter setting, where we assume the energy consumption for FCEB increases by 20%. In this case, the number of buses used in the optimal solution does not change. For cold temperature settings (3), we base our assumptions of an increase in energy consumption on [7]. We observe that for almost all technologies and lines, the same number of buses as in the base setting suffices in the cold temperature setting.

We expect our model to perform also well in the multi-depot case. We currently evaluate our 3- and 2-index approaches for the MDEVSP with joint lines on combined instances data from [5] and [2]. We combine them by adding the electric part from [2] to the instances of [5]. The number of service trips varies between 500, 1000 and 1500 trips, and either 4 or 8 depots are used. First results indicate that solving the 2-index formulation generates faster results than solving the 3-index one.

**Acknowledgments** This research has been partially funded by the Climate and Energy Fund (FFG-Project FO999900979).

## References

[1] European Commission, Directorate-General for Mobility and Transport, "EU transport in figures: statistical pocketbook 2023", Publications Office of the European Union, 2023.

[2] K. Gksiokalitis, C. Iliopoulou and K. Kepaptsoglou, "An exact approach for the multi-depot electric bus scheduling problem with time windows", *European Journal of Operational Research* 306, 189-206, 2023.

[3] M. Jiang, Y. Zhang and Y. Zhang, "A Branch-and-Price Algorithm for Large-Scale Multidepot Electric Bus Scheduling", *IEEE Transactions on Intelligent Transportation Systems*, 1-14, 2022.

[4] T. Liu and A. Ceder, "Battery-electric transit vehicle scheduling with optimal number of stationary chargers", *Transportation Research Part C* 114, 118-139, 2020.

[5] A.-S. Pepin, G. Desauniers, A. Hertz and D. Huisman, "A comparison of five heuristics for the multiple depot vehicle scheduling problem", *Journal of Scheduling* 12, 12-30, 2009.

[6] S. S. G. Perumal, R. M. Lusby and J. Larsen, "Electric bus planning & scheduling: A review of related problems and methodologies", *European Journal of Operational Research* 301, 395-413, 2022.

[7] ViriCiti, "ViriCity Report: E-bus performance", https://viriciti.com/wp-content/uploads/2020/07/ViriCiti-E-Bus-Performance-Report-July2020.pdf, last accessed 09.2021, 2020.

[8] W. Wu, Y. Lin, R. Liu and W. Jin, "The multi-depot electric vehicle scheduling problem with power grid characteristics", *Transportation Research Part B* 155, 322-347, 2022.

# An evaluation of common modeling choices for the vehicle routing problem with stochastic demands

**Ymro Hoogendoorn**
Econometric Institute,
Erasmus University Rotterdam, Rotterdam, the Netherlands
Email: y.n.hoogendoorn@ese.eur.nl

**Remy Spliet**
Econometric Institute,
Erasmus University Rotterdam, Rotterdam, the Netherlands
Email: spliet@ese.eur.nl

## 1 Introduction

We investigate commonly made modeling choices for the vehicle routing problem with stochastic demands (VRPSD). The VRPSD is the NP-hard problem of designing routes for vehicles of limited capacity that satisfy stochastic demands of customers, whose realization is only learned upon arrival at each customer. A recourse action must be taken in case the remaining load of a vehicle is insufficient to serve the current customer. A classical recourse action is to return to the depot to restock and return to the customer to resume the originally planned route. Another recourse action, is to decide that a vehicle restocks earlier in order to prevent such a stock-out. The objective is to construct the routes such that the total expected routing costs are minimized.

The following three modeling choices are commonly made in the scientific literature: 1) imposing that the total expected demand of customers on a route may not exceed the capacity of the vehicle, referred to as the expected capacity constraints (ECCs), see e.g. [1, 2, 3, 4, 5, 6, 7], 2) imposing that the number of routes is fixed, referred to as the fixed route constraint (FRC), see e.g. [1, 3, 5, 6, 7, 8, 9], and 3) using a demand distribution with a support that contains negative-valued realizations, see e.g. [1, 3, 7]. We discuss these modeling choices next.

The ECCs were introduced in [1], arguing that "*otherwise some routes will systematically fail while on others vehicles will be highly underutilized*". However, the net effect of these constraints yields structurally more visits of the depot in total, i.e., guaranteed and expected number of stochastic visits due to recourse combined. Similarly, the FRC fixes only the number of guaranteed visits while leaving the number of potential visits due to recourse free. Unless there is a good reason to distinguish between guaranteed and potential depot visits, the FRC can be considered as rather arbitrary. Although imposing unnecessary constraints may lead to an undesirable increase of the optimal objective value, the computation time to find a solution might be positively affected. Finally, in e.g. [1, 3, 7], numerical experiments are presented in which the demand of customers is assumed to follow a normal distribution. The normal distribution is appealing for the computational tractability of evaluating the total expected costs. However, the support of the normal distribution consists of the real numbers, which means that demand has a nonzero probability of being negative. Such a realization cannot occur in most practical applications, and may not have a meaningful interpretation. For these three common modeling choices that may not conform with practical necessity in some application, we investigate their effect on the optimal objective value and computation time of finding an optimal solution with a state-of-the-art algorithm.

## 2 Theoretical results

First, we consider four variants of the VRPSD, resulting from imposing or not imposing the ECCs and the FRC. Both for the case that the ECCs are or are not imposed, we have proven that the increase in the optimal solution value from additionally imposing the FRC is arbitrarily large in the worst-case. For the case that the FRC is imposed, we have proven that additionally imposing the ECCs also results in an arbitrarily large increase of the optimal objective value in the worst-case. For the case that the FRC is not imposed, it is still an open question what the worst-case increase in the optimal objective value is that results from imposing the ECCs. However, we have proven that this worst-case increase is at least by a factor of three.

To provide insight into the effect of allowing demand distributions with negative demand, we compare the objective value obtained by using a distribution that allows negative realizations to approximate a corresponding truncated, censored or folded distribution. These seem natural counterparts of distributions with negative realizations: with a censored distribution negative realizations are interpreted as 0; with a truncated distribution negative realizations are resampled until positive; with a folded distribution negative realizations are seen as positive realizations with equal magnitude. We provide bounds on the difference in objective values. Our bounds become tighter when the probability of negative demand decreases and when route lengths decrease. This implies that using a distribution which allows for negative realizations to approximate a censored, truncated or folded distribution, might not lead to large errors in case of a distribution with sufficient probability mass on positive realizations and when the routes are short.

## 3 Numerical results

To assess the impact of imposing the ECCs and FRC, we have performed numerical experiments in which we use a state-of-the-art exact algorithm from [7] to solve benchmark instances of the VRPSD from [3, 5, 10]. Out of the 316 considered benchmark instances, when the ECCs are imposed, 290 and 286 instances are solved to optimality within the time limit of one hour when the FRC is additionally imposed or not, respectively. When the ECCs are not imposed, only 68 and 44 instances are solved to optimality when the FRC is additionally imposed or not. We conclude that imposing the ECCs provides great computational advantages. To a lesser extent, imposing the FRC also offers some computational advantages, particularly when the ECCs are not imposed. For most instances, the optimal objective value does not deviate much depending on whether the ECCs and FRC are imposed. In fact it is oftentimes the same. However, there are some outlier instances with relatively large increases of the optimal objective value. The largest increase that we observed, is an increase of more than 12% from imposing the ECCs. Because the computational advantage of imposing the ECCs are so pronounced, we suggest that there is a case to be made for imposing the ECCs even when not necessitated by practice. In this case, care has to be taken because we have observed instances in which the optimal objective value deteriorates substantially, although in most instances we have not seen a substantial impact. We do not suggest imposing the FRC if there is no practical necessity.

To assess the impact of using a demand distribution that allows for negative realizations, we evaluate optimal solutions found for the instances of [3], in which a normal distribution is used to model demand. We have numerically evaluated our before mentioned bounds on the difference in optimal objective value, but found evidence that these bounds might not be very tight for these instances. We have also compared the objective value to that of using a censored, truncated and folded normal distribution. The largest observed difference in objective value is 0.01% for both the censored and folded case, and 0.08% for the truncated case. These differences are quite small and may indicate that for this set of instances it is appropriate to use a normal distribution to approximate a censored, truncated or folded normal distribution.

## References

[1] G. Laporte, F. Louveaux, and L. Van hamme, "An integer $L$-shaped algorithm for the capacitated vehicle routing problem with stochastic demands", *Operations Research* 50(3), 415-423, 2002.

[2] C. Christiansen, and J. Lysgaard, "A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands", *Operations Research Letters* 35(6),

773-781, 2007.

[3] O. Jabali, W. Rei, M. Gendreau, and G. Laporte, "Partial-route inequalities for the multi-vehicle routing problem with stochastic demands", *Discrete Applied Mathematics* 177, 121-136, 2014.

[4] C. Gauvin, G. Desaulniers, and M. Gendreau, "A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands", *Computers & Operations Research* 50, 141-153, 2014.

[5] F. Louveaux, and J. Salazar-González, "Exact approach for the vehicle routing problem with stochastic demands and preventive returns", *Transportation Science* 52(6), 1463-1478, 2018.

[6] M. Salavati-Khoshghalb, M. Gendreau, O. Jabali, and W. Rei, "An exact algorithm to solve the vehicle routing problem with stochastic demands under an optimal restocking policy", *European Journal of Operational Research* 273(1), 175-189, 2019.

[7] Y. Hoogendoorn, and R. Spliet, "An improved integer L-shaped method for the vehicle routing problem with stochastic demands", *INFORMS Journal on Computing* 35(2), 423-439, 2023.

[8] M. Gendreau, G. Laporte, and R. Séguin, "An exact algorithm for the vehicle routing problem with stochastic demands and customers.", *Transportation science* 29(2), 143-155, 1995.

[9] C. Hjorring, and J. Holt, "New optimality cuts for a single-vehicle stochastic routing problem". *Annals of Operations Research* 86, 569-584, 1999.

[10] A. Florio, R. Hartl, and S. Minner, "New exact algorithm for the vehicle routing problem with stochastic demands", *Transportation Science* 54(4), 1073-1090, 2020.

# Real-Time Routing Cost Predictions for Time Slot Management

**Gustavo Hurovich, Lucas Veelenturf, Niels Agatz**
Rotterdam School of Management,
Erasmus University, Rotterdam, The Netherlands
Email: hurovich@rsm.nl

## 1 Introduction

With the rapid growth of e-commerce, urbanization and consumer expectations for swift and reliable deliveries, the final leg of the supply chain has become a critical factor in customer satisfaction and business success. By "last mile", we refer to the final (and costliest) segment of the delivery process, where goods are transported from distribution centers to the end-users' doorsteps.

Within last-mile literature, Attended Home Delivery (AHD) has attracted significant attention. Here, the presence of the customer at the delivery location is required, either because some of the products are perishable or require refrigeration (this is usual in the groceries sector), or due to the high monetary value of the products delivered. Additionally, in the e-grocery sector, it is usual to consolidate multiple orders into the same delivery route before making the actual deliveries, which allows for more efficient and sustainable planning of the delivery routes. Here, the last mile delivery can be divided into three stages, as illustrated in Figure 1. First, there is the order taking, where the customer agrees with the retailer on fulfilment options, and then chooses the desired products. In this stage, the business aims to accept a maximal set of customers, subject to capacity constraints. After a cut-off time, no more orders are accepted and the route planning stage starts; orders are assigned to vehicles and routes are drawn by solving a Vehicle Routing Problem with Time Windows (VRPTW). Ideally, the orders accepted in the first stage allow for efficient routes, and there is enough capacity to accommodate all of them. Finally, in the delivery stage, orders are picked and loaded into the respective vehicles, and deliveries are carried out.

Between the first two stages, the offer of time slots plays a pivotal role. The choices made by customers in terms of time slots (which largely depend on what they are offered) have a large impact in the routing efficiency. E-grocers typically try to steer demand during the order intake phase by strategically modifying the menu of time slot options

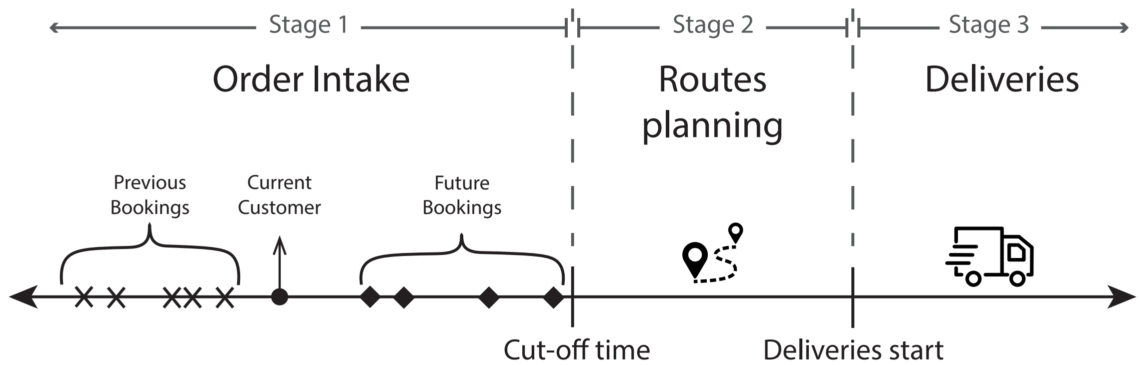



Figure 1: The three stages in periodic attended home delivery.

shown to customers. The set of tools and strategies employed to do this is referred to as demand management in the literature ([1], [2]). The primary goal is to guide customers towards choosing options that are individually acceptable for them while also being efficient and sustainable from the company's perspective. While some of the demand management decisions can be made in advance, others can only be made once there is an actual customer attempting to make a purchase. In the latter case, we call them real-time decisions, since the immediacy expected by customers while booking imposes tight constraints on the available computation time.

## 2 Formal Problem

Currently, in most e-grocer companies, whenever a customer initiates a purchasing session, the offered procedure is as follows: $(i)$ The seller provides a menu of possible time slots, each one with an associated price; $(ii)$ The customer chooses their preferred one, or none at all and leaves; $(iii)$ The customer selects their desired products and confirms the order. For the seller, in order to steer the customer's behaviour, it is important to have a good estimation of the costs associated with each time slot choice. Given the set of previous orders, we define the (myopic) marginal routing cost of accepting a customer $c$ in time slot $t$ as the difference between the total length of the optimal routing schedules, with this customer and without him at all. Formally, we define it as $\text{marginalCost}(S, c, t) = \text{routingCost}(S \cup \{(c, t)\}) - \text{routingCost}(S)$, where $\text{routingCost}(S)$ corresponds to the cost of an optimal schedule of routes that serves all customers in $S$.

There exist in literature both exact methods and heuristics to obtain the solution of a VRPTW. However, in this context, the computational time constraint is extremely tight. In this setting, and to have a smooth customer flow on the website, the retailers require these computations to be done in half a second at most. Such a constraint makes even simple heuristics computationally challenging.

# 3 Methodology

Given the computational time constraint, we propose the use of Supervised Machine Learning models to produce estimations of marginal costs. Concretely, any such model should be able to take the set of previous customers with their choices, and output a real number for each possible time slot, representing the marginal cost of each choice. We will train the models offline, using synthesized instances, and quickly evaluate them in real-time, when a customer initiates a purchasing session. These types of models already proved to be effective for determining when a time slot is feasible or not (see [3]).

## 3.1 Training

In order to train our models, we synthesize several input instances, and label them with their corresponding marginal routing costs. With that end, we proceed as follows: (1) Generate a pool of customer locations, following widely-used (clustered) distributions. (2) Draw a subset of $k + 1$ of those locations, representing the set of $k$ previous customers $S$ and one new incoming customer $c$. (3) Assign a time slot $T(s)$ to each customer $s \in S$. (3) Solve the VRPTW without the new customer and obtain a base cost. (4) For each available time slot $t$, assign $t$ to the incoming customer and solve the VRPTW including him. (5) Label the instance $(S, T, c, t)$ with the obtained marginal cost. To solve the VRPTW instances, we used a state-of-the-art open-source VRPTW solver based on Hybrid Genetic Search, which provides high quality solutions in a reasonable amount of time.

## 3.2 Models and Features

We will use a insertion heuristic as a benchmark, since that is the preferred method in the literature [2], and compare model performance against it. The proposed models include Random Forests, XGBoost, Neural Networks (NN), and Graph Neural Networks (NN), each with different architecture configurations and hyperparameter selection. We also explore the use of different feature sets, ranging from the whole set of raw customer locations and time slots chosen to aggregated features such as costumer density and proportion of customers in each slot, among others.

# 4 Initial Results and next steps

The implementation on this project is in progress, and we can share some initial results. We initially evaluated some vanilla (out of the box) models (Random Forest, XGBoost and NN), and we obtained a baseline, which will be used not only to compare performance across models, but also on different architecture configurations and to make hyperparameter tuning. In terms of features, the base configuration used the raw customer locations, and we used a fixed number of customers and time windows. We can summarize the initial results in Table 1. In total, 62500 instances were labeled, out of which 80% were used as training data and the remaining as test. Training was done minimizing the Mean Squared Error (MSE), and we also show the Mean absolute percentage error (MAPE).

The Mean Absolute Errors obtained by the vanilla models are not good, which is expected based on the rudimentary setup used. There are clear paths for improvement, which are being explored by us. Both on the model side, including feature engineering and hyperparameter tuning, and with the instance generation, which was done in a simple way in the baseline and we believe has a large impact on the performance of the models.

| Model | Configuration | MAE | MAPE | $R^2$ |
|---|---|---|---|---|
| Random Forest | 100 Trees | 759.2 | 0.763 | 0.325 |
| | 1000 Trees | 757.66 | **0.754** | **0.331** |
| XGBoost | 100 Trees | **748.15** | 1.051 | 0.304 |
| | 1000 Trees | 778.58 | 3.011 | 0.246 |
| NN | Two Layers | 797.58 | 0.896 | 0.246 |

Table 1: Baseline results for each model and configuration. Dependent variable values in the training set range between 0 and 10000, corresponding to 4hs of extra travel time.

## References


[1] N. Agatz, A. Campbell, M. Fleischmann and M. Savelsbergh, "Time slot management in attended home delivery", *Transportation Science* 45(3), 435-449, 2011.

[2] A. Campbell and M. Savelsbergh, "Decision Support for Consumer Direct Grocery Initiatives", *Transportation Science* 39(3), 313–27, 2005.

[3] L. van der Hagen, N. Agatz, R. Spliet, T.R. Visser and L. Kok, "Machine Learning–Based Feasibility Checks for Dynamic Time Slot Management", *Transportation Science*, 2022.

# Incorporating Neighborhood Interactions in Bike Sharing Rebalancing

**Christian Håkon Torsten Inngjerdingen[1], Simen Aksland Møller[1], Steffen J.S. Bakker[1], Henrik Andersson[1], and Mohamed Ben Ahmed[1,*]**

[1]Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Alfred Getz veg 3, 7491 Trondheim, Norway

## 1 Introduction

We investigate a bike sharing system (BSS) comprising a fixed number of bike stations within an urban area, and a homogeneous fleet of bicycles. Each station has a fixed number of locks, thus imposing a constraint on the number of bicycles that can be simultaneously stationed at any given station. Within this system, two situations are considered undesirable: congestion and starvation. Starvation is defined as the situation in which a customer requests a bicycle, but none are available at the station. In response, the customer can either search for a bicycle at a nearby station (referred to as roaming for bikes) or consider using an alternative means of transportation. On the other hand, congestion occurs when a customer intends to return a bicycle, but the station lacks available parking capacity. In such instances, the customer must find an alternative station with accessible parking slots. The impact of these conditions extends beyond the station experiencing violation itself, creating a demand spillover effect that can influence neighboring stations. To avoid or recover from such situations, BSS operators employ service vehicles for the continuous redistribution of bicycles among stations. This rebalancing process involves two critical decisions. The first decision is the selection of stations to visit, which is guided by real-time data, historical usage patterns, and factors such as station occupancy and user demand. The second decision pertains to determining the exact number of bicycles to load or unload at each visited station, requiring a delicate balance between addressing station imbalances and optimizing resource utilization. The rebalancing problem is inherently dynamic and stochastic. It is dynamic because the distribution of bicycles between stations is not known beforehand; instead, it continually evolves throughout the day as user demand fluctuates. Simultaneously, it is a stochastic problem as future demands for bicycles and locks at stations remain uncertain, although they are assumed to follow a known distribution.

While the bike rebalancing problem has garnered increasing attention in the literature, a substantial portion of existing contributions tends to neglect interactions between stations and the spillover effect of demand [2]. Instead, these contributions often focus on addressing violations at individual stations in isolation, implementing rebalancing strategies centered around visiting stations with the highest potential to mitigate violations at specific locations. Incorporating spillover effects into the decision-making process would involve redistributing bicycles based on minimizing the aggregate number of violations, encompassing not only the violated station but also nearby stations. This perspective represents a holistic approach to bike rebalancing that recognizes the interconnectedness of stations, aiming to enhance the overall efficiency and effectiveness BSS.

This paper investigates the Dynamic Stochastic Bike Rebalancing Problem with Neighborhood Interactions (DS-BRPRNI), an extension of the more general Dynamic Bike Rebalancing Problem (DBRP). DS-BRPRNI deals with uncertain demands at bike stations, with the primary objective of mitigating violations on a system level, as opposed to treating each station independently. Another distinctive feature is the incorporation of long roaming options within BSSs. To solve this problem, we introduce a metaheuristic named the "Explorative Preferred Iterative Look-Ahead Technique" (X-PILOT), building upon the PILOT method presented in [4]. Our method extends the original one by considering coordination between multiple vehicles and allowing more extensive heuristic exploration. Solutions obtained from this method are evaluated using a discrete-event simulator mimicking real-life BSSs. Computational experiments using historical data from major urban centers demonstrate that rebalancing with the X-PILOT method significantly enhances BSS efficiency compared to benchmark policies, including the original PILOT method [4].

## 2 Solution method

For conciseness, this section exclusively outlines the X-PILOT method, omitting details about the discrete-event simulator. The method constructs a rebalancing plan for each vehicle, specifying the stations to be visited along with the quantities of bikes to be loaded and unloaded at each station. Vehicle routes commence from a designated station, typically the current position but potentially random. The method initializes the process by calculating load/unload quantities at the starting station using a constructive algorithm. These calculations consider known demand rates at the current station and neighboring stations, accounting for users' roaming behaviors. The subsequent step involves determining the next station to visit. The selection process branches on three distinct stations, extending further to two subsequent nodes, resulting in six heuristic plans. Station selection is based on a criticality score computed for each station. Figure 1 provides an overview of the X-PILOT method. Evaluation of these plans follows, with the most favorable branching selected, quantities decided, and branching continuing from the newly added station. Criticality scores are continuously updated after each visit. A tabu list is maintained, including visited and non-relevant stations. The method accommodates multiple vehicles, provided the tabu list is regularly updated. The process repeats until the planning horizon concludes or a time limit is reached. The notable advantage lies in anticipating future rebalancing actions, optimizing present and future decisions. It introduces a parameterized branching level for heuristic exploration, a significant enhancement compared to the PILOT method, which permits only one-level branching.

To address demand uncertainty, we adopt a problem-based scenario generation approach. This method involves the generation of multiple scenarios, each representing distinct demand realizations at stations, achieved through sampling from historical data. Subsequently, the X-PILOT is executed for each scenario, and the resulting plans are evaluated across all scenarios using the discrete-event simulator. The plan that consistently performs well across the spectrum of scenarios is ultimately selected. This approach is advantageous in that it promotes the development of more robust rebalancing strategies, as decisions are optimized over a wide range of scenarios rather than being solely tailored to a single demand realization.

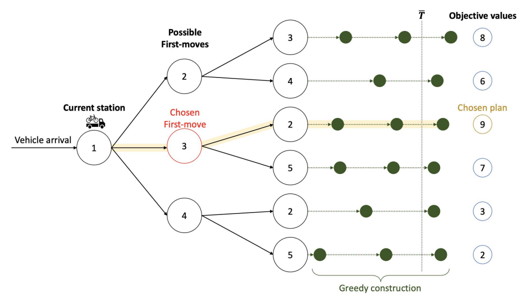



Figure 1: Overview of the X-Pilot.

## 3 Computational study

Computational experiments were conducted using data instances derived from actual BSSs located in Trondheim, Bergen, Oslo, and New York. Each data instance provides information on the number of bike stations within the BSS, their respective locations, capacities, and historical trip data. Additionally, details regarding the rebalancing vehicles, such as their capacities and travel distances, are also included in the data set. Both the metaheuristic and the simulator were implemented in Python and executed on a machine with a dual 2.4GHz Intel Xeon Gold 5115 CPU processor and 96GB of RAM. To assess the efficiency of the X-PILOT method, we introduce two evaluation metrics, namely, *successful events* and *failed events*. Successful events encompass all events of realized bike pick-ups and deliveries at each station, and successful roamings for bikes and locks within a defined neighborhood of 350 meters. Failed events account for events of starvations occurring at stations and roaming for locks that extend beyond the 350-meter neighborhood. In our analysis, we compare the performance of our X-PILOT method to the two greedy heuristics (GP & GPNI) described in [1] and the PILOT method implemented by [3]. Figure 2 presents a comparison of the number of failed events for each rebalancing policy, which has been tested across three distinct data instances. Note that, for the last instance, the method by [3] failed to solve the problem within a reasonable time limit.

In Figure 2, the results clearly demonstrate that X-PILOT yields the most favorable overall performance over a simulated horizon of ten days. In contrast, the GP heuristics exhibit the least favorable performance while the GPNI and Kloimüllner PILOT methods closely follow. In all scenarios, the application of X-PILOT for rebalancing leads to improvements in the BSS's service rates. These improvements are especially pronounced in larger and more imbalanced systems. For instance, in the case of Oslo instances, the service rate increases from 89.77% on average to 92.65% when X-PILOT is used. While this increase may not seem substantial in terms of percentages, it translates into a significant reduction of up to 75,000 failed events over the course of a season, underlining the substantial impact of the X-PILOT method.

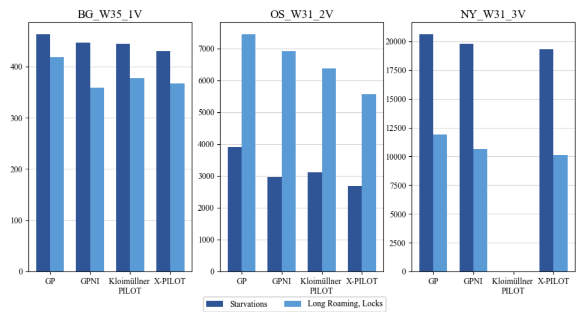



Figure 2: Number of failed events for different rebalancing policies on three data instances.

## References


[1] S. J. Bakker, A. Djupdal, L. Natvig, H. Andersson, K. Fagerholt, and M. Jahre. FOMOSim: Enabling rigorous performance evaluation of shared mobility systems using an open-source simulator. Working paper, `https://github.com/EECS-NTNU/fomo.`, 2022.

[2] M. D. Gleditsch, K. Hagen, H. Andersson, S. J. Bakker, and K. Fagerholt. A column generation heuristic for the dynamic bicycle rebalancing problem. *European Journal of Operational Research*, 2022.

[3] C. Kloimüllner, P. Papazek, B. Hu, and G. R. Raidl. Balancing bicycle sharing systems: An approach for the dynamic case. In *Evolutionary Computation in Combinatorial Optimisation*, pages 73–84. Springer Berlin Heidelberg, 2014.

[4] S. Voßs, A. Fink, and C. Duin. Looking ahead with the Pilot method. *Annals of Operations Research*, 136:285–302, 2005.

# Exact Solution of the Single Picker Routing Problem with Scattered Storage

**Stefan Irnich**
Logistics Management,
Johannes Gutenberg University, Mainz, Germany
Email: irnich@uni-mainz.de

**Katrin Heßler**
Global Data and AI, F.LCA - Global Technology and Data
Schenker AG, Essen, Germany
Email: katrin.hessler@schenker.de

## 1 Introduction

Warehouse activities include receiving, storing, picking, packing, and shipping operations [7]. Excellent surveys introduce warehouse operations planning including storage assignment, warehouse layout planning, zoning, routing, and batching [1, 19]. In this work, we address picking operations in manual (non-automated) warehouses where pickers move through the warehouse in order to collect articles from the storage locations (picker-to-parts). de Koster et al. [4] highlight that more than 80% of all order-picking systems in Western Europe are low-level picker-to-parts picking systems. Order picking denotes the process of retrieving inventory items (articles) from their storage locations in response to specific customer requests [4, 12]. Manual order picking is certainly very labor-intensive, and the literature gives different estimations for the effort: Typically, 60% of all labor activities in the warehouse result from order picking, and its cost can be estimated to be as much as 55% of the total warehouse operating expense [5, 18]. Frazelle [6] estimates that order picking contributes to up to 50% of the total warehouse operating costs. These figures explain why research on order picking operations is extensive and of high practical relevance.

In its pure form, the *single picker routing problem* (SPRP) seeks a minimum-length picker tour given the warehouse layout and the pick positions from where articles must be collected. The SPRP can be considered solved: On the one hand, the seminal work of Ratliff and Rosenthal [16] assuming a single-block parallel-aisle warehouse shows that a minimum-length picker tour can be computed with dynamic programming in linear time [9]. On the other hand, the SPRP is practically well-solved with routing policies that are rule-based heuristics such as traversal (a.k.a. S-shape), midpoint, largest gap [8], return, composite [15]. The application of heuristic routing policies is well justified in settings where pickers cannot perform all types of optimal tours, which can be complicated, counter-intuitive, and difficult to memorize. Instead, pickers perform tours defined by some simple rules. A little bit more involved is the routing when the policy combined is applied [17]. Both exact and heuristic techniques have been extended into many different directions, e.g., to other warehouse layouts [2, 14, 17], non-identical start and end points [11, 13], and multiple end points [3].

When one or several articles are pickable from more than one pick position, the warehouse operates as a *scattered storage* warehouse or *mixed shelves* warehouse. Recent works [1, 20, 21] stress that scattered storage is predominant in modern e-commerce warehouses of companies like Amazon or Zalando. The main advantage of this storage strategy is "that items of demanded SKUs are found close by irrespective of the position within the warehouse [so that] the distance to be covered for order picking is reduced this way" [20, p. 139]. The *SPRP with scattered storage* (SPRP-SS) is an integrated operational planning problem characterized by two levels of decisions. The picker routing constitutes the lower-level decision, i.e., the lengths of different picker tours must be computed to evaluate higher-level decisions. The higher-level decision is, for each requested article, the selection of one or several storage positions from where a sufficient number of this article can be collected. If the selection has been made, the resulting picker routing problem is the SPRP. However, both levels are interdependent and the SPRP-SS is known to be NP-hard [20], even if optimal routing is replaced by one of the above simple heuristic routing policies [10].

## 2 Contributions

The focus of our work is on algorithmic improvements for exactly solving the SPRP-SS. The effective solution algorithm we propose relies on the following underlying modeling approach: For the SPRP, every feasible picker tour is a path in the state space of the dynamic-programming approach of Ratliff and Rosenthal [16], and vice versa. The underlying assumption is that all pick positions are known and given. For the SPRP-SS, however, since the picker routing problem is a subproblem of the integrated operational planning problem, the fulfillment of the given pick list creates a new situation in which the *selection* of pick positions becomes essential. Our leading idea is to extend the state space of Ratliff and Rosenthal so that the selection aspect is fully modeled. In the extended state space, every feasible picker tour is still a path. However, not every path fulfills that the requested number of articles can be collected. This requirement to make consistent decisions regrading demand covering can be modeled with additional constraints in a shortest-path problem, which can no longer be solved with dynamic programming. We show that this model can be solved well as a binary program with the help of established *mixed-integer (linear) programming* (MIP) solvers.

## References


[1] Nils Boysen, René de Koster, and Felix Weidinger. Warehousing in the e-commerce era: A survey. *European Journal of Operational Research*, 277(2):396–411, 2019. doi: 10.1016/j.ejor.2018.08.023.

[2] Melh Çelk and Haldun Süral. Order picking under random and turnover-based storage policies in fishbone aisle warehouses. *IIE Transactions*, 46(3):283–300, 2014. doi: 10.1080/0740817x.2013.768871.

[3] René de Koster and Edo van der Poort. Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions. *IIE Transactions*, 30(5):469–480, 1998. doi: 10.1023/a:1007599307171.

[4] René de Koster, Tho Le-Duc, and Kees Jan Roodbergen. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2):481–501, 2007. doi: 10.1016/j.ejor.2006.07.009.

[5] Jolyon Drury. Towards more efficient order picking. Technical report, The Institute of Materials Management, Cranfield, UK, 1988.

[6] E. Frazelle. *World-Class Warehousing and Material Handling*. McGraw-Hill, New York, 2002.

[7] Jinxiang Gu, Marc Goetschalckx, and Leon F. McGinnis. Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177 (1):1–21, 2007. doi: 10.1016/j.ejor.2006.02.025.

[8] Randolph W. Hall. Distance approximations for routing manual pickers in a warehouse. *IIE Transactions*, 25(4):76–87, 1993. doi: 10.1080/07408179308964306.

[9] Katrin Heßler and Stefan Irnich. A note on the linearity of Ratliff and Rosenthal's algorithm for optimal picker routing. *Operations Research Letters*, 50(2):155–159, 2022. doi: 10.1016/j.orl.2022.01.014.

[10] Laura Korbacher, Katrin Heßler, and Stefan Irnich. The single picker routing problem with scattered storage: Modeling and evaluation of routing and storage policies. Technical Report LM-2023-01, Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Mainz, Germany, 2023. URL https://logistik.bwl.uni-mainz.de/files/2023/02/LM-2023-01.pdf.

[11] Maximilian Löffler, Nils Boysen, and Michael Schneider. Picker routing in AGV-assisted order picking systems. *INFORMS Journal on Computing*, 34(1):440–462, January 2022. doi: 10.1287/ijoc.2021.1060.

[12] Makusee Masae, Christoph H. Glock, and Eric H. Grosse. Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics*, 224:107564, 2020. doi: 10.1016/j.ijpe.2019.107564.

[13] Makusee Masae, Christoph H. Glock, and Panupong Vichitkunakorn. Optimal order picker routing in a conventional warehouse with two blocks and arbitrary starting and ending points of a tour. *International Journal of Production Research*, 58(17): 5337–5358, 2020. doi: 10.1080/00207543.2020.1724342.

[14] Ömer Öztürkoğlu, Kevin R. Gue, and Russell D. Meller. Optimal unit-load warehouse designs for single-command operations. *IIE Transactions*, 44(6):459–475, 2012. doi: 10.1080/0740817x.2011.636793.

[15] Charles G. Petersen. An evaluation of order picking routeing policies. *International Journal of Operations & Production Management*, 17(11):1098–1111, 1997. doi: 10.1108/01443579710177860.

[16] H. Donald Ratliff and Arnon S. Rosenthal. Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research*, 31(3):507–521, 1983. doi: 10.1287/opre.31.3.507.

[17] Kees Jan Roodbergen and RenÉ de Koster. Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39(9):1865–1883, jan 2001. doi: 10.1080/00207540110028128.

[18] J.A. Tompkins, J.A. White, Y.A. Bozer, E.H. Frazelle, and J.M.A. Tanchoco. *Facilities Planning*. John Wiley & Sons, Hoboken, NJ, 2003.

[19] Teun van Gils, Katrien Ramaekers, An Caris, and René B.M. de Koster. Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research*, 267(1):1–15, 2018. doi: 10.1016/j.ejor.2017.09.002.

[20] Felix Weidinger. Picker routing in rectangular mixed shelves warehouses. *Computers & Operations Research*, 95:139–150, 2018. doi: 10.1016/j.cor.2018.03.012.

[21] Felix Weidinger, Nils Boysen, and Michael Schneider. Picker routing in the mixed-shelves warehouses of e-commerce retailers. *European Journal of Operational Research*, 274(2):501–515, 2019. doi: 10.1016/j.ejor.2018.10.021.

# A data-driven location-routing optimization for sustainable medical waste management

**Ginger Y. Ke**
Faculty of Business Administration,
Memorial University of Newfoundland, St. John's, Canada
Email: gke@mun.ca

**Jiahong Zhao** **Jianfeng Chen**
School of Civil and Transportation Engineering,
Guangdong University of Technology, Guangzhou, China

## 1 Introduction

COVID-19 has inspired a lot of effort in research on uncertainties. As the pandemic is coming to an end, the quantity, composition, and spatial distribution of waste have been altered, with relatively more generations from smaller sources. The uncertainties are also shifted to quotidian operations, especially emergency services that are directly related to the reduction of infectious risks. Moreover, the expansion of the medical waste management system has a notable impact in this context due to the increasing number of facilities and vehicles, which consequentially lead to a series of environmental issues, such as greenhouse gas emission, natural energy consumption, and global climate change [1]. As a result, challenges posed to the urban waste systems persist but call for a renewed focus on ensuring sustainable waste management considering regular daily waste collections and emergency response operations. Motivated by the new developments in the post-pandemic era, we herein propose a bi-objective chance-constrained optimization model aiming to construct a cost-efficient and environmentally-friendly medical waste management system with proper facility locations and routing plans.

## 2 Mathematical Model

### 2.1 Objective functions

Two objectives, minimization of cost and risk, are considered in our location-routing problem. To that end, the total cost includes the expenditure on facility locations and operations, fuel charges for waste collection, cost of decarbonization, and procurement expenses of vehicles. The total risk, on the other hand, includes the risks at each disposal facility and on each vehicle route under uncertain emergency response time.

## 2.2 An integrated dynamic pollution-population (DPP) risk assessment

The system risk evaluates harmful impacts on the surrounding population caused by possible incidents at disposal facilities (site risk) or on transportation paths (edge risk). Our proposed risk assessment model jointly considers multiple factors, including the source (pollution of leakage, i.e., the amount of waste on-site or en route), consequence (exposed population within a certain radius), and dynamics (risk amplification due to delays in emergency response).

## 2.3 Fuzzy chance-constrained emergency response time

In practice, the emergency response time is normally uncertain due to the unstable traffic conditions in the urban road network. Two sets of chance constraints are enclosed to ensure that the probabilities of meeting the required response time $\tau$ to sites and edges are higher than a predetermined threshold $\theta$. Take the facility node $i \in \mathcal{F}$ for example, the chance constraint is $\Pr\left\{\tilde{T}_{hi} \leq \tau\right\} \geq \theta$, where $\tilde{T}_{hi}$ is the uncertain response times for hazmat $h \in \mathcal{H}$. Taking the idea of [2], the emergency response time is considered uncertain in the form of fuzzy numbers in the trapezoidal possibility distributions, and then the necessity measure is applied to handle the above chance constraints.

## 2.4 A modified two commodity flow formulation

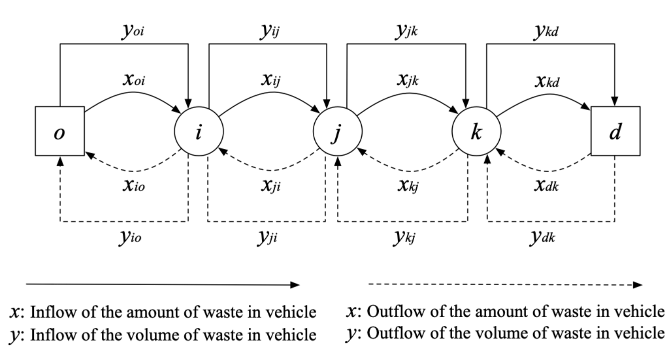



Figure 1: Flows of two-commodity flow

The two-commodity flow model uses copies of the origin nodes to transfer tours to routes, avoiding the sub-tour elimination in the traditional 3-index vehicle routing formulations, and hence discloses a higher convergence rate and better computational efficiency. In this work, we make two modifications to this model so to better serve our research purpose (see Figure 1 for detailed flows).

First, following a UNEP joint report [3], any compression and squeezing are prohibited during the medical waste collection, and so it is crucial to ensure the volume restriction, which is set to be 2/3 of the full vehicle volume. Secondly, a precise calculation of cost (fuel and decarbonization) and risk (edge pollution source) requires the knowledge of vehicle loads when leaving generation nodes, but the regular two-commodity flow model fails to achieve this. We introduce two additional decision variables and three constraint sets to compute the transportation direction on each edge and the accumulative vehicle load.

# 3 A Data-Driven Bi-Objective Solution Procedure

## 3.1 Traffic prediction by Back Propagation Neural Network (BPNN)

To evaluate the unstable emergency response time, we herein employ the real-time congestion index provided by BaiduMap. We collect the historical data on each edge for at least four consecutive working days, and predict the traffic congestion index by using BPNN. Then, after a comprehensive statistical analysis, the four prominent points of uncertain emergency response time between the emergency facility and edges can be obtained.

## 3.2 An NN-NSGA-II algorithm

According to the unique characteristics of our proposed model, we revised the original NSGA-II with a nearest-neighbor decoding procedure (NN-NSGA-II).

In more detail, we define each chromosome as a two-row matrix, respectively containing a permutation of medical waste generation nodes and a permutation of disposal facility candidates. To start, the facility nearest to the first generation node is chosen to be the origin of a route. From this facility, the route goes through the first node, and then moves to the nearest node that is covered by emergency services. The process continues until either the facility or the vehicle capacity limit is violated, and thus we obtain a nearest neighbor-based vehicle route. Proceeding to the first unassigned generation node for the next routes until all nodes are exhausted. The total cost and risk can be computed given the final location-routing plan.

Applying non-dominated sorting, and using crowding distance for fitness evaluation, chromosomes are selected with tournament selection and operated with precedence operations crossover and reverse variation. Finally, the Taguchi design approach is adopted to seek the best parameter values.

# 4 A Real-World Case Study

A series of experiments are conducted based on the real network in Shanghai, China. NN-NSGA-II efficiently derives 58 non-dominated solutions, where the recommended plan was selected via the linear programming technique for multidimensional analysis of preference proposed by [4].

Our analyses show that the optimal waste management location-routing system obtained from our model can sufficiently enlarge the system capacity and reduce both risks, without inducing extra cost. With the use of BPNN, the number of facilities decreases by 20%, and the network emergency coverage can be increased by 22%. A higher emergency confidence level leads to lower edge coverage and higher risk values. In the meantime, edges that can be selected for routing are more limited, which results in longer travel distances and larger costs. Through a comparison with two widely applied risk methods, DPP can achieve higher equity through the degree of confidence in emergency services, and ensures that every edge in the network can be properly covered if any incident occurs.

## 5 Conclusion

Recognizing the important role of emergency response, we develop a bi-objective chance-constrained optimization model to seek the best waste processing plan such that both the cost and risk are simultaneously minimized. Through a data-driven BPNN approach, the dynamics of traffic flow and congestion are predicted and fed into a set of fuzzy chance constraints to ensure the effective coverage of the emergency response. With a modified NSGA-II method, numerical experiments based on a real-world network are employed to test the proposed model. Managerial insights are revealed to facilitate practical decision making in urban medical waste management.

## References

[1] Ghannadpour, S. F., Zandieh, F., and Esmaeili, F. "Optimizing triple bottom-line objectives for sustainable health-care waste collection and routing by a self-adaptive evolutionary algorithm: A case study from Tehran province in Iran". *Journal of Cleaner Production*, 287 , 125010, 2021.

[2] Pishvaee, M., Razmi, J., and Torabi, S. "Robust possibilistic programming for socially responsible supply chain network design: A new approach". *Fuzzy Sets and Systems*, 206 , 1–20, 2012.

[3] Institute for Global Environmental Strategies. "Waste management during the COVID-19 pandemic: From response to recovery". URL: https://www.unep.org/resources/report/waste-management-during-covid-19-pandemic-response-recovery. 2020.

[4] Wang, Z. and Rangaiah, G. P. "Application and analysis of methods for selecting an optimal solution from the pareto-optimal front obtained by multiobjective optimization". *Industrial & Engineering Chemistry Research*, 56, 560–574, 2017.

# The Impact of Service Levels in Stochastic Production Routing with Adaptive Routing

**Ali Kermani, Jean-François Cordeau, Raf Jans**
Department of Logistics and Operations Management
HEC Montréal and CIRRELT, Canada
Email: ali.kermani@hec.ca, jean-francois.cordeau@hec.ca, raf.jans@hec.ca

## 1 Introduction

The Production Routing Problem (PRP) encompasses both the Lot Sizing Problem (LSP) and the Inventory Routing Problem (IRP) to enhance supply chain integration and reduce costs associated with poor coordination. The PRP typically involves decisions about the production, inventory management, and delivery of a single product made in a plant and distributed to multiple customers over a finite and discrete time horizon (see, e.g., Adulyasak et al. [2] and Hrabec et al. [4] for an overview of the relevant literature).

Considering demand uncertainty can help companies to minimize the costs associated with stockouts and customer dissatisfaction. Adulyasak et al. [1] investigate both two-stage and multi-stage formulations of the stochastic PRP (SPRP) with demand uncertainty. In their proposed formulation, routing decisions are made in the first stage, limiting flexibility by requiring customer visits in all scenarios, regardless of whether deliveries are needed. In Kermani et al. [5], we study the potential cost savings associated with flexible routing in a two-stage SPRP under demand uncertainty where the routing decisions are second-stage variables. In the present paper, we consider service level constraints, which are an effective strategy to deal with uncertainty, especially when satisfying demand from third-party suppliers or imposing penalty costs for unmet demand is not a viable option. The advantages and limitations of different service levels in the LSP are discussed in [3, 6]. Surprisingly, this aspect remains unexplored in the PRP, despite its relevance to numerous real-world situations. More precisely, we model four distinct service levels (referred to as $\alpha$, $\beta$, $\gamma$, and $\delta$), each designed to address different metrics based on specific assumptions [6]. In our study, we introduce a two-stage SPRP with adaptive routing (SPRP-AR) and service level constraints to address all these service levels and compare their effects on the SPRP.

## 2 Problem Definition

In this problem, a single product is produced and delivered to a set of customers over a finite planning horizon. We generate a finite set of scenarios to address demand uncertainty based on a given nominal demand and a discrete uniform distribution using Monte Carlo simulation. We assume a homogeneous fleet of capacitated vehicles with a defined capacity that pick up the product from the plant and deliver it to customers in each period, returning to the plant at the end of their tour. A fixed setup cost is considered whenever production takes place, alongside a unit production cost per item produced. Additionally, we account for unit holding costs for goods carried to the next period, as well as routing costs associated with the edges traversed by vehicles.

The objective is to minimize the total cost, consisting of the first-stage setup costs and the expected value of the second-stage production, holding, and routing costs. The production quantity is a second-stage variable, limited by the plant's production capacity. This variable can be adapted to the realized demand, while setup decisions are the only first-stage decisions in the problem, made at the beginning of the planning horizon. The amount of products that can be held at each node for future demands is also limited by the maximum inventory level of the node. Additionally, we bound the delivery amount for each vehicle by its capacity and prevent split deliveries to customers. We consider a distinct formulation for each service level, addressing associated conditions (see, e.g., [3, 6]). For the $\alpha$ service level, we bound the probability of stockouts for a customer in a period. For the $\beta$ service level, we impose restrictions on the expected backorder divided by the average demand. In the case of the $\gamma$ service level, we enforce a predetermined ratio of expected backlog to expected demand. Lastly, for the $\delta$ service level, we restrict the expected backlog divided by the maximum expected backlog.

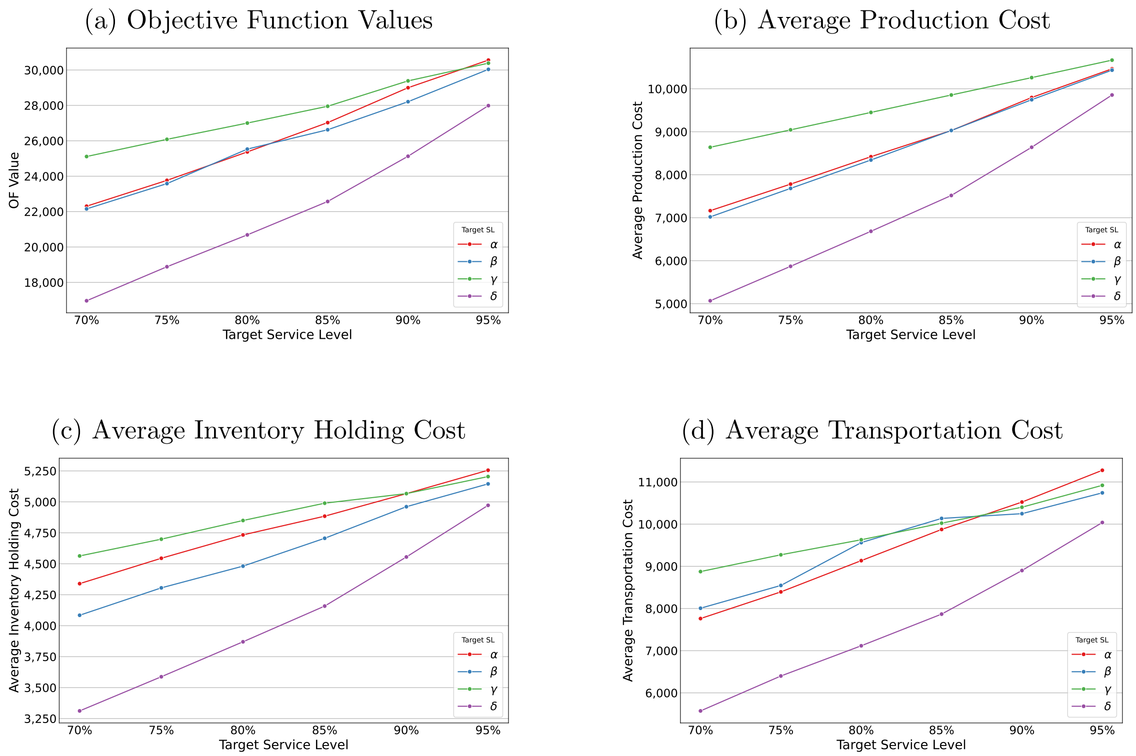
## 3 Solution Algorithm

We introduce an iterative matheuristic algorithm (IMH) designed to solve the SPRP-AR with service level constraints. The core principle of this algorithm lies in the decomposition of the original problem into more manageable subproblems. The goal of the first phase is to swiftly generate setup decisions. To achieve this, we employ a two-level LSP that includes the production plant and a single aggregated customer. At this stage, we compute the transportation cost as the cost associated with the Traveling Salesman Problem (TSP) across all nodes and apply an aggregate delivery approach to the vehicles. All binary variables, excluding the setup decisions, are relaxed at this stage, as our sole aim is to identify setup decisions. Moving on to the second phase, we maintain the setup decisions established in the first phase and proceed to solve an SPRP considering an aggregate delivery quantity for the vehicles. The goal of this phase is to determine if the setup decisions from the first stage can lead to a feasible solution for the second-stage variables. Additionally, we incorporate service level constraints to align our solution with the predefined service levels.

In the third phase, we solve a Restricted Inventory Routing Problem (RIRP) for each scenario. Here, we continue with fixed setup decisions and impose an upper bound (UB) on backlogs to ensure service level feasibility. We allow flexibility in inventory, delivery, and production quantities to refine our solutions further. Upon completing these phases, we enter an intensification phase, where we take into account approximate visit costs. We introduce visit variables to the second phase and iterate over the new problem, followed by the third phase, continually updating the approximate visit costs to explore potential enhancements. If no further improvements are obtained or the intensification phase reaches its stopping criteria, we introduce a local branching constraint in the first phase. This initiates a repeat of the procedure to generate a new setup decision and continue the process until a stopping criterion is met.

## 4 Results

We conducted computational experiments using the dataset introduced in [1] with high transportation costs to compare the impact of different service levels. Specifically, we present results for the smaller dataset that consists of 5 to 30 customers (with intervals of 5), 3 periods, 1 to 3 vehicles, and 100 scenarios. We analyze 6 different values for each service level, ranging from 70% to 95% with a 5% interval. This results in 108 instances for each type of service level and 432 instances in total. Figure 1a presents the cost comparison of the objective function for different service levels with different target values. We can observe that the least strict service level is the $\delta$ service level, followed by the $\beta$ and $\gamma$ service levels. However, the $\alpha$ service level is an event-based service level which is mostly close to the $\beta$ service level in terms of objective function value. It is also obvious that by increasing the target value of the service level, the objective function cost increases, while the difference of the objective function between different service levels tends to decrease. In Figure 1b, Figure 1c, and Figure 1d, we provide details of the different parts of the recourse variables of the objective function, including average production, inventory, and transportation costs. One can observe that all parts follow the same trend, while the transportation cost tends to vary more, especially for the $\alpha$ and $\beta$ service levels.




(a) Objective Function Values (b) Average Production Cost (c) Average Inventory Holding Cost (d) Average Transportation Cost

Figure 1: Cost comparison for different service levels

# References

[1] Y. Adulyasak, J.-F. Cordeau, and R. Jans, "Benders Decomposition for Production Routing Under Demand Uncertainty", *Operations Research* 63(4), 851-867, 2015.

[2] Y. Adulyasak, J.-F. Cordeau, and R. Jans, "The production routing problem: A review of formulations and solution algorithms", *Computers & Operations Research* 55, 141-152, 2015.

[3] S. Helber, F. Sahling, K. Schimmelpfeng, "Dynamic capacitated lot sizing with random demand and dynamic safety stocks", *OR Spectrum* 35(1), 75–105, 2013.

[4] D. Hrabec, L.M. Hvattum, A. Hoff, "The value of integrated planning for production, inventory, and routing decisions: A systematic review and meta-analysis", *International Journal of Production Economics* 248, 108468, 2022.

[5] A. Kermani, J.-F. Cordeau, and R. Jans, "A Progressive Hedging-based Matheuristic for the Stochastic Production Routing Problem with Adaptive Routing", Submitted to *Computers & Operations Research*, 2023.

[6] N. Sereshti, Y. Adulyasak, and R. Jans, "The value of aggregate service levels in stochastic lot sizing problems", *Omega* 102, 102335, 2021.

# Scheduled Service Network Design with Packing Considerations

**Sara Khodaparasti**
DIGEP, Polytechnic University of Turin, Turin, Italy
Email: sara.khodaparasti@polito.it

**Maria Elena Bruni and Teodor Gabriel Crainic**
CIRRELT & DIMEG, University of Calabria, Cosenza, Italy
School of Management & CIRRELT, UQAM, Montreal, Canada
**Guido Perboli and Walter Rei**
CIRRELT & DIGEP, Polytechnic University of Turin, Turin, Italy
School of Management & CIRRELT, UQAM, Montreal, Canada

## 1 Introduction

Tactical planning for consolidation-based carriers is a complex problem, usually addressed by the *Service Network Design* (*SND*) methodology. The SND literature generally disregards some crucial issues, however, e.g., how cargo is loaded into transportation vehicles or storage facilities - called *capacity units* in the following -, and the selection of the proper number and type of capacity units for each selected service within the designed network.

Such considerations may have a substantial impact on the system performance and should be carefully tackled to avoid underestimating costs or generating infeasible itineraries exceeding the capacity of the capacity units of the selected services [1]. To better represent the physical and operational attributes of capacity units and the utilization demand flows make of that capacity, one must address the challenges of replacing the standard aggregated flow capacity constraints with more realistic packing ones [1].

There are very few papers in the SND literature explicitly integrating packing constraints into SND models. [3] present new formulations for the Scheduled SND problem, where shipments cannot be split, while consolidation is desirable to reduce the number of homogeneous vehicles used when multiple shipments dispatch simultaneously on the same direct service. Given the somewhat simple direct-service structure, the set of commodity clusters which can be associated to each physical link (i.e., travel together on it) may be generated a priori. The packing and the SND-related decisions may then be addressed separately. [2] combine two classical problems, transportation and variable size and cost bin packing, aiming to ship multiple types of commodities on different types of vehicles moving between the supply and demand nodes of a bipartite network, while minimizing the transportation and resource-acquisition cost.

We present a unified problem setting, the *Service Network Design Problem with Packing Considerations* (SNDPC), addressing simultaneously decisions on the selection of scheduled services (on a general time-space network), the selection of the type and number of capacity units to load and move the origin-destination demands, the assignment of demand flows to the loading units, and the construction of the demand itineraries within the selected service network.

Differently from [3], Scheduled SND and packing decisions are addressed simultaneously. The problem setting is general and the freight is moved via different itineraries and using a heterogeneous fleet of vehicles. Combining two $NP$-hard combinatorial problems, network design and packing, does not make the problem easier to address and requires careful investigation.

## 2 Problem description and mathematical formulation

Consider a physical network $\mathcal{G}^{PH} = (\mathcal{N}^{PH}, \mathcal{A}^{PH})$, where $\mathcal{N}^{PH}$ is the set of nodes (representing transfer and consolidation terminals, which include the origins and destinations of demand), and $\mathcal{A}^{PH}$ is the set of links (e.g., road, rail, and river) joining these nodes.

Each potential *service* $\sigma \in \Sigma$ is defined by a route in $\mathcal{G}^{PH}$ linking its origin $o(\sigma)$ to its destination $d(\sigma)$, without intermediate stops, as well as a schedule giving the *departure* from origin and *arrival* at destination times, $\alpha(\sigma)$ and $\beta(\sigma)$, respectively. The service is composed of a number of capacity units, or *bins*, to be selected within a particular set $J_\sigma$, where $J_\sigma \cap J_{\sigma'} = \emptyset, \forall \sigma \neq \sigma' \in \Sigma$; $J = \cup_{\sigma \in \Sigma} J_\sigma$. The bins may represent containers or vehicles, or any other transportation medium, of different types. Each bin type $\pi \in \Pi$ is characterized by a capacity $Q_\pi$ and a fixed selection/usage cost $c_\pi$. $\Pi$ stands for the set of bin types, and $\phi(j) \in \Pi$ gives the type of bin $j \in J$. Each service is characterized by a maximum total number of bins, $U_\sigma$, and maximum number of bins of type $\pi$, $N_{\pi\sigma}$, which may be assigned to it, a fixed selection cost $f_\sigma$, and unit bin cost $c_{\phi(j)}$, $j \in J_\sigma$, including the loading, unloading, and transportation costs of freight within the bin on the service.

Each *demand* $k \in \mathcal{K}$ represents a request to transport a set of *items*, $I(k)$ with $I = \cup_{k \in \mathcal{K}} I(k)$, from its origin $o(k)$ to its destination $d(k)$. The items are available at time $\alpha(k)$ and need to be delivered to the final destination at time $\beta(k)$. Each item $i \in I(k)$ is characterized by a size $v_i$ (expressed in the same unit as the bin capacity), the size of the demand, $d_k$, being the summation of the size of its items. The demand may be split among different services, or put into different bins of the same service, as long as the temporal requirements are satisfied. In all cases, items arriving at a terminal different from their destination are unloaded from the bins of the preceding service, eventually held at the terminal for a while at a unit holding cost $h_k$, and then loaded into different bins associated with different departing services, continuing their journey to the final destination. To design the service network, it is therefore necessary not only to select the services to be operated, but also to determine the assignment of items to bins, which adds another layer of complexity to the problem. Hence, the main goal of the *SNDPC* is to select a set of services and the capacity units of various types to be associated to each service to satisfy the demand at minimum cost.

We model the *SNDPC* on a time-space network, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, built by extending the physical network $\mathcal{G}^{PH}$ along the dimension of time for the fixed duration of the schedule length, discretized into *time periods* $t \in \mathcal{T}$ of equal length. Operations at terminals in different periods are modeled with different nodes of the form $(n, t) \in \mathcal{N}$. There are two types of arcs in $\mathcal{A}$. A *service arc*, joining nodes $(n, t)$ and $(n', t')$, models the operation of a (single-leg) service between its origin $o(\sigma) = n$ and destination $d(\sigma) = n'$, starting at time $\alpha(\sigma) = t$ and arriving at time $\beta(\sigma) = t'$. A *holding arc*, joining nodes $(n, t)$ and $(n, t+1)$, models the possibility of holding items at node $n$ from period $t$ to $t+1$. $\mathcal{A}^{\Sigma}$ and $\mathcal{A}^{H}$ stand for the sets of service and holding arcs, respectively, with $\mathcal{A} = \mathcal{A}^{\Sigma} \cup \mathcal{A}^{H}$. Note that, one has $\mathcal{A}^{\Sigma} = \Sigma$ for the single-leg case.

We consider the following sets of decision variables: $y_\sigma \in \{0,1\}, \sigma \in \Sigma$, for the selection of service $\sigma$; $z_j \in \{0,1\}, j \in J_\sigma$, $\sigma \in \Sigma$, selects or not bin $j$ of service $\sigma$; $x^i_{aj} \in \{0,\, 1\}, a \in \mathcal{A}^{\Sigma}, j \in J_{\sigma_a}, i \in I$, represents the possible assignment of item $i$ to bin $j$ of service $\sigma$ (arc $a$); $w^i_a \in \{0,\, 1\}, a \in \mathcal{A}^H$, $i \in I$, indicates if item $i$ is held on arc $a$. The *SNDPC* model:

$$\text{Minimize} \sum_{\sigma\in\Sigma} f_\sigma y_\sigma + \sum_{j\in J} c_{\phi(j)} z_j + \sum_{a\in\mathcal{A}^H}\sum_{k\in\mathcal{K}} h_k\Big(\sum_{i\in I(k)} w^i_a\Big) \tag{1}$$

$$\text{s.t.} \quad \sum_{a\in\mathcal{A}^+_{(n,t)}}\sum_{j\in J_{\sigma_a}} x^i_{aj} + \sum_{a\in\mathcal{A}^+_{(n,t)}} w^i_a - \Big(\sum_{a\in\mathcal{A}^-_{(n,t)}}\sum_{j\in J_{\sigma_a}} x^i_{aj} + \sum_{a\in\mathcal{A}^-_{(n,t)}} w^i_a\Big)$$

$$= \begin{cases} 1, & \text{if } (n,t) = \big(o(k), \alpha(k)\big), \\ -1, & \text{if } (n,t) = \big(d(k), \beta(k)\big), \quad \forall (n,t)\in\mathcal{N}, \forall k\in\mathcal{K}, \forall i\in I(k) \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

$$\sum_{i\in I} v_i x^i_{aj} \le Q_{\phi(j)} z_j, \forall a\in\mathcal{A}^{\Sigma}, \forall j\in J_{\sigma_a} \tag{3}$$

$$\sum_{j\in J_\sigma} z_j \le U_\sigma y_\sigma, \forall \sigma\in\Sigma \tag{4}$$

$$\sum_{\substack{j\in J_\sigma \\ \phi(j)=\pi}} z_j \le N_{\pi\sigma}, \forall\sigma\in\Sigma, \forall\pi\in\Pi \tag{5}$$

$$y_\sigma \in \{0,1\}, \forall\sigma\in\Sigma, \;\; z_j\in\{0,1\}, \forall j\in J_\sigma, \forall\sigma\in\Sigma \tag{6}$$

$$x^i_{aj} \in \{0,1\}, \forall a\in\mathcal{A}^{\Sigma}, \forall j\in J_{\sigma_a}, \forall i\in I, \;\; w^i_a\in\{0,1\}, \forall a\in\mathcal{A}^H, \forall i\in I \tag{7}$$

where $\mathcal{A}^+_{(n,t)} = \{a = \big((n'', t''), (n', t')\big) \in \mathcal{A} | n'' = n,\, t'' = t\}$ and $\mathcal{A}^-_{(n,t)} = \{a = \big((n', t'), (n'', t'')\big) \in \mathcal{A} | n'' = n,\, t'' = t\}$, for each $(n, t) \in \mathcal{N}$ and $\sigma_a \in \Sigma$ denotes the service associated with arc $a \in \mathcal{A}^{\Sigma}$. The objective function (1) minimizes the total cost of the selected services, the bins used, and the holding of items at terminals. Constraints (2) ensure that each item is routed from its origin node to its destination node, respecting the temporal constraints. Constraints (3) enforce a feasible assignment of items to bins, respecting the bin capacity. Constraints (4) represent the limits on the global capacity of each service. Constraints (5) limit the total number of bins of each type for each service. Finally, constraints (6)-(7) express the nature of the variables.

## 3 Conference Presentation

We will present a comprehensive view of the topic, identify issues and challenges of different variants of the problem, and discuss mathematical formulations.

We will also present the numerical results, obtained using an off-the-shelf commercial solver, on two sets of instances built on networks from SNDLib (http://sndlib.zib.de) and the SND literature [4]. The sensitivity analysis will focus on the performance of the solver (and the state-of-the-art enumeration algorithm it offers) with respect to the instance dimension and various settings of the problem parameters (e.g., schedule length, number and characteristics of capacity units, the flexibility of the demand due dates, the split/no split demand requirements), as well as to the impact of these variations on the structure of the solutions, in comparison with the results produced by the Scheduled SND formulation with a classic modelling of the service-capacity restrictions.

We will conclude with an overview of possible avenues for the design of solution methods tailored for the particular problem structure and able to address large instances.

## References

[1] Bruni M.E., Crainic T.G., Perboli G., *Bin Packing Methodologies for Capacity Planning in Freight Transportation and Logistics*, In *Contributions to Combinatorial Optimization and Applications*, Crainic T.G., Frangioni A., Gendreau M. (Eds.), Springer, 2023, forthcoming.

[2] Flamand T., Iori M., Haouari M., The transportation problem with packing constraints, *Computers & Operations Research* 157:106278, 2023.

[3] Hewitt M., Lehuédé F., New Formulations for the Scheduled Service Network Design Problem, *Transportation Research Part B: Methodological*, 172:117-33, 2023.

[4] Lanza G., Crainic T.G., Rei W., Ricciardi N., Scheduled service network design with quality targets and stochastic travel times, *European Journal of Operational Research*, 288(1):30-46, 2021.

# Online stochastic optimization for real-time transfer synchronization in public transit networks

**Laura Kolcheva**
Department of Mathematics and Industrial Engineering,
Polytechnique Montréal, CIRRELT & GERAD, Montreal, Canada

**Antoine Legrain**
Department of Mathematics and Industrial Engineering,
Polytechnique Montréal, CIRRELT & GERAD, Montreal, Canada
Email: antoine.legrain@polymtl.ca

**Martin Trépanier**
Department of Mathematics and Industrial Engineering,
Polytechnique Montréal, CIRRELT, Montreal, Canada

## 1 Introduction

Public transportation (PT) systems are increasingly important in the context of urban growth, traffic congestion and sustainable development. PT networks are designed in multiple phases: planning, operation, and control. While changes in network design or operation systems are often expensive and difficult to implement, innovative control strategies offer a more cost-effective solution to improve the overall performance of PT networks.

Research shows that the speed and protection of transfers is one of the key factors influencing passengers' willingness to use PT ([1]). Transfers are generally synchronized during scheduling, but buses operate in a stochastic environment and can deviate from their timetables which leads to missed transfers and the possible loss of users. This is why there is a growing interest in transfer synchronization strategies for PT systems, especially in real-time. The increasing real-time availability of data on passenger demand from smart cards or bus occupancy sensors, on planned transfers from travel apps, and on vehicle locations from GPS allow significant improvements in understanding the state of PT networks. Dynamic predictions shed light on the impacts real-time control might have on users along all segments of their trips. Transfer synchronization aims towards a network-wide optimization with less myopic decisions.

We investigate the transfer synchronization problem for buses in a dense urban network through control tactics. We integrate predictions of future states of the PT system using both real-time data and historical data made available by the "Société de Transport de Laval" (STL). We bring the following contributions to the field: 1) Implementation of real-time control tactics for the synchronization problem using an arc-flow formulation ; 2) Solving large instances containing whole bus lines and many transfer points in real time. 3) Use of three online optimization algorithms for the transfer synchronization problem, and comparison of their performances. 4) Testing on a real large-scale data set from a dense PT network.

## 2 Problem description

Three control tactics are implemented alone or simultaneously in order to synchronize transfers and minimize passenger travel times. The holding tactic makes a bus wait at a stop after all passengers have boarded or alighted the vehicle. Holding is a very efficient tactic to avoid deviation from schedules, bus-bunching as well as to synchronize transfers. The holding tactic reduces operational speed and adds additional travel time for passengers onboard vehicles and waiting time for passengers wanting to board further along the line. Secondly, we use the skip-stop and skip-segment tactics. Skipping one or more consecutive stops can help reduce bus travel times or catch up delays with respect to schedules. Skipping stops has an immediate effect which is avoiding dwell times at stops, and a more long-term effect from the limiting of the number of passengers aboard the bus. Buses with fewer passengers spend statistically less time at stops. The stop-skipping tactic reduces the travel time aboard the bus and the waiting time of passengers waiting further along the line. On the other hand, stop-skipping can strongly inconvenience passengers wishing to board/alight on stops that are skipped, especially for low-frequency bus lines. Finally, we also use the speed control or speedup tactic. Speed control is an inter-stop tactic. This tactic helps decrease bus travel times without negatively impacting passengers wishing to board or alight the bus. Speed control is not always applicable in real life because of traffic congestion and speed regulations. The speedup tactic decreases travel times for passengers onboard vehicles or waiting further along the line. The speedup tactic can also allow to catch up delays in schedule and avoid missing synchronized transfers. When tactics are used, deviations from the schedules must be limited. Unlike in the literature, all stops are control stops which means tactics can be implemented at any stop or between any two stops. This allows for a more efficient control but generates more variables. Finally, the impact of tactics on all stages of passenger trips are considered.

## 3 Solution method

**Arc-flow model** An arc-flow model is formulated for the offline transfer synchronization problem using control tactics. The model minimizes total passenger travel time by improving transfer times while constraining deviation from the schedule.

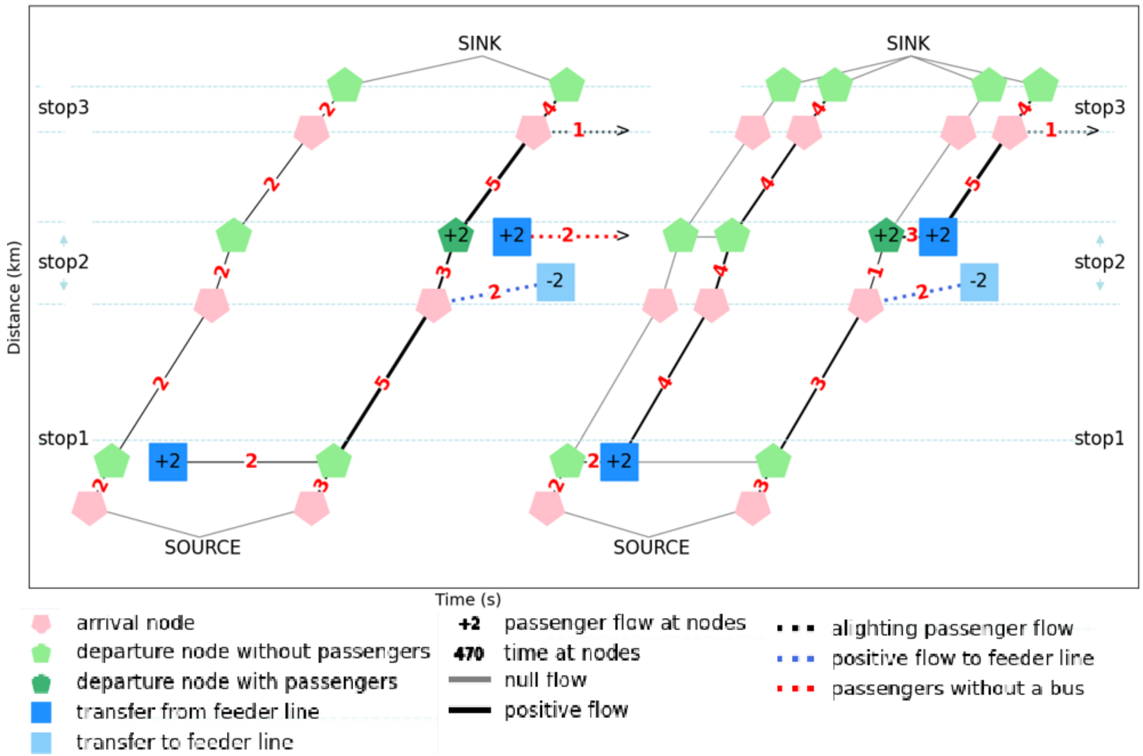



Figure 1: Arc-flow example: Left-no tactics, right-with tactics.

All tactics are integrated into a time-expanded graph of the arc-flow formulation as presented in Figure 1. In the model, we consider a main line on which we can apply tactics and feeder lines which are considered fixed. The model considers a control horizon that can range between only a few stops to the entirety of the main line.

**Online stochastic optimization** Three online stochastic algorithms ('Mean', 'Consensus' and 'Regret') [2] are adapted for the online transfer synchronization problem and evaluated in a simulation environment. Those online models can profit from the gradual reveal of real-time information and are designed to test and validate the results of the deterministic offline model. At each re-optimization, a control horizon containing some buses, stops and passengers is defined. This includes buses on the main line as well as feeder line vehicles that will transfer passengers at stops in the control horizon. Once the control horizon is defined, we collect available real-time data relevant to items in the horizon. Using sampling, we generate scenarios representing possible future states of the elements considered in the control horizon. We solve the offline model for each scenario providing decisions on tactics to use at all control stops in the control horizon : hold, skip-stop or speed control. We then apply the tactics - selected by one of three algorithms - only on the next stop of the control horizon. When a bus reaches the next stop, we start a new step and thus a new re-optimization in the simulation framework ; all future control tactics are re-evaluated at every iteration. The computation times at each re-optimization must stay low to allow an implementation in real time.

# 4 Experiments

Our experiments are based on data provided by the STL. Laval is a city in Canada with a population of 436,000. The PT network of the STL contains 46 bus lines and more than a thousand bus stops. The offline deterministic algorithm with perfect information has also been implemented to serve as target for the other algorithms. The implementation of no tactics is used as a baseline. Figure 2 shows results from computations on instances from line 42, a high frequency line with many passengers.

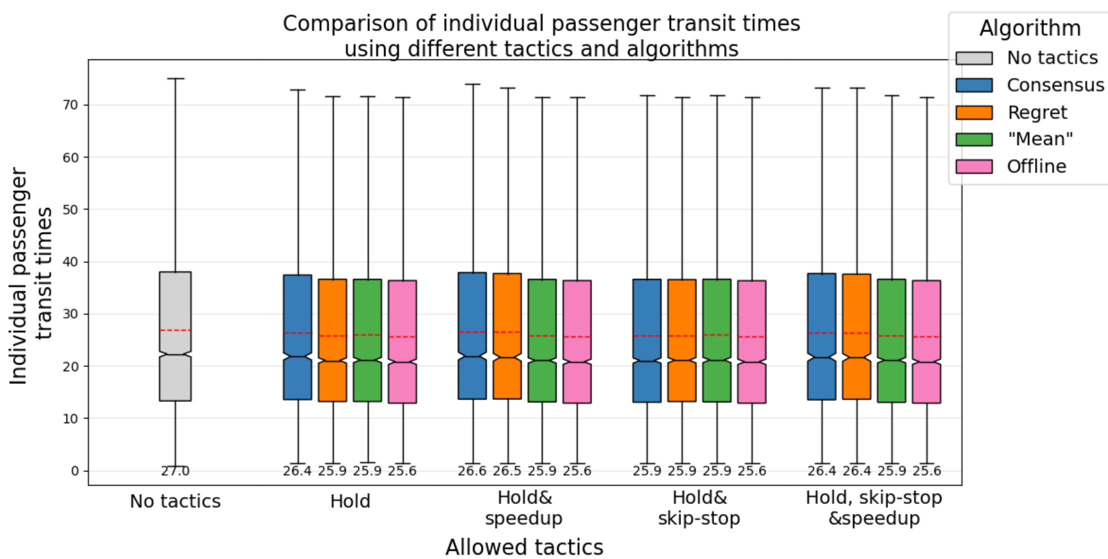



Figure 2: Total passenger travel times for different algorithms and tactics for the line 42.

# References

[1] A. A. Ceder, Y. Hadas, M. McIvor, and A. Ang. Transfer synchronization of public transport networks. *Transportation Research Record*, 2350(1):9–16, 2013.

[2] P. V. Hentenryck and R. Bent. *Online Stochastic Combinatorial Optimization*. The MIT Press, 2009.

# Resilient Relay Logistics Network Design Using $k$ Shortest Paths

**Onkar Kulkarni*, Mathieu Dahan, Benoit Montreuil**
Physical Internet Center, School of Industrial and Systems Engineering,
Georgia Institute of Technology, Atlanta, USA
*Corresponding Author: Onkar Kulkarni Email: onkar.kulkarni@gatech.edu

## 1 Introduction

The recent surge in e-commerce and world trade has led the parcel delivery industry to be one of the fastest-growing industries in the world. As the industry handles very large parcel volumes, the parcel delivery operations require meticulous planning, starting from designing the underlying logistics networks. Such networks majorly comprise hub facilities where parcels are sorted and then consolidated to ship towards their respective destinations. These parcel delivery operations traditionally rely on long-haul delivery trips that take a toll on the mental and physical health of delivery drivers, and present unsustainable working conditions. As a direct consequence, drivers are reluctant to work for such parcel delivery companies, leading to a major driver shortage in the industry. To remedy this situation, a potential solution consists of building relay facilities to fulfill demand via short-haul segments through relay transportation [1, 2, 3]. In relay logistics, the delivery drivers can advance commodities for half of their daily driving limit from one relay to the next, and then return to the original relay—ideally with other commodities—before reaching their home by the end of the day.

Logistics networks regularly face disruptions of varied nature, ranging from frequent and low-impact events such as congestion delays on roads, to low-probability high-impact events such as hurricanes. Such disruptions cause delivery delays, increased logistics costs, and a dip in customer satisfaction. While the optimization literature on relay network design does not consider disruption risks, the literature on logistics network resilience primarily designs small-scale non-relay logistics networks that can mitigate supply-demand disruptions. Since network structure has been shown to impact network resilience, we then aim to address the following research question: *How to design efficient and resilient logistics hub network configurations for relay transportation?* Drawing inspiration from the Physical Internet [3, 4], we aim to design large-scale hyperconnected relay logistics networks through topology optimization.

## 2 Model Formulation

We consider a logistics service provider interested in designing a large-scale logistics hub network for efficient and resilient relay transportation. We consider the initial planning phase of the design process and assume that the service provider has limited information regarding future demand and disruption risks. We introduce the problem of *k-Shortest Path Network Design* (*k*-SPND), which consists of locating logistics hubs to connect each origin-destination (O-D) pair with at least $k \geq 1$ routes of minimum total lengths. The premise is that by connecting O-D pairs with multiple short routes, it will then be possible to cost-effectively transport commodities with appropriate consolidation given the realized demand. In addition, if a multi-day disruption occurs at a hub or a transportation leg, then the service provider will be capable of transporting commodities via a different route, with a marginal impact on delivery cost and time.

Formally, let $\mathcal{P} := \mathcal{S} \times \mathcal{T}$ represent the set of origin-destination (O-D) pairs with each O-D pair $p$ having an associated demand share $d_p$. The service provider intends to open $N$ relay logistics hubs from a set of discrete candidate locations $\mathcal{H}$. Hubs are assumed to have sufficient capacity to handle large commodity volumes. We represent as $\mathcal{A} \subseteq (\mathcal{S} \cup \mathcal{T} \cup \mathcal{H})^2$ the set of potential (directed) transportation legs, which satisfy the driving time regulations to ensure a daily return for all drivers to their respective homes.

For every O-D pair $p = (s, t) \in \mathcal{P}$, we denote the set of $s - t$ paths as $\Lambda_p$ with each path $\lambda_p$ having travel time of $\tau_\lambda$. The goal of the $k$-SPND problem is then to select a subset of hub locations $\mathcal{H}_o \subseteq \mathcal{H}$ of size at most $N$ so as to minimize the demand-share-weighted total length of the $k$ shortest paths between each O-D pair in the subgraph induced by the set of nodes $\mathcal{S} \cup \mathcal{T} \cup \mathcal{H}_o$. To this end, we formulate it as a mixed-integer program (MIP) using path-based decisions. We consider for each hub $i \in \mathcal{H}$ a binary variable $y_i$ that takes a value of 1 if hub $i$ is opened, and 0 otherwise. Additionally, for every O-D pair $p = (s, t) \in \mathcal{P}$ and every $s - t$ path $\lambda \in \Lambda_p$, we define a continuous variable $z_\lambda$ that equals 1 if $\lambda$ is selected as one of the $k$ shortest $s - t$ paths in the subgraph induced by the opened hubs. We then derive the following MIP:

$$
\begin{aligned}
\min_{y,z} \quad & \sum_{p \in \mathcal{P}} \sum_{\lambda \in \Lambda_p} d_p \cdot \tau_\lambda \cdot z_\lambda && && \text{(1a)}\\
\text{s.t.} \quad & \sum_{i \in \mathcal{H}} y_i \leq N, && && \text{(1b)}\\
& \sum_{\lambda \in \Lambda_p} z_\lambda = k, && \forall p \in \mathcal{P}, && \text{(1c)}\\
& \sum_{\{\lambda \in \Lambda \,|\, i \in \lambda\}} z_\lambda \leq k \cdot |\mathcal{P}| \cdot y_i, && \forall i \in \mathcal{H}, && \text{(1d)}\\
& 0 \leq z_\lambda \leq 1, && \forall \lambda \in \Lambda, && \text{(1e)}\\
& y_i \in \{0, 1\}, && \forall i \in \mathcal{H}. && \text{(1f)}
\end{aligned}
$$

## 3 Solution Methodology

We develop two approaches for solving the large-scale MIP optimally: In the first approach, based on a tailored implementation of Benders decomposition, we provide an analytical characterization of the optimal dual solutions of the exponential-sized Benders subproblem to generate the feedback cuts. This leads to a pseudo-polynomial time approach to generate these cuts based on Yen's algorithm (for computing $k$ shortest paths), which we accelerate using breadth-first-search and shortest-path subroutines. In the second solution approach, we tailor an implementation of branch-and-price: At each node of the branch-and-bound tree, we solve the master problem—a linear program with an exponential number of variables and constraints—using column generation. Using complementary slackness we show that at each iteration of column generation, the pricing subproblem can also be solved in polynomial time using Dijkstra's algorithm in an auxiliary graph with edge lengths depending on the optimal dual variables of the restricted master problem.

## 4 Case Study and Partial Results

Using the national level data of one of the largest parcel delivery companies in China that partnered with our research team, we created 6 representative problem instances of increasing size and complexity. In every instance, the parcel demand originates at one of the existing outbound logistics facilities of a city owned by the company ($\mathcal{S}$) and is destined for one of the company's existing last-mile delivery centers ($\mathcal{T}$). As the company intends to implement relay transportation, it identified a set $\mathcal{H}$ of candidate locations to open relay hubs, given by the company's existing intercity logistics hubs or major highway intersections. For the transportation arcs ($\mathcal{A}$), we only retained the transportation legs for which the drive time does not exceed 5.5 hours since the Chinese government imposes an 11-hour daily driving limit for truck drivers. This ensures that parcels travel towards their respective destinations while drivers return home daily.

We run the developed solution approaches to solve the MIP with parameters ranging from 10 to 60 for hubs $N$ to open, and from 1 to 4 for the number of shortest paths $k$. Next, to validate the proposed $k$ shortest paths relay logistics networks, we compare their performance against relay logistics networks constructed with only cost considerations to support parcel delivery. As parcel delivery networks obtain their operational cost savings through commodity consolidations, we construct an *efficiency-optimized (E-O) network*, obtained by selecting up to $N$ hubs to open to minimize the cost of the consolidation for an average commodity demand.

We conduct a set of experiments, where we subject the networks to random hub disruptions. In each disruption scenario, occurring uniformly at random, we suppose a relay hub becomes dysfunctional and no parcel can be routed through it during the planning horizon. We consider week-long disruptions and determine a minimum-cost consolidation plan to measure the performance in that situation. We assume that if an O-D pair becomes disconnected in the relay network as a result of a disruption, the demand for that O-D pair cannot be fulfilled using relay transportation during the planning horizon.

Through the minimum-cost consolidation plan, we compute two performance metrics: average delivery costs for the fulfilled demand and the amount of unfulfilled demand through relay transportation. Figure 1 portrays the comparison results. It showcases that our proposed $k$ shortest paths networks with $k \geq 2$ outperform the efficiency-optimized networks when facing hub disruptions with respect to both performance metrics. In addition to guaranteeing the delivery of a higher proportion of parcel demand through short-haul transportation, our networks also achieve lower average delivery costs per parcel as compared to efficiency-optimized networks under disruptions. This shows that by ensuring the existence of an increased number of paths $k$ between each O-D pair, demand can more likely be fulfilled through the relay network when facing disruptions.

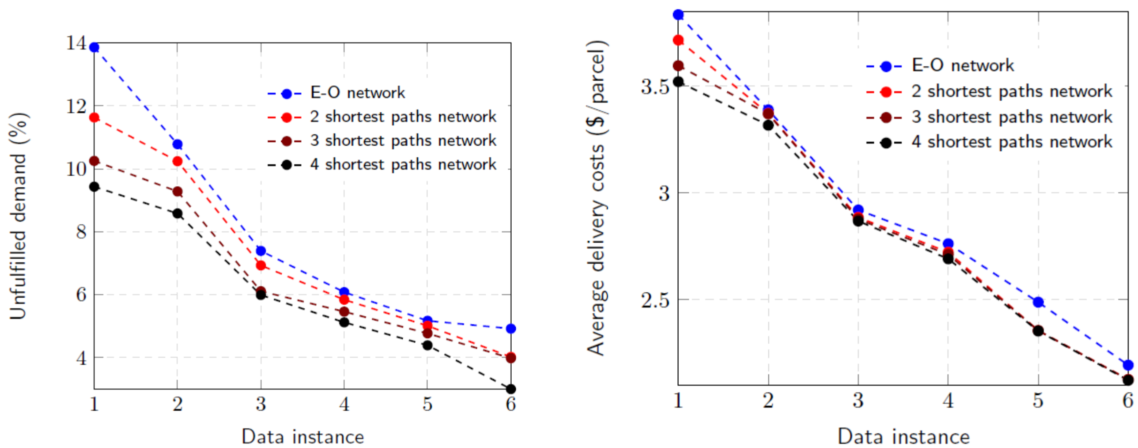



Figure 1: Comparison of network performance under 1-hub random uniform disruptions

If selected for presentation, we plan on further describing our network optimization approaches and on presenting the results regarding the efficiency comparison of the proposed with E-O networks under nominal situations, and the resilience-efficiency trade-off achieved by these networks.

## References


[1] H. Üster and H. Agrahari, "A Benders Decomposition Approach for a Distribution Network Design Problem with Consolidation and Capacity Considerations.", *Operations Research Letters* 39(2), 138-143, 2011.

[2] Z. Hu, R. G. Askin and G. Hu, "Hub relay network design for daily driver routes", *International Journal of Production Research* 57(19), 6130-6145, 2019.

[3] B. Montreuil, "Toward a Physical Internet: meeting the global logistics sustainability grand challenge", *Logistics Research* 3(2), 71-87, 2011.

[4] D. Hakimi, B. Montreuil and A. Hajji (2015). "Simulating Physical Internet Enabled Hyperconnected Semi-Trailer Transportation Systems",*Proceedings of 2nd International Physical Internet Conference.*

# Bid Construction for Urban Parcel Logistics via Combinatorial Auctions

**Simon Kwon**
School of Industrial and Systems Engineering,
Georgia Institute of Technology, Georgia, USA
Email: skwon82@gatech.edu

**Walid Klibi**
The Center of Excellence in Supply Chain (CESIT),
Kedge Business School, Bordeaux, France

**Mathieu Dahan**
School of Industrial and Systems Engineering,
Georgia Institute of Technology, Georgia, USA

**Benoit Montreuil**
School of Industrial and Systems Engineering,
Georgia Institute of Technology, Georgia, USA

## 1 Introduction

City logistics, with diverse stakeholders and conflicting interests, necessitates coordination for sustainable cities [1]. This study aligns with hyperconnected city logistics, where an orchestrator manages citywide demand flows and selects logistics service providers through a combinatorial auction. This auction consists of three stages: (1) pre-auction, selecting services for the primary auction; (2) bid construction, where bidders strategize and submit bids; and (3) winner determination, identifying auction winners [3].

This work delves into the second stage, addressing urban logistic nuances. Faced with high demand for time-sensitive operations and heightened competition among logistics service providers, less-explored aspects in the current literature, we investigate the construction of time-promised bids and the interplay between bidders, the orchestrator, and competitors. Our aim is to comprehend the competitive influence on overall profit through a stochastic bi-level optimization model. To the best of our knowledge, this is the first work addressing these urban logistics characteristics in the bid construction problem context. We explore an exact solution approach based on an optimal-value-function reformulation.

## 2 Methodology

### 2.1 Problem Overview

In a single-round, first-price reverse combinatorial auction, an urban logistics orchestrator (the auctioneer) interacts with multiple logistics service providers (bidders). The orchestrator manages flows between origin-destination (O-D) pairs, denoted as $O$, ensuring the timely delivery of demand for each O-D pair $o \in O$, with service guarantees represented by $\tau_o$. Each O-D pair $o \in O$ follows a predetermined path $\mathcal{P}_o$ designated by the orchestrator, segmented into a set of logistics activities including transport and hub processing activities. Through the auction, the orchestrator aims to allocate these activities to specific participating providers while satisfying the service guarantees of O-D pairs.

Let $L$ represent the set of logistic activities for auction, where each $l \in L$ has associated demand and time requirement options (e.g., 30, 60, and 90 mins). Demand corresponds to the expected volume and patterns for the planning horizon, while time requirements are bid promises linked to service level agreements (SLAs) denoted as $\mathcal{S}_l$ for each activity $l \in L$. The orchestrator sets bid requirements allowing single or bundled bids (multiple activities) with a limit of $K$ submissions per bidder. The auction-clearing process allocates bids to activities, aiming to meet O-D service guarantees while minimizing total allocation costs—a process known to bidders. Bidders seek to maximize profit by submitting bids specifying pairs of activities, SLAs, and corresponding bid prices.

This work focuses on a bidder's decision-making in the business setting, navigating uncertainties in demand, anticipating auction markets, and the orchestrator's responses.

### 2.2 Optimization Modeling and Solution Approach

We employ stochastic bi-level programming to address this problem, integrating bidding decisions of the bidder under consideration in the upper level and the orchestrator's decision-making process in the lower level. This accounts for bids from all bidders, including competitors, through a set of scenarios, anticipating their expected behavior.

Let $\bar{B}$ be the set of potential bids for the bidder under consideration, and $\bar{B}_l \subset \bar{B}$ be the set of bids that contain activity $l \in L$. The bidder has access to historical and known data on competitors' bids in the market. Competitors' bids and demand for logistic activities are modeled using a finite set of scenarios $\Omega$, each with probability $\phi(\omega)$ associating logistic activities with demand $d_l(\omega)$. Bids proposed by competitors in scenario $\omega$ are denoted as $\hat{B}(\omega)$. Let $B(\omega) = \bar{B} \cup \hat{B}(\omega)$ represent the set of all bids in scenario $\omega$, including both competitor bids and potential bids from the bidder under consideration. For each scenario $\omega$, bids that contain logistic activity $l \in L$ and have a corresponding SLA $s \in \mathcal{S}_l$ are denoted respectively as $B_l(\omega)$ and $B_{ls}(\omega)$. Each potential bid $b \in \bar{B}$ has associated activities $L(b)$, a maximum bid price $\bar{p}_b$, fulfillment cost $c_b(\omega)$ for each scenario $\omega$, and a specific SLA

$s \in \mathcal{S}_l$ for $l \in L(b)$. Each competitor's bid $b \in \hat{B}(\omega)$ in scenario $\omega$ comprises associated activities $L(b)$, a bid price $\hat{p}_b(\omega)$, and a specific SLA $s \in \mathcal{S}_l$ for $l \in L(b)$.

The goal of the bidder under consideration is to select a subset of bids to submit and bid prices for each bid such that the expected profit is maximized. Thus, for the upper-level problem, let binary variables $x_b$ indicate whether bid $b \in \bar{B}$ is submitted, and discrete variables $p_b$ indicate the bid price of bid $b \in \bar{B}$. For the lower-level problem, let binary variables $y_b(\omega)$ indicate whether bid $b \in B(\omega)$ in scenario $\omega$ is selected. We can formulate the bid construction problem as follows:

$$\max_{\mathbf{x},\mathbf{p},\mathbf{y}^*} \sum_{w \in \Omega} \left( \phi(\omega) \cdot \Big( \sum_{b \in \bar{B}} (p_b - c_b(\omega)) \cdot y_b^*(\omega) \Big) \right) \tag{1}$$

$$\text{s.t.} \sum_{b \in \bar{B}_l} x_b \leq 1, \quad \forall l \in L \tag{2}$$

$$\sum_{b \in \bar{B}} x_b \leq K \tag{3}$$

$$p_b \leq \bar{p}_b, \quad \forall b \in \bar{B} \tag{4}$$

where each $\mathbf{y}^*(\omega)$ satisfies:

$$\mathbf{y}^*(\omega) \in \arg\min_{\mathbf{y}(\omega)} \sum_{b \in \bar{B}} p_b \cdot y_b(\omega) + \sum_{b \in \hat{B}(\omega)} \hat{p}_b(\omega) \cdot y_b(\omega) \tag{5}$$

$$\text{s.t.} \sum_{b \in B_l(\omega)} y_b(\omega) = 1, \quad \forall l \in L \tag{6}$$

$$\sum_{l \in \mathcal{P}_o} \sum_{s \in \mathcal{S}_l} \sum_{b \in B_{ls}(\omega)} s \cdot y_b(\omega) \leq \tau_o, \quad \forall o \in O \tag{7}$$

$$y_b(\omega) \leq x_b, \quad \forall b \in \bar{B} \tag{8}$$

Equations (1) - (4) correspond to the upper-level problem, while equations (5) - (8) correspond to the lower-level problem. The upper-level problem seeks to determine a set of bids maximizing expected profit. Constraints (2) ensure that up to one bid is submitted for each logistic activity. Constraints (3) respect the auction requirement that limits the maximum number of bids to submit. Constraints (4) set the maximum bid price on bids.

For the lower-level problem, the objective of the orchestrator in (5) is to minimize the total allocation cost for each scenario $\omega \in \Omega$. Constraints (6) ensure that each logistic activity is allocated to one bid. Constraints (7) require that the allocation of bids to activities ensures the service time guarantees of each O-D pair $o \in O$. Constraints (8) link the upper- and lower-level problems, ensuring that bids are selected only if submitted.

To exactly solve the proposed bi-level model, we employ a value-function-based approach developed by [2]. This method iteratively generates bi-level feasible solutions, serving as an upper bound to the original problem. Simultaneously, the information corresponding to the lower-level variables is utilized to establish a lower bound. The algorithm terminates finitely with an optimal solution when all upper-level variables are discrete [2].

## 3 Preliminary Results

To test the proposed bid construction model and solution approach, we employed a set of synthetic urban area instances. In these scenarios, the orchestrator aims to allocate resources for 10 logistic activities, connecting 308 origin-destination (O-D) flows, with a total expected daily demand volume of approximately 30,000 parcels across the urban area. Each activity is characterized by an associated daily demand volume and SLA options. We considered three instance sizes: instance 1, associated with around 300 bids; instance 2, associated with 450 bids; and instance 3, associated with 600 bids, encompassing 30 scenarios in the optimization model. Additionally, we explored three market types with risk-averse (RA), risk-neutral (RN), and risk-seeking (RS) bidders, each characterized by different profit margins of bidders as shown in Table 1.

Table 1: *Impact of market characteristics on expected profit*

| Instance | Market Characteristics (Profit Margin) | Expected Profit ($) |
|---|---|---|
| 1 | RA (5-8%) | $59,680 |
| | RN (10-15%) | $71,098 |
| | RS (17-23%) | $80,196 |
| 2 | RA (5-8%) | $48,474 |
| | RN (10-15%) | $52,457 |
| | RS (17-23%) | $61,656 |
| 3 | RA (5-8%) | $45,488 |
| | RN (10-15%) | $51,253 |
| | RS (17-23%) | $66,460 |

Table 1 reveals that market characteristics significantly influence the expected profit of the bidder under consideration across instances. The gap in profit between the two extreme cases (RA and RS) ranges from 27% to 46%, emphasizing the need for a clear understanding and analysis of the markets when modeling the problem. In the presentation, we plan to delve into pre-processing for the translation of logistic activities into time-promised bids, provide business insights, account for urban logistic characteristics, and present numerical experiments to evaluate the performance of the proposed methodology with various urban instances.

## References

[1] T. G. Crainic and B. Montreuil, "Physical Internet enabled Hyperconnected City Logistics", *Transportation Research Procedia*, 12, 383-398, 2016.

[2] L. Lozano and J. C. Smith, "A Value-Function-Based Exact Approach for the Bilevel Mixed-Integer Programming Problem", *Operations Research*, 65(3), 768-786, 2017.

[3] J. Song and A, Regan, "Combinatorial auctions for transportation service procurement: The carrier perspective", *Transportation Research Record*, 1833(1), 40-46, 2003.

# Order Picking for E-Grocery

**Charlotte Köhler**
Department of Data Science & Decision Support
European University Viadrina, Frankfurt (Oder), Germany
Email: koehler@europa-uni.de

**Ann Melissa Campbell**
Department of Business Analytics
University of Iowa, Iowa City, United States

**Jan Fabian Ehmke**
Department of Business Decisions and Analytics
Universität Wien, Vienna, Austria

## 1 Introduction

In recent years, the e-grocery market has seen several new retailers, largely driven by the heightened demand for home delivery services during the COVID-19 pandemic. These companies often face the challenge of establishing a profitable business model, where careful planning is essential to balance tight margins with high customer expectations. The e-grocery fulfillment process can be divided into three steps: order acceptance, picking, and delivery [1]. During the order acceptance phase, retailers must decide which dynamically incoming orders to accept, taking into account their available resources for picking and delivery. In recent years, several studies have highlighted methods to enhance profitability by optimizing delivery routes during the order acceptance process. These methods suggest offering customers a limited selection of delivery time windows, e.g. [2], or setting differential pricing for various time slots to encourage customers towards more beneficial options for the retailer, e.g. [3]. While these approaches have exclusively focused on the use of delivery resources, the use of resources for picking orders for attended home deliveries has not received comparable attention [4], even though recent studies indicate that their impact on a retailer's profit is substantial and equivalent to the influence of routing decisions.[1]

[1]https://www.mckinsey.com/industries/retail/our-insights/achieving-profitable-online-grocery-order-fulfillment

This paper seeks to fill this gap by a detailed exploration of in-store picking costs in the context of e-grocery. This involves introducing a cost evaluation function designed to give retailers a more accurate and practical assessment of the resources required to pick orders to assess feasibility and maximize the number of accepted orders. By analyzing the interplay between the variety of items in an order and the associated picking times, we aim to offer a comprehensive framework to assess the feasibility of orders and explore the impact of different picking strategies on overall operational efficiency. We base our experiments on the layout of the well-known REWE grocery chain in Germany.

## 2 Exploring picking costs for e-grocery

For e-groceries, the time taken to pick items is a critical determinant of costs and use of resources, as longer picking times directly translate to increased labor expenses and fewer orders that can be picked. We define a set of items a customer orders by a set $I$. We can then represent a time to pick the items in set $I$ by a function $t = f(I)$. For a particular function, we can evaluate the feasibility and cost of arriving orders, given limits on picking time and a particular picking scheme.

We assume that all orders have a pickup or delivery window, but since all orders must be picked by the start of the window, they have a deadline $d$. Thus, in deciding whether to accept an order, it is important to evaluate whether an order can be picked by this deadline $d$ given previously accepted orders. In some applications, there may be an earliest time that an order may be picked up, but given the possibility of ice and cold storage, we will assume that that order can be picked at any time on the day of delivery before $d$. Thus, the feasibility of delivery of delivery time $d$ is based on the total picking time available before time $d$ and the previously accepted orders.

**Evaluating Order Feasibility.** For each incoming order, we must determine its feasibility concerning picking. This involves assessing whether the set of pickers can feasibly pick an additional order before the specified deadline, $d$. To make this assessment, we calculate the available slack time, which is determined by the difference between each picker's start time and the deadline, $d$, minus the total picking time of all previously accepted orders for that deadline or sooner. We repeat this for additional pickers and identify which pickers have sufficient picking time available. We explore strategies for assigning full orders to pickers and examine how varying deadline lengths, both longer and shorter, add to the complexity of evaluating the picking feasibility.

**Optimizing Order Assignment and Picking Efficiency.** Our focus also extends to enhancing the efficiency of how orders are picked. We propose three distinct policies in this regard. The first policy involves each picker completing an order individually, which represents the status quo. In addition, we explore the possibility of pickers handling

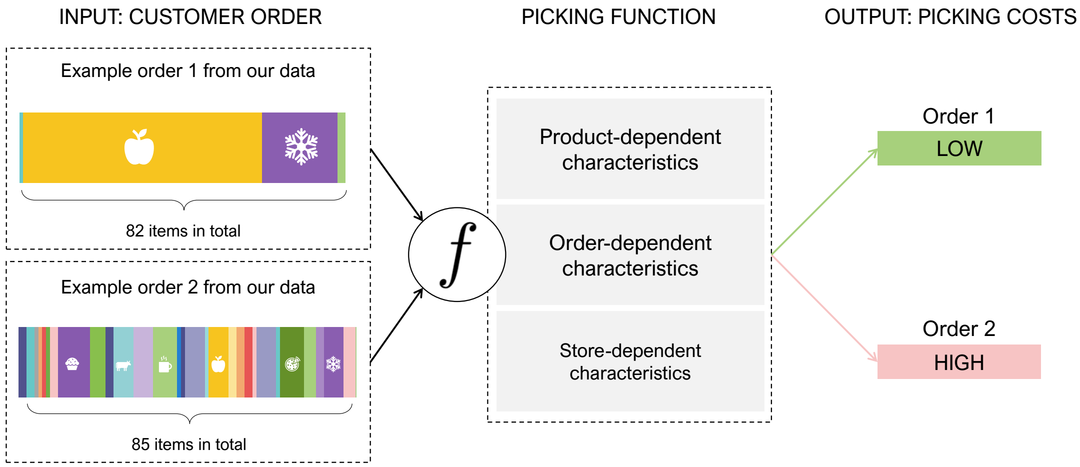



Figure 1: Determining picking costs using the picking function $f$

multiple orders concurrently. This second approach aims to minimize walking distances by enabling a picker to navigate the store just once but to fulfill several orders simultaneously. The third policy explores the concept of specialized pickers, where each picker focuses on items from a specific category, such as fresh or frozen products. Specialization is hypothesized to lead to increased picking speed, as pickers develop expertise in locating and selecting items within their designated category, thus streamlining the overall picking process.

**Picking Function.** For each order, we incorporate three key dimensions into the function $f$ to explore their impact on overall picking costs. For each order, there are *product-dependent characteristics*, such as the size and weight of each product in an order [5]. Generally, products that are smaller and lighter are simpler to pick, leading to lower picking costs. Second, there are *order-dependent characteristics*, which include the quantity of each item, the total number of different items in the order, and the total number of product categories represented by these items. Typically, orders comprising fewer distinct items are easier to pick, thus reducing picking costs. Last, there are *store-dependent characteristics*, focusing on how different store layouts affect picking processes such as walking time between product categories [6]. More compact stores with a limited range of items usually enable quicker picking, whereas larger stores with extensive inventories and layouts optimized for in-person shopping may complicate and lengthen the picking process.

In Figure 1, we present two customer orders from our data that are similar in their total item count but vary in terms of product types and the number of product categories they encompass. For each order, our picking cost function is applied to estimate the associated costs. In this example, the calculated costs reveal that the first basket incurs lower picking costs than the second basket.

## 3 Experiments and Outlook

We are conducting a comprehensive assessment of the proposed picking function, utilizing real-world data to ensure accuracy and relevance, for different order picking strategies. Our methodology involves a detailed store layout, based on REWE in Germany. Furthermore, we use historical order data from a former German e-grocer to enrich our analysis. This dataset encompasses information on over 400,000 order baskets, providing us with knowledge of the typical composition of orders. This information is instrumental in testing and validating our picking strategies and policies under realistic conditions.

With this approach, we evaluate orders as they arrive in terms of picking feasibility and cost with different picking schemes. We can also evaluate how the different order characteristics impact picking time and if some can be ignored to save computation time. Preliminary results indicate that for orders picked individually, the number of product categories represented by an order plays a larger role in the picking time than the total number of items. Store size is also an important factor.

Our findings will offer valuable contributions to the field, potentially guiding retailers in optimizing their picking processes and enhancing overall operational efficiency. The results will also help identify the picking costs and resources required for a range of baskets types. This will help identify the important costs associated with the picking side of e-grocery. The insights derived from this study will provide valuable input for a comprehensive analysis of both picking and delivery processes in the e-grocery sector.

## References


[1] A. M. Campbell and M. W. Savelsbergh, “Decision support for consumer direct grocery initiatives”, *Transportation Science* 39(3), 313-327, 2005.

[2] C. Köhler, A. M. Campbell and J. F. Ehmke, “Data-driven customer acceptance for attended home delivery”, *OR Spectrum* 1-36, 2023.

[3] R. Klein, M. Neugebauer, D. Ratkovitch, D. and C. Steinhardt, “Differentiated time slot pricing under routing considerations in attended home delivery”, *Transportation Science* 53(1), 236-255, 2019.

[4] K. Waßmuth, C. Köhler, N. Agatz and M. Fleischmann, “Demand management for attended home delivery–a literature review”, *European Journal of Operational Research*, 2023.

[5] S. Riveiro-Sanroman, M. Vazquez-Noguerol, J. A. Comesana-Benavides and J. C. Prado-Prado, “Determining the order picking time in food retailing”, *The international conference on industrial engineering and industrial management*, 2022.

[6] M. Vazquez-Noguerol, S. Riveiro-Sanroman, I. Portela-Carames and J. C. Prado-Prado, "Analyzing store features for online order picking in grocery retailing: an experimental study", *International Journal of Production Management and Engineering*, 2022.

# Cost assignment in delivery systems

**Francois Lamothe and Sandra Ulrich Ngueveu**
LAAS-CNRS, Université de Toulouse, CNRS, INP, Toulouse, France
Email: flamothe@laas.fr

## 1 Introduction

In a time when delivery systems are in rapid expansion while our economy needs to become more sustainable, the need for better analyses of such systems arises. In this work, we study the problem of assigning a servicing cost to each user of a delivery service. There is no immediate answer to this problem since the operating cost depends on the group of users (not on individual users) and economies of scale may be present. Meanwhile, this cost assignment is relevant for both the service provider who might integrate this information into its decision-making, and the user who may then better perceive its impact on the system. For example, the users may nowadays want to be able to measure the environmental cost of their delivery.

This assignment problem is equivalent to finding an appropriate cost-sharing mechanism in a cooperative game, the Traveling Salesman game. The game theory literature largely favors the Shapley value when sharing such costs because of its strong theoretical foundations [1]. The main disadvantage of the Shapley value is that it is very computationally intensive to evaluate in the context of delivery systems (or any system whose management involves an NP-hard problem). The literature contains several approximators which are still very expensive to compute in the delivery context. That is why in this work we design an approximator of the Shapley value requiring only a limited computational time. As shown in the preliminary computational results, this approximator appears to work well on the Traveling Salesman Problem instances tested compared to the approximators of the literature.

## 2 Problem setting

Given a weighted graph $G$, the goal of the Traveling Salesman Problem (TSP) is to find a cycle of minimum weight visiting all nodes of $G$.

In game theory, a cooperative game on a set of players $N$ is a function $v$ mapping each subset $S$ (also named coalition) of $N$ with a real number $v(S)$ representing the cost of servicing all players in $S$. Our interest is the Traveling Salesman Game which can be constructed from a TSP as follows. First, one node of $G$ denoted $o$ will serve as the origin of the cycles while the other nodes will be considered the players of the game. The cost $v(S)$ of a coalition of players $S$ is then the cost of the minimum weight cycle starting at $o$ and servicing all the nodes in $S$.

A cost-sharing mechanism in a cooperative game $v$ assigns a cost $\phi(i)$ to each player $i \in N$ such that the sum of all assigned costs is equal to $v(N)$. The most famous cost-sharing mechanism in game theory is the Shapley value. Indeed, this mechanism has strong theoretical foundations as it is uniquely defined by a set of basic axioms [1]. However, its main disadvantage is that it is difficult to obtain in many contexts as one needs to compute the cost $v(S)$ of all possible coalitions $S \subset N$. Even approximations of the Shapley value can be difficult to obtain. Most approximation algorithms in the literature, rely on sampling and evaluating many coalitions $S$ which can be computationally expensive when computing one cost $v(S)$ is NP-hard as in the Traveling Salesman game.

## 3 Shapley value and approximations

The Shapley value can be derived as the result of several expressions, each leading to a different approximation formula. The first formula defines the Shapley value in function of the marginal impact of the presence of a player $i$ in each subset of players $S$: $m(i,S) = v(S) - v(S \setminus \{i\})$. The Shapley value is then:

$$\phi(i) = \frac{1}{n!} \sum_{\pi \in \Pi(N)} m(i, P(i,\pi))$$

where $\Pi(N)$ is the set of permutations of $N$ and $P(i,\pi)$ is the set of predecessors of $i$ in permutation $\pi$. Following this formula, one can approximate the Shapley value by drawing a set $\Pi$ of random permutations of $N$ and averaging the marginal impacts of player $i$ over this set of permutations instead of $\Pi(N)$.

Another formula studied in [2] can be derived by rewriting the Shapley value as the result of a linear regression problem which can be solved with the following quadratic program:

$$\begin{aligned} \min_{\phi} \; & \sum_{S \subset N} \binom{n}{|S|}^{-1} \frac{n-1}{|S||N \setminus S|)} (v(S) - \phi(S))^2 \\ s.t. \; & \phi(N) = v(N) \end{aligned}$$

Following this other formula, one can approximate the Shapley value by drawing random subsets of $N$ with a probability proportional to $\binom{n}{|S|}^{-1} \frac{n-1}{|S||N \setminus S|)}$ leading to a family $\mathcal{S}$ of subsets. Then the following quadratic program is solved:

$$\min_{\phi} \sum_{S\in\mathcal{S}} (v(S) - \phi(S))^2$$
$$s.t.\ \phi(N) = v(N)$$

## 4 Our Shapley approximation

Our main contribution is the following cost sharing mechanism which approximates the Shapley value at a low computational cost. It depends only on the costs $v(\{i\} = m(i, \emptyset)$ of servicing a unique client $i$ (denoted $c_u^i$) and on the marginal cost $m(i, N)$ of client $i$ in the cycle servicing all players (denoted $c_m^i$). In this mechanism, the cost $\phi(i)$ of a player is:

$$\phi(i) = \ \lambda\ c_u^i + (1 - \lambda)\ c_m^i \text{ where } \lambda = \frac{v(N) - \sum_i c_m^i}{\sum_i c_u^i - c_m^i}$$

The coefficient $\lambda$ is chosen as the unique value which ensures $\sum_i \phi(i) = v(N)$. Rearranging the terms, this cost-sharing mechanism can be written as:

$$\phi(i) = \ c_m^i + (v(N) - \sum_i c_m^i) \frac{c_u^i - c_m^i}{\sum_i c_u^i - c_m^i}$$

In this form, the mechanism can be interpreted as assigning to each player its marginal cost $c_m^i$ and then assigning the cost remaining to be assigned proportionally to $c_u^i - c_m^i$. Note that computing all $c_m^i$ and $c_u^i$ requires only $2|N|+1$ evaluations of the cost function $v$ which is very low compared to all methods in the literature. For instance, in the approximator based on sampling permutations this would mean sampling only two permutations.

## 5 Numerical results

We now show with preliminary numerical results that our mechanism better approximates the Shapley value than the approximators in the literature when given low computational times. Our approximator will be denoted *marginal shapley* while the ones from the literature will be called *permutation shapley* and *regression shapley*. To compare the approximators, we use the following metric: denoting $\phi(i)$ the true Shapley value and $\hat{\phi}(i)$ the value of an approximator, the (average normalized) error made by the approximator is

$$\frac{1}{n} \sum_{i\in N} \frac{|\phi(i) - \hat{\phi}(i)|}{|\phi(i)|}$$

The different approximators are evaluated on TSP datasets created from instances from the literature [3]. Since we need to be able to compute the Shapley value to estimate the error of the approximators, we limited ourselves to instances with 10 customers. Our *marginal shapley* requires $2|N|+1$ evaluation of the cost function $v$ to be computed and we have given $6|N|$ evaluations to the other two methods because they rely on sampling. Despite this 3 to 1 computational advantage, Figure 1 shows that our approximator outclasses the ones from the literature on this benchmark.

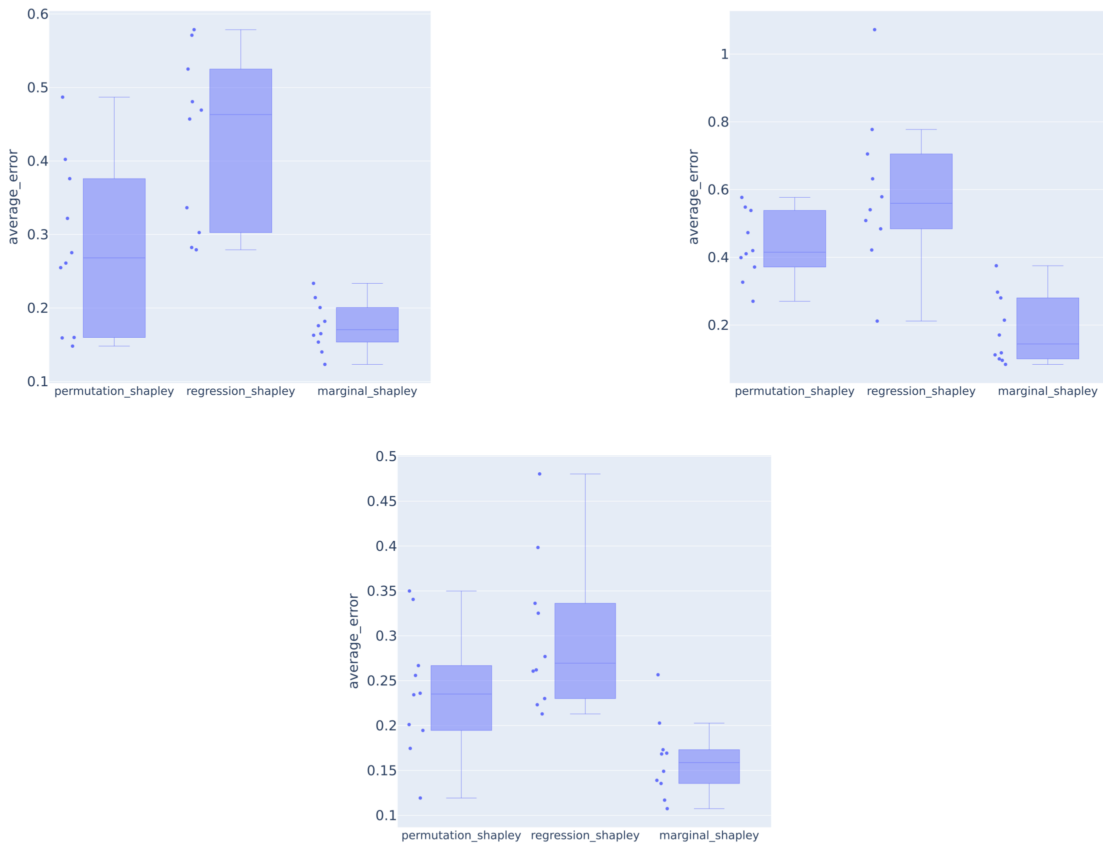



Figure 1: Error of the three approximators on three datasets from [3]

# 6 Acknowledgments

This work has been funded by the research and formation chair Responsible Retail supported by Toulouse INP, CNRS and One Stock.

# References

[1] Eyal Winter. The shapley value. *Handbook of game theory with economic applications*, 3: 2025–2054, 2002.

[2] Luis M Ruiz, Federico Valenciano, and Jose M Zarzuelo. The family of least square values for transferable utility games. *Games and Economic Behavior*, 24(1-2):109–130, 1998.

[3] Philippe Augerat. *Approche polyèdrale du problème de tournées de véhicules*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 1995.

# A Construction Matheuristic for Two-Tier Synchronized City Logistics

**Julia Lange**[*]
[*] Faculty of Business Studies and Economics,
University of Kaiserslautern-Landau, Kaiserslautern, Germany
Email: julia.lange@rptu.de

**Teodor Crainic**[**]
[**]School of Management, Université du Québec à Montréal,
and CIRRELT, Montréal, Canada

**Timo Gschwind**[*] **Walter Rei**[**]

## 1 Introduction

Sustainable city logistics planning focuses on multi-tier and multi-modal transportation with efficient consolidation and vehicle types suited for each tier. Integrating logistics services of different providers and shared transportation of multi-directional demand flows are key strategies to reduce congestion and create livable cities, see [1]. However, they are rarely addressed in the literature. This work introduces a detailed mathematical description of a day-before planning problem in two-tier multi-modal city logistics with on-time synchronization (2TM-CL-OS), where no storage exists at handover locations. While this enables the use of existing resources like supermarket parking lots in the distribution process, the requirement of delivery vehicles to meet for synchronized activities is a challenge.

The planning approach is based on two-tier scheduled service network design, see [2], where transportation services with routes, departure time windows, and capacities are given, and waiting time policies exist for customer and handover locations, the latter called satellites. Demands involve inbound (e2c), outbound (c2e), and innercity (c2c) commodity flows. The goal is to select services, including a schedule for each of them, and allocate the demands such that both operating costs and waiting times are minimized.

Since general-purpose solvers show a non-satisfactory performance in addressing the path-based mixed-integer programming (MIP) formulation of the 2TM-CL-OS, we present a construction matheuristic to evaluate this new model and discuss dependencies within the solution's structure. Thereby, we propose different approaches for variable fixing that yield promising results, and we provide conclusions on future research directions.

## 2 Mathematical Formulation

In the 2TM-CL-OS, a set of demands $d \in \mathcal{D}$ must be transported by large urban vehicles (urb) in outer-tier services $r \in \mathcal{R}$ and by small city freighters (cit) on inner-tier tours $k \in \mathcal{K}$ with handovers at satellites $z \in \mathcal{Z}$. Each demand features an origin and a destination at a customer location $i \in \mathcal{I}$ or an external zone, a volume, a handover time $h_d$, a time window, and an availability time if inbound. Each e2c and c2e demand is assigned to an inbound or an outbound connection $(r, k, z)$ representing its itinerary, see [3]. The c2c demands require an assigned tour only. Services and tours are selected for operation by binary variables $\rho_r^{\text{urb}}$ and $\rho_k^{\text{cit}}$ with corresponding fixed costs $c_r^{\text{urb}}$ and $c_k^{\text{cit}}$. A schedule with continuous starting times is determined for each selected service and tour, and thus, each demand itinerary, respecting demand time windows, travel times, waiting time allowances and synchronization requirements. The goal is to minimize the following objective function:

$$\sum_{r\in\mathcal{R}} c_r^{\text{urb}} \cdot \rho_r^{\text{urb}} + \sum_{k\in\mathcal{K}} c_k^{\text{cit}} \cdot \rho_k^{\text{cit}} + \sum_{r\in\mathcal{R}} \sum_{z\in\mathcal{Z}(r)} (\bar{\lambda}_{r,z}^{\text{urb}} - \lambda_{r,z}^{\text{urb}}) + \sum_{k\in\mathcal{K}} \left(\bar{\lambda}_k^{\text{in}} - \lambda_k^{\text{in}} + \bar{\lambda}_k^{\text{out}} - \lambda_k^{\text{out}}\right)$$
$$+ \frac{1}{100} \cdot \left[ \sum_{r\in\mathcal{R}} \sum_{z\in\mathcal{Z}(r)} \omega_{r,z}^{\text{urb}} + \sum_{k\in\mathcal{K}} (\omega_k^{\text{in}} + \omega_k^{\text{out}}) + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}(k)} \omega_{k,i}^{\text{cust}} \right]$$

The weighted sum combines the operating costs given in traveled time, the widths of the service intervals $[\lambda_{r,z}^{\text{urb}}, \bar{\lambda}_{r,z}^{\text{urb}}]$, $[\lambda_k^{\text{in}}, \bar{\lambda}_k^{\text{in}}]$, and $[\lambda_k^{\text{out}}, \bar{\lambda}_k^{\text{out}}]$ indicating handover periods as well as the waiting times of vehicles at satellites ($\omega_{r,z}^{\text{urb}}$, $\omega_k^{\text{in}}$ and $\omega_k^{\text{out}}$) and customer locations ($\omega_{k,i}^{\text{cust}}$). The set of constraints consists of four main groups:

1. *Service Network Design:* assigning demands, respecting vehicle capacities, linking assignment and selection
2. *Scheduling:* determining synchronized starting times of vehicle travels and demand handovers, respecting customer time windows and demand availability
3. *Waiting Times:* estimating vehicle waiting times, respecting waiting time allowances
4. *Satellite Capacities:* assignment and sequencing of vehicles at satellites, respecting satellite capacities

While the first two groups contain standard constraints, the waiting time estimation and the observation of the satellite capacities require tailored formulations. Two waiting time estimates are used to cover all cases of more than two vehicles meeting for handovers at the same satellite and time. Satellites are discretized into parking units for urban vehicles and city freighters, respectively, so that units can be assigned and vehicle appearances can be sequenced and timed appropriately. Since it is known that every e2c and c2e demand is handed over exactly once between one service and one tour, the mathematical model is enhanced by service interval-and selection-related valid inequalities.

## 3 Construction Matheuristic and Results

An evaluation of the mathematical description on reasonably sized instances is challenging, since general-purpose exact solvers unsurprisingly show a poor performance on the MIP model of the 2TM-CL-OS. Therefore, we propose and test a simple two-step construction matheuristic to create first insights on solving performance and dependencies. The core idea is to reduce the search space by fixing a subset of the binary selection variables for services and tours. The two-step procedure works as follows:

I. **Solve relaxed model.** Store pool of all objective-improving solutions.

II. **Solve full model.**
Fix values of selection variables based on the best solution or the pool from Step I.

The relaxed model consists of the objective function without service intervals, constraint groups 1 and 2 and simplified waiting time constraints (from group 3). Among other options tested, the following three **fixing rules** to exclude services and tours ($\rho_r^{\mathrm{urb}} = 0$ or $\rho_k^{\mathrm{cit}} = 0$) show interesting results:

- *FixUnusedRoutes:* Services and tours never selected in any solution of the solution pool are excluded.
- *FixRarelyUsedRoutes:* Services and tours rarely selected in the solutions of the pool are excluded.
- *FixPoorCombiRoutes:* Combinations of services and tours that rarely appear in high-quality solutions of the pool are excluded.

The computational experiments are conducted on randomly generated instances with 30 demands and varying numbers of satellites ($|\mathcal{Z}|$). Services are road- and rail-based, cargo bikes are used as city freighters, and waiting time allowances are equally distributed between no, moderate and long waiting permitted. All methods are implemented in Python 3.8 with Gurobi 10.0 as a general MIP solver. The time limit is 1 hour, where 10 and 50 min. are set for Step I and II, respectively.

While solving the full model stopped without any optimal solution and with a 53% median gap over all instances, the construction matheuristics show promising results. Table 1 summarizes the proportions of instances solved to optimality or proven infeasible (Opt/Inf), the median gap (Gap), and the median upper bound difference compared to the full model (Diff). Fixing unused routes reduces the optimality gap and produces solutions of better quality. However, none of the instances is solved to optimality, so that a further reduction of the search space seems reasonable. While excluding rarely used routes does this, it also shows a large amount of infeasible instances with 4 and 5 satellites due to a lack

Table 1: Computational results of construction matheuristics

| | FixUnusedRoutes | | | FixRarelyUsedRoutes | | | FixPoorCombiRoutes | | |
|---|---|---|---|---|---|---|---|---|---|
| $\lvert\mathcal{Z}\rvert$ | Opt/Inf | Gap | Diff | Opt/Inf | Gap | Diff | Opt/Inf | Gap | Diff |
| 2 | 0/0 | 25.6 | -1.1 | 100/0 | | 3.0 | 0/0 | 1.7 | -4.0 |
| 4 | 0/0 | 26.0 | -7.0 | 50/50 | | -4.3 | 50/0 | 17.3 | -5.0 |
| 5 | 0/0 | 26.8 | -1.5 | 33/33 | 0.3 | 6.2 | 33/0 | 11.6 | 4.4 |
| 10 | 0/0 | 26.1 | -3.8 | 100/0 | | 1.2 | 67/0 | 11.9 | 4.5 |

All entries except number of satellites given in percent (%).

in consideration of the two-tiered structure. This is overcome by excluding combinations of services and tours that rarely appear in high-quality solutions. Even if the amount of instances solved to optimality is decreased, the decrease in the median gap is remarkable compared to the full model, and the loss in solution quality is small.

## 4 Conclusion

The 2TM-CL-OS is a complex problem of significant importance for future city logistics systems. A mathematical formulation focusing on synchronization, waiting time estimation, and satellite capacity requirements is proposed. Results of simple construction matheuristics validate the accurate representation of structural and temporal requirements and demonstrate the applicability of the MIP model in the generation of near-optimal solutions for medium-size instances. This points to more elaborate heuristic procedures and decomposition approaches as promising future research directions to solve larger problems. In our ongoing work, we continue a broad experimental campaign to draw further methodological and managerial insights, and we aim to apply stochastic optimization to account for travel and handover time uncertainty.

## References

[1] T. G. Crainic, F. Errico, W. Rei and N. Ricciardi, "Integrating c2e and c2c traffic into city logistics planning", *Procedia - Social and Behavioral Sciences* 39, 47-60, 2012.

[2] P. Fontaine, T. G. Crainic, O. Jabali and W. Rei, "Scheduled service network design with resource management for two-tier multimodal city logistics", *European Journal of Operational Research* 294(2), 558-570, 2021.

[3] T. G. Crainic, N. Ricciardi and G. Storchi, "Models for Evaluating and PLanning City Logistics Systems", *Transportation Science* 43(4), 432-454, 2009.

# Exact Solution of the Single Picker Routing Problem with Scattered Storage

**Stefan Irnich**
Logistics Management,
Johannes Gutenberg University, Mainz, Germany
Email: irnich@uni-mainz.de

**Katrin Heßler**
Global Data and AI, F.LCA - Global Technology and Data
Schenker AG, Essen, Germany
Email: katrin.hessler@schenker.de

## 1 Introduction

Warehouse activities include receiving, storing, picking, packing, and shipping operations [7]. Excellent surveys introduce warehouse operations planning including storage assignment, warehouse layout planning, zoning, routing, and batching [1, 19]. In this work, we address picking operations in manual (non-automated) warehouses where pickers move through the warehouse in order to collect articles from the storage locations (picker-to-parts). de Koster et al. [4] highlight that more than 80% of all order-picking systems in Western Europe are low-level picker-to-parts picking systems. Order picking denotes the process of retrieving inventory items (articles) from their storage locations in response to specific customer requests [4, 12]. Manual order picking is certainly very labor-intensive, and the literature gives different estimations for the effort: Typically, 60% of all labor activities in the warehouse result from order picking, and its cost can be estimated to be as much as 55% of the total warehouse operating expense [5, 18]. Frazelle [6] estimates that order picking contributes to up to 50% of the total warehouse operating costs. These figures explain why research on order picking operations is extensive and of high practical relevance.

In its pure form, the *single picker routing problem* (SPRP) seeks a minimum-length picker tour given the warehouse layout and the pick positions from where articles must be collected. The SPRP can be considered solved: On the one hand, the seminal work of Ratliff and Rosenthal [16] assuming a single-block parallel-aisle warehouse shows that a minimum-length picker tour can be computed with dynamic programming in linear time [9]. On the other hand, the SPRP is practically well-solved with routing policies that are rule-based heuristics such as traversal (a.k.a. S-shape), midpoint, largest gap [8], return, composite [15]. The application of heuristic routing policies is well justified in settings where pickers cannot perform all types of optimal tours, which can be complicated, counter-intuitive, and difficult to memorize. Instead, pickers perform tours defined by some simple rules. A little bit more involved is the routing when the policy combined is applied [17]. Both exact and heuristic techniques have been extended into many different directions, e.g., to other warehouse layouts [2, 14, 17], non-identical start and end points [11, 13], and multiple end points [3].

When one or several articles are pickable from more than one pick position, the warehouse operates as a *scattered storage* warehouse or *mixed shelves* warehouse. Recent works [1, 20, 21] stress that scattered storage is predominant in modern e-commerce warehouses of companies like Amazon or Zalando. The main advantage of this storage strategy is "that items of demanded SKUs are found close by irrespective of the position within the warehouse [so that] the distance to be covered for order picking is reduced this way" [20, p. 139]. The *SPRP with scattered storage* (SPRP-SS) is an integrated operational planning problem characterized by two levels of decisions. The picker routing constitutes the lower-level decision, i.e., the lengths of different picker tours must be computed to evaluate higher-level decisions. The higher-level decision is, for each requested article, the selection of one or several storage positions from where a sufficient number of this article can be collected. If the selection has been made, the resulting picker routing problem is the SPRP. However, both levels are interdependent and the SPRP-SS is known to be NP-hard [20], even if optimal routing is replaced by one of the above simple heuristic routing policies [10].

## 2 Contributions

The focus of our work is on algorithmic improvements for exactly solving the SPRP-SS. The effective solution algorithm we propose relies on the following underlying modeling approach: For the SPRP, every feasible picker tour is a path in the state space of the dynamic-programming approach of Ratliff and Rosenthal [16], and vice versa. The underlying assumption is that all pick positions are known and given. For the SPRP-SS, however, since the picker routing problem is a subproblem of the integrated operational planning problem, the fulfillment of the given pick list creates a new situation in which the *selection* of pick positions becomes essential. Our leading idea is to extend the state space of Ratliff and Rosenthal so that the selection aspect is fully modeled. In the extended state space, every feasible picker tour is still a path. However, not every path fulfills that the requested number of articles can be collected. This requirement to make consistent decisions regrading demand covering can be modeled with additional constraints in a shortest-path problem, which can no longer be solved with dynamic programming. We show that this model can be solved well as a binary program with the help of established *mixed-integer (linear) programming* (MIP) solvers.

## References


[1] Nils Boysen, René de Koster, and Felix Weidinger. Warehousing in the e-commerce era: A survey. *European Journal of Operational Research*, 277(2):396–411, 2019. doi: 10.1016/j.ejor.2018.08.023.

[2] Melh Çelk and Haldun Süral. Order picking under random and turnover-based storage policies in fishbone aisle warehouses. *IIE Transactions*, 46(3):283–300, 2014. doi: 10.1080/0740817x.2013.768871.

[3] René de Koster and Edo van der Poort. Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions. *IIE Transactions*, 30(5):469–480, 1998. doi: 10.1023/a:1007599307171.

[4] René de Koster, Tho Le-Duc, and Kees Jan Roodbergen. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2):481–501, 2007. doi: 10.1016/j.ejor.2006.07.009.

[5] Jolyon Drury. Towards more efficient order picking. Technical report, The Institute of Materials Management, Cranfield, UK, 1988.

[6] E. Frazelle. *World-Class Warehousing and Material Handling*. McGraw-Hill, New York, 2002.

[7] Jinxiang Gu, Marc Goetschalckx, and Leon F. McGinnis. Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177 (1):1–21, 2007. doi: 10.1016/j.ejor.2006.02.025.

[8] Randolph W. Hall. Distance approximations for routing manual pickers in a warehouse. *IIE Transactions*, 25(4):76–87, 1993. doi: 10.1080/07408179308964306.

[9] Katrin Heßler and Stefan Irnich. A note on the linearity of Ratliff and Rosenthal's algorithm for optimal picker routing. *Operations Research Letters*, 50(2):155–159, 2022. doi: 10.1016/j.orl.2022.01.014.

[10] Laura Korbacher, Katrin Heßler, and Stefan Irnich. The single picker routing problem with scattered storage: Modeling and evaluation of routing and storage policies. Technical Report LM-2023-01, Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Mainz, Germany, 2023. URL https://logistik.bwl.uni-mainz.de/files/2023/02/LM-2023-01.pdf.

[11] Maximilian Löffler, Nils Boysen, and Michael Schneider. Picker routing in AGV-assisted order picking systems. *INFORMS Journal on Computing*, 34(1):440–462, January 2022. doi: 10.1287/ijoc.2021.1060.

[12] Makusee Masae, Christoph H. Glock, and Eric H. Grosse. Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics*, 224:107564, 2020. doi: 10.1016/j.ijpe.2019.107564.

[13] Makusee Masae, Christoph H. Glock, and Panupong Vichitkunakorn. Optimal order picker routing in a conventional warehouse with two blocks and arbitrary starting and ending points of a tour. *International Journal of Production Research*, 58(17): 5337–5358, 2020. doi: 10.1080/00207543.2020.1724342.

[14] Ömer Öztürkoğlu, Kevin R. Gue, and Russell D. Meller. Optimal unit-load warehouse designs for single-command operations. *IIE Transactions*, 44(6):459–475, 2012. doi: 10.1080/0740817x.2011.636793.

[15] Charles G. Petersen. An evaluation of order picking routeing policies. *International Journal of Operations & Production Management*, 17(11):1098–1111, 1997. doi: 10.1108/01443579710177860.

[16] H. Donald Ratliff and Arnon S. Rosenthal. Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research*, 31(3):507–521, 1983. doi: 10.1287/opre.31.3.507.

[17] Kees Jan Roodbergen and RenÉ de Koster. Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39(9):1865–1883, jan 2001. doi: 10.1080/00207540110028128.

[18] J.A. Tompkins, J.A. White, Y.A. Bozer, E.H. Frazelle, and J.M.A. Tanchoco. *Facilities Planning*. John Wiley & Sons, Hoboken, NJ, 2003.

[19] Teun van Gils, Katrien Ramaekers, An Caris, and René B.M. de Koster. Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research*, 267(1):1–15, 2018. doi: 10.1016/j.ejor.2017.09.002.

[20] Felix Weidinger. Picker routing in rectangular mixed shelves warehouses. *Computers & Operations Research*, 95:139–150, 2018. doi: 10.1016/j.cor.2018.03.012.

[21] Felix Weidinger, Nils Boysen, and Michael Schneider. Picker routing in the mixed-shelves warehouses of e-commerce retailers. *European Journal of Operational Research*, 274(2):501–515, 2019. doi: 10.1016/j.ejor.2018.10.021.

# A data-driven location-routing optimization for sustainable medical waste management

**Ginger Y. Ke**
Faculty of Business Administration,
Memorial University of Newfoundland, St. John's, Canada
Email: gke@mun.ca

**Jiahong Zhao** **Jianfeng Chen**
School of Civil and Transportation Engineering,
Guangdong University of Technology, Guangzhou, China

## 1 Introduction

COVID-19 has inspired a lot of effort in research on uncertainties. As the pandemic is coming to an end, the quantity, composition, and spatial distribution of waste have been altered, with relatively more generations from smaller sources. The uncertainties are also shifted to quotidian operations, especially emergency services that are directly related to the reduction of infectious risks. Moreover, the expansion of the medical waste management system has a notable impact in this context due to the increasing number of facilities and vehicles, which consequentially lead to a series of environmental issues, such as greenhouse gas emission, natural energy consumption, and global climate change [1]. As a result, challenges posed to the urban waste systems persist but call for a renewed focus on ensuring sustainable waste management considering regular daily waste collections and emergency response operations. Motivated by the new developments in the post-pandemic era, we herein propose a bi-objective chance-constrained optimization model aiming to construct a cost-efficient and environmentally-friendly medical waste management system with proper facility locations and routing plans.

## 2 Mathematical Model

### 2.1 Objective functions

Two objectives, minimization of cost and risk, are considered in our location-routing problem. To that end, the total cost includes the expenditure on facility locations and operations, fuel charges for waste collection, cost of decarbonization, and procurement expenses of vehicles. The total risk, on the other hand, includes the risks at each disposal facility and on each vehicle route under uncertain emergency response time.

## 2.2 An integrated dynamic pollution-population (DPP) risk assessment

The system risk evaluates harmful impacts on the surrounding population caused by possible incidents at disposal facilities (site risk) or on transportation paths (edge risk). Our proposed risk assessment model jointly considers multiple factors, including the source (pollution of leakage, i.e., the amount of waste on-site or en route), consequence (exposed population within a certain radius), and dynamics (risk amplification due to delays in emergency response).

## 2.3 Fuzzy chance-constrained emergency response time

In practice, the emergency response time is normally uncertain due to the unstable traffic conditions in the urban road network. Two sets of chance constraints are enclosed to ensure that the probabilities of meeting the required response time $\tau$ to sites and edges are higher than a predetermined threshold $\theta$. Take the facility node $i \in \mathcal{F}$ for example, the chance constraint is $\Pr\left\{\tilde{T}_{hi} \leq \tau\right\} \geq \theta$, where $\tilde{T}_{hi}$ is the uncertain response times for hazmat $h \in \mathcal{H}$. Taking the idea of [2], the emergency response time is considered uncertain in the form of fuzzy numbers in the trapezoidal possibility distributions, and then the necessity measure is applied to handle the above chance constraints.

## 2.4 A modified two commodity flow formulation

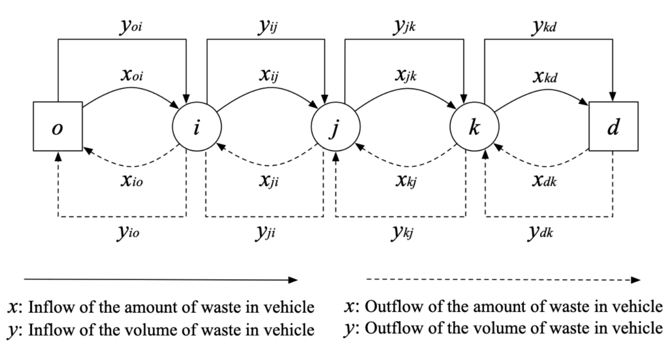



Figure 1: Flows of two-commodity flow

The two-commodity flow model uses copies of the origin nodes to transfer tours to routes, avoiding the sub-tour elimination in the traditional 3-index vehicle routing formulations, and hence discloses a higher convergence rate and better computational efficiency. In this work, we make two modifications to this model so to better serve our research purpose (see Figure 1 for detailed flows).

First, following a UNEP joint report [3], any compression and squeezing are prohibited during the medical waste collection, and so it is crucial to ensure the volume restriction, which is set to be 2/3 of the full vehicle volume. Secondly, a precise calculation of cost (fuel and decarbonization) and risk (edge pollution source) requires the knowledge of vehicle loads when leaving generation nodes, but the regular two-commodity flow model fails to achieve this. We introduce two additional decision variables and three constraint sets to compute the transportation direction on each edge and the accumulative vehicle load.

# 3 A Data-Driven Bi-Objective Solution Procedure

## 3.1 Traffic prediction by Back Propagation Neural Network (BPNN)

To evaluate the unstable emergency response time, we herein employ the real-time congestion index provided by BaiduMap. We collect the historical data on each edge for at least four consecutive working days, and predict the traffic congestion index by using BPNN. Then, after a comprehensive statistical analysis, the four prominent points of uncertain emergency response time between the emergency facility and edges can be obtained.

## 3.2 An NN-NSGA-II algorithm

According to the unique characteristics of our proposed model, we revised the original NSGA-II with a nearest-neighbor decoding procedure (NN-NSGA-II).

In more detail, we define each chromosome as a two-row matrix, respectively containing a permutation of medical waste generation nodes and a permutation of disposal facility candidates. To start, the facility nearest to the first generation node is chosen to be the origin of a route. From this facility, the route goes through the first node, and then moves to the nearest node that is covered by emergency services. The process continues until either the facility or the vehicle capacity limit is violated, and thus we obtain a nearest neighbor-based vehicle route. Proceeding to the first unassigned generation node for the next routes until all nodes are exhausted. The total cost and risk can be computed given the final location-routing plan.

Applying non-dominated sorting, and using crowding distance for fitness evaluation, chromosomes are selected with tournament selection and operated with precedence operations crossover and reverse variation. Finally, the Taguchi design approach is adopted to seek the best parameter values.

# 4 A Real-World Case Study

A series of experiments are conducted based on the real network in Shanghai, China. NN-NSGA-II efficiently derives 58 non-dominated solutions, where the recommended plan was selected via the linear programming technique for multidimensional analysis of preference proposed by [4].

Our analyses show that the optimal waste management location-routing system obtained from our model can sufficiently enlarge the system capacity and reduce both risks, without inducing extra cost. With the use of BPNN, the number of facilities decreases by 20%, and the network emergency coverage can be increased by 22%. A higher emergency confidence level leads to lower edge coverage and higher risk values. In the meantime, edges that can be selected for routing are more limited, which results in longer travel distances and larger costs. Through a comparison with two widely applied risk methods, DPP can achieve higher equity through the degree of confidence in emergency services, and ensures that every edge in the network can be properly covered if any incident occurs.

## 5 Conclusion

Recognizing the important role of emergency response, we develop a bi-objective chance-constrained optimization model to seek the best waste processing plan such that both the cost and risk are simultaneously minimized. Through a data-driven BPNN approach, the dynamics of traffic flow and congestion are predicted and fed into a set of fuzzy chance constraints to ensure the effective coverage of the emergency response. With a modified NSGA-II method, numerical experiments based on a real-world network are employed to test the proposed model. Managerial insights are revealed to facilitate practical decision making in urban medical waste management.

## References


[1] Ghannadpour, S. F., Zandieh, F., and Esmaeili, F. "Optimizing triple bottom-line objectives for sustainable health-care waste collection and routing by a self-adaptive evolutionary algorithm: A case study from Tehran province in Iran". *Journal of Cleaner Production*, 287 , 125010, 2021.

[2] Pishvaee, M., Razmi, J., and Torabi, S. "Robust possibilistic programming for socially responsible supply chain network design: A new approach". *Fuzzy Sets and Systems*, 206 , 1–20, 2012.

[3] Institute for Global Environmental Strategies. "Waste management during the COVID-19 pandemic: From response to recovery". URL: https://www.unep.org/resources/report/waste-management-during-covid-19-pandemic-response-recovery. 2020.

[4] Wang, Z. and Rangaiah, G. P. "Application and analysis of methods for selecting an optimal solution from the pareto-optimal front obtained by multiobjective optimization". *Industrial & Engineering Chemistry Research*, 56, 560–574, 2017.

# The Impact of Service Levels in Stochastic Production Routing with Adaptive Routing

**Ali Kermani, Jean-François Cordeau, Raf Jans**
Department of Logistics and Operations Management
HEC Montréal and CIRRELT, Canada
Email: ali.kermani@hec.ca, jean-francois.cordeau@hec.ca, raf.jans@hec.ca

## 1 Introduction

The Production Routing Problem (PRP) encompasses both the Lot Sizing Problem (LSP) and the Inventory Routing Problem (IRP) to enhance supply chain integration and reduce costs associated with poor coordination. The PRP typically involves decisions about the production, inventory management, and delivery of a single product made in a plant and distributed to multiple customers over a finite and discrete time horizon (see, e.g., Adulyasak et al. [2] and Hrabec et al. [4] for an overview of the relevant literature).

Considering demand uncertainty can help companies to minimize the costs associated with stockouts and customer dissatisfaction. Adulyasak et al. [1] investigate both two-stage and multi-stage formulations of the stochastic PRP (SPRP) with demand uncertainty. In their proposed formulation, routing decisions are made in the first stage, limiting flexibility by requiring customer visits in all scenarios, regardless of whether deliveries are needed. In Kermani et al. [5], we study the potential cost savings associated with flexible routing in a two-stage SPRP under demand uncertainty where the routing decisions are second-stage variables. In the present paper, we consider service level constraints, which are an effective strategy to deal with uncertainty, especially when satisfying demand from third-party suppliers or imposing penalty costs for unmet demand is not a viable option. The advantages and limitations of different service levels in the LSP are discussed in [3, 6]. Surprisingly, this aspect remains unexplored in the PRP, despite its relevance to numerous real-world situations. More precisely, we model four distinct service levels (referred to as $\alpha$, $\beta$, $\gamma$, and $\delta$), each designed to address different metrics based on specific assumptions [6]. In our study, we introduce a two-stage SPRP with adaptive routing (SPRP-AR) and service level constraints to address all these service levels and compare their effects on the SPRP.

## 2 Problem Definition

In this problem, a single product is produced and delivered to a set of customers over a finite planning horizon. We generate a finite set of scenarios to address demand uncertainty based on a given nominal demand and a discrete uniform distribution using Monte Carlo simulation. We assume a homogeneous fleet of capacitated vehicles with a defined capacity that pick up the product from the plant and deliver it to customers in each period, returning to the plant at the end of their tour. A fixed setup cost is considered whenever production takes place, alongside a unit production cost per item produced. Additionally, we account for unit holding costs for goods carried to the next period, as well as routing costs associated with the edges traversed by vehicles.

The objective is to minimize the total cost, consisting of the first-stage setup costs and the expected value of the second-stage production, holding, and routing costs. The production quantity is a second-stage variable, limited by the plant's production capacity. This variable can be adapted to the realized demand, while setup decisions are the only first-stage decisions in the problem, made at the beginning of the planning horizon. The amount of products that can be held at each node for future demands is also limited by the maximum inventory level of the node. Additionally, we bound the delivery amount for each vehicle by its capacity and prevent split deliveries to customers. We consider a distinct formulation for each service level, addressing associated conditions (see, e.g., [3, 6]). For the $\alpha$ service level, we bound the probability of stockouts for a customer in a period. For the $\beta$ service level, we impose restrictions on the expected backorder divided by the average demand. In the case of the $\gamma$ service level, we enforce a predetermined ratio of expected backlog to expected demand. Lastly, for the $\delta$ service level, we restrict the expected backlog divided by the maximum expected backlog.

## 3 Solution Algorithm

We introduce an iterative matheuristic algorithm (IMH) designed to solve the SPRP-AR with service level constraints. The core principle of this algorithm lies in the decomposition of the original problem into more manageable subproblems. The goal of the first phase is to swiftly generate setup decisions. To achieve this, we employ a two-level LSP that includes the production plant and a single aggregated customer. At this stage, we compute the transportation cost as the cost associated with the Traveling Salesman Problem (TSP) across all nodes and apply an aggregate delivery approach to the vehicles. All binary variables, excluding the setup decisions, are relaxed at this stage, as our sole aim is to identify setup decisions. Moving on to the second phase, we maintain the setup decisions established in the first phase and proceed to solve an SPRP considering an aggregate delivery quantity for the vehicles. The goal of this phase is to determine if the setup decisions from the first stage can lead to a feasible solution for the second-stage variables. Additionally, we incorporate service level constraints to align our solution with the predefined service levels.

In the third phase, we solve a Restricted Inventory Routing Problem (RIRP) for each scenario. Here, we continue with fixed setup decisions and impose an upper bound (UB) on backlogs to ensure service level feasibility. We allow flexibility in inventory, delivery, and production quantities to refine our solutions further. Upon completing these phases, we enter an intensification phase, where we take into account approximate visit costs. We introduce visit variables to the second phase and iterate over the new problem, followed by the third phase, continually updating the approximate visit costs to explore potential enhancements. If no further improvements are obtained or the intensification phase reaches its stopping criteria, we introduce a local branching constraint in the first phase. This initiates a repeat of the procedure to generate a new setup decision and continue the process until a stopping criterion is met.

## 4 Results

We conducted computational experiments using the dataset introduced in [1] with high transportation costs to compare the impact of different service levels. Specifically, we present results for the smaller dataset that consists of 5 to 30 customers (with intervals of 5), 3 periods, 1 to 3 vehicles, and 100 scenarios. We analyze 6 different values for each service level, ranging from 70% to 95% with a 5% interval. This results in 108 instances for each type of service level and 432 instances in total. Figure 1a presents the cost comparison of the objective function for different service levels with different target values. We can observe that the least strict service level is the $\delta$ service level, followed by the $\beta$ and $\gamma$ service levels. However, the $\alpha$ service level is an event-based service level which is mostly close to the $\beta$ service level in terms of objective function value. It is also obvious that by increasing the target value of the service level, the objective function cost increases, while the difference of the objective function between different service levels tends to decrease. In Figure 1b, Figure 1c, and Figure 1d, we provide details of the different parts of the recourse variables of the objective function, including average production, inventory, and transportation costs. One can observe that all parts follow the same trend, while the transportation cost tends to vary more, especially for the $\alpha$ and $\beta$ service levels.

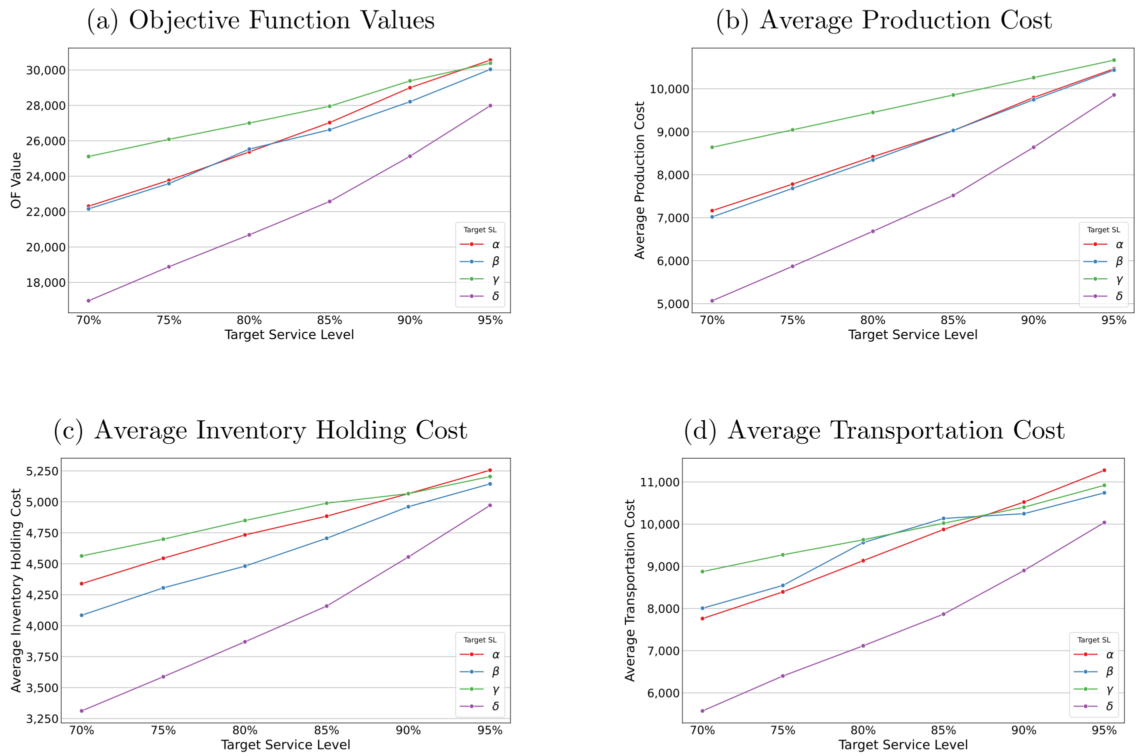



Figure 1: Cost comparison for different service levels

# References


[1] Y. Adulyasak, J.-F. Cordeau, and R. Jans, "Benders Decomposition for Production Routing Under Demand Uncertainty", *Operations Research* 63(4), 851-867, 2015.

[2] Y. Adulyasak, J.-F. Cordeau, and R. Jans, "The production routing problem: A review of formulations and solution algorithms", *Computers & Operations Research* 55, 141-152, 2015.

[3] S. Helber, F. Sahling, K. Schimmelpfeng, "Dynamic capacitated lot sizing with random demand and dynamic safety stocks", *OR Spectrum* 35(1), 75–105, 2013.

[4] D. Hrabec, L.M. Hvattum, A. Hoff, "The value of integrated planning for production, inventory, and routing decisions: A systematic review and meta-analysis", *International Journal of Production Economics* 248, 108468, 2022.

[5] A. Kermani, J.-F. Cordeau, and R. Jans, "A Progressive Hedging-based Matheuristic for the Stochastic Production Routing Problem with Adaptive Routing", Submitted to *Computers & Operations Research*, 2023.

[6] N. Sereshti, Y. Adulyasak, and R. Jans, "The value of aggregate service levels in stochastic lot sizing problems", *Omega* 102, 102335, 2021.

# Scheduled Service Network Design with Packing Considerations

**Sara Khodaparasti**
DIGEP, Polytechnic University of Turin, Turin, Italy
Email: sara.khodaparasti@polito.it

**Maria Elena Bruni and Teodor Gabriel Crainic**
CIRRELT & DIMEG, University of Calabria, Cosenza, Italy
School of Management & CIRRELT, UQAM, Montreal, Canada
**Guido Perboli and Walter Rei**
CIRRELT & DIGEP, Polytechnic University of Turin, Turin, Italy
School of Management & CIRRELT, UQAM, Montreal, Canada

## 1 Introduction

Tactical planning for consolidation-based carriers is a complex problem, usually addressed by the *Service Network Design* (*SND*) methodology. The SND literature generally disregards some crucial issues, however, e.g., how cargo is loaded into transportation vehicles or storage facilities - called *capacity units* in the following -, and the selection of the proper number and type of capacity units for each selected service within the designed network.

Such considerations may have a substantial impact on the system performance and should be carefully tackled to avoid underestimating costs or generating infeasible itineraries exceeding the capacity of the capacity units of the selected services [1]. To better represent the physical and operational attributes of capacity units and the utilization demand flows make of that capacity, one must address the challenges of replacing the standard aggregated flow capacity constraints with more realistic packing ones [1].

There are very few papers in the SND literature explicitly integrating packing constraints into SND models. [3] present new formulations for the Scheduled SND problem, where shipments cannot be split, while consolidation is desirable to reduce the number of homogeneous vehicles used when multiple shipments dispatch simultaneously on the same direct service. Given the somewhat simple direct-service structure, the set of commodity clusters which can be associated to each physical link (i.e., travel together on it) may be generated a priori. The packing and the SND-related decisions may then be addressed separately. [2] combine two classical problems, transportation and variable size and cost bin packing, aiming to ship multiple types of commodities on different types of vehicles moving between the supply and demand nodes of a bipartite network, while minimizing the transportation and resource-acquisition cost.

We present a unified problem setting, the *Service Network Design Problem with Packing Considerations* (SNDPC), addressing simultaneously decisions on the selection of scheduled services (on a general time-space network), the selection of the type and number of capacity units to load and move the origin-destination demands, the assignment of demand flows to the loading units, and the construction of the demand itineraries within the selected service network.

Differently from [3], Scheduled SND and packing decisions are addressed simultaneously. The problem setting is general and the freight is moved via different itineraries and using a heterogeneous fleet of vehicles. Combining two $NP$-hard combinatorial problems, network design and packing, does not make the problem easier to address and requires careful investigation.

## 2 Problem description and mathematical formulation

Consider a physical network $\mathcal{G}^{PH} = (\mathcal{N}^{PH}, \mathcal{A}^{PH})$, where $\mathcal{N}^{PH}$ is the set of nodes (representing transfer and consolidation terminals, which include the origins and destinations of demand), and $\mathcal{A}^{PH}$ is the set of links (e.g., road, rail, and river) joining these nodes.

Each potential *service* $\sigma \in \Sigma$ is defined by a route in $\mathcal{G}^{PH}$ linking its origin $o(\sigma)$ to its destination $d(\sigma)$, without intermediate stops, as well as a schedule giving the *departure* from origin and *arrival* at destination times, $\alpha(\sigma)$ and $\beta(\sigma)$, respectively. The service is composed of a number of capacity units, or *bins*, to be selected within a particular set $J_\sigma$, where $J_\sigma \cap J_{\sigma'} = \emptyset, \forall \sigma \neq \sigma' \in \Sigma$; $J = \cup_{\sigma \in \Sigma} J_\sigma$. The bins may represent containers or vehicles, or any other transportation medium, of different types. Each bin type $\pi \in \Pi$ is characterized by a capacity $Q_\pi$ and a fixed selection/usage cost $c_\pi$. $\Pi$ stands for the set of bin types, and $\phi(j) \in \Pi$ gives the type of bin $j \in J$. Each service is characterized by a maximum total number of bins, $U_\sigma$, and maximum number of bins of type $\pi$, $N_{\pi\sigma}$, which may be assigned to it, a fixed selection cost $f_\sigma$, and unit bin cost $c_{\phi(j)}$, $j \in J_\sigma$, including the loading, unloading, and transportation costs of freight within the bin on the service.

Each *demand* $k \in \mathcal{K}$ represents a request to transport a set of *items*, $I(k)$ with $I = \cup_{k \in \mathcal{K}} I(k)$, from its origin $o(k)$ to its destination $d(k)$. The items are available at time $\alpha(k)$ and need to be delivered to the final destination at time $\beta(k)$. Each item $i \in I(k)$ is characterized by a size $v_i$ (expressed in the same unit as the bin capacity), the size of the demand, $d_k$, being the summation of the size of its items. The demand may be split among different services, or put into different bins of the same service, as long as the temporal requirements are satisfied. In all cases, items arriving at a terminal different from their destination are unloaded from the bins of the preceding service, eventually held at the terminal for a while at a unit holding cost $h_k$, and then loaded into different bins associated with different departing services, continuing their journey to the final destination. To design the service network, it is therefore necessary not only to select the services to be operated, but also to determine the assignment of items to bins, which adds another layer of complexity to the problem. Hence, the main goal of the *SNDPC* is to select a set of services and the capacity units of various types to be associated to each service to satisfy the demand at minimum cost.

We model the *SNDPC* on a time-space network, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, built by extending the physical network $\mathcal{G}^{PH}$ along the dimension of time for the fixed duration of the schedule length, discretized into *time periods* $t \in \mathcal{T}$ of equal length. Operations at terminals in different periods are modeled with different nodes of the form $(n, t) \in \mathcal{N}$. There are two types of arcs in $\mathcal{A}$. A *service arc*, joining nodes $(n, t)$ and $(n', t')$, models the operation of a (single-leg) service between its origin $o(\sigma) = n$ and destination $d(\sigma) = n'$, starting at time $\alpha(\sigma) = t$ and arriving at time $\beta(\sigma) = t'$. A *holding arc*, joining nodes $(n, t)$ and $(n, t + 1)$, models the possibility of holding items at node $n$ from period $t$ to $t + 1$. $\mathcal{A}^{\Sigma}$ and $\mathcal{A}^{H}$ stand for the sets of service and holding arcs, respectively, with $\mathcal{A} = \mathcal{A}^{\Sigma} \cup \mathcal{A}^{H}$. Note that, one has $\mathcal{A}^{\Sigma} = \Sigma$ for the single-leg case.

We consider the following sets of decision variables: $y_\sigma \in \{0, 1\}, \sigma \in \Sigma$, for the selection of service $\sigma$; $z_j \in \{0, 1\}, j \in J_\sigma$, $\sigma \in \Sigma$, selects or not bin $j$ of service $\sigma$; $x^i_{aj} \in \{0,\, 1\}, a \in \mathcal{A}^{\Sigma}, j \in J_{\sigma_a}, i \in I$, represents the possible assignment of item $i$ to bin $j$ of service $\sigma$ (arc $a$); $w^i_a \in \{0,\, 1\}, a \in \mathcal{A}^{H}$, $i \in I$, indicates if item $i$ is held on arc $a$. The *SNDPC* model:

$$\text{Minimize} \sum_{\sigma\in\Sigma} f_\sigma y_\sigma + \sum_{j\in J} c_{\phi(j)} z_j + \sum_{a\in\mathcal{A}^H} \sum_{k\in\mathcal{K}} h_k \big(\sum_{i\in I(k)} w^i_a\big) \tag{1}$$

$$\text{s.t.} \quad \sum_{a\in\mathcal{A}^+_{(n,t)}} \sum_{j\in J_{\sigma_a}} x^i_{aj} + \sum_{a\in\mathcal{A}^+_{(n,t)}} w^i_a - \Big( \sum_{a\in\mathcal{A}^-_{(n,t)}} \sum_{j\in J_{\sigma_a}} x^i_{aj} + \sum_{a\in\mathcal{A}^-_{(n,t)}} w^i_a \Big)$$

$$= \begin{cases} 1, & \text{if } (n,t) = \big(o(k), \alpha(k)\big), \\ -1, & \text{if } (n,t) = \big(d(k), \beta(k)\big), \quad \forall (n,t) \in \mathcal{N}, \forall k \in \mathcal{K}, \forall i \in I(k) \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

$$\sum_{i\in I} v_i x^i_{aj} \leq Q_{\phi(j)} z_j, \forall a \in \mathcal{A}^{\Sigma}, \forall j \in J_{\sigma_a} \tag{3}$$

$$\sum_{j\in J_\sigma} z_j \leq U_\sigma y_\sigma, \forall \sigma \in \Sigma \tag{4}$$

$$\sum_{\substack{j\in J_\sigma \\ \phi(j)=\pi}} z_j \leq N_{\pi\sigma}, \forall \sigma \in \Sigma, \forall \pi \in \Pi \tag{5}$$

$$y_\sigma \in \{0,1\}, \forall \sigma \in \Sigma,\; z_j \in \{0,1\}, \forall j \in J_\sigma, \forall \sigma \in \Sigma \tag{6}$$

$$x^i_{aj} \in \{0,1\}, \forall a \in \mathcal{A}^{\Sigma}, \forall j \in J_{\sigma_a}, \forall i \in I,\; w^i_a \in \{0,1\}, \forall a \in \mathcal{A}^H, \forall i \in I \tag{7}$$

where $\mathcal{A}^+_{(n,t)} = \{a = \big((n'', t''), (n', t')\big) \in \mathcal{A} | n'' = n,\, t'' = t\}$ and $\mathcal{A}^-_{(n,t)} = \{a = \big((n', t'), (n'', t'')\big) \in \mathcal{A} | n'' = n,\, t'' = t\}$, for each $(n, t) \in \mathcal{N}$ and $\sigma_a \in \Sigma$ denotes the service associated with arc $a \in \mathcal{A}^{\Sigma}$. The objective function (1) minimizes the total cost of the selected services, the bins used, and the holding of items at terminals. Constraints (2) ensure that each item is routed from its origin node to its destination node, respecting the temporal constraints. Constraints (3) enforce a feasible assignment of items to bins, respecting the bin capacity. Constraints (4) represent the limits on the global capacity of each service. Constraints (5) limit the total number of bins of each type for each service. Finally, constraints (6)-(7) express the nature of the variables.

## 3 Conference Presentation

We will present a comprehensive view of the topic, identify issues and challenges of different variants of the problem, and discuss mathematical formulations.

We will also present the numerical results, obtained using an off-the-shelf commercial solver, on two sets of instances built on networks from SNDLib (http://sndlib.zib.de) and the SND literature [4]. The sensitivity analysis will focus on the performance of the solver (and the state-of-the-art enumeration algorithm it offers) with respect to the instance dimension and various settings of the problem parameters (e.g., schedule length, number and characteristics of capacity units, the flexibility of the demand due dates, the split/no split demand requirements), as well as to the impact of these variations on the structure of the solutions, in comparison with the results produced by the Scheduled SND formulation with a classic modelling of the service-capacity restrictions.

We will conclude with an overview of possible avenues for the design of solution methods tailored for the particular problem structure and able to address large instances.

## References


[1] Bruni M.E., Crainic T.G., Perboli G., *Bin Packing Methodologies for Capacity Planning in Freight Transportation and Logistics*, In *Contributions to Combinatorial Optimization and Applications*, Crainic T.G., Frangioni A., Gendreau M. (Eds.), Springer, 2023, forthcoming.

[2] Flamand T., Iori M., Haouari M., The transportation problem with packing constraints, *Computers & Operations Research* 157:106278, 2023.

[3] Hewitt M., Lehuédé F., New Formulations for the Scheduled Service Network Design Problem, *Transportation Research Part B: Methodological*, 172:117-33, 2023.

[4] Lanza G., Crainic T.G., Rei W., Ricciardi N., Scheduled service network design with quality targets and stochastic travel times, *European Journal of Operational Research*, 288(1):30-46, 2021.

# Online stochastic optimization for real-time transfer synchronization in public transit networks

**Laura Kolcheva**
Department of Mathematics and Industrial Engineering,
Polytechnique Montréal, CIRRELT & GERAD, Montreal, Canada

**Antoine Legrain**
Department of Mathematics and Industrial Engineering,
Polytechnique Montréal, CIRRELT & GERAD, Montreal, Canada
Email: antoine.legrain@polymtl.ca

**Martin Trépanier**
Department of Mathematics and Industrial Engineering,
Polytechnique Montréal, CIRRELT, Montreal, Canada

## 1 Introduction

Public transportation (PT) systems are increasingly important in the context of urban growth, traffic congestion and sustainable development. PT networks are designed in multiple phases: planning, operation, and control. While changes in network design or operation systems are often expensive and difficult to implement, innovative control strategies offer a more cost-effective solution to improve the overall performance of PT networks.

Research shows that the speed and protection of transfers is one of the key factors influencing passengers' willingness to use PT ([1]). Transfers are generally synchronized during scheduling, but buses operate in a stochastic environment and can deviate from their timetables which leads to missed transfers and the possible loss of users. This is why there is a growing interest in transfer synchronization strategies for PT systems, especially in real-time. The increasing real-time availability of data on passenger demand from smart cards or bus occupancy sensors, on planned transfers from travel apps, and on vehicle locations from GPS allow significant improvements in understanding the state of PT networks. Dynamic predictions shed light on the impacts real-time control might have on users along all segments of their trips. Transfer synchronization aims towards a network-wide optimization with less myopic decisions.

We investigate the transfer synchronization problem for buses in a dense urban network through control tactics. We integrate predictions of future states of the PT system using both real-time data and historical data made available by the "Société de Transport de Laval" (STL). We bring the following contributions to the field: 1) Implementation of real-time control tactics for the synchronization problem using an arc-flow formulation ; 2) Solving large instances containing whole bus lines and many transfer points in real time. 3) Use of three online optimization algorithms for the transfer synchronization problem, and comparison of their performances. 4) Testing on a real large-scale data set from a dense PT network.

## 2 Problem description

Three control tactics are implemented alone or simultaneously in order to synchronize transfers and minimize passenger travel times. The holding tactic makes a bus wait at a stop after all passengers have boarded or alighted the vehicle. Holding is a very efficient tactic to avoid deviation from schedules, bus-bunching as well as to synchronize transfers. The holding tactic reduces operational speed and adds additional travel time for passengers onboard vehicles and waiting time for passengers wanting to board further along the line. Secondly, we use the skip-stop and skip-segment tactics. Skipping one or more consecutive stops can help reduce bus travel times or catch up delays with respect to schedules. Skipping stops has an immediate effect which is avoiding dwell times at stops, and a more long-term effect from the limiting of the number of passengers aboard the bus. Buses with fewer passengers spend statistically less time at stops. The stop-skipping tactic reduces the travel time aboard the bus and the waiting time of passengers waiting further along the line. On the other hand, stop-skipping can strongly inconvenience passengers wishing to board/alight on stops that are skipped, especially for low-frequency bus lines. Finally, we also use the speed control or speedup tactic. Speed control is an inter-stop tactic. This tactic helps decrease bus travel times without negatively impacting passengers wishing to board or alight the bus. Speed control is not always applicable in real life because of traffic congestion and speed regulations. The speedup tactic decreases travel times for passengers onboard vehicles or waiting further along the line. The speedup tactic can also allow to catch up delays in schedule and avoid missing synchronized transfers. When tactics are used, deviations from the schedules must be limited. Unlike in the literature, all stops are control stops which means tactics can be implemented at any stop or between any two stops. This allows for a more efficient control but generates more variables. Finally, the impact of tactics on all stages of passenger trips are considered.

## 3 Solution method

**Arc-flow model** An arc-flow model is formulated for the offline transfer synchronization problem using control tactics. The model minimizes total passenger travel time by improving transfer times while constraining deviation from the schedule.

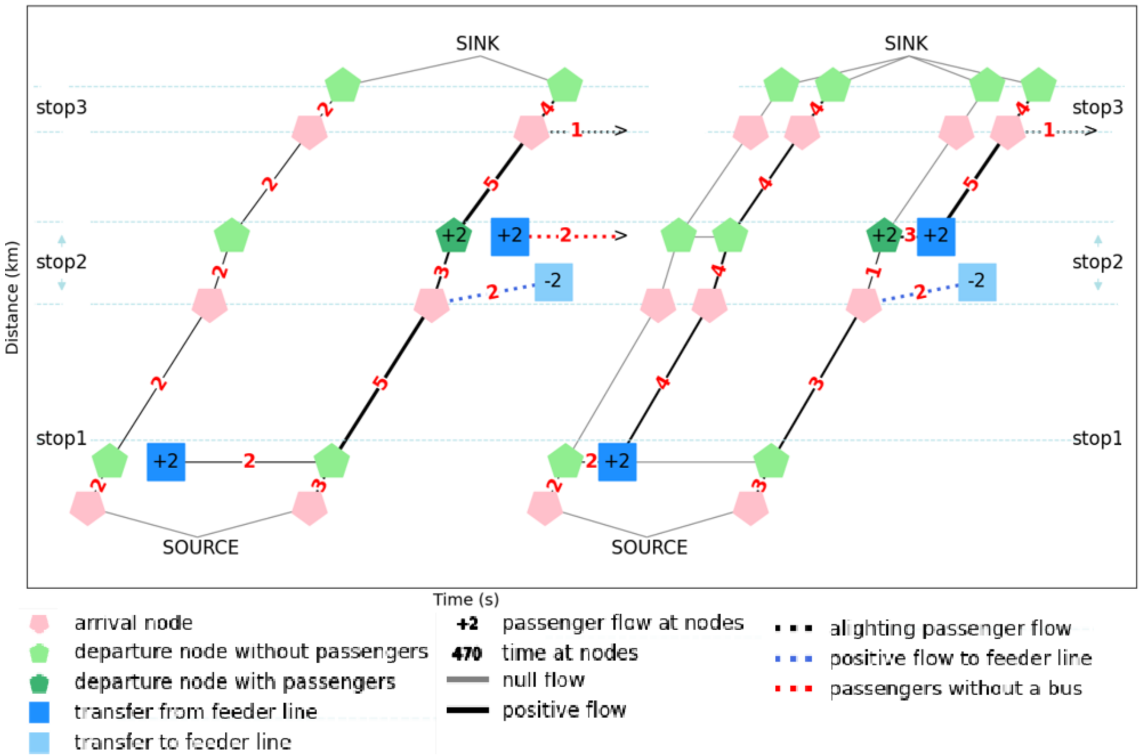



Figure 1: Arc-flow example: Left-no tactics, right-with tactics.

All tactics are integrated into a time-expanded graph of the arc-flow formulation as presented in Figure 1. In the model, we consider a main line on which we can apply tactics and feeder lines which are considered fixed. The model considers a control horizon that can range between only a few stops to the entirety of the main line.

**Online stochastic optimization** Three online stochastic algorithms ('Mean', 'Consensus' and 'Regret') [2] are adapted for the online transfer synchronization problem and evaluated in a simulation environment. Those online models can profit from the gradual reveal of real-time information and are designed to test and validate the results of the deterministic offline model. At each re-optimization, a control horizon containing some buses, stops and passengers is defined. This includes buses on the main line as well as feeder line vehicles that will transfer passengers at stops in the control horizon. Once the control horizon is defined, we collect available real-time data relevant to items in the horizon. Using sampling, we generate scenarios representing possible future states of the elements considered in the control horizon. We solve the offline model for each scenario providing decisions on tactics to use at all control stops in the control horizon : hold, skip-stop or speed control. We then apply the tactics - selected by one of three algorithms - only on the next stop of the control horizon. When a bus reaches the next stop, we start a new step and thus a new re-optimization in the simulation framework ; all future control tactics are re-evaluated at every iteration. The computation times at each re-optimization must stay low to allow an implementation in real time.

## 4 Experiments

Our experiments are based on data provided by the STL. Laval is a city in Canada with a population of 436,000. The PT network of the STL contains 46 bus lines and more than a thousand bus stops. The offline deterministic algorithm with perfect information has also been implemented to serve as target for the other algorithms. The implementation of no tactics is used as a baseline. Figure 2 shows results from computations on instances from line 42, a high frequency line with many passengers.

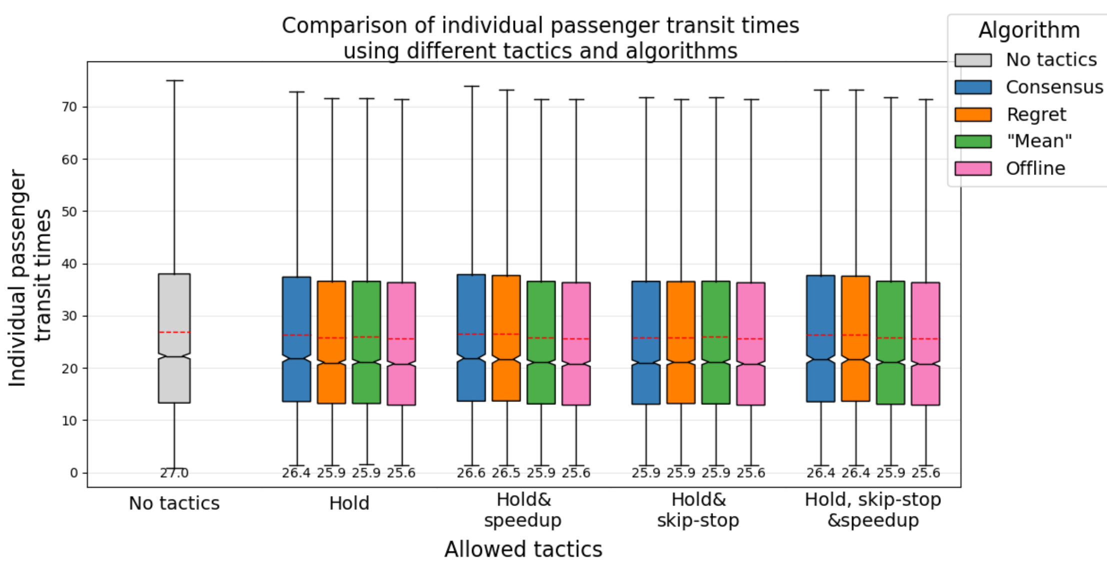



Figure 2: Total passenger travel times for different algorithms and tactics for the line 42.

## References

[1] A. A. Ceder, Y. Hadas, M. McIvor, and A. Ang. Transfer synchronization of public transport networks. *Transportation Research Record*, 2350(1):9–16, 2013.

[2] P. V. Hentenryck and R. Bent. *Online Stochastic Combinatorial Optimization*. The MIT Press, 2009.

# Resilient Relay Logistics Network Design Using $k$ Shortest Paths

**Onkar Kulkarni*, Mathieu Dahan, Benoit Montreuil**
Physical Internet Center, School of Industrial and Systems Engineering,
Georgia Institute of Technology, Atlanta, USA
*Corresponding Author: Onkar Kulkarni Email: onkar.kulkarni@gatech.edu

## 1 Introduction

The recent surge in e-commerce and world trade has led the parcel delivery industry to be one of the fastest-growing industries in the world. As the industry handles very large parcel volumes, the parcel delivery operations require meticulous planning, starting from designing the underlying logistics networks. Such networks majorly comprise hub facilities where parcels are sorted and then consolidated to ship towards their respective destinations. These parcel delivery operations traditionally rely on long-haul delivery trips that take a toll on the mental and physical health of delivery drivers, and present unsustainable working conditions. As a direct consequence, drivers are reluctant to work for such parcel delivery companies, leading to a major driver shortage in the industry. To remedy this situation, a potential solution consists of building relay facilities to fulfill demand via short-haul segments through relay transportation [1, 2, 3]. In relay logistics, the delivery drivers can advance commodities for half of their daily driving limit from one relay to the next, and then return to the original relay—ideally with other commodities—before reaching their home by the end of the day.

Logistics networks regularly face disruptions of varied nature, ranging from frequent and low-impact events such as congestion delays on roads, to low-probability high-impact events such as hurricanes. Such disruptions cause delivery delays, increased logistics costs, and a dip in customer satisfaction. While the optimization literature on relay network design does not consider disruption risks, the literature on logistics network resilience primarily designs small-scale non-relay logistics networks that can mitigate supply-demand disruptions. Since network structure has been shown to impact network resilience, we then aim to address the following research question: *How to design efficient and resilient logistics hub network configurations for relay transportation?* Drawing inspiration from the Physical Internet [3, 4], we aim to design large-scale hyperconnected relay logistics networks through topology optimization.

## 2 Model Formulation

We consider a logistics service provider interested in designing a large-scale logistics hub network for efficient and resilient relay transportation. We consider the initial planning phase of the design process and assume that the service provider has limited information regarding future demand and disruption risks. We introduce the problem of *$k$-Shortest Path Network Design* ($k$-SPND), which consists of locating logistics hubs to connect each origin-destination (O-D) pair with at least $k \geq 1$ routes of minimum total lengths. The premise is that by connecting O-D pairs with multiple short routes, it will then be possible to cost-effectively transport commodities with appropriate consolidation given the realized demand. In addition, if a multi-day disruption occurs at a hub or a transportation leg, then the service provider will be capable of transporting commodities via a different route, with a marginal impact on delivery cost and time.

Formally, let $\mathcal{P} := \mathcal{S} \times \mathcal{T}$ represent the set of origin-destination (O-D) pairs with each O-D pair $p$ having an associated demand share $d_p$. The service provider intends to open $N$ relay logistics hubs from a set of discrete candidate locations $\mathcal{H}$. Hubs are assumed to have sufficient capacity to handle large commodity volumes. We represent as $\mathcal{A} \subseteq (\mathcal{S} \cup \mathcal{T} \cup \mathcal{H})^2$ the set of potential (directed) transportation legs, which satisfy the driving time regulations to ensure a daily return for all drivers to their respective homes.

For every O-D pair $p = (s,t) \in \mathcal{P}$, we denote the set of $s-t$ paths as $\Lambda_p$ with each path $\lambda_p$ having travel time of $\tau_\lambda$. The goal of the $k$-SPND problem is then to select a subset of hub locations $\mathcal{H}_o \subseteq \mathcal{H}$ of size at most $N$ so as to minimize the demand-share-weighted total length of the $k$ shortest paths between each O-D pair in the subgraph induced by the set of nodes $\mathcal{S} \cup \mathcal{T} \cup \mathcal{H}_o$. To this end, we formulate it as a mixed-integer program (MIP) using path-based decisions. We consider for each hub $i \in \mathcal{H}$ a binary variable $y_i$ that takes a value of 1 if hub $i$ is opened, and 0 otherwise. Additionally, for every O-D pair $p = (s,t) \in \mathcal{P}$ and every $s-t$ path $\lambda \in \Lambda_p$, we define a continuous variable $z_\lambda$ that equals 1 if $\lambda$ is selected as one of the $k$ shortest $s-t$ paths in the subgraph induced by the opened hubs. We then derive the following MIP:

$$
\begin{aligned}
\min_{y,z} \quad & \sum_{p\in\mathcal{P}} \sum_{\lambda\in\Lambda_p} d_p \cdot \tau_\lambda \cdot z_\lambda && && \text{(1a)}\\
\text{s.t.} \quad & \sum_{i\in\mathcal{H}} y_i \leq N, && && \text{(1b)}\\
& \sum_{\lambda\in\Lambda_p} z_\lambda = k, && \forall p \in \mathcal{P}, && \text{(1c)}\\
& \sum_{\{\lambda\in\Lambda \,|\, i\in\lambda\}} z_\lambda \leq k \cdot |\mathcal{P}| \cdot y_i, && \forall i \in \mathcal{H}, && \text{(1d)}\\
& 0 \leq z_\lambda \leq 1, && \forall \lambda \in \Lambda, && \text{(1e)}\\
& y_i \in \{0,1\}, && \forall i \in \mathcal{H}. && \text{(1f)}
\end{aligned}
$$

## 3 Solution Methodology

We develop two approaches for solving the large-scale MIP optimally: In the first approach, based on a tailored implementation of Benders decomposition, we provide an analytical characterization of the optimal dual solutions of the exponential-sized Benders subproblem to generate the feedback cuts. This leads to a pseudo-polynomial time approach to generate these cuts based on Yen's algorithm (for computing $k$ shortest paths), which we accelerate using breadth-first-search and shortest-path subroutines. In the second solution approach, we tailor an implementation of branch-and-price: At each node of the branch-and-bound tree, we solve the master problem—a linear program with an exponential number of variables and constraints—using column generation. Using complementary slackness we show that at each iteration of column generation, the pricing subproblem can also be solved in polynomial time using Dijkstra's algorithm in an auxiliary graph with edge lengths depending on the optimal dual variables of the restricted master problem.

## 4 Case Study and Partial Results

Using the national level data of one of the largest parcel delivery companies in China that partnered with our research team, we created 6 representative problem instances of increasing size and complexity. In every instance, the parcel demand originates at one of the existing outbound logistics facilities of a city owned by the company ($\mathcal{S}$) and is destined for one of the company's existing last-mile delivery centers ($\mathcal{T}$). As the company intends to implement relay transportation, it identified a set $\mathcal{H}$ of candidate locations to open relay hubs, given by the company's existing intercity logistics hubs or major highway intersections. For the transportation arcs ($\mathcal{A}$), we only retained the transportation legs for which the drive time does not exceed 5.5 hours since the Chinese government imposes an 11-hour daily driving limit for truck drivers. This ensures that parcels travel towards their respective destinations while drivers return home daily.

We run the developed solution approaches to solve the MIP with parameters ranging from 10 to 60 for hubs $N$ to open, and from 1 to 4 for the number of shortest paths $k$. Next, to validate the proposed $k$ shortest paths relay logistics networks, we compare their performance against relay logistics networks constructed with only cost considerations to support parcel delivery. As parcel delivery networks obtain their operational cost savings through commodity consolidations, we construct an *efficiency-optimized (E-O) network*, obtained by selecting up to $N$ hubs to open to minimize the cost of the consolidation for an average commodity demand.

We conduct a set of experiments, where we subject the networks to random hub disruptions. In each disruption scenario, occurring uniformly at random, we suppose a relay hub becomes dysfunctional and no parcel can be routed through it during the planning horizon. We consider week-long disruptions and determine a minimum-cost consolidation plan to measure the performance in that situation. We assume that if an O-D pair becomes disconnected in the relay network as a result of a disruption, the demand for that O-D pair cannot be fulfilled using relay transportation during the planning horizon.

Through the minimum-cost consolidation plan, we compute two performance metrics: average delivery costs for the fulfilled demand and the amount of unfulfilled demand through relay transportation. Figure 1 portrays the comparison results. It showcases that our proposed $k$ shortest paths networks with $k \geq 2$ outperform the efficiency-optimized networks when facing hub disruptions with respect to both performance metrics. In addition to guaranteeing the delivery of a higher proportion of parcel demand through short-haul transportation, our networks also achieve lower average delivery costs per parcel as compared to efficiency-optimized networks under disruptions. This shows that by ensuring the existence of an increased number of paths $k$ between each O-D pair, demand can more likely be fulfilled through the relay network when facing disruptions.

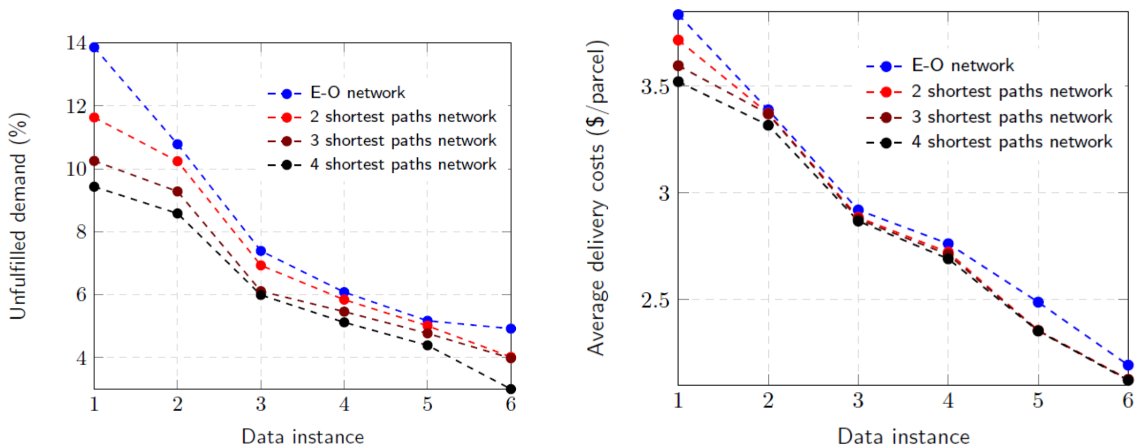



Figure 1: Comparison of network performance under 1-hub random uniform disruptions

If selected for presentation, we plan on further describing our network optimization approaches and on presenting the results regarding the efficiency comparison of the proposed with E-O networks under nominal situations, and the resilience-efficiency trade-off achieved by these networks.

# References


[1] H. Üster and H. Agrahari, "A Benders Decomposition Approach for a Distribution Network Design Problem with Consolidation and Capacity Considerations.", *Operations Research Letters* 39(2), 138-143, 2011.

[2] Z. Hu, R. G. Askin and G. Hu, "Hub relay network design for daily driver routes", *International Journal of Production Research* 57(19), 6130-6145, 2019.

[3] B. Montreuil, "Toward a Physical Internet: meeting the global logistics sustainability grand challenge", *Logistics Research* 3(2), 71-87, 2011.

[4] D. Hakimi, B. Montreuil and A. Hajji (2015). "Simulating Physical Internet Enabled Hyperconnected Semi-Trailer Transportation Systems",*Proceedings of 2nd International Physical Internet Conference.*

# Bid Construction for Urban Parcel Logistics via Combinatorial Auctions

**Simon Kwon**
School of Industrial and Systems Engineering,
Georgia Institute of Technology, Georgia, USA
Email: skwon82@gatech.edu

**Walid Klibi**
The Center of Excellence in Supply Chain (CESIT),
Kedge Business School, Bordeaux, France

**Mathieu Dahan**
School of Industrial and Systems Engineering,
Georgia Institute of Technology, Georgia, USA

**Benoit Montreuil**
School of Industrial and Systems Engineering,
Georgia Institute of Technology, Georgia, USA

## 1 Introduction

City logistics, with diverse stakeholders and conflicting interests, necessitates coordination for sustainable cities [1]. This study aligns with hyperconnected city logistics, where an orchestrator manages citywide demand flows and selects logistics service providers through a combinatorial auction. This auction consists of three stages: (1) pre-auction, selecting services for the primary auction; (2) bid construction, where bidders strategize and submit bids; and (3) winner determination, identifying auction winners [3].

This work delves into the second stage, addressing urban logistic nuances. Faced with high demand for time-sensitive operations and heightened competition among logistics service providers, less-explored aspects in the current literature, we investigate the construction of time-promised bids and the interplay between bidders, the orchestrator, and competitors. Our aim is to comprehend the competitive influence on overall profit through a stochastic bi-level optimization model. To the best of our knowledge, this is the first work addressing these urban logistics characteristics in the bid construction problem context. We explore an exact solution approach based on an optimal-value-function reformulation.

## 2 Methodology

### 2.1 Problem Overview

In a single-round, first-price reverse combinatorial auction, an urban logistics orchestrator (the auctioneer) interacts with multiple logistics service providers (bidders). The orchestrator manages flows between origin-destination (O-D) pairs, denoted as $O$, ensuring the timely delivery of demand for each O-D pair $o \in O$, with service guarantees represented by $\tau_o$. Each O-D pair $o \in O$ follows a predetermined path $\mathcal{P}_o$ designated by the orchestrator, segmented into a set of logistics activities including transport and hub processing activities. Through the auction, the orchestrator aims to allocate these activities to specific participating providers while satisfying the service guarantees of O-D pairs.

Let $L$ represent the set of logistic activities for auction, where each $l \in L$ has associated demand and time requirement options (e.g., 30, 60, and 90 mins). Demand corresponds to the expected volume and patterns for the planning horizon, while time requirements are bid promises linked to service level agreements (SLAs) denoted as $\mathcal{S}_l$ for each activity $l \in L$. The orchestrator sets bid requirements allowing single or bundled bids (multiple activities) with a limit of $K$ submissions per bidder. The auction-clearing process allocates bids to activities, aiming to meet O-D service guarantees while minimizing total allocation costs—a process known to bidders. Bidders seek to maximize profit by submitting bids specifying pairs of activities, SLAs, and corresponding bid prices.

This work focuses on a bidder's decision-making in the business setting, navigating uncertainties in demand, anticipating auction markets, and the orchestrator's responses.

### 2.2 Optimization Modeling and Solution Approach

We employ stochastic bi-level programming to address this problem, integrating bidding decisions of the bidder under consideration in the upper level and the orchestrator's decision-making process in the lower level. This accounts for bids from all bidders, including competitors, through a set of scenarios, anticipating their expected behavior.

Let $\bar{B}$ be the set of potential bids for the bidder under consideration, and $\bar{B}_l \subset \bar{B}$ be the set of bids that contain activity $l \in L$. The bidder has access to historical and known data on competitors' bids in the market. Competitors' bids and demand for logistic activities are modeled using a finite set of scenarios $\Omega$, each with probability $\phi(\omega)$ associating logistic activities with demand $d_l(\omega)$. Bids proposed by competitors in scenario $\omega$ are denoted as $\hat{B}(\omega)$. Let $B(\omega) = \bar{B} \cup \hat{B}(\omega)$ represent the set of all bids in scenario $\omega$, including both competitor bids and potential bids from the bidder under consideration. For each scenario $\omega$, bids that contain logistic activity $l \in L$ and have a corresponding SLA $s \in \mathcal{S}_l$ are denoted respectively as $B_l(\omega)$ and $B_{ls}(\omega)$. Each potential bid $b \in \bar{B}$ has associated activities $L(b)$, a maximum bid price $\bar{p}_b$, fulfillment cost $c_b(\omega)$ for each scenario $\omega$, and a specific SLA

$s \in \mathcal{S}_l$ for $l \in L(b)$. Each competitor's bid $b \in \hat{B}(\omega)$ in scenario $\omega$ comprises associated activities $L(b)$, a bid price $\hat{p}_b(\omega)$, and a specific SLA $s \in \mathcal{S}_l$ for $l \in L(b)$.

The goal of the bidder under consideration is to select a subset of bids to submit and bid prices for each bid such that the expected profit is maximized. Thus, for the upper-level problem, let binary variables $x_b$ indicate whether bid $b \in \bar{B}$ is submitted, and discrete variables $p_b$ indicate the bid price of bid $b \in \bar{B}$. For the lower-level problem, let binary variables $y_b(\omega)$ indicate whether bid $b \in B(\omega)$ in scenario $\omega$ is selected. We can formulate the bid construction problem as follows:

$$\max_{\mathbf{x},\mathbf{p},\mathbf{y}^*} \sum_{w \in \Omega} \left( \phi(\omega) \cdot \Big( \sum_{b \in \bar{B}} (p_b - c_b(\omega)) \cdot y_b^*(\omega) \Big) \right) \tag{1}$$

$$\text{s.t.} \sum_{b \in \bar{B}_l} x_b \leq 1, \quad \forall l \in L \tag{2}$$

$$\sum_{b \in \bar{B}} x_b \leq K \tag{3}$$

$$p_b \leq \bar{p}_b, \quad \forall b \in \bar{B} \tag{4}$$

where each $\mathbf{y}^*(\omega)$ satisfies:

$$\mathbf{y}^*(\omega) \in \arg\min_{\mathbf{y}(\omega)} \sum_{b \in \bar{B}} p_b \cdot y_b(\omega) + \sum_{b \in \hat{B}(\omega)} \hat{p}_b(\omega) \cdot y_b(\omega) \tag{5}$$

$$\text{s.t.} \sum_{b \in B_l(\omega)} y_b(\omega) = 1, \quad \forall l \in L \tag{6}$$

$$\sum_{l \in \mathcal{P}_o} \sum_{s \in \mathcal{S}_l} \sum_{b \in B_{ls}(\omega)} s \cdot y_b(\omega) \leq \tau_o, \quad \forall o \in O \tag{7}$$

$$y_b(\omega) \leq x_b, \quad \forall b \in \bar{B} \tag{8}$$

Equations (1) - (4) correspond to the upper-level problem, while equations (5) - (8) correspond to the lower-level problem. The upper-level problem seeks to determine a set of bids maximizing expected profit. Constraints (2) ensure that up to one bid is submitted for each logistic activity. Constraints (3) respect the auction requirement that limits the maximum number of bids to submit. Constraints (4) set the maximum bid price on bids.

For the lower-level problem, the objective of the orchestrator in (5) is to minimize the total allocation cost for each scenario $\omega \in \Omega$. Constraints (6) ensure that each logistic activity is allocated to one bid. Constraints (7) require that the allocation of bids to activities ensures the service time guarantees of each O-D pair $o \in O$. Constraints (8) link the upper- and lower-level problems, ensuring that bids are selected only if submitted.

To exactly solve the proposed bi-level model, we employ a value-function-based approach developed by [2]. This method iteratively generates bi-level feasible solutions, serving as an upper bound to the original problem. Simultaneously, the information corresponding to the lower-level variables is utilized to establish a lower bound. The algorithm terminates finitely with an optimal solution when all upper-level variables are discrete [2].

## 3 Preliminary Results

To test the proposed bid construction model and solution approach, we employed a set of synthetic urban area instances. In these scenarios, the orchestrator aims to allocate resources for 10 logistic activities, connecting 308 origin-destination (O-D) flows, with a total expected daily demand volume of approximately 30,000 parcels across the urban area. Each activity is characterized by an associated daily demand volume and SLA options. We considered three instance sizes: instance 1, associated with around 300 bids; instance 2, associated with 450 bids; and instance 3, associated with 600 bids, encompassing 30 scenarios in the optimization model. Additionally, we explored three market types with risk-averse (RA), risk-neutral (RN), and risk-seeking (RS) bidders, each characterized by different profit margins of bidders as shown in Table 1.

Table 1: *Impact of market characteristics on expected profit*

| Instance | Market Characteristics (Profit Margin) | Expected Profit ($) |
|---|---|---|
| | RA (5-8%) | $59,680 |
| 1 | RN (10-15%) | $71,098 |
| | RS (17-23%) | $80,196 |
| | RA (5-8%) | $48,474 |
| 2 | RN (10-15%) | $52,457 |
| | RS (17-23%) | $61,656 |
| | RA (5-8%) | $45,488 |
| 3 | RN (10-15%) | $51,253 |
| | RS (17-23%) | $66,460 |

Table 1 reveals that market characteristics significantly influence the expected profit of the bidder under consideration across instances. The gap in profit between the two extreme cases (RA and RS) ranges from 27% to 46%, emphasizing the need for a clear understanding and analysis of the markets when modeling the problem. In the presentation, we plan to delve into pre-processing for the translation of logistic activities into time-promised bids, provide business insights, account for urban logistic characteristics, and present numerical experiments to evaluate the performance of the proposed methodology with various urban instances.

## References

[1] T. G. Crainic and B. Montreuil, "Physical Internet enabled Hyperconnected City Logistics", *Transportation Research Procedia*, 12, 383-398, 2016.

[2] L. Lozano and J. C. Smith, "A Value-Function-Based Exact Approach for the Bilevel Mixed-Integer Programming Problem", *Operations Research*, 65(3), 768-786, 2017.

[3] J. Song and A, Regan, "Combinatorial auctions for transportation service procurement: The carrier perspective", *Transportation Research Record*, 1833(1), 40-46, 2003.

# Order Picking for E-Grocery

**Charlotte Köhler**
Department of Data Science & Decision Support
European University Viadrina, Frankfurt (Oder), Germany
Email: koehler@europa-uni.de

**Ann Melissa Campbell**
Department of Business Analytics
University of Iowa, Iowa City, United States

**Jan Fabian Ehmke**
Department of Business Decisions and Analytics
Universität Wien, Vienna, Austria

## 1 Introduction

In recent years, the e-grocery market has seen several new retailers, largely driven by the heightened demand for home delivery services during the COVID-19 pandemic. These companies often face the challenge of establishing a profitable business model, where careful planning is essential to balance tight margins with high customer expectations. The e-grocery fulfillment process can be divided into three steps: order acceptance, picking, and delivery [1]. During the order acceptance phase, retailers must decide which dynamically incoming orders to accept, taking into account their available resources for picking and delivery. In recent years, several studies have highlighted methods to enhance profitability by optimizing delivery routes during the order acceptance process. These methods suggest offering customers a limited selection of delivery time windows, e.g. [2], or setting differential pricing for various time slots to encourage customers towards more beneficial options for the retailer, e.g. [3]. While these approaches have exclusively focused on the use of delivery resources, the use of resources for picking orders for attended home deliveries has not received comparable attention [4], even though recent studies indicate that their impact on a retailer's profit is substantial and equivalent to the influence of routing decisions.[1]

[1]https://www.mckinsey.com/industries/retail/our-insights/achieving-profitable-online-grocery-order-fulfillment

This paper seeks to fill this gap by a detailed exploration of in-store picking costs in the context of e-grocery. This involves introducing a cost evaluation function designed to give retailers a more accurate and practical assessment of the resources required to pick orders to assess feasibility and maximize the number of accepted orders. By analyzing the interplay between the variety of items in an order and the associated picking times, we aim to offer a comprehensive framework to assess the feasibility of orders and explore the impact of different picking strategies on overall operational efficiency. We base our experiments on the layout of the well-known REWE grocery chain in Germany.

## 2 Exploring picking costs for e-grocery

For e-groceries, the time taken to pick items is a critical determinant of costs and use of resources, as longer picking times directly translate to increased labor expenses and fewer orders that can be picked. We define a set of items a customer orders by a set $I$. We can then represent a time to pick the items in set $I$ by a function $t = f(I)$. For a particular function, we can evaluate the feasibility and cost of arriving orders, given limits on picking time and a particular picking scheme.

We assume that all orders have a pickup or delivery window, but since all orders must be picked by the start of the window, they have a deadline $d$. Thus, in deciding whether to accept an order, it is important to evaluate whether an order can be picked by this deadline $d$ given previously accepted orders. In some applications, there may be an earliest time that an order may be picked up, but given the possibility of ice and cold storage, we will assume that that order can be picked at any time on the day of delivery before $d$. Thus, the feasibility of delivery of delivery time $d$ is based on the total picking time available before time $d$ and the previously accepted orders.

**Evaluating Order Feasibility.** For each incoming order, we must determine its feasibility concerning picking. This involves assessing whether the set of pickers can feasibly pick an additional order before the specified deadline, $d$. To make this assessment, we calculate the available slack time, which is determined by the difference between each picker's start time and the deadline, $d$, minus the total picking time of all previously accepted orders for that deadline or sooner. We repeat this for additional pickers and identify which pickers have sufficient picking time available. We explore strategies for assigning full orders to pickers and examine how varying deadline lengths, both longer and shorter, add to the complexity of evaluating the picking feasibility.

**Optimizing Order Assignment and Picking Efficiency.** Our focus also extends to enhancing the efficiency of how orders are picked. We propose three distinct policies in this regard. The first policy involves each picker completing an order individually, which represents the status quo. In addition, we explore the possibility of pickers handling

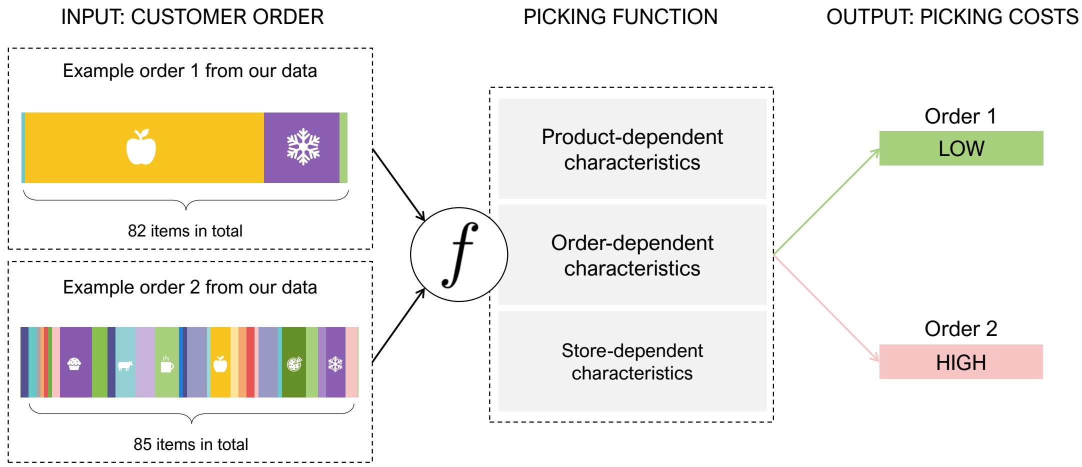



Figure 1: Determining picking costs using the picking function $f$

multiple orders concurrently. This second approach aims to minimize walking distances by enabling a picker to navigate the store just once but to fulfill several orders simultaneously. The third policy explores the concept of specialized pickers, where each picker focuses on items from a specific category, such as fresh or frozen products. Specialization is hypothesized to lead to increased picking speed, as pickers develop expertise in locating and selecting items within their designated category, thus streamlining the overall picking process.

**Picking Function.** For each order, we incorporate three key dimensions into the function $f$ to explore their impact on overall picking costs. For each order, there are *product-dependent characteristics*, such as the size and weight of each product in an order [5]. Generally, products that are smaller and lighter are simpler to pick, leading to lower picking costs. Second, there are *order-dependent characteristics*, which include the quantity of each item, the total number of different items in the order, and the total number of product categories represented by these items. Typically, orders comprising fewer distinct items are easier to pick, thus reducing picking costs. Last, there are *store-dependent characteristics*, focusing on how different store layouts affect picking processes such as walking time between product categories [6]. More compact stores with a limited range of items usually enable quicker picking, whereas larger stores with extensive inventories and layouts optimized for in-person shopping may complicate and lengthen the picking process.

In Figure 1, we present two customer orders from our data that are similar in their total item count but vary in terms of product types and the number of product categories they encompass. For each order, our picking cost function is applied to estimate the associated costs. In this example, the calculated costs reveal that the first basket incurs lower picking costs than the second basket.

## 3 Experiments and Outlook

We are conducting a comprehensive assessment of the proposed picking function, utilizing real-world data to ensure accuracy and relevance, for different order picking strategies. Our methodology involves a detailed store layout, based on REWE in Germany. Furthermore, we use historical order data from a former German e-grocer to enrich our analysis. This dataset encompasses information on over 400,000 order baskets, providing us with knowledge of the typical composition of orders. This information is instrumental in testing and validating our picking strategies and policies under realistic conditions.

With this approach, we evaluate orders as they arrive in terms of picking feasibility and cost with different picking schemes. We can also evaluate how the different order characteristics impact picking time and if some can be ignored to save computation time. Preliminary results indicate that for orders picked individually, the number of product categories represented by an order plays a larger role in the picking time than the total number of items. Store size is also an important factor.

Our findings will offer valuable contributions to the field, potentially guiding retailers in optimizing their picking processes and enhancing overall operational efficiency. The results will also help identify the picking costs and resources required for a range of baskets types. This will help identify the important costs associated with the picking side of e-grocery. The insights derived from this study will provide valuable input for a comprehensive analysis of both picking and delivery processes in the e-grocery sector.

## References

[1] A. M. Campbell and M. W. Savelsbergh, "Decision support for consumer direct grocery initiatives", *Transportation Science* 39(3), 313-327, 2005.

[2] C. Köhler, A. M. Campbell and J. F. Ehmke, "Data-driven customer acceptance for attended home delivery", *OR Spectrum* 1-36, 2023.

[3] R. Klein, M. Neugebauer, D. Ratkovitch, D. and C. Steinhardt, "Differentiated time slot pricing under routing considerations in attended home delivery", *Transportation Science* 53(1), 236-255, 2019.

[4] K. Waßmuth, C. Köhler, N. Agatz and M. Fleischmann, "Demand management for attended home delivery–a literature review", *European Journal of Operational Research*, 2023.

[5] S. Riveiro-Sanroman, M. Vazquez-Noguerol, J. A. Comesana-Benavides and J. C. Prado-Prado, "Determining the order picking time in food retailing", *The international conference on industrial engineering and industrial management*, 2022.

[6] M. Vazquez-Noguerol, S. Riveiro-Sanroman, I. Portela-Carames and J. C. Prado-Prado, “Analyzing store features for online order picking in grocery retailing: an experimental study”, *International Journal of Production Management and Engineering*, 2022.

# Cost assignment in delivery systems

**Francois Lamothe and Sandra Ulrich Ngueveu**
LAAS-CNRS, Université de Toulouse, CNRS, INP, Toulouse, France
Email: flamothe@laas.fr

## 1 Introduction

In a time when delivery systems are in rapid expansion while our economy needs to become more sustainable, the need for better analyses of such systems arises. In this work, we study the problem of assigning a servicing cost to each user of a delivery service. There is no immediate answer to this problem since the operating cost depends on the group of users (not on individual users) and economies of scale may be present. Meanwhile, this cost assignment is relevant for both the service provider who might integrate this information into its decision-making, and the user who may then better perceive its impact on the system. For example, the users may nowadays want to be able to measure the environmental cost of their delivery.

This assignment problem is equivalent to finding an appropriate cost-sharing mechanism in a cooperative game, the Traveling Salesman game. The game theory literature largely favors the Shapley value when sharing such costs because of its strong theoretical foundations [1]. The main disadvantage of the Shapley value is that it is very computationally intensive to evaluate in the context of delivery systems (or any system whose management involves an NP-hard problem). The literature contains several approximators which are still very expensive to compute in the delivery context. That is why in this work we design an approximator of the Shapley value requiring only a limited computational time. As shown in the preliminary computational results, this approximator appears to work well on the Traveling Salesman Problem instances tested compared to the approximators of the literature.

## 2 Problem setting

Given a weighted graph $G$, the goal of the Traveling Salesman Problem (TSP) is to find a cycle of minimum weight visiting all nodes of $G$.

In game theory, a cooperative game on a set of players $N$ is a function $v$ mapping each subset $S$ (also named coalition) of $N$ with a real number $v(S)$ representing the cost of servicing all players in $S$. Our interest is the Traveling Salesman Game which can be constructed from a TSP as follows. First, one node of $G$ denoted $o$ will serve as the origin of the cycles while the other nodes will be considered the players of the game. The cost $v(S)$ of a coalition of players $S$ is then the cost of the minimum weight cycle starting at $o$ and servicing all the nodes in $S$.

A cost-sharing mechanism in a cooperative game $v$ assigns a cost $\phi(i)$ to each player $i \in N$ such that the sum of all assigned costs is equal to $v(N)$. The most famous cost-sharing mechanism in game theory is the Shapley value. Indeed, this mechanism has strong theoretical foundations as it is uniquely defined by a set of basic axioms [1]. However, its main disadvantage is that it is difficult to obtain in many contexts as one needs to compute the cost $v(S)$ of all possible coalitions $S \subset N$. Even approximations of the Shapley value can be difficult to obtain. Most approximation algorithms in the literature, rely on sampling and evaluating many coalitions $S$ which can be computationally expensive when computing one cost $v(S)$ is NP-hard as in the Traveling Salesman game.

## 3 Shapley value and approximations

The Shapley value can be derived as the result of several expressions, each leading to a different approximation formula. The first formula defines the Shapley value in function of the marginal impact of the presence of a player $i$ in each subset of players $S$: $m(i, S) = v(S) - v(S \setminus \{i\})$. The Shapley value is then:

$$\phi(i) = \frac{1}{n!} \sum_{\pi \in \Pi(N)} m(i, P(i, \pi))$$

where $\Pi(N)$ is the set of permutations of $N$ and $P(i, \pi)$ is the set of predecessors of $i$ in permutation $\pi$. Following this formula, one can approximate the Shapley value by drawing a set $\Pi$ of random permutations of $N$ and averaging the marginal impacts of player $i$ over this set of permutations instead of $\Pi(N)$.

Another formula studied in [2] can be derived by rewriting the Shapley value as the result of a linear regression problem which can be solved with the following quadratic program:

$$\begin{aligned} &\min_{\phi} \sum_{S \subset N} \binom{n}{|S|}^{-1} \frac{n-1}{|S||N \setminus S|)} (v(S) - \phi(S))^2 \\ &s.t.\ \phi(N) = v(N) \end{aligned}$$

Following this other formula, one can approximate the Shapley value by drawing random subsets of $N$ with a probability proportional to $\binom{n}{|S|}^{-1} \frac{n-1}{|S||N \setminus S|)}$ leading to a family $\mathcal{S}$ of subsets. Then the following quadratic program is solved:

$$\min_{\phi} \sum_{S \in \mathcal{S}} (v(S) - \phi(S))^2$$
$$s.t.\ \phi(N) = v(N)$$

## 4 Our Shapley approximation

Our main contribution is the following cost sharing mechanism which approximates the Shapley value at a low computational cost. It depends only on the costs $v(\{i\} = m(i, \emptyset)$ of servicing a unique client $i$ (denoted $c_u^i$) and on the marginal cost $m(i, N)$ of client $i$ in the cycle servicing all players (denoted $c_m^i$). In this mechanism, the cost $\phi(i)$ of a player is:

$$\phi(i) = \ \lambda\ c_u^i + (1 - \lambda)\ c_m^i \text{ where } \lambda = \frac{v(N) - \sum_i c_m^i}{\sum_i c_u^i - c_m^i}$$

The coefficient $\lambda$ is chosen as the unique value which ensures $\sum_i \phi(i) = v(N)$. Rearranging the terms, this cost-sharing mechanism can be written as:

$$\phi(i) = \ c_m^i + (v(N) - \sum_i c_m^i) \frac{c_u^i - c_m^i}{\sum_i c_u^i - c_m^i}$$

In this form, the mechanism can be interpreted as assigning to each player its marginal cost $c_m^i$ and then assigning the cost remaining to be assigned proportionally to $c_u^i - c_m^i$. Note that computing all $c_m^i$ and $c_u^i$ requires only $2|N|+1$ evaluations of the cost function $v$ which is very low compared to all methods in the literature. For instance, in the approximator based on sampling permutations this would mean sampling only two permutations.

## 5 Numerical results

We now show with preliminary numerical results that our mechanism better approximates the Shapley value than the approximators in the literature when given low computational times. Our approximator will be denoted *marginal shapley* while the ones from the literature will be called *permutation shapley* and *regression shapley*. To compare the approximators, we use the following metric: denoting $\phi(i)$ the true Shapley value and $\hat{\phi}(i)$ the value of an approximator, the (average normalized) error made by the approximator is

$$\frac{1}{n} \sum_{i \in N} \frac{|\phi(i) - \hat{\phi}(i)|}{|\phi(i)|}$$

The different approximators are evaluated on TSP datasets created from instances from the literature [3]. Since we need to be able to compute the Shapley value to estimate the error of the approximators, we limited ourselves to instances with 10 customers. Our *marginal shapley* requires $2|N|+1$ evaluation of the cost function $v$ to be computed and we have given $6|N|$ evaluations to the other two methods because they rely on sampling. Despite this 3 to 1 computational advantage, Figure 1 shows that our approximator outclasses the ones from the literature on this benchmark.

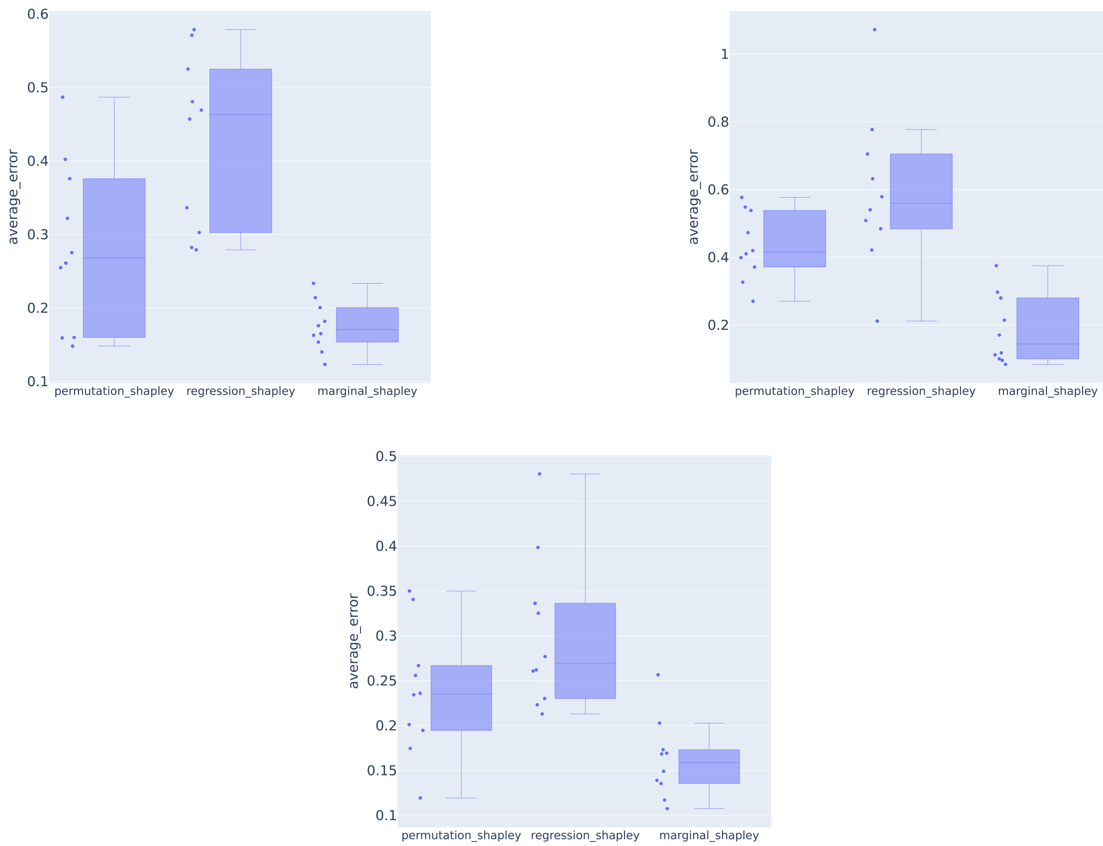



Figure 1: Error of the three approximators on three datasets from [3]

# 6 Acknowledgments


This work has been funded by the research and formation chair Responsible Retail supported by Toulouse INP, CNRS and One Stock.


# References


[1] Eyal Winter. The shapley value. *Handbook of game theory with economic applications*, 3: 2025–2054, 2002.

[2] Luis M Ruiz, Federico Valenciano, and Jose M Zarzuelo. The family of least square values for transferable utility games. *Games and Economic Behavior*, 24(1-2):109–130, 1998.

[3] Philippe Augerat. *Approche polyèdrale du problème de tournées de véhicules.* PhD thesis, Institut National Polytechnique de Grenoble-INPG, 1995.

# A Construction Matheuristic for Two-Tier Synchronized City Logistics

**Julia Lange**[*]
[*] Faculty of Business Studies and Economics,
University of Kaiserslautern-Landau, Kaiserslautern, Germany
Email: julia.lange@rptu.de

**Teodor Crainic**[**]
[**]School of Management, Université du Québec à Montréal,
and CIRRELT, Montréal, Canada

**Timo Gschwind**[*] **Walter Rei**[**]

## 1 Introduction

Sustainable city logistics planning focuses on multi-tier and multi-modal transportation with efficient consolidation and vehicle types suited for each tier. Integrating logistics services of different providers and shared transportation of multi-directional demand flows are key strategies to reduce congestion and create livable cities, see [1]. However, they are rarely addressed in the literature. This work introduces a detailed mathematical description of a day-before planning problem in two-tier multi-modal city logistics with on-time synchronization (2TM-CL-OS), where no storage exists at handover locations. While this enables the use of existing resources like supermarket parking lots in the distribution process, the requirement of delivery vehicles to meet for synchronized activities is a challenge.

The planning approach is based on two-tier scheduled service network design, see [2], where transportation services with routes, departure time windows, and capacities are given, and waiting time policies exist for customer and handover locations, the latter called satellites. Demands involve inbound (e2c), outbound (c2e), and innercity (c2c) commodity flows. The goal is to select services, including a schedule for each of them, and allocate the demands such that both operating costs and waiting times are minimized.

Since general-purpose solvers show a non-satisfactory performance in addressing the path-based mixed-integer programming (MIP) formulation of the 2TM-CL-OS, we present a construction matheuristic to evaluate this new model and discuss dependencies within the solution's structure. Thereby, we propose different approaches for variable fixing that yield promising results, and we provide conclusions on future research directions.

## 2 Mathematical Formulation

In the 2TM-CL-OS, a set of demands $d \in \mathcal{D}$ must be transported by large urban vehicles (urb) in outer-tier services $r \in \mathcal{R}$ and by small city freighters (cit) on inner-tier tours $k \in \mathcal{K}$ with handovers at satellites $z \in \mathcal{Z}$. Each demand features an origin and a destination at a customer location $i \in \mathcal{I}$ or an external zone, a volume, a handover time $h_d$, a time window, and an availability time if inbound. Each e2c and c2e demand is assigned to an inbound or an outbound connection $(r, k, z)$ representing its itinerary, see [3]. The c2c demands require an assigned tour only. Services and tours are selected for operation by binary variables $\rho_r^{\text{urb}}$ and $\rho_k^{\text{cit}}$ with corresponding fixed costs $c_r^{\text{urb}}$ and $c_k^{\text{cit}}$. A schedule with continuous starting times is determined for each selected service and tour, and thus, each demand itinerary, respecting demand time windows, travel times, waiting time allowances and synchronization requirements. The goal is to minimize the following objective function:

$$\sum_{r\in\mathcal{R}} c_r^{\text{urb}} \cdot \rho_r^{\text{urb}} + \sum_{k\in\mathcal{K}} c_k^{\text{cit}} \cdot \rho_k^{\text{cit}} + \sum_{r\in\mathcal{R}} \sum_{z\in\mathcal{Z}(r)} (\bar{\lambda}_{r,z}^{\text{urb}} - \lambda_{r,z}^{\text{urb}}) + \sum_{k\in\mathcal{K}} \left(\bar{\lambda}_k^{\text{in}} - \lambda_k^{\text{in}} + \bar{\lambda}_k^{\text{out}} - \lambda_k^{\text{out}}\right)$$
$$+ \frac{1}{100} \cdot \left[ \sum_{r\in\mathcal{R}} \sum_{z\in\mathcal{Z}(r)} \omega_{r,z}^{\text{urb}} + \sum_{k\in\mathcal{K}} (\omega_k^{\text{in}} + \omega_k^{\text{out}}) + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}(k)} \omega_{k,i}^{\text{cust}} \right]$$

The weighted sum combines the operating costs given in traveled time, the widths of the service intervals $[\lambda_{r,z}^{\text{urb}}, \bar{\lambda}_{r,z}^{\text{urb}}]$, $[\lambda_k^{\text{in}}, \bar{\lambda}_k^{\text{in}}]$, and $[\lambda_k^{\text{out}}, \bar{\lambda}_k^{\text{out}}]$ indicating handover periods as well as the waiting times of vehicles at satellites ($\omega_{r,z}^{\text{urb}}$, $\omega_k^{\text{in}}$ and $\omega_k^{\text{out}}$) and customer locations ($\omega_{k,i}^{\text{cust}}$). The set of constraints consists of four main groups:

1. *Service Network Design:* assigning demands, respecting vehicle capacities, linking assignment and selection
2. *Scheduling:* determining synchronized starting times of vehicle travels and demand handovers, respecting customer time windows and demand availability
3. *Waiting Times:* estimating vehicle waiting times, respecting waiting time allowances
4. *Satellite Capacities:* assignment and sequencing of vehicles at satellites, respecting satellite capacities

While the first two groups contain standard constraints, the waiting time estimation and the observation of the satellite capacities require tailored formulations. Two waiting time estimates are used to cover all cases of more than two vehicles meeting for handovers at the same satellite and time. Satellites are discretized into parking units for urban vehicles and city freighters, respectively, so that units can be assigned and vehicle appearances can be sequenced and timed appropriately. Since it is known that every e2c and c2e demand is handed over exactly once between one service and one tour, the mathematical model is enhanced by service interval-and selection-related valid inequalities.

## 3 Construction Matheuristic and Results

An evaluation of the mathematical description on reasonably sized instances is challenging, since general-purpose exact solvers unsurprisingly show a poor performance on the MIP model of the 2TM-CL-OS. Therefore, we propose and test a simple two-step construction matheuristic to create first insights on solving performance and dependencies. The core idea is to reduce the search space by fixing a subset of the binary selection variables for services and tours. The two-step procedure works as follows:

I. **Solve relaxed model.** Store pool of all objective-improving solutions.

II. **Solve full model.**
Fix values of selection variables based on the best solution or the pool from Step I.

The relaxed model consists of the objective function without service intervals, constraint groups 1 and 2 and simplified waiting time constraints (from group 3). Among other options tested, the following three **fixing rules** to exclude services and tours ($\rho_r^{\text{urb}} = 0$ or $\rho_k^{\text{cit}} = 0$) show interesting results:

· *FixUnusedRoutes:* Services and tours never selected in any solution of the solution pool are excluded.

· *FixRarelyUsedRoutes:* Services and tours rarely selected in the solutions of the pool are excluded.

· *FixPoorCombiRoutes:* Combinations of services and tours that rarely appear in high-quality solutions of the pool are excluded.

The computational experiments are conducted on randomly generated instances with 30 demands and varying numbers of satellites ($|\mathcal{Z}|$). Services are road- and rail-based, cargo bikes are used as city freighters, and waiting time allowances are equally distributed between no, moderate and long waiting permitted. All methods are implemented in Python 3.8 with Gurobi 10.0 as a general MIP solver. The time limit is 1 hour, where 10 and 50 min. are set for Step I and II, respectively.

While solving the full model stopped without any optimal solution and with a 53% median gap over all instances, the construction matheuristics show promising results. Table 1 summarizes the proportions of instances solved to optimality or proven infeasible (Opt/Inf), the median gap (Gap), and the median upper bound difference compared to the full model (Diff). Fixing unused routes reduces the optimality gap and produces solutions of better quality. However, none of the instances is solved to optimality, so that a further reduction of the search space seems reasonable. While excluding rarely used routes does this, it also shows a large amount of infeasible instances with 4 and 5 satellites due to a lack

Table 1: Computational results of construction matheuristics

| | FixUnusedRoutes | | | FixRarelyUsedRoutes | | | FixPoorCombiRoutes | | |
|---|---|---|---|---|---|---|---|---|---|
| $\|\mathcal{Z}\|$ | Opt/Inf | Gap | Diff | Opt/Inf | Gap | Diff | Opt/Inf | Gap | Diff |
| 2 | 0/0 | 25.6 | -1.1 | 100/0 | | 3.0 | 0/0 | 1.7 | -4.0 |
| 4 | 0/0 | 26.0 | -7.0 | 50/50 | | -4.3 | 50/0 | 17.3 | -5.0 |
| 5 | 0/0 | 26.8 | -1.5 | 33/33 | 0.3 | 6.2 | 33/0 | 11.6 | 4.4 |
| 10 | 0/0 | 26.1 | -3.8 | 100/0 | | 1.2 | 67/0 | 11.9 | 4.5 |

All entries except number of satellites given in percent (%).

in consideration of the two-tiered structure. This is overcome by excluding combinations of services and tours that rarely appear in high-quality solutions. Even if the amount of instances solved to optimality is decreased, the decrease in the median gap is remarkable compared to the full model, and the loss in solution quality is small.

## 4 Conclusion

The 2TM-CL-OS is a complex problem of significant importance for future city logistics systems. A mathematical formulation focusing on synchronization, waiting time estimation, and satellite capacity requirements is proposed. Results of simple construction matheuristics validate the accurate representation of structural and temporal requirements and demonstrate the applicability of the MIP model in the generation of near-optimal solutions for medium-size instances. This points to more elaborate heuristic procedures and decomposition approaches as promising future research directions to solve larger problems. In our ongoing work, we continue a broad experimental campaign to draw further methodological and managerial insights, and we aim to apply stochastic optimization to account for travel and handover time uncertainty.

## References


[1] T. G. Crainic, F. Errico, W. Rei and N. Ricciardi, “Integrating c2e and c2c traffic into city logistics planning”, *Procedia - Social and Behavioral Sciences* 39, 47-60, 2012.

[2] P. Fontaine, T. G. Crainic, O. Jabali and W. Rei, “Scheduled service network design with resource management for two-tier multimodal city logistics”, *European Journal of Operational Research* 294(2), 558-570, 2021.

[3] T. G. Crainic, N. Ricciardi and G. Storchi, “Models for Evaluating and PLanning City Logistics Systems”, *Transportation Science* 43(4), 432-454, 2009.

# Continuous Time Formulation to Scheduled Service Network Design with Stochastic Travel Times

**Giacomo Lanza**
Department of Computer Science,
University of Pisa, Pisa, Italy
Email: giacomo.lanza@unipi.it

**Teodor Gabriel Crainic**
CIRRELT and School of Management,
Université du Québec à Montréal, Montréal, Canada

**Mauro Passacantando**
Department of Business and Law,
University of Milan-Bicocca, Milan, Italy

**Maria Grazia Scutellà**
Department of Computer Science,
University of Pisa, Pisa, Italy

## 1 Problem description

Freight consolidation is one of the many strategies carriers apply to lower transportation costs and consequent service prices. It involves combining freight associated with multiple customers onto common vehicles or convoys (e.g., a truck, a container ship, a freight train). Postal and small-package transportation companies, less-than-truckload motor carriers, railroads, maritime liner navigation companies offer similar services [1]. The design of a consolidation-based service network is a complex planning process involving interrelated and interdependent decisions traditionally faced at tactical level by carriers and supported by service network design (SND) methodology. Its scope is to produce an operation plan - specifying the services to operate (route, stops, frequency, vehicle type, capacity, schedule), and how to move the freight (services used and terminals visited) through that service network - that achieves the economic and quality targets of the carrier. The plan is made for a certain time interval or schedule length (e.g., a week), and applied repeatedly over a longer time period, called planning horizon (e.g., a season)[1].

There is a significant body of literature on deterministic SND models [1], where the involved parameters (e.g., the volume of demand, costs, profit, travel times, service times) are considered as readily available, built on point forecasts. Nevertheless, once established, customers' and carriers' expectations are that the service network adheres as much as possible to what the carriers externally announce, despite the variations that may be observed in the parameters (with respect to the estimations used in the planning phase) during daily operations. These may cause services not operating as specified in the published schedule (thus not respecting the arrival/departure times at/from each stop) and commodities not reliably arriving at destinations (thus not respecting the due date agreed-upon with customers). Variations can significantly impact network performance, forcing carriers to apply costly real-time adjustments to meet customers' requests within the agreed standards.

Tactical planning problems are indeed inherently stochastic, involving making choices in an environment with not fully known information. Specifically, some decisions must be made at a time when only statistical distributions are known regarding some parameters, whose complete information will be available only after medium-term decisions are already made. Several authors highlighted how addressing SND through a stochastic programming approach may provide flexibility in solutions to hedge, or at least to limit, the bad effects caused by fluctuations, by explicitly considering uncertainty into the formulation [2]. Nevertheless, such an approach has been less proposed in the literature for SND, and only a few examples are available (for instance, in [3, 4]).

Our focus lies on travel times, i.e., the time required by a service to travel from one terminal to the next one along its route, considering the fluctuations that may arise in daily operations, i.e., the so-called randomness type of uncertainty [2]. These fluctuations can occur in different contexts and applications, may be due to several factors, such as traffic congestion or adverse weather conditions, and may have various impacts depending on the transportation modes. Consequences may include disruption of the planned service-to-service transfers, thus hindering efficient consolidation of commodities, and difficulties in meeting delivery deadlines for commodities. Ultimately, delays can negatively impact the reputation and revenues of the freight carrier.

We investigate a SND model in the context of consolidation-based freight carriers, by proposing a two-stage stochastic programming formulation that explicitly considers the stochastic nature of travel time on the links connecting terminals. We consider demand as a deterministic parameter characterized by origin, destination, volume, entry and due date. A set of transportation services that potentially could be offered by the carrier is given. Each potential service is defined by its origin and destination terminals, route, and schedule, namely, departure time at origin, departure and arrival times at intermediate stops (if any), and arrival time at destination. We qualify these times, and the associated inter-terminal travel times, as *usual* as they correspond to ideal estimation without perturbations or delays. Selection of services is based on such information and on the available probability distributions of travel times on each of the legs composing their routes. Depending on the selected services and on the outcome of the travel time random variables, remaining decisions must be made. These relate to the adjustment of the departure times of the selected services (within the limits of operational feasibility), the routing of commodities through the selected services, and outsourcing (we consider the entire delivery of a commodity from its origin to its destination). Additionally, the costs associated with such decisions are computed as well as the possible delays related to both service operations (with respect to the schedule) and commodities arrival times at destination (with respect to the agreed upon time of deliveries with customers).

The goal of the proposed formulation is to define a robust service network able to reduce the consequences brought by the fluctuations in travel times associated with delays in operations, which can have ripple effects on the feasibility and profitability of the plan generating additional cost for the carrier.

## 2 Problem formulation

In [5], a deterministic SND problem is formulated as a network design model on a flat graph. Terminals are represented by nodes, and service legs by arcs. The stochastic SND problem here addressed is formulated extending such a network design modelling idea. In particular, time is continuously represented, with the model introducing continuous variables for arrival and departure times of services and commodities. This method mitigates issues of traditional time-space network formulations, particularly for large-scale instances where fine time discretization increases problem dimensions, leading to intractability in exact solutions. Specifically, the problem is modelled as a two-stage (planning and recourse) stochastic programming model. First stage decisions concern the selection of the services and are made by considering the usual travel times characterizing the schedule of the potential services. Such decisions are made with the objective of minimizing the fixed service-selection costs plus the expected costs associated with second stage decisions, which relate to adjustment of service departures, routing commodities, and outsourcing. The latter depend on the travel time realizations of the associated travel time probability distributions. Once first stage decisions are made, the realizations of the uncertain parameter values is revealed at each application of the plan (we may have a more precise estimation on travel times based on the system conditions before operations begin). This new information is then used to determine the departure times of the selected services, the routing of commodities through the selected services, and outsourcing, plus the possible delays in services operations and deliveries.

Thus, first stage decision variables are the classical binary design variables representing service selections. To determine departure times of the selected services, the routing of commodities, and the possibility to exploit outsourcing, several second stage variables are introduced (classical commodity flow variables are included). The mathematical model comprises usual flow conservation constraints, non-conventional linking-capacity constraints, constraints related to service and commodity time management.

Uncertainty is approximated with a finite set of scenarios, wherein each scenario contains a realization of travel times and has a probability of occurring. Through this set, the stochastic program is formulated as a deterministic mixed integer linear program and the expectation in the objective function can be expressed as a linear function.

## 3 The Odysseus presentation

We present a preliminary analysis conducted using off-the-shelf optimization software on small to medium-sized instances. From a computational standpoint, we assess the performance of the proposed formulation in addressing this particular class of problem, specifically in relation to continuous time representation. Next, we quantify the benefits of explicitly considering stochastic travel time in the SND model. Additionally, we highlight the features that solutions exhibit when varying parameter settings to hedge against time fluctuations. Finally, we discuss potential future research avenues.

## References


[1] T.G. Crainic and M. Hewitt, "Service Network Design", in *Network design with applications to transportation and logistics*, T.G. Crainic, M. Gendreau and B. Gendron, Ch.12, 347-382, Springer, Boston 2021.

[2] M. Hewitt, W. Rei, and S. Wallace, "Stochastic Network Design", in *Network design with applications to transportation and logistics*, T.G. Crainic, M. Gendreau and B. Gendron, Ch.10, 283-315, Springer, Boston 2021.

[3] A.G. Lium, T.G. Crainic and S. Wallace, "A study of demand stochasticity in service network design", *Transportation Science* 43(2), 144-157, 2009.

[4] G. Lanza, T.G. Crainic, W. Rei and N. Ricciardi, "Scheduled service network design with quality targets and stochastic travel times", *European Journal of Operational Research* 288(1), 30-46, 2021.

[5] G. Lanza, T.G. Crainic, M. Passacantando, M.G. Scutellà, "A Continuous Time Physical Graph based Formulation to Scheduled Service Network Design", in *Optimization in Green Sustainability and Ecological Transition*, Bruglieri et al. (eds), vol. 12, AIRO Springer Series, 2023, doi: 10.1007978-3-031-47686-0_24.

# Branch-Cut-And-Price Algorithm for Vehicle Routing Problem with Drones

**Lichau Sylvain**
Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, F-33400 Talence, France,
Email: sylvain.lichau@u-bordeaux.fr
**Ruslan Sadykov**
Univ. Bordeaux, CNRS, INRIA, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France
**Julien François**
Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, F-33400 Talence, France,
**Rémy Dupas**
Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, F-33400

## 1 Introdution

The online shopping growth has increased exponentially these last years, in particular during the Covid period, leading to a particular interest in the last miles delivery problem. The integration of drones to vehicle routing problems is interesting for several reasons. First, from an economic point of view, paralleling deliveries enable faster deliveries, and therefore cheaper. The drone's energy is also considered cheaper than the gas for trucks. Second from an ecological point of view, using drones instead of trucks decreases the pollution emitted by a route. Finally, from a societal point of view, by reducing the number of customers delivered by trucks in cities, the traffic congestion will decrease significantly.

Many delivery companies are trying to integrate drones to deliveries, but face problems regarding security, public risk beliefs and legislation. One of the first successful implementation of a couple drone/truck delivery, to our knowledge, has been established in France in 2016, to a difficult-to-reach area[1]. More recently, Amazon is claiming that they will start using drone deliveries in Italy, the UK and a location in the US by the end of 2024[2]. Because of the relatively recent and theoretical aspect of the problem, modelling the drone/truck interaction is a challenge. That is why, the operation research aspect of the problem is studied under many different assumptions [3], [4].

We believe the key to integrate drones to vehicle routing problems is to balance simplicity and efficiency. Indeed, a model where drones can do en route operations will improve efficiency, but will also be hard to implement on a real-scale application; for responsibility reasons and technical reasons. For these reasons, we believe that the Two-Echelon Routing Problem with Drones (2E-VRP-D)[6] is a well-designed problem, enabling the use of multi-parallel routing, while keeping human operator near the drones in case of problems. It also maximizes the use of drones by making the number of drones on each truck a decision variable.

The contributions of this paper are the following: we suggest a new exact Branch-Cut-and-Price algorithm for the 2E-VRP-D that enumerates drone routes, using a dynamic program to only keep the routes that might be used in an optimal solution, we propose an adaptation of the Rounded Capacity Cuts designed for the 2E-VRP-D, we present numerical results on literature instances.

## 2 Problem Definition

The problem we are addressing is the 2E-VRP-D. In this problem, a set of vehicles and a set of drones have to serve a set of customers from a depot. Each customer is characterized by a demand, a service time (dependent on the type of vehicle operating the delivery), a deadline and its availability for drone delivery. The service at a customer delivered by a vehicle happens in parallel of the deliveries made by drone from this customer. The number of drones carried by each truck is a decision variable. The useful capacity of the trucks depends of the number of carried drones, each drone and its necessary equipment having a weight. While trucks and drones move at different speeds, they adhere to the same distance metric.

The drones can only carry one package per flight, and are synchronized with the trucks. A drone can fly only when its assigned truck is stationed at a customer. Drones can take off multiple time from the same customer. They also have a maximum capacity. An energy function limits the maximal flight distance, introduced by [5] as the hovering power consumption of an h-rotor drone—an upper bound on the general power consumption of an h-rotor drone. It is assumed that drones are fully recharged upon reaching the truck.

## 3 Methodology

We use the algorithm defined in [7]. In this generic algorithm, pricing subproblems are modelled as Resource Constraint Shortest Path problems (RCSP). In the master problem, the objective function is the minimization of the total duration of the routes, and the constraints enforce that each customer is visited exactly once, and that the total number of trucks and drones used is feasible.

To model the pricing problem, we enumerate the possible drone routes from each customer. We highlight the fact that finding the optimal drone route from a customer $c$ to deliver a set of other customers with $k$ drones, knowing the truck arrival date at $c$ is equivalent to solving the scheduling problem, with $k$ parallel identical machines, deadlines and with the objective of minimizing the makespan, $P|\bar{d}_j|C_{\max}$. The truck arrival date is required because the deadline of the clients implies that an optimal scheduling solution might not always be feasible. However, an optimal solution at time $t$ remains optimal at time $t+\Delta, \Delta \in \mathcal{R}^+$ if the solution remains feasible at time $t+\Delta$. This holds because the optimal scheduling solutions can always be without waiting time. Hence we look for the Pareto front of the scheduling problem maximizing the maximal truck arrival date while minimizing the makespan.

The idea of the dynamic program is to explore the combination of two solutions such that the sum of their number of machines is equal the number of machines in the problem, and the set of sets of customers delivered by the solutions is a partition of the original set of customers to deliver. We use a dynamic program to find all the optimal drone routes from each customer. We split the nodes of all the customers in the RCSP graph in two types of nodes, the first type keeping all the incoming edges and the second all the outgoing edges. We add an arc from the first node to the second for each optimal drone routes from the customer represented by the node.

# 4 Results

To compare our results, we use the same sets of instances as in [6], a 3 hour time limit and run the solver on machines with comparable processors.

| Instances | Zhou et al.[6] | | Our | |
|---|---|---|---|---|
| Size | Solved | Time (s) | Solved | Time (s) |
| 10 | 20/20 | 0.4 | 20/20 | 0.49 |
| 15 | 20/20 | 0.7 | 20/20 | 1.04 |
| 25 | 15/20 | 3561.5 (1148.6) | 20/20 | 87.34 |
| 35 | 4/9 | 7408 (3168.1) | 9/9 | 696.26 |

Table 1: Numerical results comparison

We compare the results of our algorithm against the state-of-the-art literature on the 2E-VRP-D computed on similar processors. The size field refers to the number of customers in the instance, the solved field is the number of instances solved to optimality over the number of instances in the set, and the time field is the average time taken by the algorithm to solve the instances. The number in parentheses is the average on instances solved optimally, whereas the other number is the average on all the instances of the set. Despite a small extra time to solve the small instances, our algorithm is able to solve all the medium size instances proposed in [6], and the average solving time is also significantly lower.

## 5 Conclusion

In conclusion, we suggest a new exact branch-cut-and-price algorithm to solve the 2E-VRP-D, which is greatly more efficient than the previous state-of-the-art algorithm. The central idea of this method is the enumeration of the drone routes. With this new method, we will be able to solve larger instances; we hope to double the size of instances from the literature.

## References


[1] DPDgroup, ”Innovation for delivering difficult-to-reach areas”, https://www.dpd.com/fr/en/sending-parcels/our-innovative-services/drone-delivery/

[2] Amazon Staff, ”Amazon is launching ultra-fast drone deliveries in Italy, the UK, and a third location in the U.S.”, https://www.aboutamazon.com/news/operations/amazon-prime-air-drone-delivery-updates

[3] G. Macrina, L. Di Puglia Pugliese, F. Guerriero, G. Laporte, ”A literature review: Drone-aided routing”, *Transportation Research Part C: Emerging Technologies*, 102762, 2020

[4] S.H. Chung, B. Sah, J. Lee, ”Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions”, 105004, 2020

[5] K. Dorling, J. Heinrichs, G. G. Messier, S. Magierowski, ”Vehicle routing problems for drone delivery”, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 70-85, 2016

[6] H. Zhou, H. Qin, C. Cheng, L.-M Rousseau, ”An exact algorithm for the two-echelon vehicle routing problem with drones”, *Transportation Research Part B: Methodological*, 124-150, 2023

[7] A. Pessoa, R. Sadykov, E. Uchoa, F. Vanderbeck, ”A generic exact solver for vehicle routing and related problems”, *Mathematical Programming*, 483-523, 2020

# Agricultural fleet vehicle routing problem with implements

**Aitor López-Sánchez**
CETINIA, University Rey Juan Carlos, Madrid Spain
UMR 9189 - CRIStAL, Centrale Lille, Univ. Lille, CNRS, Inria, France
Email: aitor.lopez@urjc.es

**Frederic Semet**
UMR 9189 - CRIStAL, Centrale Lille, Univ. Lille, CNRS, Inria, France

**Marin Lujak**
CETINIA, University Rey Juan Carlos, Madrid Spain

**Holger Billhardt**
CETINIA, University Rey Juan Carlos, Madrid Spain

## 1 Introduction

Agriculture is undergoing a technological revolution to meet rising global demand. Autonomous vehicles are integral to modern agricultural practices, yet Agricultural Vehicle Routing Problems primarily address homogeneous fleets with a single type of task and one type of crop. Real-world agriculture, however, involves diverse tractors performing various tasks, such as ploughing, fertilization, fumigation, and harvesting, using attached implements. Coordinating routes for these mixed fleets is crucial for optimizing task execution and resource allocation in contemporary agriculture.

The coordination of two vehicle classes (tractors and implements) simultaneously in agriculture to our knowledge remains unexplored. The concept of movement synchronization, where changes in one route affect others, involves nonautonomous vehicles relying on autonomous vehicles for spatial movement. Such a synchronization is clearly required in agriculture, where an implement is used with a tractor for a time period [1,2]. Various approaches exist in the literature, including one allowing detachment and reattachment during the route, as in the Vehicle Routing Problem with Trailers and Transshipments [3,4]. A different approach, proposed by [5], is to avoid assuming consistent associations between autonomous and non-autonomous vehicles. The Active Passive Vehicle Routing Problem introduces a scenario with active and passive vehicles, where the active vehicles, that dis-place the passive ones, may change, thus contributing to addressing the synchronization challenge in an agricultural setting [6].

## 2 Problem formulation

In this section, we present the set-partitioning formulation of the Agricultural Fleet Vehicle Routing Problem with Implements (AFVRPI), solved through column generation. Let us consider a fleet comprising both tractors and implements ($\mathcal{F} = \mathcal{V} \cup \mathcal{M}$), covering routes on the transportation network. The transportation network is represented by $(\mathcal{N}, \mathcal{A})$, where nodes in $\mathcal{N}$ consist of four distinct sets: $\mathcal{N}_{tasks}$ for agriculture tasks, $\mathcal{N}_{depots}$ for tractor and implement depots, $\mathcal{N}_{detach}$ for detaching nodes, and $\mathcal{N}_{attach}$ for attaching nodes. Arcs in $\mathcal{A}$ denote spatial and temporal connectivity, with arc distances represented as $d_{ij}$. The set of transfer arcs, denoted as $(d, a) \in \mathcal{A}_{transfer}$, includes arcs where implements can be detached $d \in \mathcal{N}_{detach}$/attached $a \in \mathcal{N}_{attach}$ to tractors. Each task has a given demand, service time, and time window.

Based on task-implement and vehicle-implement compatibilities, each vehicle in the fleet $f \in \mathcal{F}$ has a subgraph representation $(\mathcal{N}^f, \mathcal{A}^f)$ indicating the nodes and arcs it can visit. Routes for tractors and implements are elementary paths within their respective subgraphs. The route-based formulation incorporates binary variables $\delta_p^v$ for feasible tractor routes $p \in \Omega^v$ and $\lambda_q^m$ for feasible implement routes $q \in \Theta^m$. The cost of a route for a tractor $v$ is denoted as $c_p^v$, and for an implement $m$, it is denoted as $c_q^m$. Positive integers $a_{ijp}^v$ represent the number of times arc $(i, j) \in \mathcal{A}^v$ is traversed by tractor $v$ on route $p$, while $b_{ijq}^m$ represents the same for implement $m$ on route $q$. Moreover, $T_{ip}^v$ is the time spent at node $i$ if the node is visited with the vehicle $v$ in the path $p$.

The goal of the AFVRPI is to find a set of feasible routes for tractors and implements that visit all the tasks minimizing the overall cost and respecting the movement constraints. An implement route $q_m, m \in \mathcal{M}$ can be part of the solution if each arc $(i, j) \in \mathcal{A}^m$ corresponds to a compatible tractor travelling the same arc, except for transfer arcs. The restricted master formulation is for the AFVRPI is the following. The objective function (1) is to minimize the total cost of all selected routes. The assignment constraints (2) and (3) are the one-on-one vehicle-implement-task assignment constraints. The arc-coordination constraints (4) require that if an implement travels an arc, it must be coupled to a vehicle. The transfer constraints (5) set the minimum transfer time from an implement to $\tau$. The vehicle and implement constraints (6) impose the assignment of one route to each tractor and implement.

$$
\begin{aligned}
\min \quad & \sum_{v\in\mathcal{V}}\sum_{p\in\Omega^v} c_p^v\delta_p^v + \sum_{m\in\mathcal{M}}\sum_{q\in\Theta^m} c_q^m\lambda_q^m && && (1)\\
\text{s. t.} \quad & \sum_{v\in\mathcal{V}}\sum_{p\in\Omega^v} a_{jp}^v\delta_p^v \le 1 && \forall j\in\mathcal{N} && (2)\\
& \sum_{m\in\mathcal{M}}\sum_{q\in\Theta^m} b_{kq}^m\lambda_q^m = 1 && \forall k\in\mathcal{N}_{tasks} && (3)\\
& \sum_{q\in\Theta^m} b_{ijq}^m\lambda_q^m \le \sum_{v\in\mathcal{V}}\sum_{p\in\Omega^v} a_{ijp}^v\delta_p^v && \forall m\in\mathcal{M}, \forall (i,j)\in\mathcal{A}^m\setminus\mathcal{A}_{transfer} && (4)\\
& \sum_{v\in\mathcal{V}}\sum_{p\in\Omega^v} (T_{ap}^v - T_{dp}^v)\delta_p^v \ge \tau \sum_{q\in\Theta^m} b_{adq}^m\lambda_q^m && \forall m\in\mathcal{M}, \forall (d,a)\in\mathcal{A}_{transfer}, && (5)\\
& \sum_{p\in\Omega^v}\delta_p^v = 1, \quad \forall v\in\mathcal{V}, \sum_{q\in\Theta^m}\lambda_q^m = 1 && \forall m\in\mathcal{M}, && (6)\\
& \delta_p^v\in\{0,1\},\ \forall v\in\mathcal{V}, \forall p\in\Omega^v, \quad \lambda_q^m\in\{0,1\}, && \forall m\in\mathcal{M}, \forall q\in\Theta^m. && (7)
\end{aligned}
$$

# 3 A column generation approach

To solve the RPM introduced in Section 2, we develop a column generation heuristic, taking into account the independent subproblems associated with each tractor and implement. Each tractor subproblem is an Elementary Shortest Path Problem with Resource Constraints (ESPPRC) incorporating linear costs [6]. This subproblem considers two resources: the distance, restricted by vehicle autonomy, and the time, constrained by task time windows. Since the linear costs only depend on the transfer arcs, as shown in equation (5), we discretize the time to visit those nodes and arcs, by adding duplicated nodes with fixed time windows. Each tractor $v \in \mathcal{V}$ initiates its route from a given depot $s^v \in \mathcal{N}_{depots}$ and ends at the depot $e^v \in \mathcal{N}_{depots}$ by the end of the planning horizon. The implement subproblems are also ESPPRCs, but they only consider the demand constraints. The demand constraint depends on the implements and the tasks that are compatible with it. Some implements do not have a capacity, such as pruning or ploughing, some have a small capacity and need to be recharged (e.g., implements associated with fertilizer spreading), and some have a large capacity and cannot be recharged. For each type of implements, we use a specific optimized ESPPRC algorithm.

Leveraging the distinctive implementation of the subproblem for each vehicle type, the column generation approach proves highly suitable for solving it in a distributed and asynchronous manner, as outlined in [7], improving the convergence speed. Finally, an upper bound is obtained by solving the integer problem with the columns generated so far.

# 4 Preliminary computational results

Table 1: Preliminary computational results

| Instance | | | MIP model | | | Column generation | | |
|---|---|---|---|---|---|---|---|---|
| $\lvert\mathcal{V}\rvert$ | $\lvert\mathcal{I}\rvert$ | $\lvert\mathcal{N}_{tasks}\rvert$ | UB | LB | t(s)/gap(%) | UB | gap(%) | t(s) |
| 5 | 5 | 30 | 418 | 401 | 4.23 % | 418 | 4.23 % | 1 s |
| 5 | 5 | 30 | 389 | 353 | 10.19 % | 392 | 11.04 % | 7 s |
| 5 | 5 | 30 | 383 | - | 200 s | 383 | 0.00 % | 32 s |
| 5 | 5 | 40 | 446 | 419 | 6.44 % | 487 | 16.22 % | 40 s |
| 5 | 5 | 40 | 412 | 340 | 21.17 % | 425 | 25 % | 2 s |
| 5 | 5 | 40 | 425 | 412 | 3.15 % | 449 | 8.98 % | 6 s |
| 5 | 5 | 50 | 492 | - | 566 s | 503 | 2.23 % | 92 s |
| 5 | 5 | 50 | 460 | 411 | 11.92 % | 484 | 17.7 % | 60 s |
| 5 | 5 | 50 | 436 | 428 | 1.86 % | 481 | 12.3 % | 64 s |
| 5 | 10 | 30 | 647 | - | 2831 s | 647 | 0.00 % | 27 s |
| 5 | 10 | 30 | 659 | - | 20 s | 659 | 0.00 % | 18 s |
| 5 | 10 | 30 | 580 | - | 1650 s | 582 | 0.34 % | 252 s |
| 5 | 10 | 40 | 705 | - | 2253 s | 705 | 0.00 % | 136 s |
| 5 | 10 | 40 | 653 | - | 295 s | 653 | 0.00 % | 12 s |
| 5 | 10 | 40 | 679 | - | 409 s | 679 | 0.00 % | 90 s |
| 5 | 10 | 50 | 755 | - | 3027 s | 758 | 0.39 % | 262 s |
| 5 | 10 | 50 | 673 | 664 | 1.35 % | 674 | 1.5 % | 396 s |
| 5 | 10 | 50 | 623 | 611 | 1.96 % | 630 | 3.1 % | 212 s |

Table 1 shows preliminary results obtained by running our asynchronous column generation algorithm on 18 small instances. The instances differ in the distances between tasks and the compatibilities of vehicles, implements and tasks. We compare our results with those obtained by solving a MIP formulation with the commercial solver Gurobi 10.0.2. We report the upper and lower bounds returned by Gurobi after one hour of CPU time, and the upper bound returned by our column generation approach. We also indicate the computational time and the optimality gap for both methods. The optimality gap is computed as $\frac{UB-LB}{LB} \cdot 100$ when the optimal solution is not obtained with the MIP model. Our column generation approach obtains solutions faster for all instances and close to the Gurobi solutions in most instances. However, the efficiency tends to decrease when the number of tasks increases and the number of implements is limited. As a future research direction, we plan to improve the efficiency of our algorithm on large instances to outperform the solution of the MIP with a solver. Last, we aim to analyze how different capacities, time windows and compatibility parameters influence the exchange of vehicles and the quality of solutions.

## References


[1] R. Soares, A. Marques, P. Amorim, and S. N. Parragh, "Synchronisation in vehicle routing: classification schema, modelling framework and literature review," *European Journal of Operational Research*, 2023.

[2] M. Drexl, "Synchronization in vehicle routing—a survey of vrps with multiple synchronization constraints," *Transportation Science*, vol. 46, no. 3, pp. 297–316, 2012.

[3] M. Drexl, "Applications of the vehicle routing problem with trailers and transshipments," *European Journal of Operational Research*, vol. 227, no. 2, pp. 275–283, 2013.

[4] M. Drexl, "Branch-and-cut algorithms for the vehicle routing problem with trailers and transshipments," *Networks*, vol. 63, no. 1, pp. 119–133, 2014.

[5] B. Domínguez-Martín, I. Rodríguez-Martín, and J.-J. Salazar-Gonzalez, "The driver and vehicle routing problem," *Computers & Operations Research*, vol. 92, pp. 56–64, 2018.

[6] C. Tilk, N. Bianchessi, M. Drexl, S. Irnich, and F. Meisel, "Branch-and-price-and-cut for the active-passive vehicle-routing problem," *Transportation Science*, vol. 52, no. 2, pp. 300–319, 2018.

[7] S. Basso and A. Ceselli, "Distributed asynchronous column generation," *Computers & Operations Research*, vol. 146, p. 105894, 2022.

# Reoptimization in Picker-to-Parts Warehouses in E-Commerce: Asymptotic Analysis

**Catherine Lorenz**
Chair of Management Science / Operations and Supply Chain Management
University of Passau, Passau, Germany

**Alena Otto**
Chair of Management Science / Operations and Supply Chain Management,
University of Passau, Passau, Germany, Email: alena.otto@uni-passau.de

**Michel Gendreau**
Department of Mathematics and Industrial Engineering and CIRRELT,
Polytechnique Montréal, Montréal, Canada

## 1 Motivation

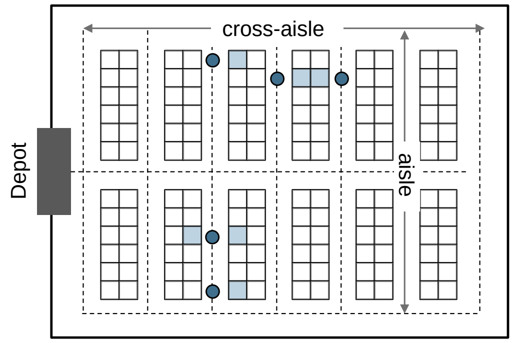
The spread of express deliveries forms a major trend in e-commerce logistics. For operational planning of order picking in warehouses this essentially means that the orders arrive 'on-the-fly' and dynamic, adjustable policies are required to keep the tight delivery promise. One of most prominent dynamic policies for order picking proposed to date is *reoptimization (Reopt)*, which optimizes picking operations each time, a new order arrives [cf. 1, 2]. However, very little is known about the performance gap of *Reopt* to optimality in the dynamic setting of order picking operations. It is important to close this gap.

In this paper, we analyze a widespread setting of e-commerce warehouses — picker-to-parts warehouses with a picker and a pushcart (Figure 1a). For the arising *Online Order Batching, Sequencing, and Routing Problem (OOBSRP)*, which is defined Section 2, we show that *Reopt* is almost surely asymptotically optimal under quite general assumptions (Section 3). Moreover, our experiments in Section 4 illustrate that *Reopt* is close to optimality already for small instances. We provide an outlook in Section 5.

## 2 Outline of the problem

We consider a zone of a warehouse with a single picker. The area has a standard rectangular grid design (Figure 1b). It consists of two or more cross-aisles as well as of several aisles that divide the shelves and provide entry to the picking locations. The picker, equipped

(a) Pushcart




(b) A typical warehouse layout

*Note.* A warehouse with three cross-aisles and seven aisles. Access points for visualized selected picking locations are marked with circles.

Figure 1: Illustration of the considered e-commerce warehouse setting

with a pushcart, collects the items ordered by customers and delivers them to the *depot*, located at an arbitrary position in the area. In a single tour, which starts and ends in the depot, the picker can serve up to $c \in \mathbb{N}$ orders at a time, since the pushcart is equipped with $c$ bins, each of which can harbor the items of a single order. Observe that, once started, the order has to be completed within the same tour. Each *item* is associated with exactly one *picking location*, so that we use these terms interchangeably.

We denote an OOBSRP *instance* with $n \in \mathbb{N}$ orders as $I(n)$. The orders arrive dynamically and form independent random vectors $R = (R_1, R_2, .., R_n)$ of *arrival times* and $O = (O_1, O_2, .., O_n)$ of *picking locations*. We assume that orders $O_j, j \in \{1, \dots, n\}$, in the sequence $O$ are *independent and identically distributed (i.i.d.)* multivariate random variables $O_j = (K_j, S_j^1, ..., S_j^{K_j})$. Thereby, $K_j$ describes the number of items in the $j^{\text{th}}$ order, which is unknown in advance; $(S_j^1, ..., S_j^{K_j})$ are locations that the picker has to visit.

The objective is to minimize the *makespan*, i.e., the time to collect all the ordered items and return to the depot. *Reopt* follows an optimal picking plan for the currently known set of not yet completed orders by taking the following decisions: *(i)* partitioning orders into mutually disjoint sets *(batches)* of at most $c$ orders each, which are picked within one tour; *(ii)* sequencing these batches; *(iii)* for each tour *(batch)*, routing the picker to collect all the items of the respective orders by starting and ending in the depot. Since *Reopt* reoptimizes at each arrival of a new order, it has to respect the current position of the picker and, if the picker is in the field, the number of occupied bins as well as orders with already partially picked items. For more details on OOBSRP, see the survey of [3].

## 3 Analytical results

We will discuss two different models for the arrival times of orders. The first modeling scheme attributes the *order statistics property* to the arrival times and can represent many stationary or variable demand patterns that realistically occur in e-commerce. The second model represents the prevalent method for modeling stationary demand, employing a homogenous Poisson process to characterize the order arrivals.

**Model 1.** *The arrival times of the orders are i.i.d. realizations* $Y_j, j \in [n]$ *of a generic continuous random variable* $Y \geq 0$ *with an arbitrary, given distribution and mean* $\mu_Y < \infty$. *Thereby, the* $i^{th}$ *arrival time is the* $i^{th}$ *order statistic:* $R_j = Y_{(j)}$, $Y_{(1)} \leq Y_{(2)} \leq ... \leq Y_{(n)}$.

For instance, considering $Y \sim uniform([0,t])$ allows to model constant arrival patterns over one working shift of length $t$, while assigning a normal- or multimodal distribution to $Y$, enables modeling demand with one or several peak-times.

For an instance $I(n)$, let denote *Reopt*$(I(n))$ and $CIOPT(I(n))$ the random variables that represent the makespan provided by *Reopt*, and the optimal makespan achievable when all orders and their arrivals are known a priori (*Complete Information Optimum*), respectively.

**Theorem 3.1.** *Given Model 1 for the arrival times and the stochastic assumptions from Section 2, Reopt is almost surely (a.s.) an asymptotically optimal policy:*

$$\lim_{n\to\infty} \frac{\mathit{Reopt}(I(n))}{CIOPT(I(n))} = 1 \quad a.s.$$

*Proof.* A detailed proof of Theorem 3.1 will be provided during the talk.

**Model 2.** *The order arrival times follow a homogeneous Poisson process of rate* $\lambda$.

In a technical proof combining Theorem 3.1 with the *Order Statistics Property*, we could show that *Reopt* stochastically converges to an optimal policy as the order volume increases in a fixed-duration working shift, assuming Model 2 for order arrivals. Independently, we could establish the asymptotic a.s. optimality of *Reopt* for Model 2, given a sufficiently small arrival rate $\lambda$ ensuring controlled growth of uncompleted order queues.

## 4 Empirical results

In this section, we empirically verify the hypothesis that *Reopt* consistently converges to an optimal policy *with high probability* for Poisson order arrivals with *any* rate, as the number of orders increases. Additionally, we show that the *speed of convergence* is fast.

We use a dynamic programming formulation to receive the solutions of both *Reopt* and $CIOPT$. In total, 300 instances of different size $n$ have been solved, assuming orders

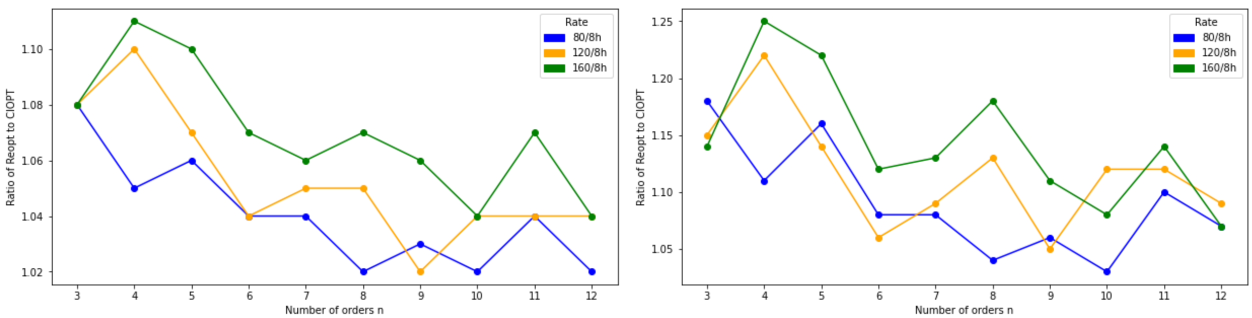



Figure 2: Average- (left) and worst- (right) observed ratio of *Reopt* to $CIOPT$

arrive according to Model 2 with three different rates, 80/120/160 orders per 8 h (100 instances each). A warehouse of dimensions 50 x 69 m, with 10 aisles and 3 cross-aisles was considered. Each order contains a random amount of 1-6 items whose picking locations are scattered uniformly at random in the storage area, requiring a picking time of 10 secs. each. The picker walks at a speed of 0.5 m/s, pushing a cart with batching capacity $c = 2$. The graphs in Figure 2 show the average- and worst ratios $\frac{Reopt(I(n))}{CIOPT(I(n))}$ with the increasing number of orders $n$.

## 5 Conclusion

Our analytical results show an excellent asymptotical performance of the *Reopt* policy in relevant-for-practice settings. Our computational experiments illustrate that the speed of convergence of *Reopt* is fast. In future research, we assess the *worst*-case performance of *Reopt* and examine further picking systems.

## References

[1] W. Lu, D. McFarlane, V. Giannikas, and Q. Zhang, "An algorithm for dynamic order-picking in warehouse operations," *European Journal of Operational Research*, vol. 248, no. 1, pp. 107–122, 2016.

[2] V. Giannikas, W. Lu, B. Robertson, and D. McFarlane, "An interventionist strategy for warehouse order picking: Evidence from two case studies," *International Journal of Production Economics*, vol. 189, pp. 63–76, 2017.

[3] E. G. Pardo, S. Gil-Borrás, A. Alonso-Ayuso, and A. Duarte, "Order batching problems: Taxonomy and literature review," *European Journal of Operational Research*, vol. 313, no. 1, pp. 1–24, 2024.

# Multi-objective optimization model for a sustainable closed-loop supply chain of the returnable packaging sector considering Extended Producer Responsibility

**Jonathan Lozano-Oviedo**

PhD Program in Engineering Systems

Universidad de Chile, Beauchef 851, Santiago, Chile.

E-mail: jonathanlozanoviedo@gmail.com

**Cristian E. Cortés**

Department of Civil Engineering

Universidad de Chile and Instituto Sistemas Complejos de Ingeniería (ISCI), Blanco Encalada 2002, Santiago, Chile.

E-mail: ccortes@ing.uchile.cl

**Pablo Maya-Duque**

Department of Industrial Engineering

Universidad de Antioquia, Medellín, Colombia.

E-mail: pablo.maya@udea.edu.co

**Pablo A. Rey**

Departamento de Industria, Facultad de Ingeniería and Instituto Universitario de Investigación y Desarrollo Tecnológico (IDT), Universidad Tecnológica Metropolitana, Santiago, Chile.

Email: prey@utem.cl

**Juan G. Villegas**

Department of Industrial Engineering

Universidad de Antioquia, Medellín, Colombia.

E-mail: juan.villegas@udea.edu.co

## 1. Introduction

Sustainable closed-loop supply chains (SCLSCs) are gaining prominence, partially motivated by the emerging measures and guidelines that governments and organizations are adopting in the issues of sustainability (economic, environmental and social), the circular economy (closing the cycle with mechanisms such as, reuse, remanufacturing, repair, among others.), and Extended Producer Responsibility (EPR). According to [1], one of the main sources of waste generation within the supply chain is packaging. Moreover, as pointed out in [2], most packaging is thrown away after a single use. The use of returnable packaging embraces the concept of a closed-loop supply chain and maximizes the lifecycle of packaging materials. Some of the benefits of using returnable packaging, as identified by [3] and [4], include the reduction of packaging material waste and $CO_2$ emissions, while increasing the efficiency of packaging product usage. However, the management of returnable packaging systems demands costly reverse logistics operations, such as collection after use, storage, cleaning, sorting, and recycling [5].

Despite being a promising alternative, Sustainable Closed-Loop Supply Chains (SCLSCs) have received limited attention in terms of economic, environmental, and social sustainability in the returnable packaging sector. Therefore, there exists a research gap that merits more in-depth exploration [6]. Specifically, optimization techniques can contribute to supporting decision-making at the strategic and tactical levels within returnable packaging SCLSCs. Furthermore, they could aid in addressing and modelling specific economic, social, and environmental goals in the medium and long terms.

Several publications in the literature have explored the optimization of returnable packaging Sustainable Closed-Loop Supply Chains (SCLSCs) (e.g., [1], [2], [5], [6]). However, these studies predominantly emphasize economic and environmental sustainability goals, neglecting the inclusion of social sustainability. The model proposed in this paper addresses this relevant aspect unexplored in the relevant literature. Furthermore, most identified studies concentrate on secondary and tertiary packaging, referring to the packaging used for consolidating and transporting products. In contrast, this paper focuses on primary packaging, which typically extends throughout the supply chain until reaching the retailer or the end consumer.

## 2. Problem description

The model proposed in this paper addresses the strategic and tactical decisions involved in designing a returnable packaging Sustainable Closed-Loop Supply Chain (SCLSC) for a bottled water company within the regulatory framework of Extended Producer Responsibility (EPR). It considers goals across the three main sustainable dimensions: economic (total costs), environmental (packaging returnability rate), and social (job opportunities). These objectives are framed within a set of circular economy mechanisms, including recycling, reuse, and energy co-processing.

The stability of the model is tested by developing a case study inspired by the bottled water beverage industry in Colombia, analysing the results obtained based on the recently introduced EPR policy for returnable plastic packaging. The model includes the following terms in the objectives functions: (objective cost function $Z_1$) raw material purchase costs, inventory holding costs, total facilities costs, total transportation costs, income caused by the deposits charged to the producer-packer for packaging not returned to the chain and penalties for non-compliance with rates for the recovery of returned packaging waste; (social-environmental objective function $Z_2$) job opportunities and a set of binary variables associated with compliance with packaging returnability rates, as illustrated below:

Minimize

$$
\begin{aligned}
Z_1 = \sum_{i\in F}\sum_{j\in M}\sum_{p\in P}\sum_{t\in T:t>0} RMC_{ipt}x_{ijpt} \\
+ \sum_{j\in M\cup D\cup C}\sum_{p\in P}\sum_{t\in T:t>0} IC_{jpt}s_{jpt} + \sum_{i,j\in A_r}\sum_{p\in P}\sum_{t\in T:t>0} CO_{jpt}r_{ijpt} + \\
+ \sum_{i,j\in A_d}\sum_{p\in P}\sum_{t\in T:t>0} \beta_p TC_t D_{ij}x_{ijpt} + \sum_{i,j\in A_r}\sum_{p\in P}\sum_{t\in T:t>0} \beta_p TC_t D_{ij}r_{ijpt} \\
- \sum_{j\in C}\sum_{t\in T:t>0}\left(\sum_{p\in\tilde{P}} IP_{pt}w_{jpt} - \sum_{i\in L}\sum_{p\in\tilde{P}} IP_{pt}r_{jipt}\right) + \sum_{\tau\in\mathcal{T}} \mu_\tau y_\tau
\end{aligned} \tag{1}
$$

Maximize

$$
Z_2 = \sum_{i,j\in A_r}\sum_{p\in P}\sum_{t\in T:t>0} Vjo_j\beta_p r_{ijpt} - \sum_{\tau\in\mathcal{T}} y_\tau \tag{2}
$$

The model allows, among other aspects, to generate a gradual participation by type of packaging according to what is most convenient for the target functions, as shown in constraints (3); this is an important modelling feature considering that there are packages that are more or less convenient for SCLSC due to their material and possibility of returning to close the cycle:

$$
\left|\sum_{j\in C} w_{jpt} - \sum_{j\in C} w_{jp,t-1}\right| \le \mu \sum_{j\in C}\sum_{\hat{p}\in P} w_{j\hat{p},t-1} \quad \forall\, p \in P, t \in T: t > 0 \tag{3}
$$

### 3. Case study results

In the case study with a time horizon of five years, there are 3 glass packaging factories, 1 plastic packaging factory (Pet), 3 glass recyclers and 1 plastic recycler, 11 water bottling plants, 10 distribution centres, 11 retail centres, 5 returned packaging waste managers, 11 waste disposal centres, and 6 waste co-processing centres. Three types of packaging are defined: returnable (glass and Pet-reuse) and non-returnable (Pet-no reuse).

The value achieved in the objective function of total costs presents a deviation above the goal of 3.02%. The distribution of cost components presents the following distribution: purchase of raw materials (77.28%); total facility operation costs (18.80%); and the rest distributed in total transportation costs among the links of the chain. The income generated by the deposits received by the producer-packer for containers not returned to the chain allows a 38% reduction in total costs.

Transportation costs are mainly concentrated in: producers-bottlers to distributors (13.5%); distributors to retailers (19.1%); retailers towards returned packaging waste managers (37.3%); and from the returned packaging waste managers to the producers-bottlers (27.7%), however, the transportation cost between packaging manufacturers and producers-bottlers is only 1.2% of the total transportation cost, plus a negligible remaining part due to other minor transportation operations. From these results, we notice that the reuse of returned packaging is a dominant circular strategy for satisfying demand.

Approximately 51 job opportunities are generated, especially concentrated in producer-bottlers, due to the circular packaging reuse strategy. Regarding packaging EPR policies, the annual rates of utilization of returned packaging waste throughout the planning horizon are largely met when comparing the goal values in the law and the obtained values: (First year: 59% above the goal), (Second year: 51.14% above the goal), (Third year: Real 62.75% above the goal), (Fourth year: 69.83% above the goal), (Fifth year: 72.51% above the goal). These results are consistent with the absence of penalty costs for non-compliance with utilization rates. Finally, most of the Pet-reuse packaging waste that returns to the chain is assigned to producers-bottlers to be reused contributing to the satisfaction of the demand.

## 4. Conclusions

In this context, the reuse of returnable plastic packaging made of Pet material appears to be the most suitable alternative for the studied SCLSC, which belongs to the bottled water sector. This is due to the fact that this type of returnable packaging can respond particularly well to the sustainability objectives by guaranteeing compliance with the utilization rates of the EPR scheme, reducing the underlying total costs and simultaneously generating new employment opportunities. The proposed multi-objective, multi-packaging model represents a useful tool for the planning of supply chains that are required by governments to seriously consider the emerging EPR laws linked to circular economy models and the sustainable development goals set by the UN.

## References


[1] B. Sarkar, M. Ullah y N. Kim, “Environmental and economic assessment of closed-loop supply chain with remanufacturing and returnable transport items”, *Computers and Industrial Engineering*, vol. 111, pp. 148–163, 2017. https://doi.org/10.1016/j.cie.2017.07.003

[2] B. Sarkar, M. Tayyab, N. Kim y M. S. Habib, “Optimal production delivery policies for supplier and manufacturer in a constrained closed-loop supply chain for returnable transport packaging through metaheuristic approach”, *Computers and Industrial Engineering*, vol. 135, 987–1003, 2019. https://doi.org/10.1016/j.cie.2019.05.035

[3] R. Carrasco-Gallego, E. Ponce-Cueto y R. Dekker, “Closed-loop supply chains of reusable articles: a typology grounded on case studies”, *International Journal of Production Research*, vol. 50, n.º 19, 5582–5596, 2012. https://doi.org/10.1080/00207543.2011.649861

[4] C. H. Glock, “Decision support models for managing returnable transport items in supply chains: A systematic literature review”, *International Journal of Production Economics*, vol. 183, pp. 561–569, 2017. https://doi.org/10.1016/j.ijpe.2016.02.015

[5] R. Accorsi, G. Baruffaldi y R. Manzini, “A closed-loop packaging network design model to foster infinitely reusable and recyclable containers in food industry”, *Sustainable Production and Consumption*, vol. 24, 48–61, https://doi.org/10.1016/j.spc.2020.06.014

[6] Y. Zhang, A. Che y F. Chu, “Improved model and efficient method for bi-objective closed-loop food supply chain problem with returnable transport items”, *International Journal of Production Research*, 1-18, 2020. https://doi.org/10.1080/00207543.2020.1851057

# Touting occasional drivers for mid-haul delivery

**Simona Mancini**
University of Palermo, Palermo, Italy
Email: simona.mancini@unipa.it - simona.mancini@aau.at

**Margaretha Gansterer**
University of Klagenfurt, Austria

**Chefi Triki**
University of Kent, Canterbury, UK

## 1 Motivation and Background

The fashion designer outlet market became more and more popular in the last decades with dozens of shopping malls newly opened in Europe alone. Such centers are shopping villages with hundreds of fashion brand stores offering outlet stocks at strongly discounted prices. They are generally located in rather remote places, far from city centers but near to highways in order to be easily accessible by several cities in the surrounding. Recently, outlet villages started to offer online purchasing services combined with home delivery. However, the cover area of such services is wide and the company might be requested to perform deliveries in a range of 200-300 kms, which bears the risk to lose profitability. The idea behind this work is to exploit in-store customers to perform mid-haul deliveries on their way back home. This would require small detours to serve customers which are even very far from the outlet with a huge costs reduction for the company. The drawback of the system is that the presence of in-store customers willing to serve a specific area is strongly affected by uncertainty. In order to decrease the negative effect of this uncertainty, we propose to adopt a *touting* strategy to incentive potential in-store customers (i.e., customers who periodically visit the outlet) to come to the outlet on a specific day and to perform, if needed, a mid-haul delivery, by offering them a discount voucher. By accepting this offer the customers guarantee their availability to perform deliveries. In addition, the company may exploit customers who express their interest to act as occasional drivers (ODs), directly on site. Such customers may make bids to cover one or more areas. After receiving all the bids, the company decides through an auction system which ones to accept. Delivery orders which are not covered neither by touted ODs nor by same-day ODs, must be served by an owned fleet before the end of the day. While the service currently implemented in practice does not allow for same-day requests and does not guarantee the delivery within a specific deadline, we think that to improve the quality of service it would be important to allow the possibility of placing orders on short notice and to request same-day delivery for an additional fee. In this case, customers' demand would become an uncertain parameter as well, since only a subset of requests would be known in advance (i.e., before making the touting offers) whereas the remaining ones are revealed the day after, just before clearing the auction for same-day ODs. In order to simplify ODs' tasks and to increase their participation, they do not have to perform door-to-door delivery but all the requests are clustered according to centroids and must be delivered to a parcel locker or a collection point located in a strategic point of the city, such as near to the highway access. This allows also customers to quickly and easily perform deliveries not only in their home town but also to other cities located along their paths. Thus, each request is associated with a centroid, and touting offers as well as same-day OD bids are related to these centroid and not to specific requests. Furthermore, each OD (touted or same-day) can serve a given number of requests, depending on the size of the vehicle. However, we assume that only one centroid is visited. The goal of this work is to propose a mixed delivery system involving touted ODs, same-day ODs, and owned fleet and to show that this system allows to considerably reduce the company's delivery cost with respect to a pure owned fleet based system. Furthermore, we analyze the benefit of the touting strategy on the overall system performance. The term *touting*, is not completely new in the logistics literature, since it has been introduced by [1]. However, the authors use it in a completely different context and with different tools. They address the multi-period vehicle routing problem and ask customers to anticipate if an order fits with the current delivery routing plan. In their case, the company is asking the customers to collaborate by just increasing their flexibility on the delivery date, but without offering them any compensation. Instead, in our problem the company asks the customers to perform a service and offers a compensation in the form of a voucher for future purchases. While the idea of exploiting in-store customers to perform deliveries on their way back home has been already investigated in the literature (e.g., [3], [2]), the exploitation of incentives to push customers to make themselves available to perform mid-haul deliveries, has not been addressed yet.

## 2 Problem Description

The resulting decision problem can be described as follows. We have a single depot 0 representing the outlet village and a set of centroids $J$ for which the initial demand is known ($d_j$). A set of frequent customers, $I$, is available for touting. To simplify the notation, we refer to them as *touted ODs* (TODs). A set of voucher options, $V$, of different values $p_v$ are available. At most one voucher per OD can be offered. The corresponding price is actually paid by the company only in case the offer is accepted. Furthermore, once a touting offer is accepted, the voucher is paid even in case the delivery service is not needed. This allows to tout even centroids for which we initially have no demand but same-day requests are expected. For each TOD, $i$, the maximum number of requests, $c_i^t$, the home location, and the average range of spending are known. The last two parameters are used to determine the probability that this OD would accept a touting offer. In fact, the average range of spending is an indicator of the customer's income, where customers with a lower income are assumed to be more likely to be available to perform deliveries. The home location is used to determine the detour required to serve a specific centroid. We assume that the acceptance probability decreases with increasing detour. We further assume that same-day customers' demand for each centroid is directly proportional to the population of the centroid. Based on this data, a set of scenarios $S$ is considered in which, the same-day demand for each centroid $\delta_j^s$ and a binary constant $\alpha_{ijv}^s$, indicating whether a TOD $i$ accepts to serve centroid $j$ for a voucher $v$ on scenario $s$ are defined. Additionally, for each scenario we suppose to receive a set of OD bids $K^s$, where for each bid $k$ we know the occasional driver $o_k$, the maximum number of requests of this OD $c_k$, the centroid for which it has been submitted $\tau_k$, and the price offered $b_k$. At most one bid for each same-day OD, $\omega$, can be accepted. We assume that the auction to assign request to ODs take place in the early afternoon, in order to leave enough time for same-day ODs (SODs) to submit bids. Therefore, the time for delivery for the owned fleet is limited to a few hours in the end of the day. For the latter, we then generate the set of all feasible paths, $H$, visiting a subset of centroids and coming back to the outlet within its opening time. The cost of covering a path $h$ is indicated as $\gamma_h$. Since centroids are supposed to be far away from each other and the time for delivery rather limited, the number of feasible paths is relatively small. Let us define $\lambda$ as an arbitrary large constant. Let us further introduce the following binary decision variables. $T_{ijv}$ indicates if a voucher $v$ is offered to OD $i$ to serve centroid $j$, while $Z_{ijv}^s$ states if this offer is accepted or not. Decision variable $Y_k^s$ indicates if the $k^{th}$ bid submitted in scenario $s$ is selected, while $U_j^s$ states if a centroid is visited by the owned fleet or not. Finally, $W_h^s$ states if path $h$ is covered by the owned fleet in scenario $s$. All the variables involved are binary. The decision problem can be formulated as a two-stage stochastic programming model as follows.

$$\min \sum_{i\in I}\sum_{j\in J}\sum_{v\in V}\sum_{s\in S} p_v Z_{ijv}^s + \sum_{s\in S}\sum_{k\in K^s} b_k Y_k^s + \sum_{h\in H}\sum_{s\in S} \gamma_h W_k^s \tag{1}$$

$$\sum_{i\in I}\sum_{v\in V} c_i^t Z_{ijv}^s + \sum_{k\in K^s|\tau_k=j} c_k Y_k + \lambda \sum_{h\in H} \mu_{hj} W_h^s \geq d_j + \delta_j^s \quad \forall j\in J \;\; \forall s\in S \tag{2}$$

$$Z_{ijv}^s = \alpha_{ijv}^s T_{ijv} \quad \forall i\in I \;\; \forall j\in J \;\; \forall v\in V \;\; \forall s\in S \tag{3}$$

$$\sum_{i\in I}\sum_{v\in V} T_{ijv} \leq 1 \quad \forall i\in I \tag{4}$$

$$\sum_{k \in K^s | o_k = \omega} Y_k \leq 1 \quad \forall \omega \in \Omega \;\; \forall s \in S \tag{5}$$

The objective function, reported in (1), minimizes the total cost for the company, given by the sum of the touting costs, the same-day bids acceptance cost and the owned fleet cost. Constraints (2) guarantee that the demand is covered in every scenario. A touting offer can be accepted only if it has been proposed to the OD (3). Each OD can receive at most one touting offer (4) and at most one bid per SOD can be accepted (5).

## 3 Preliminary Results and Discussion

We apply the model to instances based on a real-world case involving an outlet village located in the North-West of Italy. The obtained results show the benefit of applying the proposed mixed distribution system rather than making use of the owned fleet only. We observe that allowing only SODs decreases costs by 32.74%, while using TODs only has a reduction effect of 48.49%. However, the most profitable strategy is to combine both of them reducing costs by up to 67.13%. In our instances, three types of touting vouchers are considered: low (€30), medium (€50) and high (€70). Results indicate that, in the optimal solution, most of the offered vouchers (83%) are low, while only 11.3% and 5.6% are medium and high, respectively. Hence, if the required detour to serve a centroid is small, a low voucher is sufficient to attract the customers interest. The touting strategy plays an important role since 71% of the frequent customers available are actually touted and 65% of them accept the touting offer. The most touted centroids are those with the highest population (which are more likely to receive same-day demand) and the most remote ones as for these, obviously, delivery cost by the owned fleet is very high. More results, obtained on different types of road networks will be presented at the conference. Different deterministic policies for the touting decisions will be discussed and we will benchmark their performance against the stochastic model.

## References


[1] M. Keskin, J. Branke, V. Deineko, A.K. Strauss, "Dynamic multi-period vehicle routing with touting", *European Journal of Operational Research* 310(1), 168-184, 2023.

[2] S. Mancini, M. Gansterer, "Bundle generation for last-mile delivery with occasional drivers", *Omega* 108, 102582, 2022.

[3] R. Ausseil, M. W. Ulmer, J. A. Pazour, "Online acceptance probability approximation in peer-to-peer transportation", *Omega* 123, 102993, 2024.

# Data-driven distributionally robust approach for the joint chance-constrained location routing problem with uncertain demands

**Guillaume Massonnet**
LS2N, IMT Atlantique, Nantes, France
Email: guillaume.massonnet@imt-atlantique.fr

**Juliette Medina**
CRC Services, Nantes, France
Email: j.medina@crc-services.com

## 1 Introduction and related work

Fresh food supply chains generally handle products with very limited shelf-life and thus require just-in-time production and transportation from the suppliers to the retailers. In order to meet their clients' requirements for fast deliveries, it is critical for suppliers to anticipate their supply chain organization at a strategic and tactical level to provide efficient and responsive services.

This work originates from the practical case faced by a french food producer with several facilities that supplies multiple products to the warehouses of its clients on a daily basis. The transportation of these items is subcontracted to external carriers and is performed by vehicle routing, each starting from a single facility and delivering a few (generally between 2 and 3) clients. At the time this organization is decided, real demands are unknown and susceptible to vary from day to day since clients may place their orders as late as a few hours before delivery is due. As making last-minute search for additional vehicles is both costly and risky (or even occasionally impossible), the supplier's goal is to build a predefined vehicle routing plan, identical every day, that describes which facilities and which routes are used to deliver each type of product to its clients. The objective is then to propose a plan that offers enough production and transportation capacity to satisfy the clients demands with a high probability at minimum operating costs.

We refer to the above problem as the *Multi Products Capacitated Location Routing Problem (MCLRP)* with stochastic demands, which generalizes the Location Routing Problem (see [1, 2] for detailed references). We model the target service level of the supplier for full demand satisfaction as a set of constraints that must all be satisfied with a predefined probability. Unfortunately in practice, one only has access to an empirical estimation of the true underlying distribution of random parameters through the historical data available. that allows approximate these probabilistic constraints. However when data is scarce or imprecise, standard approaches such as the Sample Average Approximation (SAA) are susceptible to achieve poor out-of-sample performances. Instead, we address this problem using a *distributionaly robust* data-driven formulation based on Wasserstein balls (see [3, 4]), which ensures more consistent results even when the data sample is limited size. We provide first numerical evidences of its effectiveness on practical cases derived from real data sets.

## 2 Mathematical formulation

In what follows, we use boldface lower-case (resp. upper-case) letters to designate vectors (resp. matrices). Let $\mathcal{I} = \{1, \ldots, I\}$ be the set of sites available to open a facility, $\mathcal{K} = \{1, \ldots, K\}$ be the set of products and $\mathcal{J} = \{1, \ldots, J\}$ be the set of final customers. The demand of a given client $j \in \mathcal{J}$ for product $k \in \mathcal{K}$ is modeled as a random variable $\tilde{\xi}_{jk}$ that may be correlated with the demand of other customers. The joint probability distribution $\mathbb{P}$ of $\tilde{\boldsymbol{\xi}} = (\tilde{\xi}_{11}, \ldots, \tilde{\xi}_{1K}, \ldots, \tilde{\xi}_{J1}, \ldots, \tilde{\xi}_{JK})$ is unknown but we assume that we have access to a historical data set of $N$ past demands $\hat{\boldsymbol{\xi}}^{(1)}, \ldots, \hat{\boldsymbol{\xi}}^{(N)}$ drawn independently from $\mathbb{P}$. For all customers $j \in \mathcal{J}$, we denote $\mathcal{K}_j$ the set of items that it may order, i.e. all $k \in \mathcal{K}$ such that $\mathbb{P}(\tilde{\xi}_{jk} > 0) > 0$.

Each facility $i \in \mathcal{I}$ can produce and ship all types of products up to a limited capacity $S_i$ and incurs a setup cost $f_i$ when it is opened. Delivery to the customers is perfomed using a enumerable set of candidate routes $\mathcal{R} = \{1, \ldots, R\}$, each starting from a single facility $i \in \mathcal{I}$ and visiting a subset of (generally up to 3) customers $\mathcal{J}_r \subseteq \mathcal{J}$. For all $i \in \mathcal{I}$, we denote $\mathcal{R}_i$ the subset of routes that start from facility $i$. Split delivery is allowed, i.e. demand $\tilde{\xi}_{jk}$ may be satisfied using one or more vehicles, from one or more of the opened facilities. In addition, a route $r \in \mathcal{R}$ may be used by up to $V_r$ vehicles, each with the same capacity $Q_r$. Any vehicle using route $r$ induces a transportation cost $c_r$.

The objective is to find a solution of minimum setup and transportation cost while ensuring that the totality of the demands is satisfied with probability at least $1 - \epsilon$. We define three types of decision variables: $x_r \in \{0 \ldots, V_r\}$ is the number of vehicles that use route $r \in \mathcal{R}$ (maximum $V_r$ vehicles), $y_i \in \{0; 1\}$ is a boolean variable that indicates whether facility $i$ is open and $z^r_{jk} \in [0, 1]$ is the fraction of demand of customer $j$ for item $k$ that is served by a vehicle using route $r$.

### 2.1 Joint chance-constrained (JCC) formulation

Let $\mathcal{X}$ be the set of variables values that satisfy jointly the deterministic constraints of the problem, that is: 1) Any selected route $r$ must start from an opened facility and be used at most $V_r$ vehicles, 2) transport is only allowed using selected routes, and 3) the sum of deliveries to customer $j$ is sufficient to cover its demand for all products in $\mathcal{K}_j$:

$$\mathcal{X} = \left\{ (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \in \mathbb{N}_+^R \times \{0,1\}^I \times \mathbb{R}_+^{RJK} : \begin{array}{rl} x_r \leq V_r y_i & \forall i \in \mathcal{I}, \forall r \in \mathcal{R}_i \\ z_{jk}^r \leq x_r & \forall r \in \mathcal{R}, \forall j \in \mathcal{J}_r, \forall k \in \mathcal{K}_j \\ \sum_{r \in \mathcal{R}} z_{jk}^r = 1 & \forall j \in \mathcal{J}, \forall k \in \mathcal{K}_j \end{array} \right\}$$

Given variables $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \in \mathcal{X}$, we define the *safe set* $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ for the CMLRP as the set of all realizations $\boldsymbol{\xi} \in \mathbb{R}_+^{JK}$ for which the capacity constraints on both the facilities and the vehicles are satisfied. Formally, we have:

$$\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \left\{ \boldsymbol{\xi} \in \mathbb{R}_+^{JK} : \begin{array}{rll} \sum_{j \in \mathcal{J}_r} \sum_{k \in \mathcal{K}_j} z_{jk}^r \xi_{jk} & \leq Q_r x_r & \forall r \in \mathcal{R} \\ \sum_{r \in \mathcal{R}_i} \sum_{j \in \mathcal{J}_r} \sum_{k \in \mathcal{K}_j} z_{jk}^r \xi_{jk} & \leq S_i & \forall i \in \mathcal{I} \end{array} \right\} \tag{1}$$

We then use (1) to define the *Joint Chance-Constrained (JCC)* version of the CMLRP:

$$\min_{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}} \sum_{i \in \mathcal{I}} f_i y_i + \sum_{r \in \mathcal{R}} c_r x_r \tag{2}$$

$$\text{s.t } \mathbb{P}\left(\tilde{\boldsymbol{\xi}} \in \mathcal{S}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\right) \geq 1 - \epsilon \tag{3}$$

$$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \in \mathcal{X} \tag{4}$$

The objective (2) aims at minimizing the sum of facilities opening and transportation. Constraint (3) ensures that the capacity of the openend facilities and used vehicles are sufficient to satisfy the demand with probability at least $1 - \epsilon$ for $\epsilon \in [0, 1]$.

### 2.2 Distibutionally robust formulation using Wasserstein ambiguity

For conciseness reasons, we cannot include the mathematical details and the resulting distributionally robust (DR) formulation of the JCC CMLRP introduced above. Instead, we describe the main ideas to derive this model. We build upon the approach presented in [5] to define and strengthen the formulation of JCC programs with right-hand side uncertainty: 1) We are able to define a relaxation of the safety set (1) based on new lifted variables $\bar{\boldsymbol{\xi}}$ that corresponds to the DR counterpart of the JCC program above, 2) we strengthen this formulation with additional constraints from the nominal chance–constrained case, 3) we compute better values for the big-M constants involved in the formulation, and 4) we propose a new constraints generation method to handle real-size problems.

## 3 Experimentations and perspectives

In order to evaluate the performances of the proposed approach, we compare the standard SAA method with the DR formulation of the JCC program, with and without constraints generation. We derive instances from real-life data sets provided by a french food producer, each consisting in 1) a set of historical demands of its customers for each type of product, 2) a set of candidate starting facilities with a shipping capacity, and 3) a set of pre-generated feasible routes. We consider various sizes of requests (small, medium, large) and capacities for the starting facilities (tight, large). Using Gurobi 10.0 with a maximum computation time of one hour, we obtain preliminary results on medium-size instances ($I = 6$, $J = 20$, $K = 3$, $R \approx 1500$) for the DR counterpart of the JCC problem that achieves a final Gurobi gap lower than 2%, which is shown to be lower than 1% with the stronger lower bound provided by the constraints generation method. Future numerical experiments include the application of our methods to larger instances to assess the scalability of this approach to industrial cases. We also plan to conduct an extended sensitivity analysis of these techniques to the size of the training sample and the radius of the Wasserstein ball. The overall objective is to derive a method allowing practitioners to estimate the impact of their target service-level on the out-of-sample performances, helping them to adjust the trade-off between costs and customer satisfaction.

## Aknowledgments

This work was funded by the French National Research Agency (ANR) LabCom project CRC Lab.

## References

[1] Michael Drexl and Michael Schneider. "A survey of variants and extensions of the location-routing problem". In: *Eur. J. of Oper. Res.* 241.2 (2015), pp. 283–308.

[2] Caroline Prodhon and Christian Prins. "A survey of recent research on location-routing problems". In: *Eur. J. of Oper. Res.* 238.1 (2014), pp. 1–17.

[3] Zhi Chen, Daniel Kuhn, and Wolfram Wiesemann. "Data-Driven Chance Constrained Programs over Wasserstein Balls". In: *Oper. Res.* 0.0 (0), null.

[4] Weijun Xie. "On distributionally robust chance constrained programs with Wasserstein distance". In: *Math. Prog.* 186.1-2 (2021), pp. 115–155.

[5] Nam Ho-Nguyen et al. "Strong formulations for distributionally robust chance-constrained programs with left-hand side uncertainty under Wasserstein ambiguity". In: *INFORMS J. on Optim.* 5.2 (2023), pp. 211–232.

# Terminal tractor electrification and charging infrastructure deployment on a container port: a Benders decomposition approach

**Jorge E. Mendoza**
HEC Montréal - CIRRELT
Email: jorge.mendoza@hec.ca

**Linxiao Wu**
Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University

**Asma Mdimagh**
Université de Montréal - CIRRELT

**Yossiri Adulyasak**
HEC Montréal - CIRRELT - GERAD

**Jean-François Cordeau**
HEC Montréal - CIRRELT - GERAD

## 1 Introduction

Terminal tractors are universally used in ports for moving containers in the yard. They are believed to be responsible for over 50% of the total greenhouse gas (GHG) emissions of container terminals (Yu u.a. 2017) and their fuel accounts for a significant part of ports' operational costs (Zhen 2014). Therefore, they are excellent candidates for electrification.

In this talk, we present a study to solve the heavy-duty truck electrification problem with charging infrastructure decisions (TEPI) for short-distance trucking operations in a container terminal. Using the trajectories of the trucks derived from historical GPS-tracking data, we formulate the TEPI as a mixed-integer linear programming model (MILP). To solve the problem on a realistic scale, we further propose a Benders decomposition algorithm. We show the efficiency of our approach and the benefits of tractor electrification for the terminal through extensive experiments carried out on instances generated based on the real-world operation of a large container terminal.

## 2 Problem Statement

We consider a fleet of $F$ homogeneous diesel-powered trucks. These trucks can be *electrified* by adding an electric train, capacitors to store energy, and a charger. Once electrified, a truck becomes hybrid, meaning that it can either run on electricity or diesel. In our specific case, electrified trucks run primarily on electricity and automatically switch to diesel energy when they deplete their capacitors. Trucks can be electrified with different configurations. A configuration is defined by a combination of the number and size of the capacitors and the speed of the charger. We denote the set of existing configurations as $\mathcal{H}$. Each configuration $h \in \mathcal{H}$ is associated a retrofitting cost $C_h^1$.

Electrified trucks are wirelessly charged using fast inductive charging stations located throughout the terminal area. For modeling the location decisions of charging facilities, the operating area of the trucks is discretized into a set of vertices denoted by $\mathcal{V}$. There is also a set $\mathcal{K}$ of configurations for the charging stations. Each station configuration is associated with a default charging rate and a construction cost $C_{v,k}^2$ at vertex $v \in \mathcal{V}$. Let $B$ denote the budget allocated for the investment in electrification.

Trucks are deployed to work on shifts (each typically covering four hours). The operations of a truck in any shift can be represented as a *route* in a directed physical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ constructed on the vertex set $\mathcal{V}$, where $\mathcal{E}$ is the set of edges, respectively. We consider a set $R$ of routes, and each route starts from the (dummy) depot $v_0 \in \mathcal{V}$, traverses a group of vertices in $\mathcal{V}$, and ends at the depot.

The TEPI considered in this study involves three inter-connected decisions: (i) the truck configuration selection problem that decides the configuration for each truck in a fleet, (ii) the charging facility location problem for enabling en-route charging in the considered area, and (iii) the charging planning problem that returns the best charging plan for each truck on a planned route. The first two are strategic problems which are made for a planning horizon associated with a depreciation period of about 10 years for the facilities. Meanwhile, the third one is an operational problem which is formulated based on the results of the first two problems and solved every time a route is planned for a truck.

Let binary variables $x_{f,h}$ denote whether $f \in [F]$ trucks are electrified with configuration $h \in \mathcal{H}$, where $[F] = \{0, 1, ..., F\}$ denotes the set of non-negative indices up to $F$. Let binary variables $y_{v,k}$ represent whether a charging station with configuration $k \in \mathcal{K}$ is set up at vertex $v \in \mathcal{V}$. Furthermore, given any decision of charging station locations, denoted by vector $\boldsymbol{y}$, let $z_{h,r}(\boldsymbol{y})$ denote the minimum cost for electricity and diesel consumption of a truck with configuration $h \in \mathcal{H}$ on route $r \in \mathcal{R}$. The TEPI can be formulated as a MILP model as follows:

$$\min \sum_{f\in[F]}\sum_{h\in\mathcal{H}} C_h^1 f x_{f,h} + \sum_{k\in\mathcal{K}}\sum_{v\in\mathcal{V}^1} C_{v,k}^2 y_{v,k} + \Gamma \sum_{f\in[F]}\sum_{r\in\mathcal{R}}\sum_{h\in\mathcal{H}} \frac{1}{|\mathcal{R}|} f x_{f,h} z_{h,r}(\boldsymbol{y}) \tag{1}$$

$$\text{s.t.} \sum_{f\in[F]}\sum_{h\in\mathcal{H}} f x_{f,h} = F \tag{2}$$

$$\sum_{k\in\mathcal{K}} y_{k,v} \leq 1 \quad \forall v\in\mathcal{V} \tag{3}$$

$$\sum_{f\in[F]}\sum_{h\in\mathcal{H}} C_h^1 f x_{f,h} + \sum_{k\in\mathcal{K}}\sum_{v\in\mathcal{V}} C_{v,k}^2 y_{v,k} \leq B \tag{4}$$

$$x_{f,h} \in \{0,1\} \quad \forall f\in[F], h\in\mathcal{H} \tag{5}$$

$$y_{v,k} \in \{0,1\} \quad \forall k\in\mathcal{K}, v\in\mathcal{V} \tag{6}$$

$$z_{h,r}(\boldsymbol{y}) \in \mathbf{Z}_{h,r}(\boldsymbol{y}) \quad \forall r\in\mathcal{R}, h\in\mathcal{H}, \tag{7}$$

where $\Gamma$ represents the number of working shifts contained in the planning horizon (which is used to project the expected operational cost for running the fleet in a shift to that in the entire planning horizon) and $\mathbf{Z}_{h,r}(\boldsymbol{y})$ represents the domain (defined by linear constraints) for the decision variables $z_{h,r}(\boldsymbol{y})$, where $h \in \mathcal{H}$ and $r \in R$.

The objective function (1) minimizes the total cost, which includes three components. The first and second terms are the costs for electrifying the fleet and setting up charging stations, respectively. The third term is the total expected operational costs for running the fleet during the planning horizon. Constraints (2) ensure that each truck is retrofitted with exactly one configuration. Constraints (3) require that each vertex can install at most one charging station. Constraint (4) sets the upper bound for the investment. The domains of decision variables are given by constraints (6) to (7), where $\mathbf{Z}_{h,r}(\boldsymbol{y})$ is defined by constraints for controlling en-route charging and tracking the electricity levels onboard the trucks, given the route $r \in \mathcal{R}$, the configuration of the trucks $h \in \mathcal{H}$, and charging station decisions $\boldsymbol{y}$.

## 3 A Benders decomposition approach

The TEPI is NP-hard. To solve this challenging problem, we develop a Benders decomposition approach. In this approach, we decompose TEPI into a master problem and a set of subproblems. In the master problem, truck electrification and charging station location decisions are made. Given the solution of the master problem, we then solve a set of subproblems, each corresponding to a configuration $h \in \mathcal{H}$ and a route $r \in \mathcal{R}$.

The approach iterates between the master problem and the subproblems. Benders optimality cuts are dynamically generated and incorporated into the master problem. The approach terminates when it converges to an optimal solution or reaches a preset threshold of the optimality gap.

The Benders decomposition approach is implemented through a branch-and-cut framework, where Benders cuts are separated at nodes with integer solutions. In addition, to further accelerate the approach, we also derive valid inequalities that lift the lower bound of the master problem.

## 4 Results

To test the performance of the solution approach, we designed instances based on the geographical and operational data from a container terminal at the Port of Montreal in Canada. The terminal runs a fleet of 50 trucks for transporting containers in its yard. To generate the instances, the yard area of the terminal was mapped into a graph with 56 vertices and 199 edges. We have also generated the truck routes using 3 months of historical GPS data of the fleet. To test the performance of the algorithm, we created 25 instances, where the number of trucks ranges from 10 to 50 and the number of routes ranges from 10 to 50 as well.

We set the time limit for solving each instance to 10 hours. The computational results demonstrate that within the given time limit, the approach can solve instances with up to 30 trucks and 50 routes to optimality and that it can obtain good-quality solutions (within a 10% optimality gap) for instances of larger scale.

To evaluate the value of electrification, we compared the total cost (including electrification and truck operation costs) and the diesel fuel consumption of each instance before and after electrification. The results demonstrate that by electrifying the trucks, the terminal can reduce the total cost and the diesel fuel consumption by 37% and 95% on average, respectively.

## References


Zhen, Lu(2014): *Container yard template planning under uncertain maritime market*199–217.

Yu, Hang / Ge, Ying En / Chen, Jihong / Luo, Lihua / Tan, Caimao / Liu, Ding(2017): *CO2 emission evaluation of yard tractors during loading at container terminals*17–36.

Horn, Michael / MacLeod, Jennifer / Liu, Meinan / Webb, Jeremy / Motta, Nunzio(2019): *Supercapacitors: A new source of power for electric cars?*93–103.

Borlaug, Brennan u.a.(2021): *Heavy-duty truck electrification and the impacts of depot charging on electricity distribution systems*, 6: 673–682.

U.S. Energy Information Administration (2022): *Annual Energy Outlook 2022* `https://www.eia.gov/outlooks/aeo/data/browser/#/?id=45-AEO2021&cases=ref2021&sourcekey=0.`, Accessed November 21, 2022.

# A parallel metaheuristic framework for the Capacitated Vehicle Routing Problem

**Federico Michelotto**
Department of Electronics, Computer Science and Systems,
University of Bologna, Bologna, Italy
Email: federico.michelotto2@unibo.it

**Demetrio Laganà**
Department of Mechanical, Energy, and Management Engineering,
University of Calabria, Rende, Italy
Email: demetriosalvatore.lagana@unical.it

**Roberto Musmanno**
Department of Mechanical, Energy, and Management Engineering,
University of Calabria, Rende, Italy
Email: roberto.musmanno@unical.it

**Daniele Vigo**
Department of Electronics, Computer Science and Systems,
University of Bologna, Bologna, Italy
Email: daniele.vigo@unibo.it

## 1 Problem Definition and Motivation

The Capacitated Vehicle Routing Problem (CVRP) is one of the most studied combinatorial optimization problems (see [6]), which can be described as follows. Given a set of customers, each of which has a known location and demand, and a fleet of homogeneous vehicles with fixed capacity, the goal of the CVRP is to find the minimum cost set of routes, so that each customer is served by exactly one vehicle and the capacity constraints for the vehicles are satisfied.

Given the inherent computational difficulty and practical relevance of the problem, several successful heuristics and metaheuristics have been proposed for its solution. However, almost all methods proposed in literature for solving the CVRP are sequential algorithms. Despite the impressive results achieved by recent metaheuristic methods on large-scale instances (FILO [1], HGS [5], SISR [2]), the sequential nature of these algorithms becomes a fundamental limitation when the goal is to efficiently solve instances with thousands or more customers.

One of the most intuitive parallel programming paradigms consists in decomposing the original problem into several sub-problems, solve them independently, and then merge the sub-solutions so that a complete solution is obtained.

To this end, Santini et al. [4] studied the effect of several decomposition schemes on two state-of-the-art sequential CVRP metaheuristics. In their setting, the decomposition phase is not applied just once, but every $d$ iterations, and the generated sub-problems are solved sequentially. They found that appropriate decomposition schemes, not only are not harmful in terms of quality of the obtained solutions, but they can even be beneficial with respect to applying no decomposition at all.

The following research questions arise naturally. How can we parallelize a sequential CVRP metaheuristic without adding significant computational overhead? How much speedup can we obtain if we parallelize, and how does it scale with respect to the number of processes and the instance size? How to deal with synchronization issues between processes?
To the best of our knowledge, the rich literature related to CVRP lacks methodological parallel approaches on state-of-the-art sequential algorithms, and this work aims to fill this gap and to answer to the above open research questions.

The objective of this work is, therefore, threefold. First, we propose an effective and paradigmatic parallel metaheuristic framework for the CVRP based on a novel dynamic decomposition scheme. Second, we show that, when dealing with very large-scale instances, the parallelization is not only a desirable feature useful to speedup the computation, but in practice, necessary when dealing with these kinds of instances. Third, we provide an easy-to-use framework publicly available to the research and industry community.

## 2 Solution Methods and Contributions

Our framework is based on the master-slave parallel programming paradigm and is designed to work with virtually any existing metaheuristic for the CVRP. Our approach consists of an iterative scheme in which the master process repeatedly assigns a subset of contiguous routes to each slave process, each of which runs a predefined metaheuristic for a certain number of iterations. Then the slave processes send the optimized routes back to the master process which recombines them with the current solution.

The main contribution of our solution method is a novel dynamic decomposition scheme for the CVRP that requires no synchronization between processes.

Given a solution, the proposed decomposition scheme represents the routes as points on a polar coordinate system, in which the center is the depot, and each route is represented by the barycenter of the customers served along the route. Every time a slave process terminates, a new subset of routes is assigned to it. The routes' selection procedure aims to select a subset of contiguous routes. In the preliminary version of this work, this is done by ordering the routes by their barycenter's angle, and selecting a contiguous subset of routes from this ordered set.

The proposed framework aims at being simple, fast and scalable. The decomposition scheme guarantees no overlapping among the subset of routes sent to the slave processes, allowing for a simple recombination procedure, and requiring no synchronization between processes.

## 3 Preliminary Computational Results

To evaluate the effectiveness of the proposed parallel scheme, we run the sequential algorithm for a fixed number of iterations (or a fixed amount of time), and we compare it with the time required by the parallel algorithm to obtain a solution at least as good as the one returned by the sequential algorithm.

The proposed parallel approach has been implemented in C++ using the OpenMPI library. For the preliminary computational evaluation, FILO [1], a state-of-the-art metaheuristic for the CVRP has been used inside the slave processes as sequential CVRP metaheuristic. Preliminary results on a small subset of large-scale instances of the CVRPLIB [3] with a number of customers ranging from 6000 to 15000, on a six cores CPU, show an average speedup of 38% with respect to the sequential version.

## References


[1] L. Accorsi and D. Vigo, "A Fast and Scalable Heuristic for the Solution of Large-Scale Capacitated Vehicle Routing Problems", *Transportation Science* 55(4), 832–856, 2021.

[2] J. Christiaens and G. Vanden Berghe, "Slack Induction by String Removals for Vehicle Routing Problems", *Transportation Science* 54(2), 417–433, 2020.

[3] CVRPLIB. Capacitated vehicle routing problem library, 2020. URL http://vrp.galgos.inf.puc-rio.br/ index.php/en/. visited on 2023-11-12.

[4] A. Santini, M. Schneider, T. Vidal, D. Vigo, "Decomposition strategies for vehicle routing heuristics", *INFORMS Journal on Computing* 35(3), 543–559, 2023.

[5] T. Vidal, "Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood", *Computers & Operations Research* 140, 105643, 2022.

[6] P. Toth and D. Vigo (eds), *Vehicle Routing: problems, methods and applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

# Solving a Joint Vehicle Routing and Generalized Assignment Problem via Column Generation

**Juan José Miranda Bront**
School of Business,
Universidad Torcuato Di Tella, Buenos Aires, Argentina
Consejo Nacional de Investigaciones Científicas y Técnicas, Buenos Aires, Argentina
Email: jmiranda@utdt.edu

**Gonzalo Lera Romero**
Department of Computer Science
Universidad de Buenos Aires, Buenos Aires, Argentina

## 1 Introduction

Marketplaces and large logistic carriers concentrate a significant share of the daily sales across different industries, providing integrated selling, payment and logistics services for third party retailers. We tackle a tactical distribution problem motivated by a real world application arising within the first mile (FM) of the logistic network of large marketplaces and carriers. Consider the following setup. Retailers, usually referred to as *sellers*, offer and sell products through a platform, which is also responsible for their distribution to the customers. Packages are regularly collected from the sellers (e.g. once a day) to enter the logistic network as the first step of the distribution process. Sellers are very heterogeneous not only regarding their business, but also in terms of volume. For large marketplaces, there is a *long tail* of sellers introducing only a few packages each day to the network (i.e., below 15-20), but still relevant in terms of the total volume. Collecting the packages from these customers is expensive from a routing perspective given the low volume of packages.

Carriers, aiming to minimize routing costs, can delegate the first mile to long-tail sellers through a network of collection points (CP) [5], usually composed convenience stores. These CPs, acting as small consolidation centers, help optimize routing costs at the expense of the sellers delivering their packages. This strategy is gaining popularity in Latin America and Asia, where it offers additional income opportunities for these CPs. Note that the CPs have a limited daily package reception capacity, and that the sellers' travel distance is relevant in engaging with the platform. Practical solutions often adopt a sequential method. Initially, long-tail customers are identified based on a fixed threshold for daily package deliveries. Subsequently, the assignment of sellers to the CPs is determined by solving a Generalized Assignment Problem (GAP) [4], and the collection for other sellers is addressed through a Vehicle Routing Problem (VRP) [6].

In our study, we explore an integrated approach where each seller can be routed or assigned to a CP independently of its demand, solving the VRP and the GAP simultaneously. We propose two set partitioning formulations and a column generation based heuristic. To the best of our knowledge, this problem has not been deeply studied in the literature. Some connected problems from the VRP literature are the VRP with private fleet and common carrier (VRPPC) [2], the VRP with partial outsourcing (VRPPO) [1] and the VRP with Shared Delivery Locations (VRPSDL) [3].

## 2 Problem definition and ILP Formulations

We formalize the integrated VRP and GAP (VRP-GAP). The network is modelled with a digraph $D = (V, A)$. Let $V = V_s \cup V_{cp} \cup \{o, d\}$ a partition of the set of locations, where $V_s$ denotes the sellers, $V_{cp}$ the CPs, and $o$ and $d$ are two distinguished vertices denoting the depot. A seller $i \in V_s$ has a demand $q_i$ of packages introduced to the network either by (i) a vehicle collecting the packages from the seller, or (ii) the seller delivering the packages itself to one of the CPs, to be defined by the carrier. Each CP $j \in V_{cp}$ has a capacity $Q_j$ of packages to be allocated daily. For the routing, an unlimited fleet of vehicles with capacity $C$ is available. Let $c_{ij}$ denote the cost (distance) of assigning seller $i \in V_s$ to CP $j \in V_{cp}$ and the routing cost if location $j$ immediately after $i$ in a route, with $i, j \in \{o, d\} \cup V_s$.

We define a *feasible assignment* $a = (S, j)$, indicating the assignment of sellers $S \subseteq V_s$ to $j \in V_{cp}$, with total demand $q(a) = \sum_{i \in S} q_i$ not exceeding $Q_j$ and cost $c_a = \sum_{i \in S} c_{ij}$. Similarly, a *feasible route* $r = (o = v_0, v_1, \ldots, v_{k-1}, v_k = d)$ satisfies $q(r) = \sum_{i=1}^{k-1} q_i \leq C$, and its cost is $c_r = \sum_{i=0}^{k-1} c_{v_i v_{i+1}}$. The VRP-GAP involves finding a set of feasible routes and feasible assignments such that each seller is either visited by exactly one route or belongs to exactly one assignment at minimum total cost. Let $\Omega$ be the set of all feasible routes, and define $b_{ir}$ as the number of times seller $i \in V_s$ is visited by $r \in \Omega$. Let binary variables $y_{ij}$ take value 1 iff seller $i \in V_s$ is assigned to CP $j \in V_{cp}$, and $\lambda_r$ to take value 1 iff route $r \in \Omega$ is selected. A first model for the VRP-GAP, named F1, reads

$$\min \sum_{i \in V_s} \sum_{j \in V_{cp}} c_{ij} y_{ij} + \sum_{r \in \Omega} c_r \lambda_r \tag{1}$$

$$\text{s.t.} \sum_{j \in V_{cp}} y_{ij} + \sum_{r \in \Omega} b_{ir} \lambda_r = 1, \qquad i \in V_s \tag{2}$$

$$\sum_{i \in V_s} q_i y_{ij} \leq Q_j, \qquad j \in V_{cp} \tag{3}$$

$$y_{ij}, \lambda_r \in \{0, 1\}, \qquad i \in V_s, j \in V_{cp}, r \in \Omega \tag{4}$$

The objective (1) minimizes the total cost of the assignments and routes selected. Constraints (2) guarantees all sellers are either visited or assigned, and constraints (3) prevent the capacity of the CPs to be exceeded.

We further define $\Psi$ as the set of feasible assignments, with $b_{ia}$ indicating whether $i \in V_s \cup V_{cp}$ belongs to assignment $a \in \Psi$, either being a seller or a CP. Let binary variables $\alpha_a$ take value 1 iff assignment $\alpha \in \Psi$ is selected. The previous model for the VRP-GAP is adapted as follows and named F2:

$$\min \sum_{a \in \Psi} c_a \alpha_a + \sum_{r \in \Omega} c_r \lambda_r \tag{5}$$

$$\text{s.t.} \sum_{a \in \Psi} b_{ia} \alpha_a + \sum_{r \in \Omega} b_{ir} \lambda_r = 1, \qquad i \in V_s \tag{6}$$

$$\sum_{a \in \Psi} b_{ja} \alpha_a \leq 1, \qquad j \in V_{cp} \tag{7}$$

$$\alpha_a, \lambda_r \in \{0, 1\}, \qquad a \in \Psi, r \in \Omega \tag{8}$$

We tackle both models using column generation. The sketch of the algorithms is as follows. Given $\bar{\Omega} \subseteq \Omega$ and $\bar{\Phi} \subseteq \Phi$, let $\bar{\pi}_i$ be the optimal dual variables associated with assignment constraints ((2) for F1 and (6) for F2) and $\bar{\sigma}_j$ the ones associated with constraints (7) for F2 in the Restricted Master Problem (RMP), respectively. For both models, the pricing problem for variables $\lambda_r, r \in \Omega$ is formulated as an Elementary Shortest Path with Resource Constraints (ESPPRC) by setting $\bar{c}_{ij} = c_{ij} - \bar{\pi}_i$ and tackled using a forward labeling algorithm. For model F2, an additional pricing problem must be considered. For $a = (S, j) \in \Phi$, identify a variable with negative reduced cost $rc_a = c_a - \sum_{i \in S} \bar{\pi}_i - \bar{\sigma}_j$, or prove that none exist. The pricing problem is formulated as a 0-1 Knapsack Problem for each CP $j \in V_{cp}$ by defining $Q_j$ as the capacity of the knapsack, and $p_i = c_{ij} - \hat{\pi}_i$ as the profit and $w_i = q_i$ as the weight for each item (seller) $i \in V_s$.

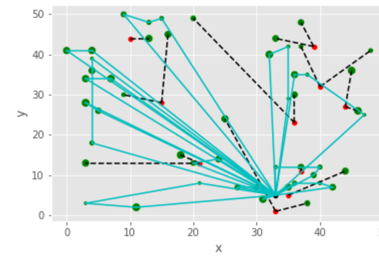
## 3 Preliminary computational results

For each formulation, we implemented a matheuristic as follows: first, solve the LP relaxation at the root node via column generation, and (ii) freeze the corresponding model and solve it using an ILP solver. We further implemented a simple BP algorithm for formulation F2, aiming to analyze the quality of the solutions found. The algorithms are implemented in C++, using CPLEX as a mathematical programming solver. We conducted computational experiments on a Intel(R) Core(TM) i5-1135G7 @ 2.40GHz and 16GB of RAM. We generated 45 random instances, with $|V_s| = 10, 20, \ldots, 50$, $|V_{cp}| = 0.2 \times n, 0.3 \times n, 0.5 \times n$, with different geographic distribution (R, C, RC) and defining the travel cost $c_{ij}$ as the Euclidean distance.

Table 1a shows the average results, aggregated by $|V_s|$. For each formulation, we report

| $\lvert V_s \rvert$ | F1 | | | F2 | | |
|---|---|---|---|---|---|---|
| | Time | %lpG | %IntG | Time | %lpG | %IntG |
| 10 | 0.03 | 12.73 | 0.00 | 0.02 | 5.65 | 1.68 |
| 20 | 0.04 | 10.66 | 0.00 | 0.04 | 1.14 | 0.41 |
| 30 | 0.09 | 11.63 | 0.58 | 0.09 | 1.33 | 0.61 |
| 40 | 0.24 | 12.29 | 0.23 | 0.26 | 1.04 | 0.64 |
| 50 | 1.97 | 13.35 | 0.73 | 2.50 | 1.39 | 0.55 |
| Totals | 0.47 | 12.13 | 0.31 | 0.58 | 2.11 | 0.78 |

(a) Average aggregated results for F1 and F2.




(b) Example of instance and solution.

the average computation time in seconds (Time), % gap of the LP relaxation (%lpG) and % gap of the integer solution found after solving the corresponding ILP (%IntG), where the gaps are computed with respect to the best known solution for the instance. For these instances, the root node can be solved very efficiently. We highlight the reduction in the %lpG shown by F2 compared to F1, which may be critical for state of the art exact algorithms. Figure 1b shows the optimal solution for an instance with 50 customers (green) and 10 CPs (red), where routes are plotted in cyan and assignments in dashed black lines. This solution illustrates that using a fixed threshold for long tail sellers is suboptimal, since the mode to serve a seller depends also on the structure of the instance.

For the conference, we will expand these results with an exact BP algorithm for each formulation as well as exploring more deeply the benefits of considering this integrated approach on a larger set of benchmark instances.

## References


[1] A. C. Baller, S. Dabia, W. E. Dullaert, and D. Vigo. The vehicle routing problem with partial outsourcing. *Transportation Science*, 54(4):1034–1052, 2020.

[2] S. Dabia, D. Lai, and D. Vigo. An exact algorithm for a rich vehicle routing problem with private fleet and common carrier. *Transportation Science*, 53(4):986–1000, 2019.

[3] S. Mancini and M. Gansterer. Vehicle routing with private and shared delivery locations. *Computers & Operations Research*, 133:105361, 2021.

[4] M. Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, 45(6):831–841, 1997.

[5] A. Snoeck, D. Merchán, and M. Winkenbach. Revenue management in last-mile delivery: state-of-the-art and future research directions. *Transportation Research Procedia*, 46:109–116, 2020.

[6] P. Toth and D. Vigo, editors. *Vehicle Routing: Problems, Methods, and Applications*. 2nd ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

# Decision Support System for Drone-based Delivery for Humanitarian Logistics Under Uncertainties

**Pitu Mirchandani**
School of Computing and Augmented Intelligence, Arizona State University, AZ, USA
**Monica Gentili**
Department of Industrial Engineering, University of Louisville, KY, USA
**Zabih Ghelichi**
Department of Industrial Engineering, University of Louisville, KY, USA

## 1. Introduction and Motivation

Immediate emergency responses are critical for vital delivering aid, especially in inaccessible areas where severe disasters damage supply lines and transportation infrastructures. When traditional land modes are inoperable, above-ground methods may be valuable. Unmanned Aerial Vehicles (UAVs), commonly known as drones, showed great potentials to swiftly deliver aid to regions inaccessible by land, offering advantages like pilot-free operation and flexibility in various applications [e.g., 1].

This research was motivated by the recent major climate and weather-related disasters across the United States, for instances., Florida's hurricanes Irma (2017) and Michael (2018), California's wildfires in 2020, and Texas' winter storms in 2021. The massive destruction on the transportation infrastructure including road networks, airports, and seaports left victims without or with limited access to essential aid items for an extended period [e.g., 2,3]. Such circumstances underscore the necessity for the development of drone-based delivery systems to enhance logistical operations in disaster response.

Drone-based delivery systems, like most logistics systems, face various sources of variability. In disaster scenarios, critical uncertainties arises from limited information about demand, including the number and locations of people needing immediate attention in the initial hours after a disaster. As system state updates and calls for assistance accumulate, decisions on drone trip scheduling must be frequently revised for efficiency. Additional uncertainties stem from the operational performance of drones, where factors such as wind and temperature fluctuations can impact flight speed, affecting flight duration and drop-off times [e.g., 4].

This research first develops a simulation-based performance evaluation model for designing an efficient drone-based aid delivery system. The goal is to create a decision support framework that assesses the performance of such systems through realistic simulations of drone delivery flights and accounting for above uncertainties. Leveraging this simulation model, we evaluate and enhance the drone-based delivery of aid items in humanitarian logistics. The proposed model considers variations in key system parameters, including updating intervals, demand rates, spatial distribution of demand, service times (travel, setup, loading, payload drop-off, and repair times), and drone energy levels requiring battery change/recharging during flight.

## 2. Problem Description

Developing decision-making models to coordinate and support the logistics for a fleet of drones for the delivery of aid packages in disaster scenarios face many challenges including *(i)* the drones' limitations, e.g., coverage range and payload capacity, *(ii)* time urgency, *(iii)* lack of information, *(iv)* unavailability of resources, and *(v)* fluidity of the situation. We consider a disaster scenario where a set $I$ of $m$ drone platforms sites are located to provide timely delivery of aid items to disaster-affected areas. Examples of such aid items include water, food, and medications. Each drone starts its trip from its dedicated platform, flies to a demand location, delivers its load, and returns to its corresponding platform. We also assume each demand location requires exactly one delivery. We assume that the drone take-off platforms are already located.

Due to the fluidity of the situation and the lack of information during the first few hours after a disaster strike, we consider a scenario where initially the set of demand locations are not known, but information about each demand location is received. This set of information includes (1) the number of demand locations, and (2) the corresponding coordinates of each demand location. The demand locations appear over time based on a spatial probability map which gives the probability of demand at each point on the plane. To efficiently schedule drone trips, we adopt the drone scheduling model introduced in [5]. This model optimally schedules and sequences a set of trips for each drone in the fleet such that a measure of disutility is minimized. In this case, the disutility is a function of the delivery and waiting time for each demand location which is non-decreasing with respect to time.

The variations in drone operational times are another source of variability. Due to the variations in the drone velocity, we assume the travel time follows a normal distribution with a known mean and variance [6]; this distribution could be updated when new data is collected. Each drone also requires a setup time for service, loading and battery swapping/charging, between two consecutive trips. The setup time of drones and drop-off time at the demand location are assumed to follow a known distribution (discussed in results section). Drones' operational flight range is limited by the maximum coverage range in terms of time and distance. Drone flights may face other non-deterministic effects. Factors such as low temperatures and battery efficiency variations can induce interruptions in the battery performance. In our simulation model, we assume such factors can impact the energy level of the drone and influence the success of completing some deliveries. To capture these effects, we assume drones may face a battery failure with a known distribution and failure rate.

## 3. Methodology

The proposed simulation model captures the variability in the following sources: (1) interval of updating the system after receiving new information, (2) demand parameters: the demand rate, the spatial distribution, (3) time parameters: travel time, setup and loading time, payload drop-off time and fixing time, and (4) drone energy level, and possibility of battery failure while delivering. Figures 1 schematically illustrates the simulation platform procedure. At every update interval, the simulation model evaluates the status of the system. In this stage, the simulation model identifies the current configuration and requirements of the system including the information about the new demands, the status of the previously received demands, the status of drones and drones' current locations. It is worthy to note that the update interval $\delta$ is a random variable independent from drones' operation. Then, this information is fed into a scheduling model which generates a set of decisions corresponding to the system status. The output of the scheduling model is an ordered list of deliveries fed back into the simulation model.

In this way, the simulation and scheduling model continuously interact with each other. Together they obtain a set of decisions from the scheduling model and the realizations of the uncertain parameters to enable the simulation model to simulate the flights and deliveries for the fleet of drones (yellow box in Figure 1). At any timestamp, the status of the demands scheduled at the previous timestamp can be labeled as either (1) served, or (2) being served, or (3) scheduled but not served yet. If a demand location is already served or being served, we remove it from the set of demand locations that are to be scheduled or rescheduled at the update time.

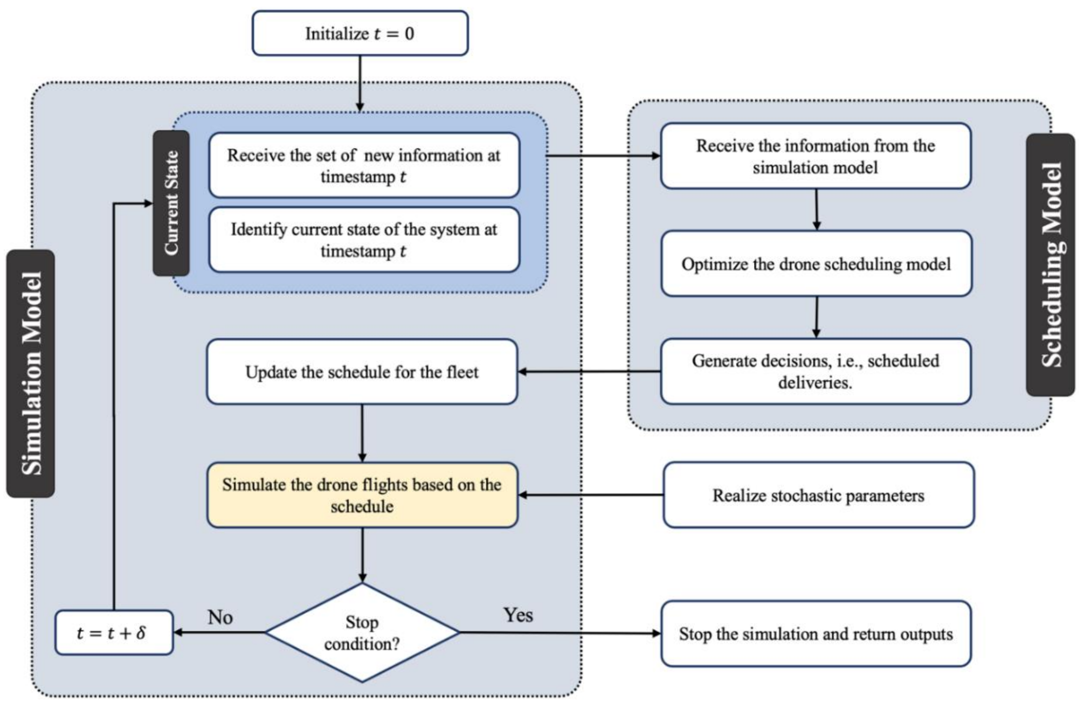



*Figure 1. An illustration of the proposed simulation model*

## 4. Preliminary Results

The performance of the proposed approach was evaluated through a series of experiments. Our goal is to show how the simulation model can be easily integrated in an optimization procedure to offer a tool that can be used for improving initial decisions. We first select $m$ drone platform locations by using the platform-location model proposed in [5], a deterministic mathematical formulation to determine the optimum locations of a predefined number $m$ of drone platforms in a disaster-affected area. A solution to this problem is a list of $m$ locations where the drone platforms must be located. We denote by $\lambda_m^*$ the set of optimum platform locations when $m$ drone platforms are selected using this deterministic model. Subsequently, we perform a set of analytical studies by evaluating the 1-opt neighbors to $\lambda_m^*$ through running the proposed simulation model. A 1-opt neighbor to the optimum solution with $m$ platforms is a list of $m$ locations which differs from $\lambda_m^*$ in 1 element.

Our analyses show there are multiple solutions which are better than the initial chosen solution obtained from the deterministic model in terms of total disutility, waiting time and percentage of served demands. The results showed that a simple 1-opt neighborhood search can improve the obtained results in all aspects. The reason for this observation lies in the fact that the simulation model accounts for higher level of uncertainties compared to the deterministic platform-location optimization model. For instance, the simulation model accounts for the variable waiting time of the demands, from when each demand manifests in the system until it is served, including the possible failure in completing the deliveries.

## 5. Conclusions

This research presents a simulation-based performance evaluation model for the designing a system for timely delivery of humanitarian aid packages via a fleet of drones. The goal was to develop a simulation system that can allow one to evaluate performance of a drone-based delivery system. This simulation system is designed to simulate the drone flights and capture different sources of variations and uncertainties in a drone-based delivery system, including (1) interval of updating the system, (2) demand parameters, (3) time parameters and, (4) drone energy level. The first experiments were designed for a case study of Central Florida [7]. The corresponding set of analyses was to show the capability of the proposed simulation-based performance evaluation model to support the decision-making by improving the solutions obtained from platform-location optimization models and then evaluating performance of the system under alternative solutions, models, and strategies. Our results and discussions in the to-be-presented paper and presentation will highlight the importance of simulation based tool for supporting decision-making under multiple uncertainties, validation of system configuration and evaluating different strategies. The second set of experiments [see 7] evaluates different system settings for randomly generated instances including the performance of multiple algorithms to solve the scheduling model, and frequency for updating the system.

## 6. References


[1]. Otto, A., N. Agatz, J. Campbell, B. Golden, and E. Pesch, *Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey.* Networks, 2018. **72**(4): p. 411-458.

[2]. *BAHAMAS: Hurricane Dorian*. 2019, Office for the Coordination of Humanitarian Affairs.

[3]. Allyn, B. and J. Beaubein, *Getting Aid To Bahamas Is A Logistical Nightmare As Support Systems 'Do Not Exist'*, in *NPR*. 2019.

[4]. Kim, S.J., G.J. Lim, and J. Cho, *Drone flight scheduling under uncertainty on battery duration and air temperature.* Computers and Industrial Engineering, 2018. **117**: p. 291-302.

[5]. Ghelichi, Z., M. Gentili, and P.B. Mirchandani, *Drone logistics for uncertain demand of disaster-impacted populations.* Transportation Research Part C: emerging technologies, 2022. **141**: p. 103735.

[6]. Kim, S.J. and G.J. Lim, *A real-time rerouting method for drone flights under uncertain flight time.* Journal of Intelligent Robotic Systems, 2020. **100**: p. 1355-1368.

[7] Ghelichi, Z., *Drone Location and Scheduling Problems in Humanitarian Logistics,* PhD Dissertation in Industrial Engineering University of Louisville, KY, 2021,

# Algorithms for solving the On-Demand Bus Routing Problem with Bus Stops Assignment

**Jorge Mortes**[a,b,*], **Martin Cousineau**[b], **Fabien Lehuédé**[a], **Jorge E. Mendoza**[b], **María I. Restrepo**[a]

[a]*Département Automatique, Productique et Informatique, IMT Atlantique, LS2N, UMR CNRS 6004, Nantes, France*

[b]*Department of Logistics and Operations Management, HEC Montréal, Canada*

*Email: jorge.mortes-alcaraz@imt-atlantique.fr

## 1 Introduction

Fixed-line buses are known to be very efficient in highly populated areas, but they become under-utilized when the demand decreases. In consequence, researchers study alternative flexible solutions called on-demand or demand-responsive bus systems [1]. By flexible, we mean that the bus routes might change from one day to another in order to satisfy the existing demand. We find in [2] one of several existing real-life ODBRP-SA applications. Additionally, in [1] some more already running systems can be found. In this work, we propose two algorithms to optimize bus routes in an on-demand public bus system that runs during nighttime (i.e., low demand) in an area with a high concentration of bus stops. Even though on-demand mobility for public bus systems has been covered in the literature [1], only a few articles propose assigning passengers' origins and destinations to bus stops or meeting points to reduce transportation costs [3, 4] and pollution.

We consider a set of transportation requests $R$, with each request $r \in R$ being defined by the origin, the destination, a set of possible pick-up bus stops $P_r$, a set of possible drop-off bus stops $D_r$, the number of passengers, and a time window defining the earliest pick-up time and the latest drop-off time. The sets $P_r$ and $D_r$ can be built following different criteria (e.g., stops within a walking range, user preferences, etc.). The time window to pick up or drop off a customer at each bus stop is modified considering the user's time spent walking towards or from the assigned bus stops. We denote by $P = \bigcup_{r \in R} P_r$, and $D = \bigcup_{r \in R} D_r$, the sets containing all pick-up and all drop-off stops, respectively. Note that if two or more requests have pick-up or drop-off stops in common, sets $P$ and $D$ contain multiple copies of those stops. We define the On-Demand Bus Routing Problem with Bus Stop Assignment (ODBRP-SA) on a directed graph $G = (N, A)$, where $N$ is the set of nodes and $A$ is the set of directed arcs. The set $N$ comprises all the bus stops and two copies of the depot denoted by $0^+$ and $0^-$ modeling the start and end of bus routes, respectively (i.e., $N = P \cup D \cup \{0^+, 0^-\}$). Let $K$ be the set of vehicles. Each vehicle $k \in K$ has a passenger capacity $Q_k$. Each node $i \in N \backslash \{0^+, 0^-\}$ is associated with a passenger load $q_i$ (positive for pickups and negative for dopoffs); with a non-negative service duration $d_i$; and with a time window $[e_i, l_i]$. Lastly, a cost $c^k_{i,j}$ is associated with vehicle $k$ traveling through arc $(i, j) \in A$. Based on these definitions, the ODBRP-SA consists of creating the set containing the shortest bus routes that accommodate all the requests while assigning a single pick-up and drop-off stop to each request, respecting the vehicle capacities, and complying with the users' time windows.

## 2 Solution Methods

To solve the ODBRP-SA we propose an algorithm based on a combination of a small and large neighborhood search (SLNS) [5] and a set covering component (SCP). SLNS [5] is a recent version of the adaptive large neighborhood search meta-heuristic [6] which alternates small destroy-repair operations for intensification with larger neighborhood searches for diversification. Our algorithm is similar to the one proposed in [7]: First, routes are generated by the SLNS and stored in a so-called pool of routes. Then, the SCP is solved to select the best set of routes from the pool of routes. After solving the SCP, the pool of routes is emptied, and the algorithm returns to the SLNS. This is run for a certain budget of time. Note that we use the best solution found in the SLNS as a warm start for the SCP and the solution found by the SCP to restart the SLNS.

The repair and destroy operators for the SLNS used in this work are the following:

- **Repair operators:** List insertion heuristics [8] that sort the requests by increasing time window width, increasing pick-up time window start, decreasing drop-off time window end, decreasing amount of passengers, and increasing and decreasing distance from the pick-up point to the depot.

- **Destroy operators:** Random and history removal for small and large destruction iterations; and string and split string removal only in small destruction iterations.

The principal contribution of this work is to integrate bus stop assignments to improve transportation costs. To do so, we compare two strategies: The first strategy, which we call the *Sampling Method* (SM), assigns bus stops as follows: first, we generate a certain amount of samples from the same instance (a sample being an ODBRP-SA instance where every customer has only one pick-up and drop-off bus stop assigned at random). Then, we run every sample with the algorithm presented above, saving the best 100 solutions of each sample in a common pool of routes (elite pool). Finally, we run the SCP model one last time with the routes from the elite pool. By doing this, we allow the possibility of assembling routes from different samples (i.e., different bus stop assignments). The advantage of this approach is that it can be used with any pickup and delivery problem with time windows (PDPTW) solver without any modification other than collecting the set of non-dominated routes. The second algorithm, which we call the *Assignment Method* (AM), assigns bus stops during the repair phase of the SLNS. More precisely, when inserting a request, the algorithm evaluates all feasible insertion positions for all pick-up and drop-off bus stops of the request and selects the combination that minimizes the travel cost increase.

## 3 Results

The instances that we use in this work are extracted from the publicly available taxi trip dataset of New York City. From each sampled taxi trip, we compute the five closest bus stops within five minutes of walking time from the origin and destination. Real-world distances and travel time matrices are obtained with the openrouteservice backend. We create 40 instances divided into four 10-instance groups containing 50, 100, 250, and 500 requests (up to 500, 1,000, 2,500, and 5,000 nodes respectively). For each algorithm, we perform 10 runs of the same duration on each instance, being the durations 15, 45, 90, and 180 minutes respectively. For the sampling method, we use 25 samples run in parallel in batches of 5 instances for a fifth of the duration.

Table 1 shows a comparison between the two proposed algorithms (SM and AM) and two baselines. The first baseline (D2D) serves the customers door-to-door (i.e., it does not assign stops), and the second one (NBS) selects the nearest stop from the origin as the pick-up point and that from the destination as the drop-off point. For each algorithm, we report the total distance traveled by the fleet and the average walking distance per user. Note that the walking distance is not reported for D2D, as the trips are door-to-door and users do not walk. The results show that simply pre-assigning users to bus stops and using a PDPTW solver (i.e., NBS and SM) can decrease up to 12% the distance traveled by the buses in comparison with D2D. Note that SM's performance does not scale well, as having more customers needs more samples to explore more bus stop combinations. Nevertheless, this could be improved by biasing the sampling mechanism and by adding the NBS instances as samples. Lastly, when comparing D2D with a dedicated algorithm (i.e., AM), being able to consolidate bus stops yields improvements up to 27% in terms of distance at the cost of a user's average walk distance of about 400m (or 5.33 minutes).

In conclusion, in this paper, we studied the assignment of transportation requests to bus stops in on-demand transportation. We observe that, by introducing reasonable walking efforts for the passengers, we create a consolidation that strongly impacts the

Table 1: Comparison of the proposed algorithms with the baselines.

| | D2D | NBS | | SM | | AM | |
|---|---|---|---|---|---|---|---|
| Set | Dist. (km) | Dist. (km) | Walk (m) | Dist. (km) | Walk (m) | Dist. (km) | Walk (m) |
| 50 | 241.83 | 225.38 | 238.52 | 217.27 | 399.03 | **194.45** | 405.34 |
| 100 | 405.67 | 370.40 | 237.86 | 358.83 | 395.71 | **309.90** | 409.75 |
| 250 | 854.01 | 760.54 | 237.83 | 727.99 | 400.171 | **617.41** | 404.99 |
| 500 | 1,593.34 | 1,390.37 | 236.46 | 1,408.42 | 397.173 | **1,135.40** | 403.67 |
| Avg. | 773.71 | (-11%) 686.67 | 237.67 | (-12%) 678.13 | 398.023 | (-27%) **564.29** | 405.94 |

length of routes, decreasing transportation costs and pollution.

## References


[1] P. Vansteenwegen, L. Melis, D. Aktaş, B. D. Galarza Montenegro, F. Sartori Vieira, and K. Sörensen, "A survey on demand-responsive public bus systems", *Transportation Research Part C: Emerging Technologies* 137, 103573, 2022.

[2] Via Transportation, 2024. *Public mobility solutions.* https://ridewithvia.com/.

[3] L. Melis and K. Sörensen, "The static on-demand bus routing problem: large neighborhood search for a dial-a-ride problem with bus station assignment", *International Transactions in Operational Research* 29(3), 1417-1453, 2022.

[4] M. Stiglic, N. Agatz, M. Savelsbergh, and M. Gradisar, "The benefits of meeting points in ride-sharing systems", *Transportation Research Part B: Methodological* 82, 36-53, 2015.

[5] T. Le Colleter, D. Dumez, F. Lehuédé, and O. Péton, "Small and large neighborhood search for the park-and-loop routing problem with parking selection", *European Journal of Operational Research*, 308(3), 1233-1248, 2023.

[6] S. Røpke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows", *Transportation Science* 40(4), 455-472, 2006.

[7] D. Dumez, C. Tilk, S. Irnich, F. Lehuédé, and O. Péton, "Hybridizing large neighborhood search and exact methods for generalized vehicle routing problems with time windows", *EURO Journal on Transportation and Logistics* 10(4), 100040, 2021.

[8] J. Christiaens and G. Vanden Berghe, "Slack induction by string removals for vehicle routing problems", *Transportation Science* 54(2), 417-433, 2020.

# A supervised machine learning approach for replenishment order decisions under transportation cost uncertainty

**Daniel Müllerklein**
School of Management,
Catholic University of Eichstaett-Ingolstadt,
Auf der Schanz 49, Ingolstadt, 85049, Germany
Email: daniel.muellerklein@ku.de

**Pirmin Fontaine**
School of Management,
Catholic University of Eichstaett-Ingolstadt,
Auf der Schanz 49, Ingolstadt, 85049, Germany
Email: pirmin.fontaine@ku.de

**Janosch Ortmann**
Département d'analytique, opérations et technologies de l'information,
Université du Québec à Montréal, 405 Rue Sainte-Catherine, Montréal, Canada
Email: ortmann.janosch@uqam.ca

## 1 Introduction

Companies worldwide seek a balance between risks and cost-efficiency in their supply chains. Due to climate change and the increase in extreme weather events, global inland waterway transport disruptions gained growing attention as low- and high-water-level situations affect the shipment carriers. As a result, they enforce contractual surcharges based on the actual water level situation at the time of shipment. Therefore, affected companies are undertaking efforts to re-design their supply chains. Two primary levers to increase resilience are increasing inventory and sourcing from multiple suppliers [1].

To improve efficiency and resilience, one open question is, therefore, how replenishment policies for multiple suppliers with lead time differences need to be adjusted to account for the transportation cost uncertainty driven by the enforced surcharges. Though disruptions are recurring, the probabilistic characteristics of timing and impact in practice are unknown. Thus, replenishment decisions need to be determined in a data-driven way.

However, determining the optimal replenishment decisions, i.e., when to order, how many units, and from which supplier is a challenging problem for decision makers that replenish inventory from multiple suppliers. While demand uncertainty has been studied extensively, the effects of transportation uncertainty, despite its impact on total costs, are poorly understood. In addition, there is still a need for data-driven approaches in inventory management [2]. We address these two gaps by formulating the problem as a stochastic Inventory Routing Problem with Direct Deliveries (SIRPDD) and introducing a new machine-learning for policy (ML4P) model with a new approach to hyperparameter tuning for data-driven replenishment decisions with multiple suppliers and transportation cost uncertainty.

## 2 Problem setting

For the SIRPDD, we consider a single product that is distributed with direct deliveries from multiple suppliers to a single customer location. Each direct delivery is done through a transportation mode that differs by supplier in terms of transportation costs and delivery times. The process is modeled in discrete time periods. The single customer location stores inventory to satisfy its demands, whereas unlimited supply capacity is available from suppliers. The customer has a known inventory storage capacity. At the beginning of each period, deliveries from suppliers arrive that were ordered delivery time periods earlier. Then, a customer's demand is fulfilled, which is known. In case the demand exceeds the available inventory, shortages occur. After the fulfillment of customer demand, replenishment orders are placed. Lastly, at the end of each time period an inventory quantity remains that forms the starting inventory of the next time period. Inventory holding and shortage costs are charged to the inventory level at the end of the period, while transportation costs are determined at the time period of arrival.

Transportation modes are prone to disruptions. These disruptions are uncertain concerning their impact on transportation cost increases, duration, and time of occurrence during the planning horizon. Hence, the decision maker needs to anticipate the relevant transportation costs in delivery time periods in the future since the surcharge is not defined when the reorder is placed but when replenishment is executed. No probability distribution for the future occurrence of disruptions is known; however, historical information is available.

Thus, a central decision maker manages inventory at the customer location and decides on the supplier to replenish from, how much to replenish, and when to replenish, given that a replenishment order can be made at each of the time periods. The overall objective is to minimize the total expected costs as sum of inventory holding, transportation, and shortage costs. Specifically, we minimize the trade-off between the costs of paying a cost premium to replenish for an alternative supplier or building inventory when no disruption occurs and the uncertain cost increase from a replenishment that is affected by a disruption.

## 3 Solution method

We develop a solution procedure through a combination of mathematical optimization with machine learning to predict optimal replenishment orders using historical data only.

This approach builds on the idea that a replenishment policy can be learned from historical optimal decisions. First, we create labels by solving the IRPDD assuming perfect information. We vary starting inventories and planning horizons to ensure a sufficient degree of the uncertainty space is covered. Then, for each order size and each supplier, we train a decision tree classifier on the labels. During training, we conduct the hyperparameter tuning. In contrast to classical machine learning-optimization frameworks that optimize for classifiers' individual prediction performance, we optimize hyperparameters to minimize the costs of applying the resulting replenishment policy directly. This evaluation is particularly important as otherwise inter-dependencies between the individual classifiers are neglected. Thus, we ensure that the hyperparameters chosen result in overall efficient and stable replenishment decisions rather than individual best-fit classifiers that are tailored to the most accurate prediction of historical disruptions.

In order to evaluate all possible combinations of values for the different hyperparameters, a large number of training and evaluation steps is required. In addition, the cost evaluation is computationally more costly than comparing test labels against test predictions. To overcome this rise in computational complexity, we develop a genetic algorithm with the effectiveness of predicted replenishment orders as fitness function to find a well-performing hyperparameter set.

## 4 Results

We test our approach in a case study with real disruption data. This case is based on a chemical company at the border of the river Rhine, Germany. We split available data into a training, validation, and evaluation set. The main supplier delivers through the river while the alternative is disruption-free but at a cost-premium. In addition, we consider two order sizes for the main supplier. Relevant features include the inventory position, historical water level, their trends, and predictions.

We compare our ML4P model against four benchmarks and the perfect information (PI) solution. The benchmarks include two machine learning models optimized for prediction performance using a decision tree (ML-DT) and a neural network (ML-NN) as classifier. In addition, we compare against a standard reorder point reorder quantity (s,Q)

| | | PI | ML4P | ML-DT | ML-NN | (s,Q) | RA |
|---|---|---|---|---|---|---|---|
| Obj | Costs | 323.79 | 476.61 | 566.72 | 713.73 | 582.18 | 599.58 |
| | Increase | 0% | 47% | 75% | 120% | 80% | 85% |
| Orders | Large main | 14 | 24 | 23 | 54 | 0 | 0 |
| | Normal main | 97 | 74 | 80 | 18 | 147 | 0 |
| | Alternative | 12 | 14 | 9 | 0 | 0 | 147 |

Table 1: Performance of ML4P on evaluation set against benchmarks.

policy and the risk avoidance of only ordering from the alternative supplier (RA). All machine learning models are trained on the training data split while the ML4P model is tuned on the validation set. Table 1 summarizes the results on the evaluation set.

Overall, the ML4P model significantly outperforms all benchmarks. Specifically, cost savings of 18% against the (s,Q) industry standard as well as 16% cost savings through the different hyperparameter tuning objective can be achieved. The remaining 47% gap to the perfect information solution highlights the high problem complexity and degree of uncertainty.

## 5 Conclusions

In summary, the contributions are: We have introduced a new problem setting for the inventory replenishment problem with multiple suppliers, lead time differences, and transportation cost uncertainty to decide when to source, which quantity, from which supplier. We have developed a new machine learning framework, ML4P, that directly optimizes the effectiveness of the different replenishment decisions within the hyperparameter tuning. We have tested the ML4P in a case using real-life disruption data and shown that it significantly outperforms standard replenishment policies and traditional machine learning frameworks for inventory replenishment. Finally, we have highlighted managerial implications on the data-driven inventory replenishment with multiple suppliers and cost uncertainty, leveraging public databases for waterway transport disruption uncertainty.

## References

[1] S. Hosseini,D. Ivanov, and A. Dolgui, Review of quantitative methods for supply chain resilience analysis, *Transportation Research Part E: Logistics and Transportation Review*, 125(1), 285-307, 2019.

[2] J. Svoboda and S. Minner, Typology and literature review on multiple supplier inventory control models, *European Journal of Operational Research*, 293(1), 1-23, 2021.

# Combining scheduled and responsive approaches for daytime bike-sharing redistribution

**Bruno A. Neumann-Saavedra**
Decision Support Group,
Technische Universität Braunschweig, Germany
Email: b.neumann-saavedra@tu-bs.de

**Dirk C. Mattfeld**
Decision Support Group,
Technische Universität Braunschweig, Germany

**Mike Hewitt**
Quinlan School of Business
Loyola University, Chicago, United States

## 1 Introduction

Bike-sharing systems (BSSs) provide an eco-friendly means of public transport with which a user can rent a bike from one service location, make a trip, and return that bike at any service location. A common consequence of those one-way trips is that many users may fail to rent or return bikes due to shortages of bikes and empty parking spaces, respectively. Therefore, BSS operators face the challenge to redistribute bikes among service locations during the day. The so-called daytime redistribution thus seeks to ensure that users can rent and return bikes where and when they need them. The literature proposes two general approaches to perform daytime redistribution. We call the first one "scheduled redistribution" and the second one "responsive redistribution". For an extensive literature review, the reader is referred to [1].

Scheduled redistribution involves planning regular visits to service locations in advance to pick up and deliver bikes. Under this approach, historical riding data is used to derive demand patterns that help to predict when service locations require service. It is intended that the resulting redistribution schedule is performed as prescribed day after day. Given that resources for performing daytime redistribution are usually scarce, scheduled redistribution is a cost-effective manner of guiding daily operations to service locations that frequently run full or empty. However, scheduled redistribution may not always align with the real-time needs of service locations. Suppose a truck visits a service location as prescribed for picking up a large bike quantity. However, the truck cannot pick up the prescribed bike quantity due to the low bike inventory at that service location. This shortfall may have a cascading effect over time. Suppose in a subsequent visit to a service location the truck is expected to deliver a large bike quantity. However, since the truck could not pick up sufficient bikes during the previous visit, the truck now arrives at that service location with an empty or near-empty truckload, being unable to deliver sufficient bikes. Therefore, considering a rigid redistribution may lead to an inferior solution quality, see [2] for a deep discussion.

Responsive redistribution, on the other hand, entails real-time monitoring of bike inventory by operational control, see, for example, [3]. This approach dispatches trucks to service locations where redistribution is needed the most at a given point in time. Following this approach, rebalancing drivers react flexibly to inventory fluctuations. However, responsive redistribution can be costly due to more frequent dispatches to trucks compared to scheduled redistribution. Additionally, responsive decisions are short-sighted and do not anticipate global future system dynamics suitably.

In this work, we propose a hybrid approach that combines the efficiency of scheduled redistribution with the adaptability of responsive redistribution. In this hybrid approach, we generate a redistribution schedule a priori as a guideline for truck operations. Then, when the trucks are operating, we compare the expected short-term benefit, say, the shortages of bikes and empty parking spaces that deviation can potentially save, of following the redistribution schedule with the best detour suggested by responsive redistribution. Such a detour deviates from the redistribution detour at a point in time to visit an alternative service location, returning later to the original schedule. In this way, we consider not only the effect at the service location the truck is currently placed, but also evaluate the extent to which a detour affects future actions.

## 2 An illustrative example of decision-making

Consider the example depicted in Figure 1. The vertical rectangles represent five service locations from $n_1$ to $n_5$. The white and gray square inside each rectangle indicates that the parking space is empty or occupied, respectively. The horizontal rectangles represent the trucks, $v_1$ and $v_2$, where the number of gray squares indicates the truckload. In the point of time $t = 10$, the truck $v_1$ is located in service location $n_1$, whereas the truck $v_2$ is located in service location $n_3$.

The solid lines represent the scheduled paths, depicting the option that the truck follows the redistribution schedule. The dashed lines represent the responsive paths, depicting the option that the truck deviates from that schedule, say, a detour. In the example, we see both responsive paths suggest visiting the service location $n_4$. We presume only

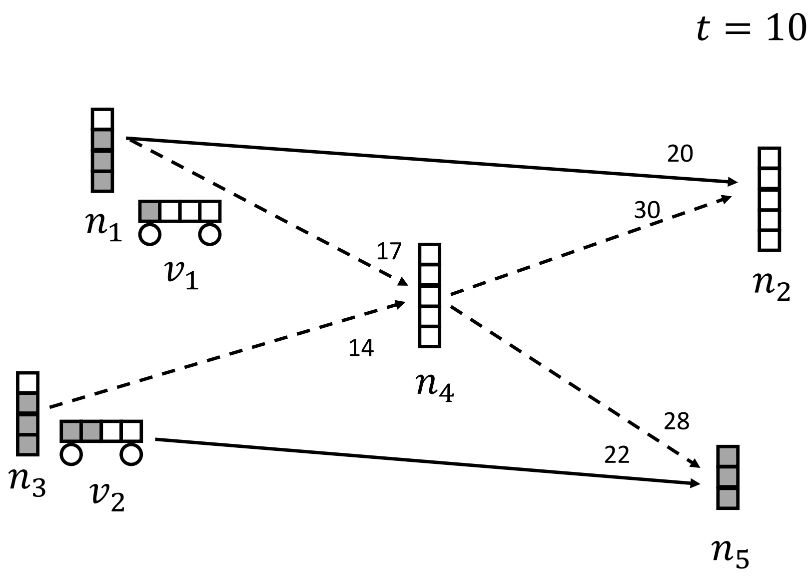



Figure 1: Example.

one of those responsive paths can be taken. The number above each line represents the expected time in which the truck would arrive at the respective service location.

The truck $v_1$ might pick up bikes from the service location $n_1$ and follow the responsive path. Doing so, however, involves that vehicle delivery bikes to service location $n_4$ and arrive (probably) empty to the service location $n_2$. Thus, for that truck, the selection of a path depends on the number of failed rentals one can potentially save at those service locations. The situation for truck $v_2$ is different since the service location $n_5$ is already full. This means that the service location $n_5$ might not require redistribution with urgency. Therefore, it makes sense that the truck $v_2$ picks up bikes from the service location $n_3$ and then follows the responsive path to visit the service location $n_4$. Consequently, the truck $v_1$ would follow the responsive path.

## 3 Modeling approach

In this work, we follow the multi-truck Sequential Decision Process (SDP) proposed by [3] to model daytime redistribution. At a decision state, that is, when a truck arrives at a service location, a decision is made about the path each truck has to follow, ensuring that no trucks are located in a service location simultaneously. Decisions are based on short-term forecast about the expected number of shortages of bikes and empty parking spaces. Once a decision is made, the SDP updates bike inventories as well as the position and loads of trucks until the next decision state is triggered. Given an initial state, the optimal policy minimizes the expected number of shortages over all possible decisions.

## 4 Managerial insights

To assess the performance of our hybrid approach, we consider three BSSs located in San Francisco (SF), Minneapolis (MN), and Boston (BO), with different characteristics in terms of number of service locations and regularity in demand. To sum up, SF is a small BSS compared to MN and BO. Regarding demand, MN displays a highly irregular demand because of trips related to leisure and shopping, wheras SF and BO display more regular demand due to commuting. For more details, the reader is kindly referred to [2].

In particular, our hybrid approach usually decides to follow scheduled paths when they are prescribed in such a way that they redistribute bikes from full to empty service locations. These highly-utilized scheduled paths visit service locations with high and regular demand, thus the expected benefit of following them is significant. For this reason, the redistribution schedules prescribed by SF and BO perform particularly well, especially between rush hours. Conversely, low-utilized scheduled paths are an indicator that they visit service locations where the daily redistribution needs are irregular. In those cases, the hybrid approach tends to prefer responsive paths as they better reflect the short-term needs of service locations based on the current bike inventory. This is particularly true for MN, where a smaller number of scheduled paths lead to a suitable benefit.

## References


[1] C. S. Shui and W. Y. Szeto, “A review of bicycle-sharing service planning problems.”, *Transportation Research Part C: Emerging Technologies* 117, 102648, 2020.

[2] B. A. Neumann-Saavedra, D. C. Mattfeld and M. Hewitt, “Assessing the operational impact of tactical planning models for bike-sharing redistribution.”, *Transportation Research Part A: Policy and Practice* 150, 216-235, 2021.

[3] J. Brinkmann, M. W. Ulmer and D. C. Mattfeld, “The multi-vehicle stochastic-dynamic inventory routing problem for bike sharing systems.”, *Business Research* 13, 69-92, 2020.

# Choice-Driven Service Network Design and Pricing in a Competitive Environment

**Adrien Nicolet**
Department of Maritime and Transport Technology,
Delft University of Technology, The Netherlands
Email: A.Nicolet@tudelft.nl

**Bilge Atasoy**
Department of Maritime and Transport Technology,
Delft University of Technology, The Netherlands

## 1 Context

In intermodal freight transport, the Service Network Design (SND) problem is of key importance, as it covers most of the tactical decisions such as the itineraries to be served, the offered frequencies and how demand should be assigned to these services [1]. In the existing SND literature, only a handful of works cover pricing decisions and preferences of shippers. In this work, we make use of "choice-based optimization" [2] to combine a SND pricing problem with a mode choice model representing the decisions of shippers. Therefore, we develop a Choice-Driven Service Network Design and Pricing (CD-SNDP) model, which includes a Mixed Logit model to consider heterogeneous behaviors of shippers directly in the decision-making of the transport operator.

## 2 Proposed methodology

The proposed CD-SNDP is a generalization of the bi-level Joint Design and Pricing formulation of Tawfik and Limbourg [3]. Firstly, it generalizes the network structure as cycles and services with multiple stops are allowed. Secondly, the shippers' objective is also generalized as they do not only proceed to a minimization of their costs, but instead maximize their utilities. These utilities contain other attributes beside the costs, such as frequency, accessibility, etc. Thirdly, our formulation generalizes the demand representation as it can accommodate some unobserved attributes (via randomly distributed error terms) and shippers' heterogeneity (through the Mixed Logit formulation). Finally, the service frequency is made endogenous to the optimization model along with the price.

The upper level of the CD-SNDP represents an inland barge operator, whose objective is to maximize their profits under fleet size and capacity constraints. The lower level describes the utility maximization of shippers. They have four transport alternatives: the barge operator, a competing inland waterway carrier, train, and truck. We estimated the utility functions for these alternatives in a previous work, where a Mixed logit formulation was introduced to represent the heterogeneous cost sensitivities of shippers [4]. This bi-level SNDP problem can be reformulated into a single level linear problem using the strong duality theorem, big M technique and expressing frequencies as binary variables.

With the Mixed logit formulation, we introduce a stochastic version of the Service Network Design and Pricing problem. The problem is then solved using Sample Average Approximation (SAA). We compare four models: a benchmark where shippers are purely cost minimizers, a version of the CD-SNDP using only the deterministic part of the utility functions, and two stochastic variations: one with a Multinomial Logit (MNL) including randomly distributed error terms and one with the Mixed Logit. The obtained solutions are assessed through an out-of-sample simulation, which imitates the behavior of a real population of shippers. In this simulation, the optimal service frequencies and prices obtained by the models are plugged into the utility functions of this real population of shippers and the actual profits can be computed.

## 3 Key results

We apply the proposed methodology to a 3-node intermodal network between the port of Rotterdam and the two inland ports of Duisburg and Bonn, using input data from [5] and [6]. The profits obtained by the four models are compared together with the actual profits resulting from the out-of-sample simulation in Figure 1.

Compared to the benchmark, the deterministic version of our CD-SNDP returns actual profits that are more than 2.5 times higher. This is because the choice-driven model also considers frequency in the mode choice, so the operator can charge higher prices by compensating them with a higher frequency. Only the costs are considered in the benchmark, making such a trade-off impossible. The expected and actual profits are also closer with the CD-SNDP as the shippers' representation used in the optimization is closer to the real behavior.

Adding stochastic elements to the CD-SNDP allows to further increase the actual profits and reduce the gap between expected and actual profits. Indeed, with the addition of error terms (MNL) and the heterogeneous cost sensitivity of shippers (Mixed Logit), the representation of shippers within the optimization becomes closer and closer to reality. Compared to the version with MNL, the actual profits of the stochastic version with Mixed Logit are only marginally improved. However, the latter version provides the most

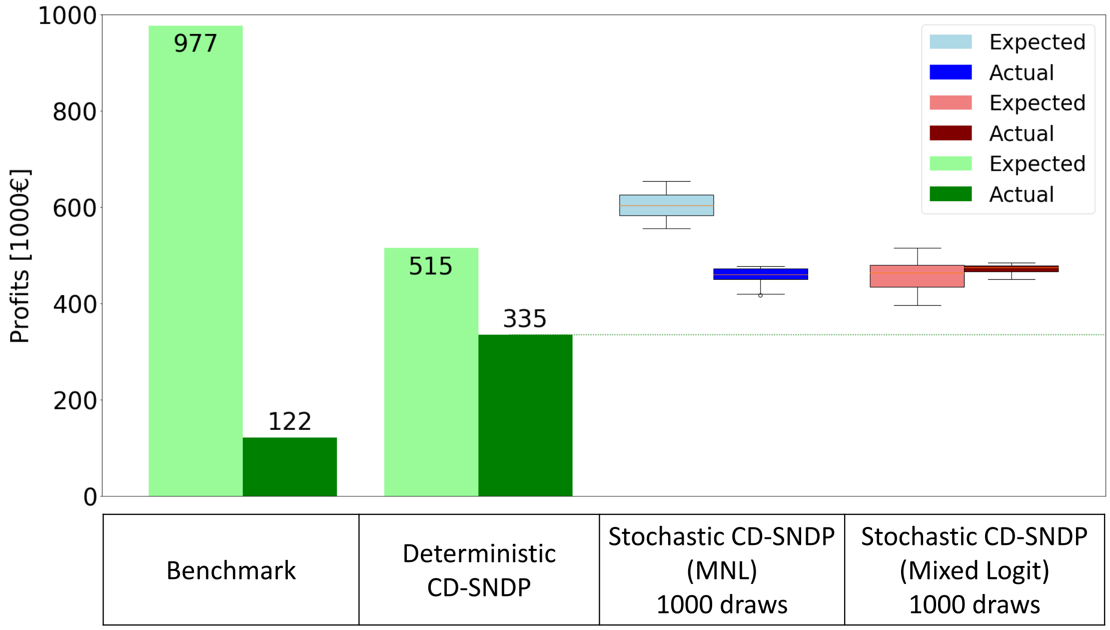



Figure 1: Comparison of profits obtained by the different models

accurate estimation of the profits, where all the other versions considerably overestimate the expected profits.

## 4 Conclusions and Further Work

With this work, we show that it is highly beneficial for a transport operator to include the information they have about the demand during the design of their services. Indeed, the profits achieved by our CD-SNDP are substantially higher than the benchmark, even if the embedded mode choice model is simply deterministic. This is because the benchmark's assumption that shippers are purely cost-minimizers neglects other attributes that still play a role in the decision-making of shippers, such as the service frequencies. Moreover, making use of stochastic CD-SNDP exploits the potential of the model further. Indeed, perfect and complete information about the shippers is not available to the operator, so that their demand model will miss some aspects that play a role in the shippers' choices. These aspects can indirectly be accounted for by adding random error terms in the model. Finally, quantifying and incorporating the heterogeneous preferences of shippers allows to get a better prevision of the profits. All in all, including more information about the shippers while designing and pricing the services results in considerable gains for the transport operator.

Now that the efficiency of our methodology has been shown with a 3-node network, the immediate next step is to apply the CD-SNDP to a larger instance. Moreover, we assumed that the competition is fixed and exogenous. It means that the competing carriers do not react to the services and prices set by the operator. But the competitors will also seek to improve their services and profits, even more so if they lose market share to the operator. To capture these market dynamics, a competition model is being developed, where two or more operators iteratively solve a Service Network Design and Pricing model so as to maximize their own profits. This model will enable to experiment with different degree of information available to the operators. Indeed, the situation with full and perfect information is convenient theoretically, but never occurs in reality. That is why we want to investigate the influence of limited information on the decisions and interactions of transport operators.

## Acknowledgement

This research is supported by the project "Novel inland waterway transport concepts for moving freight effectively (NOVIMOVE)". This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 858508.

## References

[1] T.G. Crainic, "Service network design in freight transportation", *European Journal of Operational Research* 122(2), 272–288, 2000.

[2] M. Pacheco Paneque, *A general framework for the integration of complex choice models into mixed integer optimization*, Ph.D. thesis, EPFL, 2020.

[3] C. Tawfik, and S. Limbourg, "A bilevel model for network design and pricing based on a level-of-service assessment", *Transportation Science* 53(6), 1609–1626, 2019.

[4] A. Nicolet, R. Negenborn, and B. Atasoy, "A logit mixture model estimating the heterogeneous mode choice preferences of shippers based on aggregate data", *IEEE Open Journal of Intelligent Transportation Systems* 3, 650-661, 2022.

[5] I. Majoor, et al., "D.2.2: Novimove transport model architecture and data collection", *Technical report*, 2021.

[6] P. Shobayo, A. Nicolet, E. van Hassel, B. Atasoy, and T. Vanelslander, "Conceptual development of the logistics chain flow of container transport within the Rhine-Alpine corridor", *Proceedings of the European Transport Conference (ETC)*, 1-17, 2021.

# Collective Distribution Network Design Problem

**Daniel M. Ocampo-Giraldo, Ana María Anaya-Arenas, Janosch Ortmann**
Département AOTI, École des Sciences de la Gestion
Université du Québec à Montréal and CIRRELT, Canada
Email: ocampo_giraldo.daniel_mauricio@courrier.uqam.ca

## 1 Introduction and Problem Description

A collective distribution network is made of stakeholders who agree to share resources such as vehicles, warehousing space, workforce and information in favor of cost reduction, improvement of goods exchange, maximization of vehicle utilization, enterprise carbon footprint reduction, and so on. Thus, the design of a collective distribution network must decide on the distribution and usage of such shared resources.

In this network we consider a set of companies. Each company has several suppliers and customers and needs to transport a quantity of commodities across the network. In addition, each supplier and customer has a restrictive operational time when commodities can be loaded or unloaded, as well as perishability constraints. To share transportation resources, we assume that transporters can meet at different places (e.g. warehouses and distribution centers) where they can consolidated commodities or transfer them to another vehicle if appropriate. Regardless of the nature of the activity, we refer to these transshipment points as **hubs**. Thus, hubs are facilities that members of the collective distribution network share to be used as transshipment points or consolidation points for all network members' commodities. Each possible hub location has its own set-up requirements, and the number of resources shared is different (i.e., each company provides a different area and workforce to the network). Henceforward, in a collective distribution network, hubs are facilities where commodities transshipment can take place, with different set-up costs and capacities. Nevertheless, commodities exchange can be fulfilled without passing through a hub.

In summary, the collective distribution network design problem looks to minimize the set-up cost for the hubs and the distribution cost for the commodities under operational constraints concerning capacity (hub and vehicle), operational time and driver workload. In this distribution network design problem there are two types of decisions: (i) which hubs out of a finite set of potential ones should be opened and (ii) which vehicle routes should be built to serve the origin-destination commodities flow demand using a given fleet.

Collective distribution network design is similar to the Many-to-Many Location Routing Problems (MMLRP) because both of them decide on hub location and flow distribution. Nevertheless, the design of a collective distribution network has flexibility that goes beyond the hub location's usual assumptions. System flexibility, moreover, implies the inclusion of the time dimension to ensure system operation, i.e., commodities consolidation must occur prior to the delivery route. As mentioned by [1] time and synchronization play a key role in hub location problems, but there is little or no research involving synchronization in MMLRPs. The same lack of research is highlighted by [4] in their literature review of intermediate facilities in freight transportation. Synchronization in vehicle routing problems has been the subject of several studies ( a survey can be found in [2]), although, when location decisions should be taken most papers ignore synchronization ([3]). A two-echelon location-routing problem with synchronization is studied by [5]. They study a problem faced by a postal service company that serves customers outside densely populated areas, which motivates the need to locate intermediate facilities and synchronize vehicles traveling between a terminal and facilities with those traveling between facilities and customers.

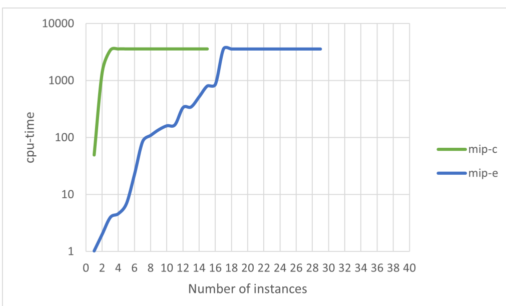
## 2 Mathematical Modelling

For the design of this distribution network, we propose two different Mixed Integer Programming (MIP) models that jointly consider location and routing decisions. The first model is a flow model that uses continuous variables to track lorries' time. These variables are used to decide when and which vehicle serves each commodity transportation request. Also, synchronization constraints are guaranteed through those variables. We name this model as ***mip-c*** formulation. Our second approach is a MIP model that uses time-extended network, where time decisions are given jointly with flow decisions and it uses discrete time intervals over the planning time horizon. We refer to the latter model as ***mip-e*** formulation. In both cases, the time definite synchronization of flow and vehicles are explicitly integrated in the formulation to assure that time restrictions are respected.

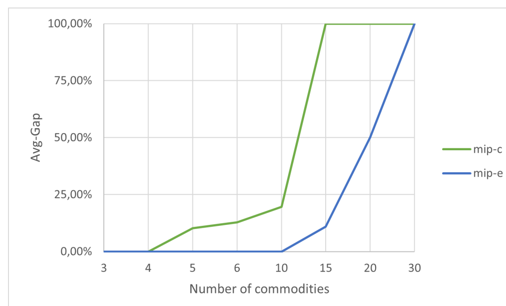
Both formulations have their advantages and shortcomings, for example, *mip-c* model size (number of variables and constraints) stays manageable when the number of transportation requests increases, but its linear programming (LP) relaxation leads to weak lower bounds, increasing the computational cost of the tree exploration. On the other hand, *mip-e* model allows the management of synchronization constraints through flow conservation constraints, that induce strong LP relaxation bounds. Nevertheless, the model size rapidly grows with the number of transportation requests, the extension of the time-horizon and the granularity of the time discretization.

# 3 Results and Discussion

Mathematical formulations are tested over a set of 40 instances adapted from [6]. A hub set-up cost is included to assess the value of the location decisions over the vehicles' routes when both decisions are taken by the model. Instances size goes from 3 up to 30 transportation requests and always considers four possible hub locations. These instances are solved with the two models using Gurobi solver, with a time limit of 3600 seconds. The results obtained from these tests are summarized in Figure 1. The reported results correspond to instances where a solution that is different from 100% optimality gap is obtained within the time limit. Figure 2 compares the hub's contribution to the solution found by *mip-e* formulation.

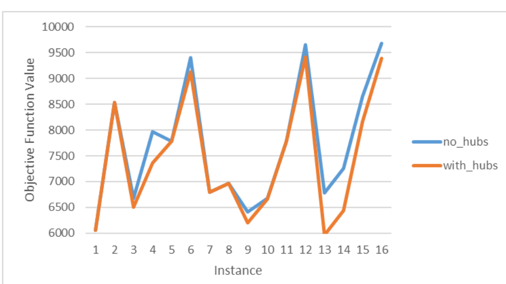



(a) Number of instances solved

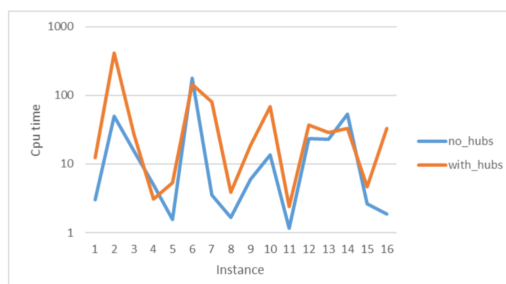



(b) Average Optimality Gap

Figure 1: mip-c Vs mip-e




(a) Routing cost




(b) Solution time

Figure 2: Hubs contribution

Preliminary results show that *mip-e* has a better performance than *mip-c* in terms of solution quality and computational efficiency. For example, within the one-hour time limit *mip-e* solves 28 instances to optimality, while *mip-c* can only solve 14 (Figure 1a). Moreover, *mip-c* cannot solve to optimality instances of more than 10 commodities, while with *mip-e* can (Figure 1b). These preliminary results show that *mip-e* exploits better the structure of the problem, reducing the search space. However, both models face scalability issues as none of the instances with more than 20 commodities can be solved within the time limit. Regarding the solution structure, the use of transshipments leads to lower routing costs (Figure 2a), but at the expense of higher computational time (Figure 2b).

## 4 Conclusion

This work has explored a new problem of collective distribution network design, where multiple partners work in a shared network with hubs and routing decisions. The problem aims to maximize the use of limited resources while satisfying the demand of the origin-destination pairs. The selected hubs can act as commodities' transshipment points between vehicles and the plan include explicitly synchronization and tight time restrictions for multiple commodities. We have proposed two mixed-integer programming models for the problem, *mip-e* and *mip-c*, and evaluated their performance on a set of instances. The preliminary results indicate that *mip-e* has a better performance than *mip-c* in terms of solution quality and computational efficiency. However, both models have scalability issues when the problem size grows. Therefore, to solve real-world scenarios, we plan to develop and test alternative solution methods.

## References


[1] Sibel A Alumur et al. "Perspectives on modeling hub location problems". In: *European Journal of Operational Research* 291.1 (2021), pp. 1–17.

[2] Michael Drexl. "Synchronization in vehicle routing - A survey of VRPs with multiple synchronization constraints". In: *Transportation Science* 46 (3 2012), pp. 297–316. ISSN: 15265447. DOI: 10.1287/trsc.1110.0400.

[3] Michael Drexl and Michael Schneider. "A survey of variants and extensions of the location-routing problem". In: *European Journal of Operational Research* 241.2 (2015), pp. 283–308.

[4] G. Guastaroba, M. G. Speranza, and D. Vigo. "Intermediate facilities in freight transportation planning: A survey". In: *Transportation Science* 50 (3 2016), pp. 763–789. ISSN: 15265447. DOI: 10.1287/trsc.2015.0631.

[5] Seyed Mostafa Mirhedayatian et al. "A two-echelon location-routing problem with synchronisation". In: *Journal of the Operational Research Society* 72 (1 2021), pp. 145–160. ISSN: 14769360. DOI: 10.1080/01605682.2019.1650625.

[6] Afonso Sampaio et al. "Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers". In: *Networks* 76.2 (2020), pp. 232–255.

# The hub line location problem with time-definite deliveries

**Esteban Ogazón, Ángel Ruiz**
Department of Operations and Decisions Systems, Faculty of Business Administration, Université Laval, Quebec City, Canada; and Interuniversity Research Centre on Enterprise Networks, Logistics, and Transportation (CIRRELT), Canada
Email: estebanogazon@gmail.com
**Ana María Anaya-Arenas**
Department of Analytics, Operations, and IT, ESG-UQAM, Interuniversity Research Centre on Enterprise Networks, Logistics, and Transportation (CIRRELT), Canada

## 1 Introduction

The hub location problem (HLP) stands at the core of operational research for optimizing network design, especially within the domain of logistics. In this context, the urgency of time-definite deliveries has intensified with the rise of e-commerce and same-day delivery services. The efficiency of these logistics networks is significantly influenced by the topology of the network design [1]. Selecting an optimal structure is crucial for the efficacy of the operations, as it dictates the flow and cost of transportation within the network. A notable development is the hub line location problem (HLLP) [2]. Its objective is to determine the most cost-effective configuration of $p$ hubs connected linearly by $p - 1$ arcs, simplifying routing and scheduling. Motivated by the requirements of modern regional transportation networks, this paper introduces an innovative extension to the Hub Line Location Problem (HLLP) by incorporating time-definite delivery constraints, resulting in the formulation of the Time-Definite Hub Line Location Problem (TD-HLLP). This formulation upholds the advantages of the traditional hub line model while addressing the operative needs to ensure punctuality in service delivery, a factor commonly overlooked by hub location studies [3].

Different regional distribution contexts, as those of e-commerce or healthcare networks, require that to warrant a service level of a few hours, managers need to carefully plan how pickups and deliveries need to be synchronized in the daily operations. For instance, in a regional healthcare network, the healthcare facilities demand/produce perishable commodities or items of urgent transportation (e.g., food, biomedical samples, sterilized equipment, etc.) with high frequency. Here, the required interconnexion between facilities could be of even less than four hours. On other side, in a regional network of independent food growers, the need of interconnectivity could go from five to seven hours. In an e-commerce network, service level can be up to 12 hours. To approach these problems and to effectively introduce the time-definite delivery factor into the HLLP, it becomes imperative to make decisions that go beyond selecting which hubs are connected. We must also determine how and when they are visited by the transportation vehicles. We hence study the network design with one and with two vehicles, that begin their routes at each end of the line and traverse it in opposite directions, to assure efficient coverage and timely delivery. Our study proposes two contributions. First, the formulation of the Time-Definite Hub Line Location Problem (TD-HLLP), with a novel set-packing approach to enhance efficiency in time-sensitive deliveries. Second, we analyze the hub-line structure performance under different operational scenarios using instances inspired by the transportation needs of a real regional network.

## 2 Problem definition

The HLLP-TD is formulated under the premise of minimizing the total transportation costs within a fully connected network. It is formalized over a complete graph $G = (D, A)$, where $D$ is the set of nodes and $A$ is the set of arcs. In this network, each node origin $i \in D$ needs to send a single unit demand to every other node in the network (destination $j \in D$). The problem seeks the most efficient design that assures a time-definite delivery for every node in the network. Consolidation and synchronization of flow become crucial as the economies of scale are archived through shipment consolidation in selected hubs. The problem decides the location of $p$ transhipment facilities (hubs) in the network over a set of possible locations ($H \subseteq D$). We consider a single allocation setting and the strict use of hubs to interconnect nodes, where $x_{ik}$ is a binary variable that represents the decision of allocating node $i \in D$ to hub $k \in H$ and $x_{kk} = 1$ represents the location of a hub in node $k \in H$. Here, each node in the network is to be connected to exactly one hub and direct deliveries are not allowed. The model decides also on the interconnection of hubs (the line structure), where every hub must be part of the inter-hub route, with either one or two vehicles.

The service level that can be assured depends on the geographical distribution of the network and the logistics requirements of the commodities to move. We consider that the service level at these regional networks can be of a three to eight hours. Therefore, the selection of the hub line needs to warrant that the *collection*, *transhipment,* and *delivery operations* will respect a high service level. Hence, the HLLP-TD considers explicitly the following process. First, every node has a period of accumulation of demand. Subsequently, the *collection process* starts simultaneously at all nodes by direct deliveries to their entry point to the line. The collection period for a hub consists then of receiving all the items coming from their assigned spokes. Once a hub completes the collection period, the *transhipment process* starts in an inter-hub vehicle, requiring a time $m_k$ to process the items at hub $k \in H$. Therefore, a route can pass by a hub only after its collection process has been completed. Finally, the *delivery process* consists of sending from each hub the received items to their destined spokes. A hub can initiate the delivery process only if the route has reached the hub from both directions, ensuring a synchronized and coordinated transportation process.

We implement a set-partitioning approach to model the inter-hub routing. Let $r \in R$ be the set of routes, each composed of a set of potential hubs $H^r \subseteq H$. Binary variable $z^r = 1$ represents the use of route $r \in R$. Each metro-line route is traversed in two directions, denoted by $\theta \in \Theta$. The set of directed arcs connecting pairs of hubs in route $r$ in direction $\theta$ is defined by $A^{r\theta}$, indexed by $(k, l): k, l \in H^r$. Figure 1 depicts a metro-line routing structure with three hubs $(k_1, k_2, k_3)$ and their corresponding spokes. To ensure that all items traversing the inter-hub route are consolidated in the vehicle traveling in their corresponding direction, we compute the time at which the vehicle complies with the route progression policy. The earliest time at which the collection and processing of all the items is completed at hub $k$ using route $r$ is given by consolidation time $\alpha_k^{r\theta}$. We can then compute the earliest time at which the collection is finished, and the vehicle is ready to perform the transshipment at every hub. We develop two formulations adapted to consider a limited fleets of one or two vehicles respectively.

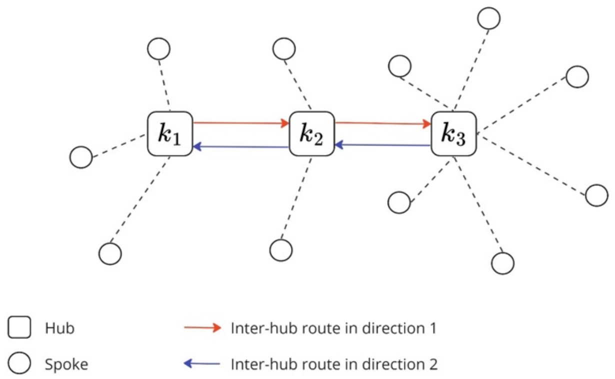


Figure 1. Metro-line structure

## 3. Preliminary results

To model realistic scenarios for our initial tests, we use real data instances from healthcare institutions from the province of Quebec. The size of each instance is described by the number of nodes $|D|$ and the average travel time between nodes in minutes $\bar{t}$. We solve the

HMLLP for $p = (3, 4, 5)$, and Table 1 reports the total cost obtained for each instance with service level $T = (240, 360)$. Missing values represent instances with no feasible solution for a given configuration. All instances were solved using Gurobi 10.0.1 in less than 5 seconds.

Table 1. Summary of computational results

| Instance | $\|D\|$ | $\bar{t}$ | $p = 3$ | | $p = 4$ | | $p = 5$ | |
|---|---|---|---|---|---|---|---|---|
| | | | $T = 240$ | $T = 360$ | $T = 240$ | $T = 360$ | $T = 240$ | $T = 360$ |
| 1 | 25 | 73.6 | - | 2799.9 | - | 2783.7 | - | 2782.1 |
| 2 | 27 | 78.6 | - | 2237.5 | - | 2148.9 | - | 2119.2 |
| 3 | 28 | 77.9 | 2733.7 | 2588.1 | - | 2530.5 | - | 2497.7 |
| 4 | 36 | 80.2 | - | 5019.1 | - | 5017.7 | - | 5017.4 |
| 5 | 41 | 85.0 | - | 6102.3 | - | 6099.1 | - | 6098.9 |
| 6 | 47 | 91.4 | - | 6506.5 | - | 6503.2 | - | 6503.0 |
| 7 | 51 | 90.0 | 411.4 | 411.4 | 394.4 | 394.4 | 393.3 | 393.3 |

Our preliminary results confirm the importance of including explicitly the time synchronization constraints to warrant the service level. Without it, the network selected can be infeasible (as it is the case for a service level of four hours in most of the instances). Moreover, the choice of hubs and its connections can be contingent to the service level as can be seen in instance three.

# 4. Conclusion

In this paper we extend the HLLP to include time-definite deliveries and achieve economies of scale through shipment consolidation and synchronization (the HLLP-TD). Our formulation integrates key operational aspects like vehicle routing and consolidation time tracking. This allows managers to assess feasible service levels and comparing the efficiency of the network structure under different time constraints. This approach advances logistics optimization for time-sensitive deliveries. Our future work will explore the practical implementation and benefits of this structure in real-world logistics scenarios.

# References

[1] I. Contreras, E. Fernández, and A. Marín, "Tight bounds from a path based formulation for the tree of hub location problem," *Comput. Oper. Res.*, vol. 36, no. 12, pp. 3117–3127, Dec. 2009, doi: 10.1016/j.cor.2008.12.009.

[2] E. Martins de Sá, I. Contreras, J.-F. Cordeau, R. Saraiva de Camargo, and G. de Miranda, "The Hub Line Location Problem," *Transp. Sci.*, vol. 49, no. 3, pp. 500–518, Aug. 2015, doi: 10.1287/trsc.2014.0576.

[3] S. A. Alumur, J. F. Campbell, I. Contreras, B. Y. Kara, V. Marianov, and M. E. O'Kelly, "Perspectives on modeling hub location problems," *Eur. J. Oper. Res.*, vol. 291, no. 1, pp. 1–17, May 2021, doi: 10.1016/j.ejor.2020.09.039.

# Dynamic Discretization Discovery Algorithm for Airline Timetable Development and Fleet Assignment with Passenger Choice

**Ritesh Ojha**
Georgia Institute of Technology, Atlanta, USA
Email: rojha8@gatech.edu

**Luke Marshall**
Microsoft Research Redmond, Washington, USA

**Introduction** The airline timetable development and fleet assignment problem under endogenous passenger choice, abbreviated as ATFP, aims to create passenger-friendly flight timetables while optimizing fleet utilization for profitability [1]. Designing a flight timetable for even one major airline is an enormous undertaking. Airlines typically make incremental adjustments to existing schedules due to administrative constraints and computational challenges, which limit comprehensive redesigns. These adjustments involve critical decisions like setting flight departure times, assigning aircraft to flights, and satisfying demand for various itineraries. State-of-the-art methods for ATFP involve multi-stage heuristic algorithms that cannot be used by legacy airline carriers. We develop an exact dynamic discretization discovery (DDD) algorithm that iteratively solves relatively small mixed-integer programs (MIPs) formulated over partially time-expanded networks to generate near-optimal solutions; the effectiveness of the algorithm relies on a sophisticated lower-bound MIP model and a novel upper-bound construction procedure. We also develop effective *arc-based* and *path-based* network refinement strategies that enable the DDD algorithm to effectively refine the partially time-expanded network while generating only a small fraction of the complete time-expanded network.

**Problem Description** Consider a set of airports $I$. The host airline operates non-stop flights between airport pairs called *segments*, denoted by $S$. Each segment $s \in S$ has a travel time $\tau_s$ and requires $n_s$ flights to be operated by the host airline daily; every two flight departures in a segment must be separated by a minimum time denoted by $g_s$. Passengers demand to travel between origin-destination pairs or *markets* denoted by $M$. The host airline serves a set of passenger types $P$ and a set of fare classes $L$. Each market $m \in M$ has a set of itineraries $\mathcal{R}_m$ and the price of a ticket for itinerary $r \in \mathcal{R}_m$ and for fare class $\ell \in L$ is $\rho_{r\ell}$. Each market $m \in M$ has demand $d_{mp}$ for passenger type $p \in P$.

The host airline operates a set of fleet types $F$ with count $\kappa_f$ and capacity $Q_f$ for $f \in F$. Let $\alpha^0_{mp}$ (or $\tilde{\alpha}^0_{mp}$) denote the total (or adjusted total) attractiveness of all itineraries of other airlines and of the no-fly alternative in market $m$ for passenger type $p$. Similarly, let $\alpha_{rp\ell}$ (or $\tilde{\alpha}_{rp\ell}$) denote the total (or adjusted total) attractiveness of itinerary $i$ and fare class $\ell$ offered by the host airline for passenger type $p$.

The problem is formulated over a complete time-expanded network $\widehat{G} = (\widehat{N}, \widehat{A})$ where $\widehat{N}$ represents the set of nodes and $\widehat{A}$ represents the set of arcs in the network during a planning horizon $\mathcal{T}$. $\widehat{G}$ has a discretization parameter $\Delta$ which is equal to the greatest common divisor of all time-related parameters in the problem and $\mathcal{T} = [\alpha\Delta, \beta\Delta]$. Each node $(i,t) \in \widehat{N}$ denotes an airport $i \in I$ at time $t \in \widehat{T}_i$, where $\widehat{T}_i = \{\alpha, \alpha+1, \cdots, \beta\}$ is the set of time points at airport $i \in I$. The arc set $\widehat{A}$ includes a set of flight arcs $\widehat{A}_F$, where each arc is of the form $(i,t) \to (j,t')$ where $(i,j) \in S$, $t \in \widehat{T}_i, t' = t + \tau_{ij} \in \widehat{T}_j$, and a set of ground arcs $\widehat{A}_H$ where each arc is of the form $(i,t) \to (i, t+\Delta)$ where $i \in I$ and $t, t+\Delta \in \widehat{T}_i$; $\widehat{A} = \widehat{A}_F \cup \widehat{A}_H$. Let $\widehat{A}_F(s)$ denote the set of flight arcs in segment $s \in S$. Let $\widehat{R}_m(a)$ denote the set of all itineraries of the host airline in market $m \in M$ that uses the flight arc $a \in \widehat{A}_F(s)$ on segment $s \in S$. As the demand is cyclic, i.e., it repeats daily, $\widehat{A}$ also includes *ground wrap-around* arcs for each airport that connect the last time point with the first time point at the airport and *flight wrap-around* that represent the *red-eye* flights. $\delta^+_{it}$ (or $\delta^-_{it}$) denotes the set of outgoing (or incoming) arcs from node $(i,t) \in \widehat{N}$. Let $t_c$ be the time at which the number of aircrafts in the network is counted. Then, $\widehat{A}_F(t_c)$ denotes the set of active flight arcs (meaning, the corresponding aircraft is either in the air or departs from the origin airport of the segment $(i,j)$) at $t_c$ and $\widehat{A}_H(t_c)$ denotes the set of ground arcs that originate at $t_c$ in the complete time-expanded network. We define $\widehat{D}_{st}$ to denote the set of flight arcs departing in segment $s \in S$ in the interval $[t, t+g_s)$. Finally, we define $\widehat{A}(s)$ to denote the set of all flight arcs in segment $s \in S$.

Binary decision variables $x_{af}$ must be set to 1 if flight arc $a\widehat{A}_f$ uses an aircraft of fleet type $f \in F$ and 0 otherwise. Continuous variables $x_{af}$ count the number of aircraft of fleet type $f \in F$ on ground arc $a \in \widehat{A}_H$. Continuous variables $w^0_{mp} \geq 0$ denote the sum of the market shares of the other airlines' itineraries and the no-fly alternative in each market $m \in M$ for passenger type $p \in P$. Continuous variables $w_{rp\ell} \geq 0$ denote the market shares of passenger type $p \in P$ corresponding to the combination of itinerary $r$ and fare class $\ell$. The cost of operating an aircraft of fleet type $f \in F$ on arc $a \in \widehat{A}$ is $c_{af}$. We assume that there is no cost of waiting for an aircraft at an airport, therefore, $c_{af} = 0 \; \forall a \in \widehat{A}_H, f \in F$.

The MIP model for ATFP is shown in Model 1. The objective function in (1a) minimizes the host airline's total operating loss ($cost - revenue$). Constraints (1b) ensure the flow balance of each fleet type at each time-space node.

$$\text{Minimize} \quad \sum_{a\in\widehat{A}}\sum_{f\in F} c_{af}x_{af} - \sum_{m\in M}\sum_{p\in P}\sum_{r\in\mathcal{R}_m}\sum_{\ell\in L} d_{mp}\rho_{r\ell}w_{rp\ell} \tag{1a}$$

$$\text{s.t.} \quad \sum_{a\in\delta_{it}^{+}} x_{af} - \sum_{a\in\delta_{it}^{-}} x_{af} = 0 \quad \forall\ (i,t)\in\widehat{N}, f\in F \tag{1b}$$

$$\sum_{a\in\widehat{A}_H(t_c)} x_{af} + \sum_{a\in\widehat{A}_F(t_c)} x_{af} \le \kappa_f \quad \forall f\in F \tag{1c}$$

$$\sum_{f\in F}\sum_{a\in\widehat{D}_{ijt}} x_{af} \le 1 \quad \forall\ (i,j)\in S, t\in\widehat{T}_i \tag{1d}$$

$$\sum_{f\in F}\sum_{a\in\widehat{A}(s)} x_a = n_s \quad \forall\ s\in S \tag{1e}$$

$$\alpha_{mp}^{0} w_{rp\ell} \le \alpha_{rp\ell} w_{mp}^{0} \quad \forall\ m\in M, r\in\mathcal{R}_m, p\in P, \ell\in L \tag{1f}$$

$$\sum_{m\in M}\sum_{p\in P}\sum_{\ell\in L}\sum_{r\in\widehat{R}_m(a)} d_{mp}w_{rp\ell} \le \sum_{f\in F} Q_f x_{af} \quad \forall\ s\in S, a\in\widehat{A}(s) \tag{1g}$$

$$\left(\frac{\widehat{\alpha}_{mp}^{0}}{\alpha_{mp}^{0}}\right) w_{mp}^{0} + \sum_{r\in\mathcal{R}_m}\sum_{\ell\in L}\left(\frac{\widehat{\alpha}_{rp\ell}}{\alpha_{rp\ell}} w_{rp\ell}\right) = 1 \quad \forall\ m\in M, p\in P \tag{1h}$$

$$\sum_{f\in F} x_{af} \ge w_{rp\ell} \quad \forall\ s\in S, a\in\widehat{A}(s), m\in M, r\in\widehat{R}_m(a), p\in P, \ell\in L \tag{1i}$$

$$x_{af}\in\{0,1\} \quad \forall\ f\in F, a\in\widehat{A}_F \tag{1j}$$

$$x_{af}\ge 0 \quad \forall\ f\in F, a\in\widehat{A}_H \tag{1k}$$

$$w_{mp}^{0}\ge 0 \quad \forall\ m\in M, p\in P \tag{1l}$$

$$w_{rp\ell}\ge 0 \quad \forall\ m\in M, r\in\mathcal{R}_m, p\in P, \ell\in L \tag{1m}$$

Constraints (1c) ensure that the fleet size is respected for each fleet type. Constraints (1d) ensure that a minimum time duration should separate any two consecutive departures in a market. Constraints (1e) ensure that the flight frequency in each market is satisfied. Constraints (1f) ensure that the market share of each itinerary-fare class combination is proportional to its attractiveness; these constraints embed the linearized version of a discrete-choice model of passengers booking decisions as described in [1]. Constraints (1g) denote the capacity constraint of the aircraft assigned to a flight. Constraints (1h) ensure that the market share across all itineraries sum to one. Constraints (1i) ensure that passengers cannot be assigned to an itinerary if any of its flight legs is not operated. Constraints (1j)-(1m) define the domain and range of variables.

**Solution Methodology** As shown in Figure 1, the DDD algorithm starts with a partially time-expanded network $G = (N, A)$ that satisfies some properties and $N \subseteq \widehat{N}$. A lower-bound MIP model (LBM) is formulated over $G$ and requires careful modeling of the flight travel times and constraints (1d) and (1j) to ensure that its optimal objective value is a lower-bound to the optimal objective value of ATFP. We formulate a novel flight-matching model (that matches inbound flights with outbound flights at each airport in the LBM optimal solution) with actual flight travel time and side constraints to check if the LBM optimal solution can be converted into a feasible solution to ATFP. If yes, then the algorithm fixes the flight schedule and calculates the maximum revenue using a passenger-choice-based linear program; the difference between the revenue and the operating cost (obtained from the fixed flight schedule) gives the upper bound on the objective of ATFP. If not, then the algorithm uses arc-based or path-based refinement ideas to identify new time-space nodes to add to $G$, update the set of arcs $A$, and re-solve the LBM. The algorithm terminates when a solution with a desired quality (MIP optimality gap) is available.

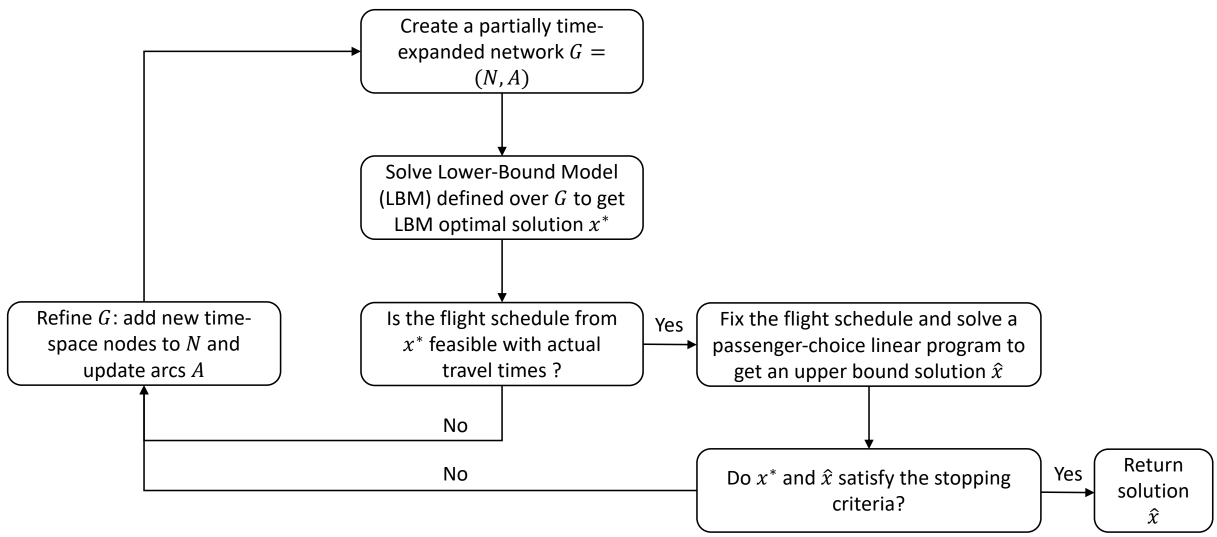



Figure 1: Dynamic Discretization Discovery Algorithm

**Results** The exact algorithm finds the optimal solution and is $8\times$ faster than directly using a commercial solver on small and medium-scale instances obtained from real-life data of *Alaska Airlines*; the commercial solver cannot prove optimality for the small instances even in 15 minutes. The algorithm is computationally efficient because it solves small MIPs (LBM) in each iteration and finds the optimal solution while generating only $10-30\%$ of the complete time-expanded network. The network refinement procedure uses the LBM solution and the instance data to identify and add effective time-space nodes to the partially time-expanded network; the network size remains small ($< 30\%$), but the bounds improve. We are currently creating large-scale instances from publicly available data from the Federal Aviation Administration (FAA) and Bureau of Transportation Statistics.

## References


[1] K. Wei, V. Vaze, and A. Jacquillat, "Airline timetable development and fleet assignment incorporating passenger choice," *Transportation Science*, vol. 54, no. 1, pp. 139–163, 2020

# A routing-based policy framework for assessing freight consolidation effects on small city livability: The case of Bergen, Norway

**Cosku Can Orhan, Stein W. Wallace, Julio Cesar Goez, Mario Guajardo**
Department of Business and Management Science
NHH Norwegian School of Economics, Bergen, Norway
Email: cosku.orhan@nhh.no

## 1 Introduction

The growing urban population and increasing e-commerce sales cause more and more parcel deliveries in urban cities. While some carriers are well consolidated and achieve efficient vehicle utilization, the ongoing uberification trend has increased the number of delivery vehicles driving in urban areas with loads less than vehicle capacities. As a result, the negative impacts of freight distribution on the attractiveness and livability of cities grow progressively (i.e., increased traffic congestion and noise pollution). Two classes of freight policy measures are suggested by [1] to reduce these negative impacts. While solutions driven by carriers form the first class of measures, regulations that restrict carriers' access to urban areas constitute the second. The former requires carrier cooperation and need substantial support from the public sector. The most well-known solution among these is urban consolidation centers (UCCs), which foster freight consolidation within a target area and, consequently, achieve shorter trips and stopping times, fewer stops, and greater vehicle utilization. Despite its benefits, the literature highlights that a substantial number of UCCs fail in practice [2]. One of reasons for the failure lies in these centers' financial performance [3], which are initially run by public money to stimulate cooperation among carriers. Although this initial funding encourages carriers to collaborate, carriers opt out as soon as they are expected to pay for the service (i.e., [4]). Hence, several studies have focused on achieving financially sustainable UCCs (i.e., [5], [6], [7]). The main conclusion drawn from these studies is that financially sustainable UCCs are challenging to obtain, but increasing freight volumes for consolidation, decreasing operational costs, and implementing administrative policies may establish financial viability.

**1.1. Motivation.** This drawn conclusion is based on case studies for large cities, such as Frankfurt [5] and Copenhagen [6]. However, the suggested conditions to achieve financially viable UCCs in these studied settings may not be applicable for small cities due to the limited volume of freight for consolidation. Moreover, such costly infrastructure-led solutions are often considered to be unviable in small cities unless they already exist [8]. The second class of freight measures is, therefore, more applicable to be employed in small cities to achieve consolidation. Although obtaining insights for small cities is essential, there is limited guidance that authorities can rely on apart from pilot studies, whose results are known only after the experiment and may reduce the faith in achieving consolidation. Moreover, extrapolation from the existing literature, which is often concentrated on larger cities, may not provide reasonably robust answers due to, for instance, differences in the structure of the distribution systems (i.e., one-level vs. two-echelon) or variations in road network configurations (i.e., topography, the city being medieval or traffic patterns).

**1.2. Problem statement.** In this work, we focus on guiding authorities in small cities and realize what is important to achieve with consolidation to inform policy-making. In this context, we are not interested in how to achieve consolidation as this decision is political and can change over time as technology moves forward. Instead, we take the role of authorities and analyze the value of consolidation by varying the number and size of carriers in the market. In this regard, we use carriers' logistics models as micro models to answer macro questions for authorities. Consequently, we ask whether consolidation implies a marginal change or a major shift in terms of number and length of stops, and total driven distance. As a case study, we analyze Bergen, a small medieval city in Norway with a complicated street patterns.

## 2 Data

We collect road networks, road objects, and road restrictions from the National Road Database administrated by the Norwegian Public Roads Administration and define the road network topology of Bergen both for driving and walking. We primarily focus on freight distribution within the city center, which we define based on the postcode classifications of the Norwegian Mapping and Cadastre Authority. We identify the locations of all apartments in the city center from Norwegian Mapping and Cadastre Authority and connect these apartments with our road network. Based on the estimated population of each apartment from [9] and real demand volumes and patterns in Bergen (from Post-Nord, a major carrier in Norway), we derive the probability of each apartment receiving a package per day and, consequently, generate instances that represent the daily freight distribution in Bergen. For each instance we generate, we implement Dijkstra's algorithm to find the shortest path between each apartment, both for driving and walking.

## 3 Routing-based policy guidance framework

To answer the aforementioned questions, we provide a routing-based framework consisting of three interconnected stages. The proposed approach is based on adopting clustering and vehicle routing models of carriers as micro models to grasp macro impacts of consolidation for authorities. In this context, we do not rely on carriers' models to determine the optimal delivery process, such as finding the optimal driving and walking paths. Instead, we use them to gather sufficient operational details to derive strategic insights, informing authorities about consolidation for the purpose of their policy-making. The stages of the framework are briefly introduced below.

**Cluster formation:** We cluster the apartments based on a walkable distance, together with the carrying capacity of the courier, such that in each cluster each pair of apartments is within a walkable distance from each other.
**Vehicle routing:** We solve a capacitated vehicle routing problem to identify the total driven distance between clusters.
**Courier delivery:** We solve a capacitated delivery vehicle routing problem and calculate the distance traveled by the courier, which is further used together with off-loading and service times to estimate the length of stops.

## 4 Results

Our findings indicate that setting up a scheme in which small carriers do not directly deliver goods to end customers, but instead relies on a few large carriers to manage the distribution, leads to greater improvements in distance and stop metrics compared to the market transition from a few large carriers to a single, fully consolidated carrier. Moreover, as the number of small carriers increases, their inefficiencies progressively grow, representing what we are facing today with the growing internet trade and ongoing uberification.

## 5 Conclusion

Our contribution in this work is two-fold. First, we provide a routing-based framework for small city authorities to analyze the impact of freight consolidation for their own configurations and environments. Second, we provide a computational study with real data from Bergen, Norway.

## Acknowledgments

This work is supported by The Research Council of Norway (Grant no. 308790).

## References


[1] Dolati Neghabadi, P., Evrard Samuel, K. & Espinouse, M. Systematic literature review on city logistics: overview, classification and analysis. *International Journal Of Production Research.* **57**, 865-887 (2019)

[2] Quak, H. & Tavasszy, L. Customized solutions for sustainable city logistics: the viability of urban freight consolidation centres. *Transitions Towards Sustainable Mobility.* pp. 213-233 (2011)

[3] Browne, M., Sweet, M., Woodburn, A. & Allen, J. Urban Freight Consolidation Centres. Final Report. Transport Studies Group, University of Westminster for the Department for Transport, 2November 2005. (2005)

[4] Marcucci, E. & Danielis, R. The potential demand for a urban freight consolidation centre. *Transportation.* **35**, 269-284 (2008)

[5] Elbert, R. & Friedrich, C. Simulation-based evaluation of urban consolidation centers considering urban access regulations. *2018 Winter Simulation Conference (WSC).* pp. 2827-2838 (2018)

[6] Heeswijk, W., Larsen, R. & Larsen, A. An urban consolidation center in the city of Copenhagen: A simulation study. *International Journal Of Sustainable Transportation.* **13**, 675-691 (2019)

[7] Heeswijk, W., Mes, M., Schutten, J. & Zijm, W. Evaluating urban logistics schemes using agent-based simulation. *Transportation Science.* **54**, 651-675 (2020)

[8] Allen, J. & Huschebeck, M. BESTUFS Policy and Research Recommendations: Urban freight in small and medium sized cities, and Urban waste logistics, BESTUFS project deliverable 1.2. (BESTUFS Project,2006)

[9] Fang, J. & Opedal, E. Analysing the demand for carsharing in Bergen: an empirical approach using car-specific and demographic data. (NHH Norwegian School of Economics,2020)

# Service network design for freight transportation in a river network

**Olivier Péton**
IMT Atlantique, Laboratoire des Sciences du Numérique de Nantes
(LS2N, UMR CNRS 6004) Nantes, France
Email: olivier.peton@imt-atlantique.fr

**Oscar H. Ariztegui Beltrán**
IMT Atlantique, Laboratoire des Sciences du Numérique de Nantes
(LS2N, UMR CNRS 6004) Nantes, France
Departement of Engineering, Universidad de la Sabana, Chía, Colombia

**William J. Guerrero Rueda**
Departement of Engineering, Universidad de la Sabana, Chía, Colombia

**David L. Cortés Murcia,**
Chiper, Bogota, Colombia

## 1 Introduction

The use of rivers as transport corridors offers a cost-effective and environmentally sustainable alternative to road transport, facilitating the seamless movement of cargo from major maritime hubs to hinterland port cities. The objective of this research is to design an efficient transport service, inspired by the principles of the Physical Internet (PI), in a network that includes seaports and a set of river ports. This specific network is a missing link between local/regional road networks and deep sea transportation lines. The river ports include two types of facilities: PI hubs (transshipment ports), where transshipment operations can be performed, and other ports, where the cargo can be loaded or unloaded, but not transshipped.

In this context, we present a mixed-integer linear programming model for a service network design problem that aims to make the following decisions: (i) define the regular routes that operate in the river network, (ii) assign a heterogeneous fleet of vessels to each route and determine their frequency, and (iii) route a set of transportation demands (called commodities), with the possibility of multiple transshipments from their origin to their destination.

## 2 An overview of the mathematical formulation

The considered optimization problem is modeled with a Mixed Integer Linear Programming formulation that will be extensively presented during the conference. The model is based on a network similar to the one shown in Figure 1, including a set of sea ports, a set of PI-hubs located on rivers, and a set of river ports where no transshipment operations can be performed.

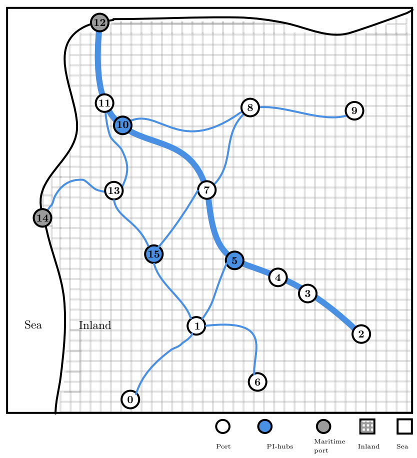



Figure 1: River-sea network

Our route-based model considers a set of candidate routes. A route is a sequence of port calls. It can be of several types: circular, butterfly, pendulum, etc. ([1]).

We consider a set of commodities. Each commodity is characterized by the triplet indicating its origin port, destination port and the quantity to be moved, respectively. Commodities cannot be split. We assume that quantity to be moved does not exceed the smallest ship's capacity.

If there is a route that visits both the origin and the destination of a commodity, then a direct trip is possible (but not mandatory). Otherwise, the commodity must be transshipped one or more times at PI-hubs. Each transshipment requires both unloading and loading the commodities, for a given cost.

Following [3], we consider several classes of ships that can sail along the river network: river-sea ships can call at both river ports and sea ports, while several capacities of river ships are defined depending on the part of the river they navigate.

The cost of a route is determined by the type of vessel that serves it. It is equal to the cost of operating an empty ship on the route, plus the fixed stopover cost at each port on the route. Each unit carried by the route on a graph arc is charged an additional cost.

The mathematical model is based on the following variables: The integer variables $x_{rv}$ represent the number of vessel trips of type $v$ on the route $r$. The binary variables $y^{+}_{kir}$ and $y^{-}_{kir}$ indicate that the commodity $k$ is loaded/unloaded at port $i$ on route $r$. The binary variables $u_{kir}$ are equal to 1 if the commodity $k$ is on route $r$ when it leaves port

$i$. Finally, $L_{ri} \geq 0$ is the total cargo on route $r$ when leaving port $i$.

The goal is to select a subset of routes and vessels that will achieve transportation of all goods at minimum cost. The sets of constraints model flow conservation, demand satisfaction, and vessel capacity constraints. In addition, a number of constraints relate to loading, unloading, and transshipment operations. For example, a commodity cannot be unloaded before it is loaded, it cannot be transshipped on the same route, etc.

## 3 Results on the Rhine/Meuse/Main case study and managerial insights

The mathematical model was tested on several case studies based on real river networks (Rhine, Danube, Mississippi and Magdalena river). The routes and the commodities were manually generated according to the traffic observed on Vesselfinder.com. The mathematical model was solved with Gurobi 10.0.1 (linux64), on a a computer with an Intel(R) processor at 3.00GHz, and using up to 8 threads.

The numerical experiments aimed at assessing a number of additional business constraints and constraint related to PI networks: (i) following the idea of relay network design (see, e.g., [4]), we assume that the set of selected routes forms a partition of the network's edges, thus prohibiting overlapping lines. (ii) we consider an upper bound on the number of transportation lines (results not presented in this abstract). (ii) we introduce capacity constraints at PI-hubs (results not presented in this abstract).

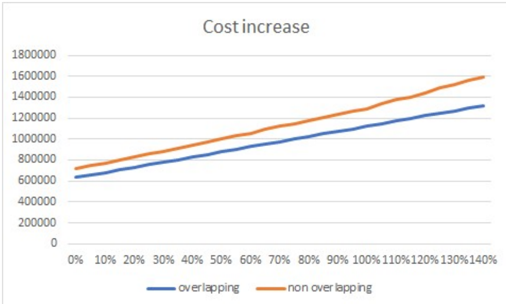
The Rhine/Meuse/Main inland waterways include the ports of Antwerp and Rotterdam, as well as a list of 3 PI-hubs and 11 river ports in the Netherlands, Germany and Belgium. At Odysseus conference, we plan to present extensive computational results for the other case studies. Here follow some results related to the impact of the non-overlapping constraints:

- Figure 2a presents the impact of a demand increase (horizontal axis) on the total cost (vertical axis). We observe that the non-overlapping constraint causes a 15% increase of the total cost on average. Economies of scale are observed: a 100% demand increase results in a 70% increase of the cost.

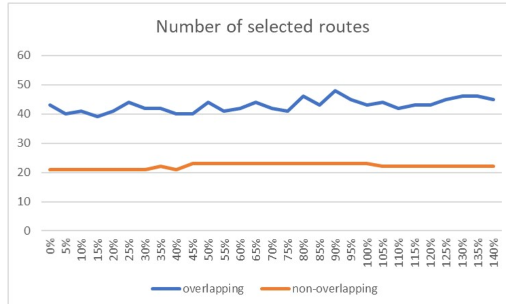
- Figure 2b shows that the number of routes is quite stable when the demand increases, both in the overlapping and non-overlapping cases.

- Figure 2c presents the CPU time needed to optimally solve the model. It shows that the set-partitioning-like structure of the overlapping constraint make the problem more difficult to solve.

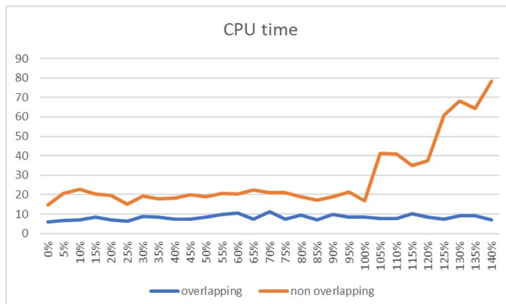
The presentation at Odysseus will also explore the sensitivity to various features of




(a) Impact of the demand increase on the total cost




(b) Impact of the demand increase on the number of routes




(c) Impact of the demand increase on the CPU time

Figure 2: Results of the Rhine / Meuse / Main case study

the problems: transportation cost, port costs and vessels capacity. It will also include the results of current research on a decomposition method to solve large instances.

# References


[1] M. Christiansen, E. Hellsten, D. Pisinger, D. Sacramento, C. Vilhelmsen, Liner shipping network design, European Journal of Operational Research, 286(1), 1–20, 2020.

[2] O.H. Ariztegui Beltrán, D. L. Cortés Murcia, W.J. Guerrero Rueda, O. Péton, M. Mohammadi, Rivers, trees, hubs and stopovers (Hub location and network design for waterborne transportation in the Physical Internet context), INFORMS TSL 2023 conference, Chicago, July 2023.

[3] J. Zheng, D. Yang, Hub-and-spoke network design for container shipping along the Yangtze River, Journal of Transport Geography, 55, 2016, 51-57.

[4] E.A. Cabral, E. Erkut, G. Laporte, R.A. Patterson, The network design problem with relays, European Journal of Operational Research, 180 (2), 2007, 834-844.

# A branch-and-price algorithm for the Min-Max Multi-trip Location Arc Routing Problem

**Isaac Plana**
Departament de Matemàtiques per a l'Economia i l'Empresa,
Universitat de València, Spain
Email: isaac.plana@uv.es

**Teresa Corberán**
Departament d'Estadística i Investigació Operativa,
Universitat de València, Spain

**José María Sanchis**
Departamento de Matemática Aplicada
Universidad Politécnica de Valencia, Spain

## 1 Introduction

The use of drones for delivery or inspection tasks is becoming more common, since they offer several advantages compared to traditional means of transportation, being able to reach places that are difficult to access or dangerous for people or conventional vehicles. Given the finite flight autonomy of drones, certain applications require the integration of ground support vehicles. These vehicles serve as launch and retrieval points for drones and facilitate tasks such as recharging or battery exchange. While the coordination of ground vehicles with drones has been extensively explored in the literature for node routing problems, as evidenced by [2], few works have addressed this problem from the perspective of arc routing problems (see, e.g., [1]).

## 2 Problem definition

Let $G = (V, E)$ be an undirected graph with a subset of required edges $E_R \subset E$ and a subset $\mathcal{D} \subset V$ of vertices representing launch points. A central depot, denoted as $0 \in \mathcal{D}$ serves as the origin and destination for ground vehicles (trucks), and may also function as a launch point. Let $E_{NR} \subset E$ be the set of non-required edges representing the possible deadheading movements of drones. Since drones can move freely between any pair of points, we consider that the graph $G = (V, E_{NR})$ is complete.

Each required edge has an associated service time, while each non-required edge has a deadheading time proportional to the Euclidean distance between its endpoints. For each launch point $d \in \mathcal{D}$, $c_{0d} > 0$ is the time needed for a truck to travel from depot 0 to point $d$ (obviously, $c_{00} = 0$). There is a fleet $\mathcal{K}$ of $P$ trucks, each one carrying a drone, that have to be assigned to a launch point, considering that two trucks cannot share the same launch point. Drones have a limited autonomy that allows them to fly for a maximum time $L$ before returning to recharge at their launch point. There is no limit on the number of flights (routes that begin and end at the launch point) that a drone can execute. The problem consists of selecting a launch point for each truck–drone pair and determining a set of flights for each drone. These flights must start and conclude at the corresponding launch point, with flight time not exceeding $L$, ensuring that all the required edges are collectively traversed. If we denote $C_k$ as the sum of the travel times for truck $k \in \mathcal{K}$ from the central depot to its designated launch point and back, plus the total flight time of its drone, the objective is to minimize the maximum of all these times, that is, $\max_{k \in \mathcal{K}}\{C_k\}$.

Note that the launch points can be considered as facilities that can be opened or not, and the costs $c_{0d}$ can be understood as their opening costs. Therefore, the problem can be considered as a *Location Arc Routing Problem* and we call it the *Min–Max Multi–trip Location Arc Routing Problem, MM–MT–LARP*. We will refer as *depots* to the potential launch points in $\mathcal{D}$.

# 3 The branch–and–price algorithm

Given that we are considering several depots and multiple flights for each drone, an MM–MT–LARP solution is expected to contain a considerable number of different routes. Therefore, we believe that a branch–and–price algorithm would be a fitting approach for effectively solving this problem.

## 3.1 Master problem

For each depot $d \in \mathcal{D}$, let us call $\mathcal{F}^d$ to the set of all feasible flights of a drone from $d$. Given a flight $t \in \mathcal{F}^d$, $c_t$ denotes the total time of $t$, and, given a required edge $e \in E_R$, $a_{te}$ is a binary parameter that takes the value 1 if flight $t$ traverses edge $e$ and 0 otherwise. To formulate the master problem (MP), we define the following variables:

- For each depot $d \in \mathcal{D}$, variable $\alpha_d$ takes the value 1 if depot $d$ is opened (that is, there is a truck that goes to this launch point to release a drone) and 0 otherwise.
- For each launch point $d \in \mathcal{D}$ and each flight $t \in \mathcal{F}^d$, variable $\lambda_{td}$ takes the value 1 if flight $t$ is used in the solution and 0 otherwise.
- Variable $z$ corresponds to the largest operation time among all the trucks.

Then, the MP corresponding to the MM-MT-LARP can be formulated as:

$$\text{Minimize} \qquad z$$

s.t.:

$$\sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{F}^d} a_{te} \lambda_{td} = 1 \qquad \forall e \in E_R, \tag{1}$$

$$\lambda_{td} \leq \alpha_d \qquad \forall d \in \mathcal{D}, \quad \forall t \in \mathcal{F}^d, \tag{2}$$

$$\sum_{d \in \mathcal{D}} \alpha_d \leq P, \tag{3}$$

$$\sum_{t \in \mathcal{F}^d} c_t \lambda_{td} + 2c_{0d}\alpha_d \leq z \qquad \forall d \in \mathcal{D}, \tag{4}$$

$$\lambda_{td} \in \{0, 1\} \qquad \forall d \in \mathcal{D}, \quad \forall t \in \mathcal{F}^d, \tag{5}$$

$$\alpha_d \in \{0, 1\} \qquad \forall d \in \mathcal{D}, \tag{6}$$

$$z \in \mathbb{R}. \tag{7}$$

Equations (1) imply that each required edge $e \in E_R$ is traversed by exactly one flight. Inequalities (2) state that to use a flight, its depot must be open, while inequality (3) assures that at most $P$ depots can be opened. Inequalities (4), together with the objective function, make sure that variable $z$ takes a value equal to the longest time of all the trucks.

It is easy to see that if $Q$ is an upper bound for the number of flights that will be used for any depot, inequalities (2) can be replaced by

$$\sum_{t \in \mathcal{F}^d} \lambda_{td} \leq Q\alpha_d \qquad \forall d \in \mathcal{D}, \tag{8}$$

which are more convenient for the formulation and solution of the pricing problem. A valid value for $Q$ can be the total number of required edges of the graph.

In the branch–and–price algorithm, we solve a linear relaxation of the MP (LMP). The initial LMP consists of inequalities (1), (3), (4), and (8), as well as bounds $\lambda_{td} \geq 0$ and $0 \leq \alpha_d \leq 1$ (note that the upper bounds for $\lambda_{td}$ are implied by (1)).

### 3.2 Pricing problem

At each node of the search tree, we solve a reduced LMP (RLMP) with only a subset of feasible flights for each depot. After solving the RLMP, we must look for new flights with negative reduced cost and add them to the RLMP, which is then solved again. To obtain these flights with negative reduced costs, we must solve a Pricing Problem (PP) for each depot $d \in \mathcal{D}$. It can be seen that, given a depot $d$, the PP can be stated as a Profitable Arc Routing Problem [3] with a single vehicle in which the profit/cost $b_e^d$ of each edge is defined as:

$$b_e^d = \begin{cases} c_e \rho_d^* + \mu_e^*, & \text{if } e \in E_R \\ c_e \rho_d^*, & \text{if } e \in E_{NR} \end{cases},$$

where $\mu_e^*$ and $\rho_d^*$ are the values of the dual variables associated with inequalities (1) and (4), respectively, for the optimal solution of the RLMP.

This PP is solved using heuristic algorithms and, if no feasible solution with negative reduced cost is found, a branch–and–cut algorithm is used. If the optimal solution of the PP has a non–negative reduced cost, the solution of the RLMP is optimal for the LMP.

### 3.3 Branching rules

If the optimal solution of the LMP is not integer, we have to branch to continue exploring the search tree. We have used the following branching rules:

- If there is an $\alpha_d$ variable with fractional value, we add $\alpha_d = 0$ and $\alpha_d = 1$.
- If there is a depot $d$ and a required edge $e$ such that $\sum_{t\in\mathcal{F}^d} a_{te}\lambda_{td}$ is fractional, we add $\sum_{t\in\mathcal{F}^d} a_{te}\lambda_{td} = 0$ and $\sum_{t\in\mathcal{F}^d} a_{te}\lambda_{td} = 1$.
- If there is a depot $d$ and a non–required edge $e$ such that $s_{de} = \sum_{t\in\mathcal{F}^d} a_{te}\lambda_{td}$ is fractional, we add $\sum_{t\in\mathcal{F}^d} a_{te}\lambda_{td} \leq \lfloor s_{de} \rfloor$ and $\sum_{t\in\mathcal{F}^d} a_{te}\lambda_{td} \geq \lceil s_{de} \rceil$.
- Ryan and Foster's branching rule [4].

## 4 Computational experiments

We are conducting computational experiments on newly generated instances with sizes ranging between 9 and 88 required edges, 2 and 6 potential depots, and 2 and 5 trucks, and comparing the results with those obtained by a branch–and–cut algorithm using an arc–based formulation. Preliminary results seem to show that the branch–and–price approach is a promising one when the number of depots and trucks is large.

## References

[1] Amorosi, L., Puerto, J., and Valverde, C. (2023). A multiple-drone arc routing and mothership coordination problem. *Computers & Operations Research*, 159, 106322.

[2] Chung, S. H., Sah, B., and Lee, J. (2020). Optimization for drone and drone–truck combined operations: A review of the state of the art and future directions. *Computers & Operations Research*, 123, 105004.

[3] Feillet, D., Dejax, P., and Gendreau, M. (2005). The Profitable Arc Tour Problem: Solution with a Branch–and–Price Algorithm. *Transportation Science* 39, 539-552.

[4] Ryan D.M. and Foster B.A. (1981), An integer programming approach to scheduling. *Computer Scheduling of Public Transport*, 1, 269-280.

# Efficient Move Evaluation and Neighborhood Exploration for Integrated Order Picking Planning Problems in Picker-to-Parts Warehouses

**Thibault Prunet[1,2], Nabil Absi[1], Valeria Borodin[3], Diego Cattaruzza[4]**

[1] Mines Saint-Etienne, Univ. Clermont Auvergne, INP Clermont Auvergne, CNRS UMR 6158 LIMOS, F-13120 Gardanne, France

[2] CERMICS, Ecole des Ponts, Marne-la-Vallée, France

[3] IMT Atlantique, LS2N, UMR CNRS 2004, Nantes, France

[4] CRIStAL Centre de Recherche en Informatique Signal et Automatique de Lille, University Lille, CNRS, Centrale Lille, Inria, UMR 9189, F-59000 Lille, France

Email: {thibault.prunet,absi}@emse.fr, valeria.borodin@imt-atlantique.fr, diego.cattaruzza@centralelille.fr

## 1 Introduction and related literature

Within warehousing, Order Picking (OP) is largely considered as the most resource-intensive activity. In manual picker-to-parts warehouses, where a walking human operator performs the picking, the OP activity alone represents 50-75% of the total operating costs [1]. While efficient solution methods have been developed for single OP planning problems, recent researches have highlighted the potential gains from integrating several level of decisions [5]. In this talk, we study the *Picker Routing Problem* (PRP), which consists in finding a minimum length tour within a warehouse, and the integrated problems that jointly optimize other decisions with the PRP. The methodological findings we introduce are generic and apply to all integrated problems where the objective is to minimize a function that is cumulative on aisle traversals (e.g., traveled distance, picking time). Among them, we test our methodology on the *Joint Order Batching and Picker Routing Problem* (JOBPRP) that aims at grouping customer orders in consolidated batches retrieved by a single route.

The OP literature uses two different modeling paradigms to design efficient algorithms. The most straightforward option models a picking route as a tour between visited locations, as it is classically done in the routing literature. In this case, the elementary modeling unit is the *visit* to a location, and its position in a route. This approach benefits from the prolific literature on routing problems. An alternative option is to model a picking route as a succession of aisle traversals, thus exploiting *problem-specific knowledge* on the warehouse layout structure. The seminal work of Ratliff and Rosenthal (1983) [4] is the first to use the *aisle* as the elementary modeling unit, disregarding the visit order in a route, and introducing a polynomial-time *Dynamic Programming* (DP) algorithm for the PRP in a single-block warehouse. It is important to emphasize that the two paradigms lead to very different models and algorithms. Nowadays, most mathematical-programming based methods with state-of-the-art results use the aisle modeling, for instance Wahlen ad Gschwind (2023) [6] for the JOBPRP. However, in terms of metaheuristics, most studies do not fully exploit problem-specific knowledge.

To the best of the authors knowledge, no prior study focuses on designing efficient move evaluation using problem-specific knowledge for integrated OP problems. Since neighborhoods can get quite complex for OP problems (i.e., multiple positions inserted in a route, or several routes modified), the efficient computation of insertion costs is far from being a computationally easy task. In this talk, we aim at addressing this gap by introducing three main methodological contributions: *1.* A novel constructive heuristic for the PRP coined the *Aisle First Cross Second* (AFCS) heuristic. The AFCS provides upper and lower bounds for route distance within a very reasonable time complexity, serving as a surrogate objective function for move evaluation when solving integrated problem. *2.* A neighborhood exploration scheme that relies on several move underestimation and overestimation routines (including the AFCS bounds) to efficiently prune the neighborhood search. *3.* A generic Large Neighborhood Search (LNS) algorithm, tested on benchmark instances from the JOBPRP literature, exhibiting promising results.

## 2 Methodology

### 2.1 Aisle First Cross Second heuristic

In this section, we provide a brief description of AFCS heuristic we use as a surrogate objective function. The AFCS is a 2-step heuristic that builds first the aisle traversals, then constructs the cross aisle traversals. Both steps are optimally solved to incur the minimum cost increment.

First, the AFCS builds the aisle traversals. This is achieved by first computing the different traversal costs (i.e., top, bottom, gap and 1-pass) for each aisle. Then, each aisle is traversed by its least-cost traversal. Some aisle traversals are adjusted to ensure the future feasibility of the route, in particular there should be an even number of full traversal in each block, otherwise a valid tour cannot be built. Corrections are made to minimize the additional distance. After this stage, the aisle traversals are fixed and provide a lower bound on the total distance.

Second, the AFCS builds the cross traversals. The procedure first adds single cross traversals in each block. Following this step, the solution consists of partial subtours that need to be connected to form a valid solution. This is achieved by the addition of double cross traversals to the solution, the resulting problem being modeled as a minimum spanning tree.

A lower bound for the PRP is derived from the first step of the AFCS heuristic, and the second step returns a valid solution, that is an upper bound. From these bounds, we derive an approximation ratio for the heuristic dependent solely on the geometry of the layout. Furthermore, we prove that the AFCS runs in linear time complexity.

### 2.2 Surrogate move evaluation and neighborhood exploration

Since exact move evaluation can be costly, we introduce a move underestimation and two overestimation procedures. As stated in the previous section, the AFCS provides lower and upper bound for the PRP, which can be used to evaluate moves. Furthermore we introduce an additional move overestimation procedure based on the dynamic programming algorithm of Ratliff and Rosenthal (1983) [4]. From these results, we propose a generic neighborhood exploration scheme that uses the bound information to efficiently prune dominated parts the search space.

### 2.3 Large Neighborhood Search for integrated problems

An LNS algorithm is developed to evaluate the performances of the proposed neighborhood schemes on integrated problems. The algorithm uses several removal operators (i.e. random, related, and aisle removal) and several insertion operators (i.e. best element best insertion, random element best insertion, largest element best insertion and k-regret). An important feature that distinguishes this LNS from those applied in the routing literature is the consideration of the *size* of a move. Indeed, inserting an order or an SKU in the solution may modify several routes, or add several visits in one route, so that ignoring move size leads to all the challenging elements being left in the pool for the end of the algorithm, when their insertion would be more challenging. The LNS algorithm is then enhanced by local search to improve its intensification capabilities.

## 3 Preliminary results and conclusion

In this section we present preliminary experiments on benchmark sets from the literature on the JOBPRP. We compare our results with the two column generation-based heuristics from Wahlen and Gscwhind (2023) [6]: SC-2 that solves the root node and input the routes in a set covering model, and BPC-DF-2, a diving heuristic. We compare our algorithm against the SC-2 and BPC-DF-2 on the JOBPRP benchmark sets from Henn and Wäscher (2012) [2], Muter and Öncan (2015) [3] that has been extended by Wahlen and Gschwind

(2023) [6], Žulj et al. (2018) [7] and Wahlen and Gswind (2023) [6]. As far as we know, the SC-2 and BPC-DF-2 propose the state-of-the-art results on all the tested JOBPRP benchmark sets, outperforming the existing literature. Table 1 reports a summary of these results, where the gap between a solution $z$ and the best known solution $z^{BKS}$ is computed as $100(z - z^{BKS})/z^{BKS}$, and running times are expressed in seconds.

Table 1: Comparison with Wahlen and Gschwind (2023) [6] on different JOBPRP benchmark sets

| | | W&G SC-2 | | W&G BPC-DF-2 | | Our method | | |
|---|---|---|---|---|---|---|---|---|
| Instance set | # Inst | gap | time | gap | time | gap | time | iterations |
| Henn and Wäscher | 5760 | 0.2 | 19.6 | **0.1** | 34.4 | 0.3 | 91.9 | 37329 |
| Muter and Öncan | 450 | 4.3 | 64.8 | 3.8 | 78.1 | **0.5** | 108.6 | 36749 |
| Žulj, Kramer and Schneider | 60 | 0.6 | 51.7 | 0.9 | 118.9 | **0.5** | 294.1 | 27784 |
| Wahlen and Gschwind | 500 | 11.9 | 115.7 | 12.4 | 118.5 | **0.8** | 293.8 | 27973 |

Preliminary experiments of Table 1 show very promising results. The main objective of this study has been met, as the algorithm performs a large number of LNS iterations within a limited computation time. Although our method does not quite reach the running times reported in [6], it proves to scale better on large instances with longer routes.

## References


[1] E. Frazelle, "World-class warehousing and material handling", McGraw-Hill Education, 2016.

[2] S. Henn and G. Wäscher, "Tabu search heuristics for the order batching problem in manual order picking systems", *European Journal of Operational Research* 222(3), 484-494, 2012.

[3] I. Muter and T. Öncan,,"An exact solution approach for the order batching problem", *IEE Transactions* 47(7), 728-738, 2015.

[4] H.D. Ratliff and A.S. Rosenthal, "Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem", *Operations Research* 31(3), 507-521, 1983.

[5] T. van Gils, K. Ramaekers, A. Caris and R.B.M. de Koster, "Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review", *European Journal of Operational Research* 267(1), 1-15, 2018.

[6] J. Wahlen and T. Gschwind, "Branch-Price-and-Cut-Based Solution of Order Batching Problems", *Transportation Science* 57(3), 756-777, 2023.

[7] I. Žulj, S. Kramer and M. Schneider, "A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem", *European Journal of Operational Research* 264(2), 653-664, 2018.

# Robust Policies for a Multi-Stage Assignment Problem under Demand Uncertainty

**Alice Raffaele, Demetrio Laganà**
Department of Mechanical, Energy, and Management Engineering,
University of Calabria, Rende, Italy
Email: alice.raffaele@unical.it

**Roberto Roberti**
Department of Information Engineering,
University of Padova, Padova, Italy

## 1 Problem Definition and Motivation

We consider a stochastic dynamic setting over a planning period $P := [0, \Pi]$ of $\Pi + 1$ days. We denote by $N$ the set of known customers who place exactly one order each during $P$. Orders can be either served or outsourced by paying a penalty $\varphi \in \mathbb{N}$. An order is made by a customer $i \in N$ on a known day $d_i \in P$, and it is associated with an uncertain demand $q_i \in Q_i := [\underline{q}_i, \overline{q}_i]$, with $q_i \in \mathbb{N}$. Each customer is associated with an *inner day window* $P_i := [d_i + e_i, d_i + \ell_i]$ which, if the customer allows for some flexibility in the delivery, is extended to the *flexible day window* $[d_i + e_i - \delta_i^B, d_i + \ell_i + \delta_i^A]$. More precisely, if $\delta_i^B$ or $\delta_i^A$ are nonzero, the order can be anticipated or delayed by paying a penalty for each day of anticipation or delay. Every day in the planning period, a fleet $K$ of vehicles, each with a resource capacity of $Q \in \mathbb{N}$, is available. We denote by $\mathcal{V} := \mathcal{V}_0 \cup \tilde{\mathcal{V}} \subseteq N \times P$ the set of all orders placed by customers during the planning period, where $\mathcal{V}_0$ is the set of pending known orders and $\tilde{\mathcal{V}}$ the one of future orders with demand uncertainty. Since each customer places exactly one order during the planning period, then we have $|\mathcal{V}| = |N|$.

At the end of the day $p \in P$, we decide which pending orders to serve on the day $p + 1$ by allowing anticipations and delays considering day windows and future orders. The goal is to minimize total costs due to fleet sizing and customer inconvenience. In particular, the former costs are due to the maximum number of vehicles exploited each day, and the latter includes the total penalty associated with anticipations, delays, and outsourcing.

This problem finds application in many contexts, such as reverse logistics, corrective maintenance, technician routing and scheduling, and freight transportation (see, e.g., Mishra *et al.* [1], Nowak and Szufel [2], or Ghezelsoflu *et al.* [3]). Our study is motivated by the setting of a company offering an on-demand waste collection service. Recently, such services have been spreading more and more to overcome the inefficiency of traditional systems, going towards smart management and thus contributing to the United Nations Sustainable Development Goals.

## 2 Model Formulations

We formulate the problem with multi-stage *Robust Optimization* (RO) and as a *Markov Decision Process* (MDP).

### 2.1 Multi-stage Robust Optimization

We represent the uncertainty model with a cardinality-constrained uncertainty set where each future customer places exactly one order during the planning period with a random demand value, but at most $\Gamma$ orders exceed their minimum value.

Given a realization of the uncertainty set (i.e., the set of all orders placed during the planning period is entirely known), we can model a deterministic multi-period assignment problem with Integer Linear Programming. In a compact formulation, we use the following variables: $a_{ikp} \in \{0, 1\}, \forall i \in N, k \in K, p \in P_i$, taking value 1 if the customer $i$ is served by the vehicle $k$ on the day $p$; $u_{kp} \in \{0, 1\}, \forall k \in K, p \in P$, taking value 1 if the vehicle $k$ is used on the day $p$; $\eta \in \mathbb{N}$, indicating the maximum number of vehicles to exploit each day (at most $|K|$). Instead, in a Set Packing formulation, we introduce $\mathcal{A}_p$ as the set of all feasible assignments in terms of vehicle capacity on the day $p \in P$ and let $\alpha_a \in \{0, 1\}$ be equal to 1 if and only if the assignment $a \in \mathcal{A}_p$ is selected in the optimal solution. We minimize the total costs due to fleet sizing and customer inconvenience. Every day, we set constraints to ensure that the loaded demand of each used vehicle $k$ does not exceed $Q_k$.

When only pending orders are known, this family of constraints has to be reformulated considering the uncertainty model. Similarly to Munari *et al.* [4], we express the robustness of each order assignment to a vehicle $k \in K$ on a day $p \in P$. We use recursive equations to decide the highest vehicle load after assigning some customers to the vehicle $k$ on the day $p$ when up to $\gamma \leq \Gamma$ orders have attained their worst-case value. This provides us with an upper bound on the multi-stage optimal solution value. Lower bounds are obtained by assuming considering pending orders only or minimum demand for all future orders.

### 2.2 Markov Decision Process

An MDP is a stochastic framework consisting of a series of *epochs* over a finite and discrete planning period, used to model sequential decision-making under uncertainty. In our case, at each epoch, the *pre-decision state* contains all available information to decide assignments for the next epoch: the current day of the planning period, the set of pending orders, and the set of customers who have not placed an order yet. Observing the pre-decision state, we select an *action* by assigning a subset of pending orders to each vehicle available on the next epoch and deciding whether to outsource any orders. We denote by $\Pi$ the set of all deterministic *policies*, where a policy $\pi \in \Pi$ is a sequence of *decision rules*. Each decision rule is a function that maps a pre-decision state into an action, leading to a deterministic *post-decision state*. When random information occurs in the form of new orders, we make a *stochastic transition* to the next epoch and pre-decision state. The goal is to find an *optimal policy* $\pi^*$ such that the expected costs to serve or outsource orders are minimum, given an initial state composed of the pending orders received on day 0.

## 3 Solution Methods and Contributions

Identifying such an optimal Markovian deterministic policy is challenging due to the curse of dimensionality. Thus, we rely on *Approximate Dynamic Programming* (ADP). Typically, the following four methods are used to devise alternate policies: *Policy Function Approximation*, adopting functions that return an action given a state, without considering any forecast; *Cost* and *Value Function Approximations*, estimating the deterministic and stochastic cost of being in a future state as the result of a decision now, respectively; and *Direct Lookahead Approximation* (DLA), optimizing over more than one time period into the future to make better decisions now. In our approach, we develop DLA methods integrating the deterministic or robust approximation models described in § 2.1. First, we solve these models with a compact formulation. Then, we employ the Set Packing formulations in a Branch-and-Price framework. Also, we design myopic policies serving customers in the order they appear, disregarding any future consequences of our decisions.

Few works in the literature combine stochastic dynamic programming and RO. Our setting is similar to Subramanyan *et al.* [5], who consider a multi-day vehicle routing problem with known customer demands and unknown days of disclosure. Instead, we deal with known days and uncertain demands, and, unlike [5], we focus on fleet sizing, include outsourcing and day windows, and do not limit to the costs of pending orders only.

## 4 Computational Results

We benchmark our methods by comparing DLA and myopic policies in a rolling-horizon fashion. We use three instance sets from Albareda-Sambola *et al.* [6] and a fourth set with actual data from the application motivating our study. Figure 1 shows the preliminary results on the smallest set with $N = 25$ customers, $P = 3$ days, $K = 2$ vehicles per day, and solving the compact formulation in the DLA policies. We vary the vehicle capacity from 25% to 100% of the original value and $\Gamma$ from 5% to 20% of $N$. In each configuration, we compare three policies with the corresponding value of the ex-post solution to compute

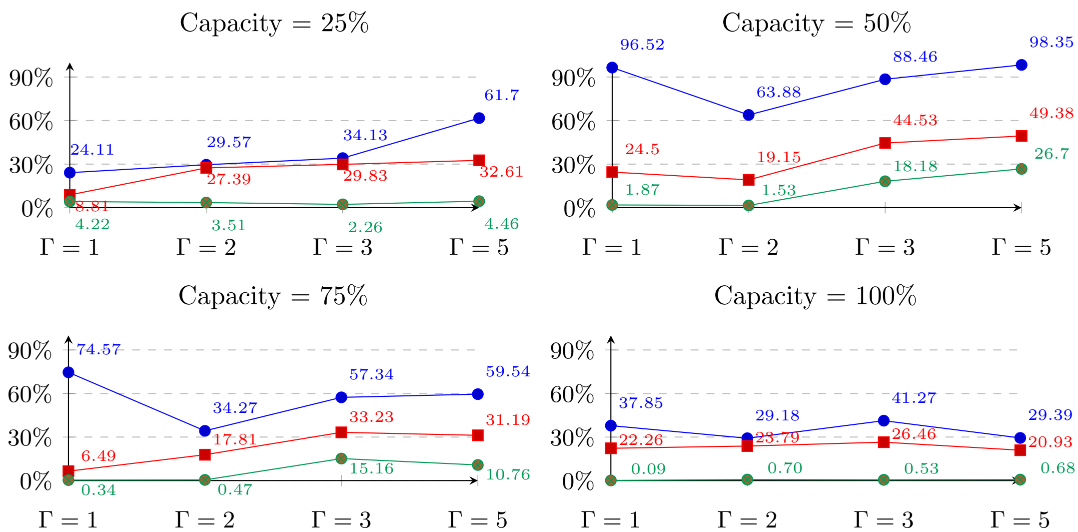



Figure 1: Preliminary results on the gap between the values of the ex-post solution and three policies: myopic, robust on the worst case, and deterministic with minimum demand.

average gaps. By tightening the vehicle capacity and increasing Γ, DLA policies seem to perform nicely and better than the myopic one. We will assess it by running further tests.

## References

[1] S. Mishra and S. P. Singh, "Designing dynamic reverse logistics network for post-sale service," *Annals of Operations Research*, pp. 1–30, 2022.

[2] M. Nowak and P. Szufel, "Technician routing and scheduling for the sharing economy," *European Journal of Operational Research*, 2023.

[3] A. Ghezelsoflu, M. Di Francesco, A. Frangioni, and P. Zuddas, "A multiperiod drayage problem with customer-dependent service periods," *Computers & Operations Research*, vol. 129, 2021.

[4] P. Munari, A. Moreno, J. De La Vega, D. Alem, J. Gondzio, and R. Morabito, "The robust vehicle routing problem with time windows: compact formulation and branch-price-and-cut method," *Transportation Science*, vol. 53, no. 4, pp. 1043–1066, 2019.

[5] A. Subramanyam, F. Mufalli, J. M. Laínez-Aguirre, J. M. Pinto, and C. E. Gounaris, "Robust multiperiod vehicle routing under customer order uncertainty," *Operations Research*, vol. 69, no. 1, pp. 30–60, 2021.

[6] M. Albareda-Sambola, E. Fernández, and G. Laporte, "The dynamic multiperiod vehicle routing problem with probabilistic information," *Computers & Operations Research*, vol. 48, pp. 31–39, 2014.

# A Discrete-Continuous Approximation Model for Optimal Facility Location in Disaster Response Logistics

**Diana G. Ramirez-Rios, Ph.D.**
Department of Industrial and Systems Engineering,
University at Buffalo, Buffalo, NY, USA
Email: dgramire@buffalo.edu

**José Holguín-Veras, Ph.D., P.E., F.ASCE**
Department of Civil and Environmental Engineering,
Rensselaer Polytechnic Institute, Troy, NY, USA

## 1 Introduction

Disaster preparedness efforts regarding the location and prepositioning of relief supplies are becoming much more crucial than ever before. Local distribution is challenging after major disasters and catastrophic events as resources are mostly or entirely destroyed in the affected region, posing unique challenges to distributing relief supplies. As a result, local supplies may not be available for the population in need. The location of these supplies is a crucial aspect of the decision, as it will determine how fast the agency responds to the needs of the affected population.

This research considers the decisions made by disaster relief organizations and local government agencies when it comes to preparing the distribution of critical relief. These decisions typically need to be made in a short period prior to the event, where locations are identified as point of distribution (PODs), so that relief can be transported to these PODs from a distribution center. This paper proposes a discrete-continuous facility location model that computes the optimal location of PODs, number of PODs, and shapes of the districts served by the PODs, their delivery frequencies, and shipment sizes such that the total social costs are minimized. The social costs considers both the private logistics costs and the economic externalities associated with the population's human suffering in the form of deprivation costs. These deprivation costs are empirically estimated and account for the time that a person is without relief supplies [1]. The proposed model analyzed both linear and non-linear (exponential) deprivation cost functions.

The problem tackled is a continuous planar facility location problem, where the POD's location can be located anywhere in the grid and demand is continuously distributed in the region. There are several advantages to this model, but also added complexity. One important advantage of using this model is that it allows for flexibility in settings where there are many and nearly uncountable potential sites to choose from as PODs. Another advantage is that it helps design districts where demand is uncertain or hard to predict, without having to go to a stochastic model formulation. The main complexity of this model is a not well defined objective function, which involves solving a double integral over the area of study. An alternative solution to these complex model formulations is to approximate these functions to simpler, closed-form analytical formulations or continuous approximation (CA) models. Most CA model formulations assume demand is homogeneously and uniformly distributed, which provides a practical way of analyzing NP-hard logistics models, e.g., vehicle routing problems, facility location, location-allocation. For instance, Losch [2] assumed uniformly distributed demand over an infinite region. Earlier location studies used CA to analyze optimal location of plants and warehouses [3], [4], [5]. These models typically determine the optimal number of facilities to locate and the facility's optimal service area that minimizes both the setup and transportation costs.

The discrete-continuous facility location model proposed in this paper includes two main components. The first is a mixed-integer programming (MIP) formulation to select the optimal PODs' location, the shapes and sizes of the districts they serve, and the distribution strategy for relief supplies to the PODs, subject to time and capacity constraints. The second is a set of continuous approximation (CA) models that compute the total deprivation costs for different shapes integrated into the MIP formulation. Because of space constraints we will explain the objective model formulation and how a CA model for rectangular districts is incorporated using linear deprivation costs.

## 2 Model Formulation

Given a continuous convex closed two-dimensional space $C \subseteq R^2$ as a potential disaster site, with its population continuously distributed with density function $\rho(x, y), (x, y) \in C$. After a disaster occurs, the population in need will travel to the nearest point of distribution (POD) to receive their relief. Suppose that all of the population in $C$ has been affected by a disaster and in need of critical relief. Let region $C$ of length $L_x$ and width $L_y$ be partitioned into $C_{ij}$ districts, where $i = 1, ..., n_x$ and $j = 1, ..., n_y$ are the cuts made along the x and y axis, respectively. PODs are located at any location $(x^p_{ij}, y^p_{ij}) \in C$ and each one serves one district $C_{ij}$. Assume one fixed location of a distribution center where relief supplies arrive and are further delivered to the PODs. Periodically, relief supplies are delivered from the distribution center to the PODs for a period $T$.

The optimal is that which minimizes the total social costs $\Phi$ that equals the logistics costs $\Omega$ plus the deprivation costs $\Gamma$. Equation 1 shows the logistics costs that includes the fixed costs of POD as a function of location, the inventory holding costs, and the distribution costs. Equation 2 shows the deprivation costs as the integral over each district's region of the deprivation costs. These costs consider all the population's deprivation costs, expressed as the product of population density $\rho(x, y)$ and the deprivation cost function $\gamma_g(\theta_g, \delta(x, y, x_{ij}^p, y_{ij}^p))$. The total social costs, explained in Equation 3 are expressed as the summation of Equations 1 and 2.

$$\Omega = \sum_i \sum_j c^F(x_{ij}^p, y_{ij}^p) + c^H \frac{q_{ij}}{2} f_{ij} + c^T D \left[ \tau^{SA} q_{ij} + 2\tau^V (x_{ij}^p, y_{ij}^p) \right] f_{ij} \tag{1}$$

$$\Gamma = \sum_i \sum_j f_{ij} \int_{lx_i} \int_{ly_j} \rho(x, y) \gamma_g \left( \theta_g, \delta(x, y, x_{ij}^p, y_{ij}^p) \right) dydx \tag{2}$$

$$\begin{aligned} \Phi = \sum_i \sum_j c^F(x_{ij}^p, y_{ij}^p) + c^H \ \frac{q_{ij}}{2} f_{ij} + c^T D \left[ \tau^{SA} \ q_{ij} + 2\tau^V \ (x_{ij}^p, y_{ij}^p) \right] f_{ij} \\ + \sum_i \sum_j f_{ij} \int_{lx_i} \int_{ly_j} \rho(x, y) \gamma_g \left( \theta_g, \delta(x, y, x_{ij}^p, y_{ij}^p) \right) dydx \end{aligned} \tag{3}$$

Equation 1 shows the logistics costs that includes three components. Let $c^F(x_{ij}^p, y_{ij}^p)$ be the installation and setup costs of the POD as a function of its location $(x_{ij}^p, y_{ij}^p)$. The inventory handling costs is the second component computed from the average inventory $\frac{q_{ij}}{2}$ of the Economic Order Quantity (EOQ) model using a periodic inventory strategy of $q_{ij}$ items shipped to the POD that serves the district $C_{ij}$, which assumes a linear consumption rate of the demand and constant lead times. With $c^H$ as the unit inventory handling cost per unit, the total inventory costs based on the average inventory on hand is $c^H \frac{q_{ij}}{2}$. The third component is the distribution costs, with $c^T D$ as the unit travel time costs. The delivery times consists of the long-haul two-way trip for the transportation of the relief supplies from the distribution center to the POD $2\tau^V(x_{ij}^p, y_{ij}^p)$ and the relief unloading, loading and distribution times $q_{ij}\tau^{SA}$. The latter times are obtained through the product of the productivity rate $\tau^{SA}$ in times per unit supply (e.g., hrs/lb), that it takes to setup, load, unload, and prepare the relief to be distributed.

Equation 2 computes the deprivation costs assumed by the population served by the POD, which is given by the product of the population density $\rho(x, y)$ and the deprivation cost function $\gamma_g(\theta_g, \delta(x, y, x_{ij}^p, y_{ij}^p))$. The deprivation costs incurred at each time of delivery depend on the time it takes the population located near $(x, y)$ to reach the POD at $(x_{ij}^p, y_{ij}^p)$ to receive the relief supplies; that is, the deprivation time $\delta(x, y, x_{ij}^p, y_{ij}^p)$.

Incorporating the deprivation costs component in the objective function as shown in Equation 3 is especially challenging. This second component involves solving a double integral that may not always be easily integrable. Assuming the region impacted by the disaster is rectangular and a Manhattan distance metric, we develop a set of CA models for distinct typical shapes (i.e., rectangle, triangle, rhombus) that represent potential service areas of PODs. The CA models are developed by integrating the deprivation cost function across a homogeneous and uniform region and represent the shape's dimensions and POD location variables. For example, the CA model of the deprivation costs incurred at a POD located at $x_{ij}^p, y_{ij}^p$ that serves rectangular district $C_{ij}$ of length $lx_i$ and width $ly_j$ are expressed in Equation 4 for the linear deprivation cost function $b^L + m^L \delta(x, y, x_{ij}^p, y_{ij}^p)$, where $\delta$ is the deprivation time associated with the delivery at each POD location.

$$\Gamma_{ij} = \rho \left[ b^L ly_j lx_i + m^L ly_j lx_i \left( \frac{x_{ij}^p + y_{ij}^p}{s^V} + q_{ij} \tau^{SA} \right) \right. \\ \left. + \frac{m^L}{s^{WP}} \left( ly_j {x_{ij}^p}^2 + lx_i {y_{ij}^p}^2 - lx_i ly_j x_{ij}^p - lx_i ly_j y_{ij}^p + \frac{ly_j lx_i^2}{2} + \frac{lx_i ly_j^2}{2} \right) \right] \tag{4}$$

These CA models are a function of location and size of the district and compute the population's total deprivation costs. These functions are incorporated in the MIP formulation to replace the "deprivation costs" component of the total social costs in Equation 3. The MIP considers multiple configurations of the rectangular region into districts that could take multiple shapes, and with each shape, its corresponding CA model.

## References


[1] J. Holguín-Veras, N. Pérez, M. Jaller, L. N. Van Wassenhove, and F. Aros-Vera, "On the appropriate objective function for post-disaster humanitarian logistics models," *Journal of Operations Management*, 31(5), 262-280, 2013.

[2] A. Losch, *The Economics of Location*, New Haven, CT, USA: Yale University Press (in German), 1940 (Transl. by W. H. Woglam and W. F. Stolper, 1954).

[3] G.F. Newell, "Scheduling, location, transportation, and continuum mechanics: some simple approximations to optimization problems," *SIAM Journal of Applied Mathematics*, 25(3), 346-360, 1973.

[4] A.M. Geoffrion, "Making better use of optimization capability in distribution system planning," *AIIE Transactions*, 11, 96-108, 1979.

[5] M. Beckmann, "A continuous model of transportation," *Econometrica*, 20(4), 643-660, 1952.

# A Learning Framework for Generating Practical Last-mile Delivery Routes

**Mohammad Hesam Rashidi**
Department of Civil & Mineral Engineering,
University of Toronto, Toronto, Canada
Email: hesam.rashidi@mail.utoronto.ca
**Mehdi Nourinejad**
Department of Civil Engineering,
York University, Toronto, Canada
Email: mehdi.nourinejad@lassonde.yorku.ca
**Matthew Roorda**
Department of Civil & Mineral Engineering,
University of Toronto, Toronto, Canada
Email: matt.roorda@utoronto.ca

## 1 Introduction

Last-mile deliveries refer to the final leg of the supply chain where goods are transported from a warehouse to end consumers' doorsteps. Last-mile routing is a combinatorial optimization problem usually formulated as a variation of the Traveling Salesman Problem (TSP). The generic TSP prescribes the *optimal* route (e.g., minimal travel time) that a truck located at a warehouse can take to visit each customer exactly once and then return to the warehouse. While routes generated by TSPs may be efficient on paper, empirical evidence suggests that they are impractical, and drivers frequently deviate from them due to missing factors in the optimization algorithm [1]. This study develops a data-driven framework that reduces the gap between theoretical optimization and practical applicability in last-mile deliveries.

Recent literature highlights the value of human-centred algorithms that incorporate the preferences and insights of the workforce into operations planning. A human-centric algorithm leverages the tacit knowledge that workers accumulate through daily experience, thereby enhancing productivity and service quality. For instance, Sun et al. [2] integrated human judgment into a bin-packing algorithm. The bin-packing problem aims to find the minimum number of containers needed to pack items while respecting capacity constraints. Sun et al. [2] enhanced Alibaba's algorithm by integrating packer experiences and behaviours to reduce non-conformity and increase productivity. Gattermann-Itschert et al. [3] developed a methodology for including planners' preferences in a railway crew scheduling problem. They used supervised Machine Learning (ML) to detect favourable duty characteristics using historical data and optimization to generate practical crew schedules.

This study develops a solution generation framework for TSPs with soft time windows. The framework is trained on the open-source dataset from Amazon's Last-mile Research

Challenge [4]. The dataset includes over 6,000 historically realized TSP instances in the United States. For every TSP instance, the actual sequence in which the driver visited the customers is documented. Each realized delivery route is classified according to one of three qualities: high, medium, or low. The route quality labels serve as a proxy, indicating the level of satisfaction for logistics planners regarding a given observed route. Our framework uses supervised ML to identify patterns between the structural characteristics of a route and its quality. We use the insights from the ML model to create a data-informed Random Insertion (RI) heuristic[5]. This research generates solutions that conform to the constraints of a TSP, are efficient on paper, and have great potential to have high quality in the real world.

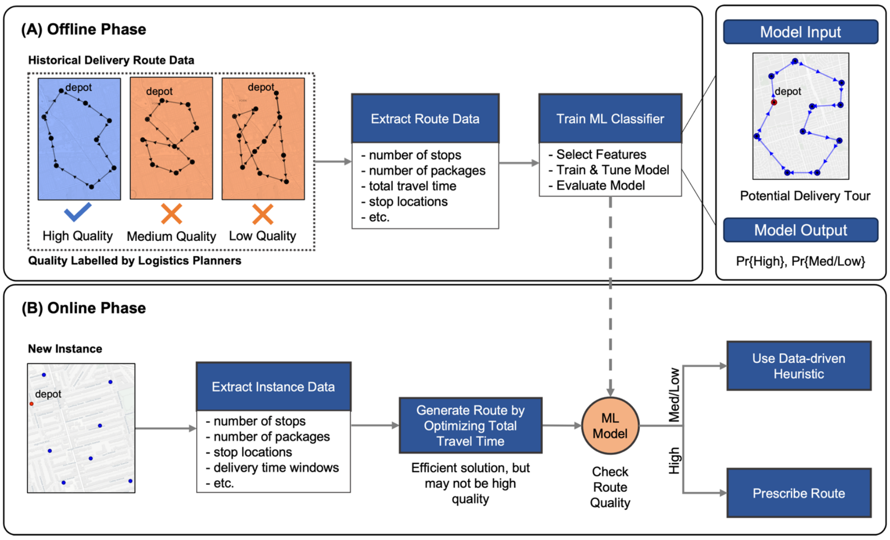



**Figure 1.** High-level solution generation framework. A travel-based solution is a TSP solution generated by optimizing total travel time.

## 2 Methodology

Figure 1 shows the high-level framework in this study. In the offline phase, the framework trains an ML classifier on the historical delivery data using supervised learning (Logit, Random Forest, and Neural Networks models). The classifier is designed to quantify the probability of a tour's high or medium/low quality in the field. We engineered two sets of features, differing in whether they change as the delivery sequence changes: instance-related (e.g., number of stops, packages, and day type) and route-related features (e.g., route duration or recipient availability likelihood). This study uses five-fold cross-validation to evaluate the model's generalization ability by iterative training and validating the model on different combinations of folds. Throughout the splitting process, stratified random sampling ensures that the proportions of varying route quality classes are preserved in all subsets.

The framework uses the trained classifier in an RI heuristic in the online phase to create solutions beyond optimizing travel time. If a solution generated by optimizing travel time (travel-based solution hereafter) classifies high quality, the solution is prescribed to the driver as is. However, if the solution does not meet the quality standards, then a solution is generated using the data-informed RI. This study generates the travel-based solution using a local search metaheuristic from Google's OR-Tools, constrained to a one-second runtime [6].

The RI dynamically inserts randomly selected unvisited stops into a partial tour. We create the initial partial route by extracting the neighbourhood visit sequence from a historically realized high-quality instance with the most matching neighbourhoods to the new instance. We use the matching neighbourhoods' visit sequence as a blueprint (i.e., initial partial route) for creating a solution for the new instance. We determine the best insertion position for an unvisited node by considering the added travel time, the turning angle caused, and the resulting backtracking distance, where all of which can be calculated within $O(1)$. We run the mentioned procedure multiple times to create a pool of solutions where we use the classifier to choose the highest quality solution in the pool. The classifier has a time complexity of $O(n)$ as it needs to traverse the route once to determine its quality.

# 3 Results

Table 1 compares the performance of the classifiers during cross-validation and testing. All models generalize well to unseen data, showing consistent performance during training and testing. All three models show similar explanatory performance; however, we chose the logit model as the classifier for creating the heuristic as it offers valuable statistical information about the features' significance and role in determining the quality of a route.

The logit classifier shows statistical evidence that high-quality routes visit neighbourhoods at different times than when the medium/low-quality routes do; a route with more similar visit times to previously realized high-quality routes is more practical and has a higher quality. Further, high-quality routes have fewer backtracks to already visited neighbourhoods and are designed to serve more packages per stop compared to medium/low-quality routes.

We exploit these insights within the data-informed RI heuristic. The drivers' routes in 42.4% of the test set (15% of the entire dataset) are labelled high-quality by Amazon. Based on the classifier's *corrected* assessment, the RI creates high-quality solutions for 68.9% of the testing set instances (i.e., a 26.5% increase from the driver-performed). The figure above is the classifier's assessment, and the classifier is not perfect. However, we are correcting for the classifier's positive and negative predictive values when reporting these values. This is justified as the classifier shows generalizability on unseen data, as shown in Table 1 (i.e., the precision and NPV scores are similar during training and testing).

The increase in the number of high-quality routes comes with a degree of trade-off with total travel time. The median travel time of the data-informed routes is 7.7% higher than the median for travel-based solutions. In comparison, the median travel time for the driver-performed routes is 4.1% higher than the travel-based solution.

The Amazon challenge used a disparity metric to evaluate the competing algorithms; their metric compared a generated route to the high-quality driver-performed route for the same instance and reported a dissimilarity score between 0 and 1. The closer to 0, the lower the disparity between the two routes [4]. Cook et al. [7] scored an average of 0.019 dissimilarity on unseen data, earning the best score in the competition, whereas our method achieved an average of 0.083. Although our method does not outperform the state-of-the-art using the disparity metric, we contribute to the literature by developing a learning framework. We learn promising route structures from the high-quality routes while also learning what to avoid from the medium/low-quality routes. Cook et al. [7] only exploit high-quality routes to generate routes which are not representative of the available dataset. Further, our classifier suggests that instance-related features affect route quality. Amazon's disparity metric scores algorithms solely on the high-quality routes with a statistically different instance structure than the medium/low-quality routes. Therefore, the disparity score, while useful, may not paint the complete picture of an algorithm's performance.

**Table 1.** Classifiers' performance during cross-validation and testing.

| Model Name | Dataset | Accuracy | Precision | Recall | F1 | NPV[1] | AUC[2] |
|---|---|---|---|---|---|---|---|
| **Logit** | CV[3] Mean | 0.703 | 0.692 | 0.600 | 0.642 | 0.710 | 0.766 |
| | Test Set | **0.728** | **0.736** | 0.603 | 0.663 | 0.723 | **0.783** |
| **Random Forest** | CV Mean | 0.696 | 0.662 | 0.649 | 0.655 | 0.723 | 0.756 |
| | Test Set | 0.708 | 0.674 | **0.664** | **0.669** | **0.735** | 0.771 |
| **Neural Networks** | CV Mean | 0.701 | 0.677 | 0.628 | 0.651 | 0.718 | 0.765 |
| | Test Set | 0.719 | 0.704 | 0.635 | 0.668 | 0.730 | 0.782 |

[1]**NPV**: Negative Predictive Value, [2]**AUC**: Area under curve, [3]**CV**: Cross Validation

# References


[1] S. Özarık, P. da Costa, and A. M. Florio, "Machine Learning for Data-Driven Last-Mile Delivery Optimization," *Transportation Science*, 0(0), 2023.

[2] J. Sun, D. J. Zhang, H. Hu, and J. A. Van Mieghem, "Predicting Human Discretion to Adjust Algorithmic Prescription: A Large-Scale Field Experiment in Warehouse Operations," *Management Science*, 68(2), 846–865, 2022.

[3] T. Gattermann-Itschert, L. M. Poreschack, and U. W. Thonemann, "Using Machine Learning to Include Planners' Preferences in Railway Crew Scheduling Optimization," *Transportation Science*, 57(3), 796–812, 2023.

[4] D. Merchán et al., "Amazon Last Mile Routing Research Challenge: Data Set," *Transportation Science*, 2022.

[5] A. M. Campbell and M. Savelsbergh, "Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems," *Transportation Science* 38 (2004).

[6] L. Perron and V. Furnon, "OR-Tools," *Google*. Accessed: Jul. 25, 2023. [Online]. Available: https://developers.google.com/optimization

[7] W. Cook, S. Held, and K. Helsgaun, "Constrained local search for last-mile routing," *Transportation Science*, 58(1), 12–26, 2024.

# The location routing problem with load-dependent travel times for cargo bikes

**Alexander Rave**
Ingolstadt School of Management, Catholic University Eichstätt-Ingolstadt, Ingolstadt, Germany, Email: ARave@ku.de

**Pirmin Fontaine**
Ingolstadt School of Management, Catholic University Eichstätt-Ingolstadt, Ingolstadt, Germany

## 1 Introduction

Last-mile delivery with traditional delivery trucks is ecologically unfriendly and leads to high road utilization. Thus, cities seek for different delivery options to solve these problems [1]. One promising option is the use of cargo bikes in last-mile delivery. These bikes are typically released at micro hubs, which are small containers located at advantageous places in the city center. Since the bike's travel speed is dependent on its remaining load as well as the gradient of the street [2], placing the hubs at valleys might cause additional work for rides. Therefore the following question arises: How high is the impact of load-dependent travel times on micro hubs' placements?

To answer this question, it is necessary to consider both the operational routing and tactical micro hub location decisions simultaneously. For this, we introduce the location routing problem with load-dependent travel times and time windows (LRPTWLTT). We formulate the problem as a mixed-integer linear program (MILP) and introduce an adaptive large neighborhood search (ALNS) to solve larger instances.

There are few publications considering load-dependent travel times. [2] focuses on the routing of cargo bikes and [3] on the routing of electric-powered vehicles. [4] consider load-dependent flight times for drone routing. However, the location decision with load-dependent travel times and cost minimization has not been addressed in the literature so far.

We contribute to the literature as follows: First, we combine the location decision with load-dependent travel times. Second, we formulate the problem as MILP and develop an ALNS. Third, we generate multiple managerial insights on both the impact of load-dependent travel times and the consideration of the location decision.

## 2 Problem setting

We assume that there is a finite set of potential micro hub locations and decide on the cost-minimal locations of micro hubs, the number of (homogeneous) cargo bikes allocated to each hub, and the bikes' routing to serve all customers once. For this, we consider a fixed cost term for each micro hub and cargo bike used and variable costs for the routing.

Multiple cargo bikes can be allocated to a single hub. Thus, once allocated to a micro hub, the cargo bike has to be launched and returned to this hub. The customers have a demand mass and each cargo bike has a weight-dependent maximum payload. There are load-dependent travel times that influence the arrival times of each customer. The load-dependent travel times depend on the air and rolling resistance, the gravity force, and frictional losses [2]. Each customer has a fixed time window, where he can be served. The bike might arrive earlier at the customer, which leads to waiting times. In addition, service time arises when a customer is served.

## 3 MILP formulation

We consider the index set for customers ($C$), micro hubs ($H$), and all nodes ($N = H \cup C$). Additionally, following the idea of [2], the bike's load is divided into several load levels defined by set $L$. Each load level $l$ is bounded by its minimum $p_l$ and the maximum $r_l$ and assumed to have an average load of $(p_l + r_l)/2$. Then, $p_0$ equals 0 and $p_{|l|}$ equals $Q$, with $Q$ being the maximum payload per bike.

When traveling from node $i$ to node $j$, certain traveling times ($t_{i,j,l}$) depend on the remaining bike's load level and the road's gradient. Each customer has a time window $[a_i, b_i]$, a demand mass $q_i$, and a service time $s_i$. In the delivery system, costs occur for each hub opened ($\hat{c}^h$), each bike used ($\hat{c}^b$), and when traveling from node $i$ to $j$ ($c^v_{i,j}$).

The main decision variables determine the routing between nodes ($x_{i,j}$), if a micro hub is opened ($v_i$), and the number of bikes allocated to a hub ($w_i$). Aligned decisions are the arrival time at a node ($y_i$). The decision variable $z_{i,j,l}$ indicates whether a load level is chosen when traveling from $i$ to $j$.

$$\min \sum_{i \in H} \left(\hat{c}^h \cdot v_i + \hat{c}^b \cdot w_i\right) + \sum_{i,j \in N} c_{i,j} \cdot x_{i,j} \tag{1}$$

subject to

$$\sum_{j \in N} (x_{i,j} - x_{j,i}) = 0 \qquad \forall i \in N \tag{2}$$

$$\sum_{j \in N} x_{i,j} = 1 \qquad \forall i \in C \tag{3}$$

$$\sum_{j \in C} x_{i,j} \leq M \cdot v_i \qquad \forall i \in H \qquad (4)$$

$$\sum_{j \in C} x_{i,j} \leq w_i \qquad \forall i \in H \qquad (5)$$

$$\sum_{j \in N} (f_{j,i} - f_{i,j}) = q_i \qquad \forall i \in C \qquad (6)$$

$$f_{i,j} \geq q_j \cdot x_{i,j} \qquad \forall i \in N, j \in C \qquad (7)$$

$$f_{i,j} \leq (Q - q_i) \cdot x_{i,j} \qquad \forall i \in C, j \in N \qquad (8)$$

$$y_i - y_j + s_i + \sum_{l \in L} t_{i,j,l} \cdot z_{i,j,l} \leq M \cdot (1 - x_{i,j}) \qquad \forall i, j \in C : i \neq j \qquad (9)$$

$$a_i \leq y_i \leq b_i \qquad \forall i \in C \qquad (10)$$

$$\sum_{l \in L} z_{i,j,l} = x_{i,j} \qquad \forall i, j \in N \qquad (11)$$

$$\sum_{l \in L} p_l \cdot z_{i,j,l} \leq f_{i,j} \leq \sum_{l \in L} r_l \cdot z_{i,j,l} \qquad \forall i, j \in N \qquad (12)$$

$$v_i \in \{0, 1\}, w_i \in \mathbb{N} \qquad \forall i \in H \qquad (13)$$

$$x_{i,j}, z_{i,j,l} \in \{0, 1\}, f_{i,j} \in \mathbb{R}^+ \qquad \forall i, j \in N, l \in L \qquad (14)$$

$$y_i \in \mathbb{R}^+ \qquad \forall i \in C \qquad (15)$$

The objective function minimizes total costs. Constraints (2) conserve flow, and Constraints (3) ensure that each customer is served exactly once. Constraints (4) define if a micro hub is opened, and Constraints (5) determine the number of bikes per hub. The remaining bike's load traveling from $i$ to $j$ is defined in Constraints (6). This load is limited to each bike's payload (Constraints (7) and Constraints (8)). Subtours are eliminated in Constraints (9). Constraints (10) ensure that the time windows are adhered to. When traveling from $i$ to $j$ exactly one load level is selected (Constraints (11)). Constraints (12) ensure that each bike's load is within the load interval. Last, variables are defined.

As only small instances can be solved when implementing the MILP, we further developed a metaheuristic solution procedure capable of solving larger instances, which we will describe in detail during our talk.

## 4 Preliminary numerical results

We consider the instances of [2] for five large cities (Fukuoka, Madrid, Pittsburgh, Seattle, and Sydney) with 100 customers. Figure 1 reports the average cost decrease in % for the five cities when solving the LRPTWLTT using our ALNS compared to the vehicle routing problem with load-dependent travel times implying fixed hub locations (blue bars) and the location routing problem implying not considering load-dependent travel times (yellow bars).

Cost decrease compared to fixed hub locations (VRPTWLTT) and not considering load-dependent travel times (LRPTW)

| | Effect of h | Effect of lo | Bevölkerungsdichte |
|---|---|---|---|
| Fukuoka | 8.8% | 7.5% | 4668 |
| Madrid | 3.6% | 9.4% | 5416 |
| Pittsburgh | 5.5% | 8.3% | 2000 |
| Seattle | 2.0% | 7.8% | 3393 |
| Sydney | 7.7% | 7.6% | 8000? |

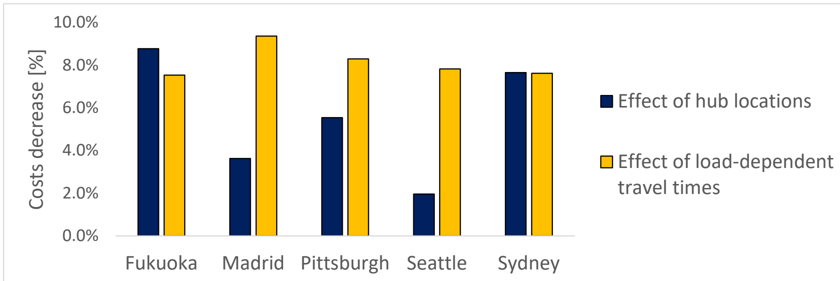



Figure 1: Cost savings when optimizing hub locations (blue bars) and when considering load-dependent travel times (yellow bars).

We find that load-dependent travel times have a significant influence on location and routing decisions. When considering load-dependent travel times, costs decrease by at least 7.5%. In addition, when placing micro hubs optimally, costs can be decreased by at least 2.0% in Seattle and at maximum by 8.8% in Fukuoka.

## 5 Conclusion

This talk introduces the location routing problem with load-dependent travel times and time windows. We formulate the problem as MILP and present an ALNS. Preliminary results of our heuristic show that load-dependent travel times have a large impact on micro hubs' placements. Details on the ALNS and the full numerical insights will be given at the conference.

## References


[1] N. Boysen, S. Fedtke and S. Schwerdfeger, "Last-mile delivery concepts: a survey from an operational research perspective", *OR Spectrum*, 43 (1), 1-58, 2021.

[2] P. Fontaine, "The vehicle routing problem with load-dependent travel times for cargo bicycles", *European Journal of Operational Research*, 300 (3), 1005-1016, 2022.

[3] Z. Wu, J. Wang, C. Chen and Y. Liu, "The load-dependent electric vehicle routing problem with time windows", *International Journal of Shipping and Transport Logistics*, 17 (1-2), 182-213, 2023.

[4] M. Nishira, S. Ito, H. Nishikawa, X. Kong and H. Tomiyama, "An integer programming based approach to delivery drone routing under load-dependent flight speed", *Drones*, 7 (5), 2023.

# Value of Real-Time Parking Information for Routing Last-Mile Delivery

**Sara Reed**
Analytics, Information, and Operations Management, University of Kansas, Lawrence, Kansas
Email: sara.reed@ku.edu

**Michael T. Lash**
Analytics, Information, and Operations Management, University of Kansas, Lawrence, Kansas

## 1 Introduction

Parking poses a challenge for last-mile delivery in urban environments [1]. One way to mitigate this challenge may be to provide the driver with real-time information on parking availability. Experimental work by the Urban Freight Lab [2] in Seattle, Washington, shows that real-time parking information reduced driving time and distance by 28% and 12%, respectively, relative to driver routes without this information. Notably, the aim of their observational study is to understand driver decision making. Our work models how this parking information may be utilized to improve parking and consolidation decisions.

The Dynamic Parking Delivery Problem with Real-Time Information (DPDP-I) is the problem of choosing a parking spot and how many customers to serve from that parking spot while en-route. We model the DPDP-I as a Markov Decision Process (MDP). Similar to other dynamic routing problems, the DPDP-I incurs the curses of dimensionality. Therefore, we develop heuristic policies and analyze the potential impacts of real-time parking information on delivery practices.

Our work focuses on two factors influencing decision making for drivers: parking availability and customer density. If parking does not pose a challenge, the driver may be efficient by serving each customer individually and driving between customer locations. However, if parking in the area is difficult, then the driver may benefit from parking once and consolidating packages into one service set for on-foot delivery. Focusing on the factor of customer density, if customers are closer together, we expect potential gains from consolidation as well. In this work, we develop a dynamic policy that identifies these interdependencies and therefore makes decisions based on these factors.

The contributions of this work can be summarized as follows:

- Introduces model to consider historical and real-time parking information for en-route decisions by the delivery driver.

- Identifies the interdependencies between parking availability and customer density in parking and consolidation decisions for the delivery driver.
- Provides computational results with real-world parking data in urban environments.

## 2 Problem Statement

The DPDP-I determines en-route decisions being made by the delivery driver tasked with serving $n$ customers. We assume the order of customer service is fixed and each customer has one package. Let $\mathcal{C}_0$ denote the initial sequence of customers to be served. En-route decisions include where to park the vehicle and which customers (up to a capacity of $q$ customers) to serve on-foot prior to returning to the vehicle. Let $P$ be the set of potential parking spots. Parking availability is stochastic and some parking spots are occupied at the time the delivery person is making the decision.

We model the delivery process as an infinite-horizon MDP with a trapping state (when all customers have been served.) The first decision epoch occurs when the delivery person leaves the depot. Otherwise, a decision epoch occurs every time the delivery person returns back to the vehicle. The state $s_k = (t_k, i_k, \mathcal{P}_k, \mathcal{C}_k)$ is defined by the time $t_k$, current parking location $i_k$, available parking locations $\mathcal{P}_k \subseteq P$, and the sequence of customers left to be served $\mathcal{C}_k$. Let $\mathcal{C}_k^i$ denote the $i$-th customer in the sequence $\mathcal{C}_k$. The action $(l, m)$ indicates that the driver parks at parking spot $l \in \mathcal{P}_k$ and serves the next $m$ customers in $\mathcal{C}_k$. Equation (1) defines the cost of taking action $(l, m)$ in state $s_k$,

$$C(s_k, (l, m)) = d(i_k, l) + m \cdot f + w\left(l, \mathcal{C}_k^1\right) + \sum_{j=1}^{m-1} w\left(\mathcal{C}_k^j, \mathcal{C}_k^{j+1}\right) + w\left(\mathcal{C}_k^m, l\right) \tag{1}$$

where $d(u, v)$ is the driving time between $u, v \in P$, $f$ is the loading time for each package, and $w(u, v)$ is the walking time between $u, v \in \mathcal{C}_0 \cup P$.

The optimal policy $\pi^*$ minimizes the total expected time of the delivery tour and can be expressed as

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmin}}\, \mathbb{E}\left[\sum_{k=0}^{\infty} C\left(S_k, a_k^\pi(S_k)\right) | S_0\right] \tag{2}$$

where $a_k^\pi(S_k)$ is the action selected by policy $\pi$ for state $S_k$.

## 3 Solution Approach

We propose heuristic policies for the DPDP-I that we evaluate relative to existing models in the literature. Specifically, we introduce the $i$-Look-Ahead Policy for $i \leq q$ and a dynamic policy based on two features of the state: parking availability and customer density.

The $i$-Look-Ahead ($i$-L-A) policy considers the next $i$ customers when deciding where to park the vehicle and which customers to serve from that parking spot. In state $s_k$, the delivery driver can serve one of the following sets once the vehicle is parked: $\{\mathcal{C}_k^1\}$, $\{\mathcal{C}_k^1, \mathcal{C}_k^2\}$,..., or $\{\mathcal{C}_k^1, ..., \mathcal{C}_k^i\}$. This policy aims to incorporate historical parking availability information in this choice. For example, if the driver only serves customer $\mathcal{C}_k^1$, we include the expected time to serve customers $\{\mathcal{C}_k^2, ..., \mathcal{C}_k^i\}$ in the next decision epoch to determine the action that minimizes expected time to service the next $i$ customers.

To develop a policy based on interdependencies between parking availability and customer density, we generate synthetic instances and evaluate the performance of the $i$-L-A policy for $i \in \{1, 2, 3\}$. Let $p_i^t$ be the probability that parking spot $i$ is available at time $t$. We generate parking availability by sampling $p_i^t$ over a uniform distribution in one of the following intervals: $(0, 0.2)$, $(0.2, 0.4)$, $(0.4, 0.6)$, $(0.6, 0.8)$, and $(0.8, 1.0)$. We also generate customer density by sampling the distance between consecutive customers over a uniform distribution in one of the following intervals (in meters): $(0, 80)$, $(80, 160)$, $(160, 240)$, $(240, 320)$, $(320, 400)$, and $(400, 480)$. For a given probability distribution and customer density, we generate 100 instances. Then, we determine which of the policies (1-L-A, 2-L-A, or 3-L-A) achieves the lowest completion time of the delivery tour on average. Table 1 provides the policy look-up table for this dynamic policy. We observe that when customers are further apart and parking does not pose a challenge, the driver does not benefit from consolidation. When parking poses a challenge, poor parking options may diminish gains from high levels of consolidation (i.e. using 2-L-A instead of 3-L-A.)

| | **Customer Density Interval** | | | | | |
|---|---|---|---|---|---|---|
| **Probability Interval** | (0, 80) | (80, 160) | (160, 240) | (240, 320) | (320, 400) | (400, 480) |
| (0, 0.2) | 2-L-A | 2-L-A | 2-L-A | 2-L-A | 2-L-A | 2-L-A |
| (0.2, 0.4) | 2-L-A | 2-L-A | 2-L-A | 1-L-A | 1-L-A | 2-L-A |
| (0.4, 0.6) | 3-L-A | 2-L-A | 1-L-A | 1-L-A | 1-L-A | 1-L-A |
| (0.6, 0.8) | 3-L-A | 1-L-A | 1-L-A | 1-L-A | 1-L-A | 1-L-A |
| (0.8, 1.0) | 3-L-A | 1-L-A | 1-L-A | 1-L-A | 1-L-A | 1-L-A |

Table 1: Dynamic policy look-up table based on synthetic instances.

## 4 Results and Conclusions

The service region for our computational study is the Downtown neighborhood of Los Angeles, California, where the daily parking meter sensor activity is publicly available [3]. We utilize this sensor information to determine the probability $p_i^t$ that parking spot $i \in P$ is available at time $t$.

We evaluate our solution approaches with two types of benchmark policies: a greedy policy and deterministic routing policies. The No-Look-Ahead policy (No-L-A) dictates that the delivery driver greedily parks at the closest available parking spot to the next customer and services up to $q$ customers that are within a specified walking radius of that parking spot. We utilize a generalization of the Modified TSP (M_TSP_$\bar{p}$) in [1] as a deterministic solution for a fixed service order that accounts for the difficulty to find parking based on the average parking time $\bar{p} \in \{0, 3, 6, 9\}$.

Figure 1 summarizes average time spent in delivery activities for each policy. The left figure shows the dynamic policy realizes the lowest completion time of the delivery tour, followed by the 1-L-A policy. The right figure shows that the dynamic policy reduces both average driving and walking times relative to the 1-L-A policy to achieve this result. We also analyze the dynamic policy on smaller service areas that may realize higher benefits from consolidation.

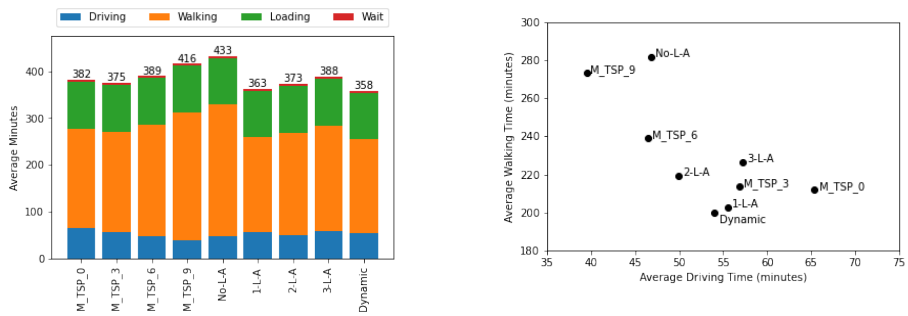



Figure 1: Average time spent in delivery activities for each policy (left), in particular walking and driving time comparisons (right).

An understanding of how to utilize real-time parking information to make en-route decisions for the delivery driver may reduce completion times 17% on average relative to a greedy approach. We show efficient trade-offs between walking and driving times may be achieved by considering parking availability and customer density in real time as opposed to deterministic modeling approaches. Understanding the value of real-time information in last-mile delivery informs a broader discussion on the value of smart city infrastructure.

## References

[1] S. Reed, A. M. Campbell, and B. W. Thomas, "Does Parking Matter? The Impact of Parking Time on Last-Mile Delivery Optimization," working paper at `https://arxiv.org/abs/2107.06788`.

[2] G. Dalla Chiara, K. Fiete Krutein, A. Ranjbari, and A. Goodchild, "Providing curb availability information to delivery drivers reduces cruising for parking," *Scientific Reports* 12, no. 1 (2022).

[3] LADOT Parking Meters Division, "LADOT Parking Meter Occupancy - Archive," accessed 3/30/2022 at `https://data.lacity.org/Transportation/LADOT-Parking-Meter-Occupancy-Archive/cj8s-ivry`.

# The Two-Echelon Multicommodity Location-Routing Problem with Stochastic and Correlated Demands

**Walter Rei[1,2], Teodor Gabriel Crainic[1,2], David Escobar-Vargas[2,3], and Stein W. Wallace[4]**
Email: rei.walter@uqam.ca

[1] Département d'analytique, opérations et technologies de l'information, École des Sciences de la Gestion, Université du Québec à Montréal, Canada

[2]CIRRELT, Canada

[3] Département d'informatique et recherche opérationnelle, Université de Montréal, Canada

[4] Centre for shipping and logistics, Norwegian School and Economics (NHH), Norway

## 1 Introduction

In modern freight transportation systems, echeloned (or tiered) networks are employed, wherein distribution activities are planned and conducted throughout the echelons, see [2, 3]. The planning of these systems presents significant challenges that must be addressed to ensure the efficient conduct of distribution activities. At the strategic and tactical levels, a crucial challenge lies in making decisions about the configuration and structure of the network, which is subsequently employed for distribution activities in conditions that vary randomly. At this conference, we will present novel methodological work focused on solving the Two-Echelon Multicommodity Location-Routing Problem with Stochastic and Correlated Demands (2E-MLRPSCD). From a practical perspective, the 2E-MLRPSCD is relevant for decision-making regarding the locations of central infrastructure and the assignment of freight flows to facilitate distribution activities throughout a transportation network when demands randomly change. Furthermore, correlation phenomena related to demands can significantly impact the optimal assignment of flows, making it a crucial aspect to consider. From a methodological perspective, the 2E-MLRPSCD defines a particularly hard combinatorial optimization problem to solve, motivating the need for efficient algorithms.

## 2 Problem definition

The 2E-MLRPSCD is defined in a system that includes three main components: *platforms* (primary facilities serving as demand origins), *satellites* (intermediate facilities), and *customers* (demand destinations). Formally, the problem is represented as a complete weighted directed graph $N = (V, A)$, with vertices $V = P \cup Z \cup C$, divided into three disjoint sets: platforms $P$, satellites $Z$, and customers $C$, see [3]. Platforms are large-sized facilities with a known set of commodities to be distributed to customers. Satellites are medium- to small-sized multimodal infrastructures that serve as intermediate facilities, allowing the consolidation and sorting of freight between the two transportation echelons involved in distributing goods to customers. Each satellite location $z \in Z$ is associated with a limited storage capacity $Q_z$ and a fixed opening cost $F_z$.

*Demand* is defined between platforms and customers, each individual demand being characterized by an origin, a destination and a requested volume to be delivered. Let $K$ denote the set of origin-destination (OD) demands. In the deterministic version of the problem, each OD demand $k \in K$, is characterized by a known volume $vol_k$, an origin $O(k)$ associated with a platform node in $P$, and a destination $D(k)$ associated with a customer node in $C$. Additionally, a fixed assignment cost $\Delta_{pzk}$ represents the cost of serving OD demand $k \in K$ through platform $p \in P$ and satellite $z \in Z$. This cost reflects the necessity to plan and secure the operational capacity required for transporting the associated commodity through the corresponding infrastructure.

Each arc $(i, j) \in A = A^1 \cup A^2$ is associated with a non-negative cost $\zeta_{ij}$ for a vehicle to travel between $i$ and $j$. Set $A^1$ denotes the arcs of the first echelon, corresponding to the connections between platforms $P$ and satellites $Z$ and between satellites. As for the set $A^2$, it comprises the arcs of the second echelon, which involve connecting the satellites $Z$ with the final customers $C$ and connecting the customers among themselves, see [3].

Deliveries are performed by two homogeneous fleets of vehicles $H = H^1 \cup H^2$ with limited load capacities $cap_1$ and $cap_2$, which are respectively available for the first and second echelon, and are able to transport any demand. Specialized fleets might be used at each echelon to accommodate the specific requirements of various applications. For instance, in city logistics systems, the fleet $H^2$ might include smaller city freighter vehicles better suited for conducting distribution activities in urban areas, see [2]. In all cases, vehicles are assumed to be available at each existing facility for each echelon, where vehicles start and end their routes.

The considered problem involves the selection of satellite facilities, the assignment of OD demands to satellites, as well as the routing of vehicles at each echelon to deliver the freight from platforms to customers, going through satellite facilities. Each OD demand that is made available at its originating platform has to be moved by a first-echelon vehicle to a given satellite to be then transferred to a second-echelon vehicle. Loads delivered at satellites are then transshipped and consolidated into second-echelon vehicles, which will perform the deliveries to the final destinations.

The 2E-MLRPSCD involves uncertainty in the volume of demand stemming from random changes occurring between correlated OD pairs. Probability distributions are assumed available to describe the random demand variations. Moreover, the problem setting involves correlations among OD pairs, where each OD pair can be either positively or negatively correlated with other distinct OD pairs. Specifically, the studied problem is characterized by two sets of OD pairs used to represent the correlation; OD pairs within each set are positively correlated, while all correlations between OD pairs in different sets are strongly negative (i.e., low demands in one set result in high demands in the other).

The 2E-MLRPSCD problem setting addresses strategic and tactical planning decisions in multiple application fields. In terms of decision-making and information processing, the design and assignment decisions during the planning stage must be defined based on an evaluation/estimation of their impact on operations, including the available recourse actions to adapt the plan to the observed demands. To account for this setting, the 2E-MLRPSCD is thus formulated as a two-stage stochastic optimization model. The first-stage decisions involve selecting the locations of satellite facilities and assigning OD demands to these satellites. These decisions are made while facing the uncertainty. In the second stage, the demand volumes are revealed, at which point, a limited set of routes for the first and second echelons vehicles are constructed in such a way that: (i) every route of the first echelon starts and ends at the same platform; (ii) every route of the second echelon starts and ends at the same satellite; (iii) all the customers' demands are satisfied either by the system or an outsource service; (iv) the load capacity of each vehicle is not exceeded; (v) each customer served by the system is visited by only one vehicle; and (vi) the total demand assigned to a satellite facility must not exceed its capacity. Therefore, the overall objective of the two-stage stochastic optimization model is to minimize the sum of the fixed location and assignment costs and the expected routing costs (the recourse actions).

## 3 A progressive hedging-based metaheuristic for the 2E-MLRPSCD

We develop a progressive hedging-based metaheuristic to solve the considered problem. Our methodology builds on the work dedicated to solving the stochastic network design problem, see [1]. From a methodological perspective, the 'classic' progressive-hedging algorithm iteratively solves the set of deterministic subproblems, which result from the scenario-based decomposition of the extensive formulation. The latter formulation arises through the application of sampling techniques to render the stochastic model solvable.

At each iteration, the PH metaheuristic solves each scenario-specific deterministic subproblem separately, thus producing a series of solutions that may differ from one another. The search then proceeds by computing a reference solution (the expected value of the best scenario-specific solutions is traditionally used), which also serves to assess the overall level of consensus among the scenario-specific solutions. The formulations of the scenario subproblems are then adjusted to incentivize agreement (i.e., to make subproblems converge to the same solution). This general process is repeated until either a consensus solution is found or another stopping criterion is reached (e.g., a computation time limit).

It is well known that the PH algorithm does not necessarily converge to an optimal solution when it is applied in the case of mixed-integer programs, such as the 2E-MLRPSCD. This thus presents an initial challenge to address. Another significant algorithmic challenge stems from the computational load associated with solving a sequence of NP-hard problems (one for each scenario) during each iteration of the PH metaheuristic. There is a clear need for an efficient guiding strategy and procedures to direct the algorithm towards reaching a high-quality consensus solution more quickly. Therefore, we introduce a PH metaheuristic with a set of algorithmic and methodological enhancements aimed at accelerating the search for an efficient implementable solution. These enhancements encompass: 1) a set of population structures to obtain alternative and diverse solutions for the scenario subproblems, 2) a set of novel scenario-selection strategies that effectively derive key insights from subproblem solutions to identify potential consensus, 3) a specialized heuristic to define a high-quality reference solution in the first PH iteration, and 4) a reset procedure to prevent the PH metaheuristic from getting trapped in local optima.

At the upcoming conference, we will present 1) the two-stage formulation for the 2E-MLRPSCD, 2) the developped PH metaheuristic (emphasizing the designed enhancements), and 3) some meaningful computational experiments. The latter will primarily focus on assessing the efficiency of the algorithmic enhancements and the impacts of explicitly considering stochastic demands, including correlations, when solving the problem.

## References


[1] T. G. Crainic, X. Fu, M. Gendreau, W. Rei and S. W. Wallace, "Progressive hedging-based metaheuristics for stochastic network design", *Networks* 58(2), 114-124, 2011.

[2] T. G. Crainic, G. Perboli and N. Ricciardi, "City Logistics", in *Network Design with Applications to Transportation and Logistics*, T. G. Crainic, M. Gendreau and B. Gendron (eds), Springer, 2021.

[3] N. Sluijk, A. M. Florio, J. Kinable, N. Dellaert and T. Van Woensel, "Two-echelon vehicle routing problems: A literature Review", *European Journal of Operational Research* 304(3), 865-886, 2023.

# Disaster Response on a Network with Stochastic Demand and Uncertain Edge Accessibility

**Jessa Rhea**
Applied Mathematical and Computational Sciences
University of Iowa, Iowa City, Iowa
Email: jessa-rhea@uiowa.edu

**Jeffrey W. Ohlmann**
Department of Business Analytics
University of Iowa, Iowa City, Iowa
Email: jeffrey-ohlmann@uiowa.edu

## 1 Motivation

As climate change alters the dynamics of our planet, it is amplifying the frequency, unpredictability, and severity of natural disasters such as hurricanes, earthquakes, wildfires, floods, and droughts. In addition, the escalation of sociopolitical tensions trigger human-induced disasters, demanding support for communities in conflict-ridden areas. Regardless of the cause, disasters severely impact the economy and the well-being of the affected communities [1]. The instability of these events has posed an escalating challenge for local grass roots organizations and governments, as traditional models of forecasting and risk assessment are proving unreliable [1]. As acute seasons of crisis have transformed into year-round chronic need for disaster relief and as identifying locations for predeployed resources becomes more imprecise, leveraging emerging technologies to address these issues is vital.

In the aftermath of a disaster, the timeliness of aid and critical care are crucial [1]. In response to these challenges, we advocate for the strategic routing of mobile facilities to distribute life-saving supplies. Representing a disaster-affected area as a network with uncertain accessibility of the edges and uncertain demand at the nodes, we consider the problem of distributing aid with a fleet of mobile facilities. As a mobile facility proceeds along its routes, it observes and serves demand while also gaining information on road accessibility that it shares with the rest of the fleet.

## 2 Problem Description

We propose a mobile facility routing problem to address the challenges by providing safe and immediate disaster response for a network in which there is uncertain edge traversability and stochastic beneficiary demand. Our problem is defined on an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, ..., V\}$ is the set of locations where a mobile facility can park and distribute aid, and $\mathcal{E} = \{(v, v') : v, v' \in \mathcal{V}\}$ is the set of edges representing roads between locations. We adopt the convention of Andreatta and Romeo [2] and assume that when a mobile facility arrives at location $v$, it observes the accessibility of all edges in the set $\mathcal{N}(v) = \{(v, v') : v' \in \mathcal{V}\}$. If an edge is accessible, we assume $t_{vv'}$, the travel time between locations $(v, v') \in \mathcal{E}$, is known. Furthermore, associated with each location $v \in \mathcal{V}$ is a uncertain amount of demand represented by random variable $\mathbf{D}_v$ which is observed upon arrival at the location.

Each mobile facility $m$ in the fleet of capacitated homogeneous mobile facilities $\mathcal{M} = \{1, ..., M\}$ begins and ends its route at a location $w_m \in \mathcal{W} \subset \mathcal{V}$, where $\mathcal{W}$ is the set of exogenously-determined warehouse locations at which mobile facilities have been strategically prepositioned [3]. The capacity of a mobile facility is $Q$ and a mobile facility $m$ may replenish its capacity to this level by returning to its warehouse $w_m$. When a mobile facility arrives at a location $v$ and observes demand, it serves as much of this demand as its available capacity permits. The objective is to maximize the amount of beneficiary demand served by the fleet over the problem horizon. We note that a mobile facility can contribute to this objective even if it doesn't serve demand at an epoch (e.g., when its capacity is entirely depleted) by determining accessibility of edges and observing beneficiary demand at locations along its route.

We formulate our problem as a Markov decision process (MDP) and describe its elements in the remainder of this section.

**States $S_k$:** The state of the system $S_k \in \mathcal{S}$ at decision epoch $k \in \{0, ..., K\}$ consists of the time, $t_k$; the destination of each mobile facility, $(\ell_m)_{m \in \mathcal{M}}$; the time each mobile facility will arrive at their destination, $(a_m)_{m \in \mathcal{M}}$; the available capacity of each mobile facility, $(q_m)_{m \in \mathcal{M}}$; the edge accessibility probabilities for each edge in the network, $(p_e)_{e \in \mathcal{E}}$; and the probability mass function of unmet beneficiary demand at each location in the network, $(F_v)_{v \in \mathcal{V}}$.

**Actions $x_k$:** An action $x_k \in \chi(S_k)$ assigns the next destination $\ell_m^x \in \mathcal{N}(\ell_m)$ for each mobile facility $m$ stationary $(a_m = t_k)$ at its destination $\ell_m$ in state $S_k$. For each mobile facility $m$ still en route $(a_m > t_k)$ to its destination $\ell_m$, the action $x_k$ sets $\ell_m^x = \ell_m$.

**Reward function $R(S_k, x_k)$:** The reward function $R(S_k, x_k)$ corresponds to the beneficiary demand served by stationary mobile facilities in $S_k$. We note that $R(S_k, x_k)$ is independent of $x_k$ because there is no immediate reward for determining a mobile facility's next destination as this reward will not be observed until the future arrival.

**Stochastic information** $W_{k+1}$**:** The stochastic information $W_{k+1}$ is the observed demand at locations where mobile facilities arrive at epoch $k+1$ and the observed accessibility of the edges incident to these nodes. A mobile facility $m$'s capacity for serving realized demand determines the duration of service, which is incorporated into arrival time $a_m$.

**Transition function** $S^M(S_k, x_k, W_{k+1})$**:** The transition function $S^M(S_k, x_k, W_{k+1}) = S_{k+1}$ determines the state at epoch $k+1$ from the state $S_k$, action $x_k$, and stochastic information $W_{k+1}$. This transition consists of a deterministic transition associated with action $x_k$ and then a stochastic transition resulting from $W_{k+1}$.

After accruing the reward $R(S_k, x_k) = \sum_{m \in \{\mathcal{M}: a_m = t_k\}} \min\{d_{\ell_m}, q_m\}$, we reflect the served beneficiary demand at epoch $k$ by updating the available capacity of each mobile facility as $q_{m,k+1} = \max\{q_{m,k} - d_{\ell_{m,k}}, 0\}$ for $m \in \{\mathcal{M} : a_m = t_k\}$ and $q_{m,k+1} = q_{m,k}$ for $m \in \{\mathcal{M} : a_m > t_k\}$.

Then, the action $x_k$ updates the destination of each mobile facility $m$, $\ell_{m,k+1} = \ell^x_{m,k}$, and the arrival time of each mobile facility $m$ at its destination, $a_{m,k+1}$. The arrival of a mobile facility at its destination triggers the time of the next decision epoch, $t_{k+1} = \min_{m \in \mathcal{M}} \{a_{m,k+1}\}$.

One or more mobile facilities arriving at their destination provides stochastic information $W_{k+1}$ in the form of observed beneficiary demand and observed edge accessibility. Specifically, for $v \in \mathcal{V}$ such that $v = \ell_{m,k+1}$ and $a_{m,k+1} = t_{k+1}$: (1) we set $F_v(d) = 1$ for the observed demand quantity $d$ at location $v$, and (2) we set the edge accessibility probabilities to 0 or 1 based on the observed information for all edges in $\mathcal{N}(v)$.

**Policies** $X^\pi$**:** A solution is a policy $X^\pi$ in the policy class $\Pi$ that assigns a feasible action to every state $S_k$, $X^\pi : S_k \mapsto x_k$.

**Objective:** The objective is to determine a policy $X^{\pi^*} \in \Pi$ that maximizes the expected discounted beneficiary demand served given initial state $S_0$. Thus, our goal is

$$\max_{X^\pi \in \Pi} \mathbb{E}^\pi \left[ \sum_{k=1}^{K} \alpha^{t_k} R(S_k, X^\pi(S_k)) | S_0 \right]$$

where $X^\pi(S_k)$ is an action for state $S_k$ subject to policy $\pi$, and $0 < \alpha \leq 1$ is a discount factor accounting for the time-sensitivity of serving beneficiary demand.

# 3 Solution Approach

We are conducting ongoing work to devise and compare competing solution approaches for our problem. In particular, we are considering the class of stochastic direct lookahead policies [4, 5]. To generate (plausible) a posteriori upper bounds for our candidate solution approaches on benchmark instances, we heuristically solve the vehicle routing problem with stochastic demand (VRPSD) that results when we consider a perfect information case in which all edge accessibility is known with certainty [6].

Serving as another source of inspiration, Becker and Batta [7] introduce the Canadian prize collection problem, which has an objective of maximizing the sum of collected (deterministic) prizes on a network with binary edge traversability. For the single-vehicle case, Becker et al. [8] present three solution approaches: an iterative prize collection heuristic, an iterative shortest path with prize collection heuristic, and an iterative shortest path with intermediate points and prize collection heuristic. We are examining the utility of these heuristics for our problem variant.

## References


[1] Russell J. Andrews and Leonidas M. Quintana. Unpredictable, unpreventable and impersonal medicine: global disaster response in the 21st century. *EPMA Journal*, 6(1):2, 2015.

[2] Giovanni Andreatta and Luciano Romeo. Stochastic shortest paths with recourse. *Networks*, 18(3):193–204, 1988.

[3] Monir Sabbaghtorkan, Rajan Batta, and Qing He. Prepositioning of assets and supplies in disaster operations management: Review and research gap identification. *European Journal of Operational Research*, 284(1):1–19, 2020.

[4] Lina Al-Kanj, Warren B Powell, and Belgacem Bouzaiene-Ayari. The information-collecting vehicle routing problem: Stochastic optimization for emergency storm response. *arXiv preprint arXiv:2301.06497*, 2023.

[5] Lawrence Thul and Warren B Powell. An information-collecting drone management problem for wildfire mitigation. *arXiv preprint arXiv:2301.07013*, 2023.

[6] Justin C Goodson, Barrett W Thomas, and Jeffrey W Ohlmann. Restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits. *Transportation Science*, 50(2):591–607, 2016.

[7] John Becker and Rajan Batta. Canadian prize collection problem. *Military Operations Research*, 28(2):55–92, 2023.

[8] John Becker, Esther Jose, Nastaran Oladzad-Abbasabady, and Rajan Batta. Search and rescue over uncertain terrain in humanitarian and military contexts: Capturing emerging targets and performing reconnaissance. In *Tutorials in Operations Research: Advancing the Frontiers of OR/MS: From Methodologies to Applications*, pages 285–313. INFORMS, 2023.

# New Results About Single Allocation Hub Location Problems

**Antonio Manuel Rodríguez Chía**
Departamento de Estadística e Investigación Operativa,
University of Cádiz, Spain
Email: antonio.rodriguezchia@uca.es

**Inmaculada Espejo Miranda**
Departamento de Estadística e Investigación Operativa,
University of Cádiz, Spain

**Juan Manuel Muñoz Ocaña**
Departamento de Estadística e Investigación Operativa,
University of Cádiz, Spain

**Alfredo Marín Pérez**
Departamento de Estadística e Investigación Operativa,
University of Murcia, Spain

**Raúl Páez Jiménez**
Departamento de Estadística e Investigación Operativa,
University of Cádiz, Spain

## 1 Introduction

Hub location problems arise in various application settings, e.g., telecommunication and transportation systems where several origin/destination sites send and receive some product. Instead of serving each origin-destination pair directly (because this sort of linkage is too expensive to be carried out), transshipment points (hubs) collect the product from the origin and distribute it to the destination. These hubs centralize the product shipment, resulting in lower transportation costs and potential savings in the overall design and operational costs of the system. Therefore, the hubs systems are designed to exploit the scale economies attainable through the shared use of high capacity links between hubs. [1] include discussions of modeling economies of scale and real-world examples of hub systems, as passenger and freight airlines, less-than-truckload and truckload transportation, postal operations, express shipment and cargo delivery, liner shipping, public transit, and computer and telecommunication networks. Moreover, new applications are appearing, as the green hubs or hub systems for medical applications including in drone delivery networks. This wide range of applications indicates the power of the hubs location problems and the need for more and better models. Many reviews about hub location problems, see [1, 2, 4, 5, 8], show the very wide range of activity in this field and the applications of these problems.

This paper deals with uncapacitated single-allocation hub location problems. In the uncapacitated single-allocation $p$-hub median problem (USApHMP), the aim is to choose $p$ hubs and assign every site to them minimizing the overall transportation costs between origins and destinations through the hubs. In the uncapacitated single-allocation hub location problem (USAHLP) a cost for setting a hub is given and the number of hubs is a decision variable. The aim is to locate the hubs and to assign the remaining sites to the hubs minimizing the overall installation and transportation costs. Both problems are NP-hard. Moreover, even if the locations of the hubs are fixed, the allocation part of the problem remains NP-hard [10].

[13] presented the first mathematical formulation for this problem. Since then, different linearization strategies have been used in the literature to handle the quadratic term in the objective function of this model. [3, 6, 7, 15], among others.

Alternative methods to linearize the binary quadratic terms in the formulation by [13] are given in [9, 11, 14]. The method described in [11] uses a row generation procedure and applies whenever Euclidean distances are used. [9] proposed exact algorithms based on Benders decomposition for solving large-scale instances. They assume that the transportation costs between hubs are proportional to the distance between them. [14] provided a convex reformulation and a branch-and-cut algorithm based on outer approximation cuts.

## 2 Our contribution

In this manuscript, a new compact formulation for uncapacitated single-allocation hub location problems with fewer variables than the previous Integer Linear Programming formulations in the literature is introduced. Our formulation works even with costs not based on distances and not satisfying triangle inequality. Moreover, some of the existing formulations for the USApHMP need to have the overall transportation cost from origin to destination disaggregated in the three components: origin-hub, hub-hub, hub-destination. Our formulations are valid for both cases, with aggregated/disaggregated transportation costs. This allows us to model more realistic cases in transportation systems where, for instance, fares are not proportional to travel distances or longer trips may have lower ticket prices than shorter trips.

Different families of valid inequalities are obtained considering extended formulations

| Formulation | Binary variables | Continuous variables | Constraints |
|---|---|---|---|
| [3] | $n^2 + n$ | $n^4$ | $n^4 + n^2 + n + 1$ |
| [15] | $n^2$ | $n^4$ | $2n^3 + n^2 + n + 1$ |
| [7] | $n^2$ | $n^3$ | $2n^2 + n + 1$ |
| [6] | $n^2$ | $n^2$ | $2n^2 + n + 1$ |
| [12] | $n^2$ | $n^2$ | $n^4 + n^2 + n + 1$ |
| Our Formulation | $n^2$ | $n$ | $2n^2 + n + 1$ |

Table 1: Number of variables and constraints for different formulations of the USApHMP

and later projecting out some of their variables by applying the Farkas' lemma. Moreover, separation procedures for these inequalities are developed. A comparison of the performance of the most recent and efficient solution methods existing in the literature [11, 9, 14] shows the efficiency of our methodology, solving large-scale instances in competitive times. Although we focus on USApHMP, the formulation can be adapted to USAHLP. Furthermore, capacitated versions could also benefit from our results, as could other variants of the problem.

Table 1 gives the number of variables and constraints of the aforementioned formulations, where $n$ is the number of sites that represent the origins and destinations. As shown in the table, our paper constitutes a contribution to the existing literature of the USApHMP by providing a new formulation that uses fewer variables than the aforementioned ones.

Different families of valid inequalities are obtained considering extended formulations and later projecting out some of their variables by applying the Farkas' lemma. Moreover, separation procedures for these inequalities are developed. A comparison of the performance of the most recent and efficient solution methods existing in the literature [11, 9, 14] shows the efficiency of our methodology, solving large-scale instances in competitive times. Although we focus on USApHMP, the formulation can be adapted to USAHLP. Furthermore, capacitated versions could also benefit from our results, as could other variants of the problem.

## References

[1] Alumur, S., Campbell, J. F., Contreras, I., Kara, B., Marianov, V., and O'Kelly, M. E. (2021). Perspectives on modeling hub location problems. *European Journal of Operational Research*, 291(1):1–17.

[2] Alumur, S. and Kara, B. (2008). Network hub location problems: the state of the art. *European Journal of Operational Research*, 190(1):1–21.

[3] Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72:387–405.

[4] Campbell, J. F. and O'Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169.

[5] Contreras, I. and O'Kelly, M. E. (2019). Hub location problems. In Laporte, G., Nickel, S., and Saldanha-da-Gama, F., editors, *Location Science*, pages 327–363. Springer, New York.

[6] Ebery, J. (2001). Solving large single allocation $p$-hub problems with two or three hubs. *European Journal of Operational Research*, 128:447–458.

[7] Ernst, A. T. and Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation $p$-hub median problem. *Location Science*, 4:139–154.

[8] Farahani, R., Hekmatfar, M., Arabani, A., and Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers and Industrial Engineering*, 4(4):1096–1109.

[9] Ghaffarinasab, N. and Kara, B. Y. (2019). Benders decomposition algorithms for two variants of the single allocation hub location problem. *Networks and Spatial Economics*, 19:83–108.

[10] Kara, B. Y. (1999). Modeling and analysis of issues in hub location problem. *Bilkent University, PhD thesis.*

[11] Meier, J., Clausen, U., Rostami, B., and Buchheim, C. (2016). A compact linearisation of Euclidean single allocation hub location problems. *Electronic notes in Discrete Mathematics*, 52:37–44.

[12] Meier, J. F. and Clausen, U. (2017). Solving single allocation hub location problems on Euclidean data. *Transportation Science*, 52(5):1035–1296.

[13] O'Kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393–404.

[14] Rostami, B., Errico, F., and Lodi, A. (2022). A convex reformulation and an outer approximation for a class of binary quadratic program. *Operations Research*, Online. https://doi.org/10.1287/opre.2021.2241.

[15] Skorin-Kapov, D., Skorin-Kapov, J., and O'Kelly, M. E. (1996). Tight linear programming relaxations of uncapacitated $p$-hub median problems. *European Journal of Operational Research*, 94:582–593.

# A Stochastic Prize Collection Methodology for Mobile Clinic Deployment Planning

**Rosemarie Santa González**

H. Milton Stewart School of Industrial and Systems Engineering

Georgia Tech, Georgia, USA

Email: rosemarie.santa@gatech.edu

**Teodor Gabriel Crainic** & **Marilène Cherkesly**

Department of Analytics, Operations and Information Technology

University of Quebec in Montréal, Montréal, Canada

**Marie-Ève Rancourt**

Department of Logistics and Operations Management

HEC, Montreal, Canada

## 1 Introduction

Around 90% of countries surveyed by the World Health Organization (WHO) reported disruptions to health services in 2021 [14]. [1] stresses that humanitarian organizations must continue to focus on the distribution of services and access to healthcare to meet the 2030 Sustainable Development Goals [12]. To correct healthcare disruptions, the WHO proposes the use of mobile clinics as a solution for humanitarian healthcare relief [13]. Mobile clinics are an intermittent modality used to improve access to healthcare when permanent health facilities are not available [5, 6]. They consist of vehicles transporting medical equipment and healthcare providers that can offer onsite healthcare services [10]. Mobile clinics are a staple in humanitarian contexts but they have become increasingly visible due to COVID-19 [3], as they are well suited to fill healthcare needs during epidemics [3, 2, 9, 8]. Studies have also shown that mobile clinics allow for prompt response and flexibility because of the ability to change locations [15], and they can be equipped to respond to several health issues [4]. Despite their benefits, mobile clinics present logistical challenges and can be operationally expensive [5]. To exacerbate the challenge of mobile clinic deployments, [16] stresses that humanitarian supply chains are unstable and unpredictable. According to [7], during humanitarian relief deployments, there is a high risk of infrastructure damage as well as a high probability of secondary disasters. To ensure a robust plan for humanitarian relief deployment, instead of using a deterministic approach uncertainty must be considered in the planning stages [7].

This study addresses the need for a decision methodology for mobile clinic deployments that considers the uncertainty faced by practitioners during the tactical planning. It adapts a Prize Collection Problem (PCP) to define the Stochastic Benefit (i.e., prize) Collection Problem (SBCP). We model mobile clinic deployments as an SBCP with a set packing formulation that seeks to maximize the total expected prize collection. Moreover, the objective proposed quantifies in monetary value the benefit offered to beneficiaries through a benefit cost ration (BCR). This is the first study to address mobile clinic deployment planning as an SBCP. To the best of our knowledge, this study is the first to propose a stochastic formulation for the PCP. Furthermore, this study is also the first to consider the impact of uncertainty on the transportation network during mobile clinic deployments for humanitarian relief. We consider three uncertain parameters: travel time, usability of the roads, and access to the locations. No studies have previously evaluated or proposed recourse policies in the context of disaster and humanitarian relief. Whereas, we propose four different recourse policies and evaluate their impact on the total benefits (i.e., prize collected minus cost) and number of locations visited. In addition, we evaluate its performance through different levels of uncertainty. Hence, this study is the first to propose and evaluate the impacts of different recourse policies on the performance of relief efforts. Our approach is sufficiently general to support any prize collection operation, notwithstanding our model is tested on real world data for the deployment of mobile clinics for humanitarian relief through a vaccination campaign.

## 2 Context and Problem Description

We test our methodology in the context of vaccination campaigns for humanitarian relief. Based on information previously shared by field practitioners [11], we define the decision process in mobile clinic deployments affected by uncertainty. First, practitioners design a tactical plan for a determined schedule length to be repeated along the planning horizon. Before applying the tactical plan, practitioners receive new informationregarding conditions that affect the transportation network (e.g., likelihood of landslides, potential war acts). The new information is translated to a numerical value that allows practitioners to decide if and how the tactical plan must be adjusted. Deciding and notifying community visits in advance about mobile clinic visits offers benefits. It ensures community availability and allows the local contact to inform residents and update practitioners. Community assessments help in choosing communities and timing, but these plans may change based on accessibility, travel times, and road conditions. If necessary, practitioners can exclude certain communities or alter visit timings. Route adjustments might involve changing the order of stops or selecting different roads, possibly visiting a subset of the originally planned communities in a new sequence. At the beginning of the planning horizon, the estimated travel time, based on the experts knowledge and distributions, is known and it is assumed that locations are accessible by usable roads. However time, access, and usability are subjected to sources of uncertainty, which can affect the effectiveness and delivery of humanitarian relief (i.e., prize and cost) [7]. Considering uncertainty at the tactical planning level facilitates accounting for fluctuations in costs, impact of estimated parameters, and actions taken to adjust the plan at each implementation.

## 3 Modeling Mobile Clinic Deployments

We model the deployment of mobile clinics under uncertainty in humanitarian contexts as an SBCP with a two-stage stochastic program. The first stage builds the initial tactical plan before the revelation of updated information. During the first stage, communities to be served are selected, as well as what routes will be used and at what time periods communities will be visited . After the first stage decisions have been taken, experts provide information updates that are translated into numerical information on accessibility to the locations, travel times, and usability of the roads. This numerical information is then used to alter the plan through the second stage decisions. Therefore, at the begining of the planning horizon before implementing the plan information is updated and the plan is adjusted. In the present study, we assume that uncertainty revelation can result in a travel time increment but not a decrease, as a reduction in travel time is not considered a negative consequence whereas more time spent on the road rather than providing services is of detrimental consequence for the collection of the prize (i.e., less beneficial for the population). The non-usability of the roads is represented by an infinite travel time for each road affected by external factors and as consequence deemed unusable. The accessibility of the community is not guaranteed during all time periods and it is captured by removing the node representing the community and paths from the network at the corresponding time period, as it is not possible to pass by the location. We propose and evaluate four different recourse policies each captured by different two-stage SBCP models. The select then route policy selects the locations to be visited during the first stage and in the second stage selects the routes and time periods at which each location will be visited. In comparison, the first stage full recourse, simple recourse, and reoptimize per time period policies share the same first stage decisions (i.e., selection, routing, and time period). However, these policies differ in the decisions taken during the second stage (i.e., after the information has been updated. Full recourse allows for a complete reselection for the current routes. Simple recourse is captured by eliminating all routes (i.e., sequence of stops and path segments) that the revelation of uncertainty renders infeasible, Yet, during the simple recourse communities are not rescheduled during a different time period. Finally, reoptimize per time period is captured by adding a constraint that ensures communities are visited on the originally selected time period only. All the proposed models allow decision makers to select communities, schedules, and routes for mobile clinic deployments. In conjunction, these models can be used to evaluate the impact of recourse decisions on the flexibility and consistency offered, as well as the prize collected and costs incurred.

## 4 Conference Presentation

At the conference, we will discuss the problem focusing in particularly on describing the uncertainty, the information revelation sources and process, as well as the possible plan-adjustment strategies. The results and analysis of an extensive experimentation campaign will also be presented. The solution approach will be tested on four different cases for vaccination campaigns in humanitarian contexts using real world data from Indonesia, Iraq, Kenya, and Malawi. We will also test different levels of uncertainty: mild, moderate, and severe. These levels of uncertainty represent different amplitudes in variation of the uncertain parameters. With a BCR technique, introduced in the vaccination literature, we will translate benefit into a monetary prize. The preliminary results show that three out of four models are sensitive to higher BCRs under moderate and severe uncertainty levels. However, the simple recourse policy is not sensitive to different BCR levels. This means that the performance of the simple recourse or a do nothing adjustment plan will not vary based on the scale of the benefit provided. The four policies will be compared against the deterministic model to measure their performance and study the impact of accounting for the adjustment costs. Preliminary results show that the route on second stage policy outperformed other policies both in profit and locations served. This policy offers the most flexibility to practitioners, however it does not allow for a detailed schedule to be announced to the communities. To counter this downside, practitioners can opt to instead use a reroute per time period to offer a higher level of reliability while reaching more communities. On the other hand, opting for a simple recourse policy can offers even higher reliability while achieving similar levels as the reroute per time period policy in terms of profit.

## References


[1] T. Adhanom Ghebreyesus. Foreword of world health statistics 2021. `https://apps.who.int/iris/bitstream/handle/10665/342703/9789240027053-eng.pdf`, 2021.

[2] D. J. Alcendor, P. D. Juarez, P. Matthews-Juarez, S. Simon, C. Nash, K. Lewis, and D. Smoot. Meharry medical college mobile vaccination program: Implications for increasing covid-19 vaccine uptake among minority communities in middle tennessee. *Vaccines*, 10(2):211, 2022.

[3] K. M. Ali. Mobile care in the pandemic. *Bull World Health Organ*, 100:94–95, 2022.

[4] T. Blackwell and M. Bosse. Use of an innovative design mobile hospital in the medical response to hurricane katrina. *Annals of emergency medicine*, 49(5):580–588, 2007.

[5] S. Du Mortier and R. Coninx. *Mobile health units in emergency operations: a methodological approach*. Humanitarian Practice Network, Overseas Development Inst., 2007.

[6] S. D. Du Mortier and R. Coninx. Mobile health units in emergency operations a methodological approach. Technical report, Overseas Development Inst., 2007.

[7] M. C. Hoyos, R. S. Morales, and R. Akhavan-Tabatabaei. Or models with stochastic components in disaster operations management: A literature survey. *Computers & Industrial Engineering*, 82:183–197, 2015.

[8] A. Leibowitz, L. Livaditis, G. Daftary, L. Pelton-Cairns, C. Regis, and E. Taveras. Using mobile clinics to deliver care to difficult-to-reach populations: A covid-19 practice we should keep. *Preventive Medicine Reports*, 24:101551, 2021.

[9] B. S. Levy. Health and peace. *Croatian medical journal*, 43(2):114–116, 2002.

[10] C. R. McGowan, L. Baxter, C. Deola, M. Gayford, C. Marston, R. Cummings, and F. Checchi. Mobile clinics in humanitarian emergencies: a systematic review. *Conflict and Health*, 14(1):4, 2020.

[11] PUI. Personal communication with dr. marwa, 2017.

[12] UN. About the sustainable development goals - united nations sustainable development. `https://www.un.org/sustainabledevelopment/sustainable-development-goals/`, 2015.

[13] WHO. Mobile clinics. `https://www.who.int/emergencies/partners/mobile-clinics`, 2016.

[14] WHO. World health statistics 2021. `https://apps.who.int/iris/bitstream/handle/10665/342703/9789240027053-eng.pdf`, 2021.

[15] N. P. Wray, T. W. Weiss, T. J. Menke, P. J. Gregor, et al. Evaluation of the va mobile clinics demonstration project. *Journal of Healthcare Management*, 44(2):133, 1999.

[16] D. K. Yadav and A. Barve. Analysis of critical success factors of humanitarian supply chain: An application of interpretive structural modeling. *International journal of disaster risk reduction*, 12:213–225, 2015.

# Tactical staffing and workforce scheduling decisions for green last-mile delivery

**Alberto Santini**
Department of Economics and Business,
Universitat Pompeu Fabra, Barcelona, Spain
Email: alberto.santini@upf.edu

**Minakshi Punam Mandal, Claudia Archetti**
Department of Information Systems, Decision Sciences and Statistics,
ESSEC Business School, Cergy, France

## 1 Introduction

A fundamental tactical question arises for logistic operators involved in Last-Mile Delivery (LMD) in the urban environment: how many couriers should they employ? On the one hand, a more extensive workforce is associated with higher staffing costs; on the other hand, using fewer couriers degrades the quality of service or forces the operator to resort to expensive outsourcing options. Demand for home delivery is highly seasonal, further complicating the challenge of choosing the correct workforce size. This extended abstract introduces a decision support system for tactical hiring decisions under demand uncertainty.

The relevance of this research question has been recently highlighted, e.g., in [1], where the authors identified staffing and fleet sizing as needing attention from the operational research community because of the "lack [of] scientific decision support". We fill this gap by considering the problem of a logistic operator delivering parcels throughout the day and facing the tactical problem of sizing its workforce. They can decide to fulfil each delivery with either a fleet of owned vehicles driven by couriers or paying a fee to an outsourcing provider. The logistic operator must then balance the tactical staffing and operational outsourcing decisions.

A central concept in our setting is that of *satellites*, i.e., locations within the city where the couriers start and end their delivery trips. Each satellite is associated with a given portion of the city, called an *area*; areas are further grouped into *regions*. We assume that hiring decisions must be taken at the tactical and regional level, i.e., a courier is hired for a specific region and an extended period. Assignment of couriers to satellites (and, therefore, to areas) happens at the operational level according to the needs of the logistic operator.

The logistic company must decide how many couriers to hire in the mid-to-long term in each region and which area to assign them in the short term to minimise the combined labour costs and expected outsourcing costs. We emphasise that we are concerned with hiring and scheduling the workforce, assuming that the company already owns a fleet of vehicles. Therefore, we do not study the problem of purchasing or leasing vehicles and consider the corresponding costs sunk. Finally, we highlight the stochastic nature of our problem: the decision-maker can estimate the number of deliveries in each area but cannot know this number precisely on the timescale required to make tactical decisions.

### 1.1 Shift stability

A distinguishing characteristic of our approach is that we explicitly incorporate shift stability to reduce the day-to-day variation in couriers' working hours. Shift instability and low wages have been identified as the main criticalities of the modern logistics industry, especially in the last mile. From the point of view of the couriers, unstable shifts cause a sensible decrease in happiness and worsen the work-life balance; moreover, the downsides of erratic shifts particularly affect already vulnerable categories, such as single parents. From the point of view of logistic operators, excessive shift instability causes higher employee turnover and lower performance, producing a net detrimental effect for the firm. Chung [3] studies the impact of variable work schedules and concludes that "despite the common assumption that their use helps firms achieve higher performance by matching the supply of labor to demand fluctuations [...] this study demonstrates otherwise", and "scholars and practitioners should reconsider the general assumption that staffing flexibility helps organisations adapt to uncertain environments".

## 2 Methodology

At a high level, we solve the following optimisation problem:

$$\begin{aligned} \text{min} \quad & \text{Staffing Costs} + \mathbb{E}\big[\text{Outsourcing Costs}\big] \\ \text{subject to} \quad & \text{Staffing Constraints} \\ & \text{Shift Stability Constraints.} \end{aligned}$$

In the following, we analyse each component. **Staffing costs** are deterministic and depend on the number of couriers we hire. **Outsourcing costs** are stochastic because demand is unknown in advance; they depend on the number of couriers we hire. We assume that all parcels must be delivered, and if we do not have enough couriers to perform all deliveries, we must outsource the remaining ones. There are two challenges to estimating outsourcing costs. First, their stochastic nature, which we tackle using a scenario-based approach and computing the expected value as the empirical mean over all scenarios. Second, for each scenario, we must determine how many parcels the hired couriers can deliver and, therefore, how many must be outsourced. To this end, we should solve many Capacitated Vehicle Routing Problems. For our tactical problem, however, such an approach would be impractical (we cannot know precise customer locations in advance even in a scenario-based model) and unnecessary (we need not know each courier's route, only how many parcels they can deliver). Instead, we propose a procedure using Figliozzi's approximation formula [2] to estimate how many parcels can be delivered by a given number of couriers. **Staffing constraints** impose global and regional-level upper bounds on the number of couriers we can hire. They derive, respectively, from the staffing budget and the number of vehicles available in each region. **Shift stability constraints** offer different degrees of flexibility when creating courier shifts. A shift is a set of consecutive periods such that if a courier works during one of them, he must work during all of them. We propose three alternative stability constraints: Flexible, Partially Flexible and Fixed. Flexible shifts can start at any time during the day; potentially, each shift could start at a different time. With Partially Flexible shifts, we limit the number of the different shift start times, but we still leave flexibility as to when the chosen start times occur. Finally, Fixed shifts can only start at two times during the day (the morning shift starts at 8 AM and the afternoon shift starts at 2 PM). Figure 1 shows examples of the shifts generated using each of the three types of stability constraints.

## 3 Conclusions

The main computational takeaways from preliminary experiments are two: a limited number of scenarios (thirty in our case) is sufficient to obtain a good approximation of the stochastic costs, and the overall optimisation procedure is extremely fast, in the order of tens of seconds. These performances allow us to conduct a thorough sensitivity analysis to understand which instance characteristics affect the results the most. The experiments are based on real-world demographic data from Paris, Lyon, Frankfurt and Berlin. We vary the demand's pattern and magnitude, the bounds in the staffing constraints, and the relative costs of hiring couriers and outsourcing parcels. The main results are as follows.

First, using completely fixed shifts that start at two predetermined times during the day results in significantly higher costs. Over all instances, the average per-parcel cost obtained using fixed shifts is 9.36% higher than the one obtained using extremely flexible schedules, in which couriers can be called into (and out of) at each two-hour period. On the other hand, a partially flexible model that uses two shifts—but allows their start times to be a decision variable—incurs costs that are only 1.89% higher than those obtained with

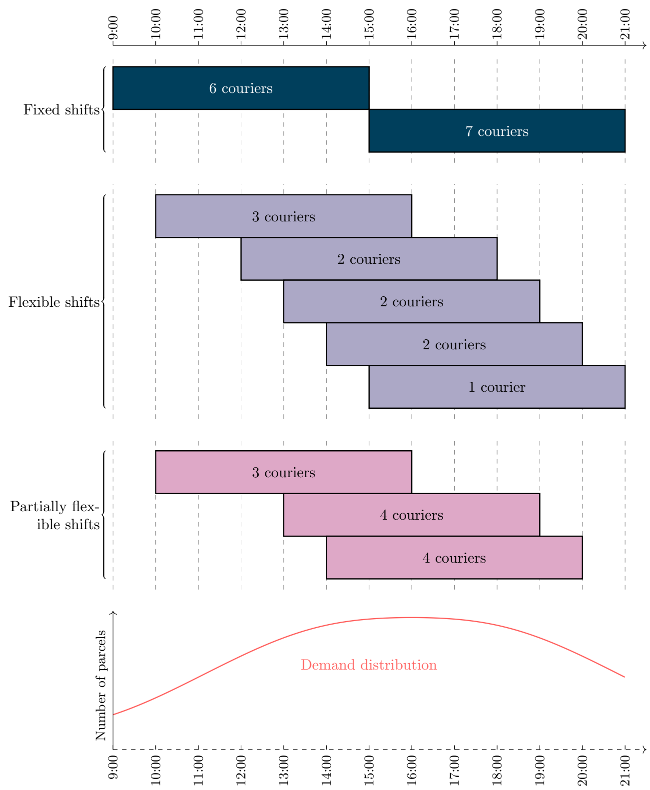



Figure 1: **Shift types.** Example of fixed (blue), flexible (purple) and partially flexible (pink) shifts for a 12-hour working day. The demand distribution at the bottom shows that the afternoon is busier than the morning.

extremely flexible schedules.

Second, the advantage of flexible schedules compared to fixed ones is more significant when the company can hire a large workforce and has a large fleet. When the company cannot hire many couriers (because it does not have enough vehicles to operate or because market conditions make labour scarce), flexible and fixed schedules yield almost the same costs. The conclusion is that stable shifts are a viable strategy for a company that has trouble finding couriers. On the one hand, stable shifts do not increase costs significantly, and on the other hand, they provide better working conditions that help attract potential employees.

## References


[1] Boysen, N., Fedtke, S., Schwerdfeger, S., 2021. Last-mile delivery concepts: a survey from an operational research perspective. OR Spectrum 43, 1–58. doi:`10.1007/s00291-020-00607-8`.

[2] Figliozzi, M.A., 2008. Planning approximations to the average length of vehicle routing problems with varying customer demands and routing constraints. Transportation Research Record 2089, 1–8. doi:`10.3141/2080-01`.

[3] Chung, H., 2022. Variable work schedules, unit-level turnover, and performance before and during the COVID-19 pandemic. Journal of Applied Psychology 107, 515–532. doi:`10.1037/apl0001006`.

# The impact of profit sharing on collaborative vehicle routing with dynamic request acceptance

**Yannick Oskar Scherr**
Department of Operations, Energy, and Environmental Management,
University of Klagenfurt, Klagenfurt, Austria, and
Department of Business Decisions and Analytics,
University of Vienna, Vienna, Austria
Email: yannick.scherr@univie.ac.at

**Margaretha Gansterer**
Department of Operations, Energy, and Environmental Management,
University of Klagenfurt, Klagenfurt, Austria

**Richard F. Hartl**
Department of Business Decisions and Analytics,
University of Vienna, Vienna, Austria

## 1 Motivation

Fast response times by logistics carriers are crucial to satisfy customers but committing to fulfill transportation requests at a time when future requests are unknown can lead to inefficient utilization of resources. These efficiency losses not only impact the individual carrier's profits but also society and the environment as a whole, e.g., in terms of traffic and emissions. Collaboration between carriers represents a measure to improve the utilization of resources [1]. Available communication and data exchange capabilities support the exchange of requests between carriers via digital platforms with negligible transaction costs, while combinatorial auctions enable horizontal collaboration in a decentralized way without requiring carriers to reveal sensitive information.

We investigate the operational problem of a less-than-truckload carrier receiving customer requests for transporting goods from pickup to delivery locations. Incoming requests must be answered immediately and are fulfilled on the next day using a fleet of vehicles. After the request acceptance phase and before the fulfillment, the carrier can trade requests with other collaborating carriers in a combinatorial auction. The focus of research on horizontal collaboration in vehicle routing so far has been on static and deterministic problem settings that do not consider customer request acceptance decisions. On the other hand, there is well established literature on the related fulfillment problems, i.e., pickup and delivery problems and their dynamic and stochastic versions, that ignore collaboration. The dynamic and collaborative problem that we consider in this work has been recently introduced in [2], in which a Markov decision process (MDP) model and heuristic solution approaches have been proposed.

We study a collaborative vehicle routing problem with dynamic request acceptance to provide the following contributions. First, we propose a two-step policy for making dynamic acceptance decisions that take into account the strategic rejection of requests that could be fulfilled. Also, overbooking, i.e., accepting requests that cannot be served before the auction takes place, is enabled. We use this policy that can be tuned with regard to its strictness in rejecting and overbooking requests to evaluate different acceptance strategies of carriers in the collaboration. Second, we propose a profit sharing mechanism that can be adjusted with regard to achieving equal auction outcomes for each collaborating carrier. Numerical experiments on symmetric (all carriers use the same strategy) and asymmetric (carriers use different strategies) settings allow us to gain insights into the stability of symmetric acceptance strategies and potential incentives to deviate from them under the influence of different profit sharing mechanisms.

## 2 Problem Description

The planning horizon of this problem is divided into two subsequent phases with the first one concerning the dynamic acceptance of requests and the second one concerning the exchange of requests. In the request acceptance phase, stochastic customer requests featuring a pickup location, delivery location, load, and revenue arrive over time. The carrier must decide immediately on accepting or rejecting an incoming request. While making those sequential decisions, the carrier takes into account the fulfillment of the accepted requests on the next day. For the fulfillment of the pickup and delivery requests, the carrier uses a fleet of vehicles, each with a limited route duration and load capacity. For accepted requests that cannot be served by the carrier, a penalty fee needs to be paid. The request acceptance market phase ends at a cutoff time before the auction takes place.

The second phase comprises the combinatorial auction that follows the 5-phase procedure proposed by [3]. First, each carrier selects the requests to submit to the auction. Based on all selected requests, the auctioneer or platform offers bundles containing subsets of those requests. The carriers submit bids for all bundles according to their preferences. By solving the winner determination problem, the auctioneer redistributes the bundles to carriers to minimize the total fulfillment costs according to the bids. Finally, the collaboration savings are shared among the individual carriers by determining payments to or from each carrier using a profit sharing mechanism. We investigate mechanisms that can be configured using a parameter $\rho$, with $0 \leq \rho \leq 1$, adjusting the emphasis of the profit sharing towards not sharing profits at all after the redistribution ($\rho = 0$) or sharing profits in an egalitarian way ($\rho = 1$). Equation 1 is used for determining payments $\psi_\gamma$ for each carrier $\gamma \in \Gamma$ based on its fulfillment costs before the auction ($\beta_\gamma$), fulfillment costs after the redistribution of requests ($\alpha_\gamma$), and the parameter $\rho$. The collaboration savings of a carrier are depicted on the left-hand side for illustration purposes.

$$\beta_\gamma - \alpha_\gamma + \psi_\gamma = \rho \frac{\sum_{\gamma' \in \Gamma}(\beta_{\gamma'} - \alpha_{\gamma'})}{|\Gamma|} + (1 - \rho)(\beta_\gamma - \alpha_\gamma) \qquad \forall \gamma \in \Gamma \tag{1}$$

Overall, the objective of each carrier is to maximize its profit by accepting and exchanging requests. The profit is composed of the revenue collected by accepting requests minus the fulfillment costs for routing and outsourcing.

## 3 Solution Approach

Each carrier's optimization problem can be modeled as an MDP covering all sequential decisions. Due to the curses of dimensionality, the MDP cannot be solved to optimality, e.g., using dynamic programming. Hence, heuristic policies are considered. For updating a tentative fulfillment plan with efficient and balanced routes, an insertion heuristic allows to quickly estimate the marginal costs of a request for supporting the acceptance, selection, and bidding decisions. For the carriers' decisions in the auction, we use strategies commonly assumed in collaborative routing literature. Carriers select the requests with the highest marginal costs to trade in the auction and all bidding decisions are made truthfully regarding the marginal costs of fulfilling the requests of a bundle.

For deciding on the acceptance or rejection of an incoming request, we propose a two-step policy. In the first step, it is decided whether to accept the request and insert it into one of the tentative routes based on the request's revenue, the marginal insertion costs, and the route's slack. If this decision is negative, it is decided in the second step whether the request should be accepted and added to the set of overbooked requests based on its revenue, the penalty fee, and an estimate for the request's attractiveness in the auction. Negative decisions in both steps result in the request being rejected. In each of the two steps, a separate linear threshold function is used that can easily be parameterized. This allows for conducting interpretable experiments that depict carriers' potential acceptance strategies with regards to strategic rejection (step 1) and overbooking (step 2).

## 4 Results and Conclusions

We conduct numerical experiments to evaluate the impact of strategic rejection and overbooking using different parameter settings for the two-step acceptance policy. The instances vary in the degree of overlap between the carriers' service areas, the request revenue pattern, and the profit sharing mechanism. Results show that the most profitable symmetric strategies are stricter in the rejection of requests than it would be required for collecting the largest revenue, as a balance with the fulfillment costs is found. The collaboration savings are smaller when strategic rejection is allowed compared to when all requests must be accepted that can be fulfilled, but allowing for overbooking of requests increases the collaboration savings. After identifying the best symmetric strategies, we analyze their stability and potential incentives for carriers to deviate from them given different profit sharing mechanisms (using different values of $\rho$). Table 1 provides average results for one set of instances, showing the impact on a carrier's profit that deviates from the best symmetric strategy in different ways.

| Deviating strategy | $\rho = 0$ | $\rho = 1/3$ | $\rho = 2/3$ | $\rho = 1$ |
|---|---|---|---|---|
| Rejecting less | 0.96% | -0.28% | -1.51% | -2.75% |
| Rejecting more | -7.65% | -7.09% | -6.53% | -5.97% |
| Disregarding overbooking | -18.19% | -10.49% | -2.80% | 4.90% |

Table 1: Impact of deviating strategies on carrier profit with different profit sharing.

We observe that a more egalitarian profit sharing prevents deviations by carriers to be less strict in the rejection of requests but also gives incentives to deviate from the best overbooking strategy. This could result in all carriers refraining from overbooking and to reduce total profit. Profit sharing that combines the individual outcome of the redistribution with an adjustment towards more equality ($\rho = 1/3$, $\rho = 2/3$) can stabilize strategies. Further results will cover deeper insights on the impact of profit sharing on carriers' strategies.

## References


[1] M. Grazia Speranza, "Trends in transportation and logistics", *European Journal of Operational Research* 3, 830–836, 2018.

[2] Y.O. Scherr, M. Gansterer and R.F. Hartl, "Request acceptance with overbooking in dynamic and collaborative vehicle routing", *European Journal of Operational Research*, to appear.

[3] S. Berger and C. Bierwirth, "Solutions to the request reassignment problem in collaborative carrier networks", *Transportation Research Part E: Logistics and Transportation Review* 46(5), 627–638, 2010.

# Optimizing Electric Vehicle Charger Locations for Ride-hailing Services through Discrete Simulation-based Optimization

**Tommaso Schettini**
HEC Montréal, École de technologie supérieure,
Email: tommaso.schettini@hec.ca

**Fausto Errico**
École de technologie supérieure

**Jorge E. Mendoza**
HEC Montréal

**Carolina Osorio**
HEC Montréal

## 1 Introduction

In recent years, the use of electric vehicles (EVs) has been on a steady upward trend and the technology has reached a sufficient level of maturity for a wide variety of use-cases, ranging from private transportation, to ride-hailing services. Consequently, the problem of planning the installation or extension of EV charging infrastructure has attracted considerable research attention in the scientific community [2]. One interesting thread in this literature relates to the design of charging networks for supporting a privately owned fleet of specialized EVs (such as taxis [1]). Following this thread, we consider the problem of planning the installation of the charging infrastructure for a fleet of autonomous and electric ride-hailing vehicles. This problem setting is based on [3] who proposed a highly effective dynamic control strategy for an autonomous fleet of ride-hailing vehicles. In their strategy, the charging infrastructure is used in two ways, to allow vehicles to reposition and wait, and to allow vehicles to recharge. However, [3] assume that the available charging infrastructure corresponds to the network of public chargers, and each charger has unlimited charging capacity. In practice, this is rarely the case and a private operator would opt to install private chargers to support their operations.

In this paper, we consider the problem of optimizing the charging and parking infrastructure for supporting an autonomous fleet of ride-hailing vehicles with the objective of maximizing its expected operational profit. The dynamic and stochastic nature of ride-hailing services makes it challenging to measure the expected operational profit of a particular charger configuration. To overcome this issue, we make use of a simulator to estimate the expected operational profit associated with each solution. In the simulation, we assume that the fleet is operated by a single profit-maximizing agent, which is responsible for optimizing the behavior of the fleet given the infrastructure configuration and the current state of the system. Furthermore, we assume that charging stations installed have a limited capacity, and operate using a FIFO policy.

When formulated this way, the problem becomes a discrete simulation-based optimization (DSO) problem. We refer to this problem as the Charger and Parking Location Problem (CPLP). Let $\mathcal{I}$ denote the set of candidate locations in which chargers and parking stations may be installed. The CPLP consists of locating $c$ charging and $r$ parking stations from $\mathcal{I}$, respectively. We use variable $\theta_i$ to describe weather charging station $i$ is installed and $\gamma_i$ to describe weather parking station $i$ is installed. We denote by $\mathcal{P}(\theta, \gamma)$ the expected profit of the fleet using the infrastructure described by variables $\theta_i$ and $\gamma_i$, $\forall i \in \mathcal{I}$. Thus, the optimization problem is formulated as follows.

$$\max \mathcal{P}(\theta, \gamma) \tag{1}$$

$$\sum_{i \in \mathcal{I}} \theta_i = c \tag{2}$$

$$\sum_{i \in \mathcal{I}} \gamma_i = r \tag{3}$$

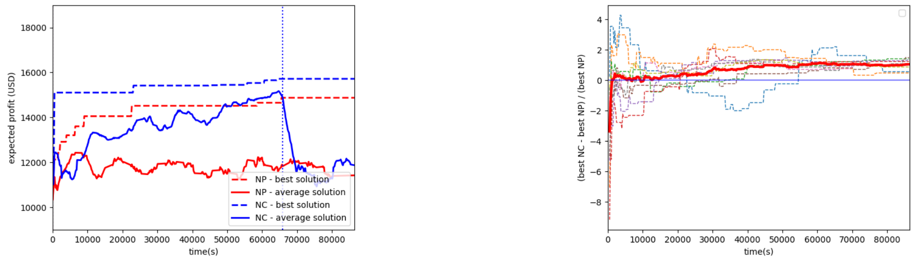
$$\theta_i \in \{0,1\}, \gamma_i \in \{0,1\} \qquad \forall i \in \mathcal{C}. \tag{4}$$

## 2 Methodology

To solve the CPLP, we implemented a novel DSO method based on the classical algorithm of Nested Partitions [4] (NP), which is a popular and versatile algorithm for DSO problems. NP employs an *adaptive random search* strategy that initially evaluates solutions uniformly sampled from the feasible space. The strategy then randomly samples and evaluates additional solutions, using information from previously evaluated solutions to bias the sampling procedure.

NP performs its search by concentrating the sampling of solutions on a gradually narrowing *promising* region, which is identified by partitioning the feasible space in subregions and iteratively selecting the promising region from a set of candidate regions. Hence, the algorithm alternates between partitioning, and sampling solutions from specific regions of the feasible space.

The design of these two components, partitioning and sampling, play a vital role in determining the effectiveness of the method. In our case, we have at our disposal a formulation of the problem (1)–(4). While this formulation cannot be directly input into a solver to compute an optimal solution due to the fact that (1) is computed using a simulator, it reveals useful information about the problem's structure. Therefore, in this paper, we propose a DSO algorithm that uses the mathematical structure of the feasible space of the CPLP to construct auxiliary integer linear programming formulations to inform partitioning and sampling.




(a) Evolution of the average evaluated solutions (b) Comparison of the best found solutions

Figure 1: Comparison of evaluated solutions by NP and NC

To partition the feasible space of the problem, we use a clustering strategy. Specifically, we use the K-means algorithm to cluster a set of randomly sampled solutions and use the resulting clusters to partition the feasible space. For our sampling strategy, we generate and solve an auxiliary formulation of the problem. Specifically, we augment the 0-1 linear programming formulation of the feasible space by including the constraints that define the promising region and a linear randomized objective function. We then solve this formulation to sample a random point from the promising region. Additionally, we inject geographical information into the K-means. We do so to ensure that the clustering will aggregate together solutions with the same general distribution of chargers. We achieve this by projecting the solutions onto an auxiliary space through the use of linear features that encode the general position of each charger. For the sake of brevity, we refer to the specific version of NP implementing the proposed partitioning and sampling approaches as Nested Clusters (NC).

## 3 Case study and computational results

To illustrate the effectiveness of our method, we consider a case study in the context of New York City. We based our simulation on the one used in [3], which we expanded to account for charging stations with limited capacity, and a FIFO policy. The parameters of the simulation were based on [3]. We generated 9 instances, considering different numbers of CSs (1,2, and 4) to be placed and different fleet sizes (20, 30, and 40 vehicles). We benchmark our approach in two ways: firstly, we compare NC to NP; secondly, we compare our strategy to a solution to the problem obtained through a non-simulation-based heuristic. In our benchmark with NP, we observe that NC is able to more effectively capture the structure of the problem, and quickly converge to an area of the search space populated with high-quality solutions. In Figure 1a we report an example run of the two algorithms. In the figure, we report on the x-axis the clock time in seconds, indicating each algorithm's progression, while the y-axis reflects the objective value (expressed in USD) of the solutions evaluated. We use a dashed line to indicate the objective function value of the best solution evaluated over the course of the algorithm's runtime, and a

| # Vehicles \ # CSs | 1 | 2 | 4 |
|---|---|---|---|
| 20 | 10.20 | 17.61 | 3.12 |
| 30 | 11.35 | 12.86 | 3.84 |
| 40 | 4.95 | 7.75 | 2.63 |

Table 1: Comparison of the best NC solutions to the non-simulation-based heuristic solution

solid line to show the rolling average of the last 20 solutions from the promising region evaluated by each algorithm. From the graph, we see that NC is able to identify an area of high-quality solutions and concentrate sampling efforts in that area, leading to solutions of higher quality being evaluated. This is also reflected in the evolution of the best-evaluated solution by the two algorithms. Due to space limitations, we could not report all tested instances, however, we remark that we achieved similar results on all other instances. In Figure 1b we report a comparison of the best solutions evaluated by the algorithm on each tested instance. In our testing, we observed an improvement of 1.05% in the best solution achieved by NC compared to NP.

Finally, in Table 1, we report the gap between the simulated objective of the final solutions obtained by NC and a solution achieved through a non-simulation-based heuristic. Specifically, we used a covering heuristic, that given a maximum service range (which is determined based on the speed of the vehicles and the willingness of the users to wait), maximizes the total demand that is in-range of chargers and repositioning stations. As we observe, the DSO strategy is able to significantly outperform the non-simulation-based solutions. We note that when the number of vehicles is set to 40, the benefits of embedding the simulator into the optimization process decrease. This is because a fleet of 40 vehicles is oversized compared to the levels of demand observed in the simulation. In that case, maximizing the demand in the proximity of the chargers is an effective strategy.

## References

[1] J. Asamer, M. Reinthaler, M. Ruthmair, M. Straub, and J. Puchinger. Optimizing charging station locations for urban taxi providers. *Transportation Research Part A: Policy and Practice*, 85:233–246, 2016.

[2] M. Kchaou-Boujelben. Charging station location problem: A comprehensive review on models and solution approaches. *Transportation Research Part C: Emerging Technologies*, 132:103376, 2021.

[3] N. D. Kullman, M. Cousineau, J. C. Goodson, and J. E. Mendoza. Dynamic ride-hailing with electric vehicles. *Transportation Science*, 56(3):775–794, 2022.

[4] L. Shi and S. Ólafsson. Nested partitions method for global optimization. *Operations research*, 48(3):390–407, 2000.

# Impact of port volume commitment in container routing while considering real-life constraints

**Marco Silva**
Department of Manufacturing Sciences and Logistics (SFL),
Ecole des Mines Saint-Etienne, Gardanne, France
Email: marco.costasilva@emse.fr

**Nabil Absi**
Department of Manufacturing Sciences and Logistics (SFL),
Ecole des Mines Saint-Etienne, Gardanne, France

**Dominique Feillet**
Department of Manufacturing Sciences and Logistics (SFL),
Ecole des Mines Saint-Etienne, Gardanne, France

**Azzedine Cheref**
CMA CGM Transport Maritime, Marseille, France

**Pauline Mamour**
CMA CGM Transport Maritime, Marseille, France

**Hamza Ben Ticha**
CMA CGM Transport Maritime, Marseille, France

## 1 Introduction

In this work, we introduce a container routing operations model for liner services in maritime transportation as a decision-support tool to assess the profitability of a liner shipping network while considering the impact of supply chain collaboration with the use of port volume commitment agreements and other real-life business constraints.

Container routing involves determining how containers are transported from their origins to their destinations using different liner service routes. Liner shipping companies publish regular frequency round-trip service routes with a fixed sequence of ports at a defined schedule, to attract cargo.

A port volume commitment agreement involves liner companies committing to port authorities regarding the minimum amount of port handling movements during a fixed period of time. In return, liner companies benefit from competitive pricing that will influence their container route selection strategy.

Prior research studies have explored different facets of the container routing problem. A comprehensive survey discussing the use of operations research methods and introducing the main challenges associated with containership routing is available in [1]. The authors emphasize that research on containership routing lags behind practice, especially given the fast growth of the container shipping sector and advancements in operations research and computer technology.

Numerous studies have focused on introducing certain main realistic liner service characteristics that can impact operations costs into problem formulations. These characteristics include container transshipment, cabotage rules, cargo transit time limits, and empty container repositioning.

Container transshipment enables the consolidation of containers into large vessels and plays an important role in container routing ([2], [3]). These should also consider the service level of the origin-to-destination transit time and maritime cabotage rules. These characteristics have been studied, for instance, in the work of [2]. The repositioning of empty containers for liner shipping companies aims to optimize the repositioning process while considering the cost and availability of empty containers. Empty containers accumulate in import-intensive regions due to a significant imbalance in world trade [4].

The study's main contribution lies in integrating various realistic constraints into a single-problem framework, previously addressed independently in other studies. These constraints encompass transshipment, cabotage rules, transit time, and empty container repositioning. In addition, we introduce new and challenging constraints, such as considering different capacities of containers (vessel slots, reefer outlets, weights, and dangerous cargo limits), port capacity limits for loading, unloading,transshipment, cargo delivery priority with penalties, soft transit time limits with penalties, and cargo commitment agreements with port authorities with penalties. The impact of volume agreements has been studied from the perspective of the port authority management, as can be seen in [5]. To our knowledge, our study represents the first attempt to include the analysis of port volume commitment agreements in the formulation of a cargo routing problem.

## 2 Methodology

Our research focuses on deriving an exact solution method for large-sized real instances. The aim is to analyzing the impact of integrating real-life constraints on the solution of our problem. The goal is also to analyze not only solutions dealing with alternatives on how to route cargo but also solutions dealing with the alternatives of different liner service routes. For instance, adding ports with volumes committed to existing routes can facilitate the achievement of the commitments. To accomplish this, we develop a column generation approach and leverage customized dynamic programming techniques, focusing on extended dominance rules, to accelerate the generation of origin-destination container paths.

A key decision involves modeling the liner service network. We adopt a model of a space-time graph model, as depicted in Figure 1. There we can verify that space dimension allows complex routes: multiple legs for the same port. The time dimension allows regular (weekly = 7 days) services. Port legs in continuous red and transshipment legs in dashed violet allow for accurate dimensioning of transit time and costs between ports.

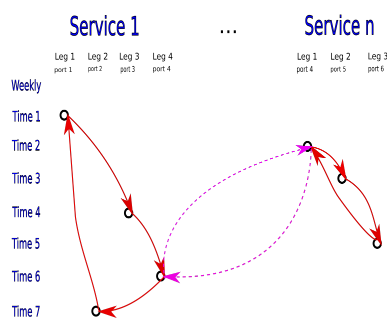



Figure 1: Space-Time Graph

This graph allows us to model our problem as a MILP using an extended formulation. The formulation we base our development is given by:

$$\min \sum_{k\in K,\, p\in P_k} c_{kp} x_{kp} \tag{1}$$

$$s.t. \quad \sum_{p\in P_k} x_{kp} = q_k \qquad \forall k \in K \tag{2}$$

$$\sum_{k\in K,\, p\in P_k} a^e_{kp} x_{kp} < capteuvessel^e \qquad \forall e \in A_v \tag{3}$$

$$\sum_{k\in K,\, p\in P_k} a^e_{kp} u_k x_{kp} < capunittrans^e \qquad \forall e \in A_t \tag{4}$$

$$\sum_{k\in K,\, p\in P_k} \sum_{e^p\equiv dst(e^p_o)=dst(e)\ v\ src(e^p_d)=dst(e)\ v\ node(e^p_t)=dst(e)} a^{e^p}_{kp} u_k x_{kp} < capunitport^e \qquad \forall e \in A_v \tag{5}$$

$$\sum_{k\in K,\, p\in P_k} a^e_{kp} w_k x_{kp} < capweightvessel^e \qquad \forall e \in A_v \tag{6}$$

$$\sum_{\substack{k\in K \\ if\ k\ Reefer}, p\in P_k} a^e_{kp} u_k x_{kp} < capplugvessel^e \qquad \forall e \in A_v \tag{7}$$

$$x_{kp} \geq 0, \quad \forall k \in K, p \in P_k, \tag{8}$$

where $x_{kp}$ is a variable that represents flow of cargo $k \in K$ through path $p \in P_k$. We minimize total maritime costs that include only port load and unloading fees plus penalties when defined (not explicit in the base formulation). Constraints (2) guarantee that cargo demands are satisfied and constraints (3)-(7) guarantee that different capacities (TEU, weight and reefer outlets for vessels, port transshipment, load, and unloading) are respected.

## 3 Results

We assess the underlying assumptions and their implications through extensive computational experiments. We show how less accurate or not integrated model assumptions can result in the design of suboptimal container routing plans.

In particular, we develop specialized visual aids (graphs) for Pareto-optimal analysis. These graphs enable decision-makers to determine the best set-up for pricing and penalties for their port volume commitment agreements.

## 4 Conclusion

We present a maritime cargo routing model that not only allows users to establish optimal cargo routing paths but also to exploit results to define the best set-up for pricing and penalties included in their suppliers' and customers' contracts, including port volume commitment contracts. We privilege practice by introducing real-life business constraints. In parallel, we developed a tailored column generation based approach with a focus on performance for solving large-scale real-life instances.

## References


[1] Q. Meng, S. Wang, H. Andersson, and K. Thun, "Containership Routing and Scheduling in Liner Shipping: Overview and Future Research Directions", Transportation Science, 48(2), 265-280, 2014

[2] S. Wang, Q. Meng, and Z. Sun, "Container routing in liner shipping", Transportation Research Prt E, 49(1), 1-7, 2013

[3] D. Koza, "Liner shipping service scheduling and cargo allocation", European Journal of Operational Research, 275(3), 897-915, 2019

[4] B. Brouer, D. Pisinger, and S. Spoorendonk, "Liner shipping cargo allocation with repositioning of empty containers", Information Systems and Operational Research,49(2), 109-124, 2011

[5] D. Syahchari, E. Kuncoro, H. Saroso, and D. Sudrajat, "Effect of Supply Chain Collaboration and Service Stakeholder Commitment on Dry Port Firm Performance", ICCMB'21, 89-93, 2021

# A Matheuristic for the Grey Zone 2E-VRP with Covering Options, Multi-trip and Synchronization

**Edgar Ricardo Silva Russi**
Computer Science & Digital Society Laboratory (LIST3N)
**Nacima Labadie**
Computer Science & Digital Society Laboratory (LIST3N)
**Caroline Prodhon**
Computer Science & Digital Society Laboratory (LIST3N)
Université de Technologie de Troyes, Troyes, France
Email: edgar.silva_russi@utt.fr

## 1 Introduction

This paper introduces The Grey Zone Two-Echelon Vehicle Routing Problem with Covering Options, Multi-trip and Synchronization for Last-mile Deliveries (2E-MTVRPTW-SS-GZ-CO). Unlike the classical Two-echelon Vehicle Routing Problem (2E-VRP), customers can be served through either Customer-to-parcel (C2P) stations or home delivery in a given time window. Moreover, we consider satellites as possible first-echelon C2P stations and grey zone customers at city limits which can be served by both an internal combustion vehicle (ICEV) or an alternative fuel vehicle (AFV) such as in [1]. The contributions of this work are multiple: First, the problem is formally defined in Section 3. In Section 4, our multi-step GRASP-based matheuristic is presented. Finally, results and conclusions are given in Section 5.

## 2 Literature Review

In 2E-VRP distribution models, to coordinate the arrival of the first and second echelon vehicles, two types of synchronization emerge: Exact synchronization which is typically used when intermediate stations do not have storage [3][4][5][6], and synchronization with precedence which is used when storage is possible at these stations [7][8][9]. Our model incorporates exact synchronization by limiting exchange times to a specific fixed value as in [2]. More information on synchronization in VRP can be found in the recent study [10]. Moreover, a second important aspect of our model is the inclusion of coverage options ( i.e.

customer-to-parcel C2P stations) in densely populated areas in order to reduce last-mile distribution costs. This concept has been previously explored in 2E-VRP schemes [11] [12][13] indicating that overall operating costs can be reduced when all customers can be served by both delivery options, due to the greater flexibility this offers in the model. In our particular case, the coverage options are not limited to the second echelon but are included in both. More detailed information on the use of C2P stations in VRP schemes can be found in [14]. To the best of our knowledge, there is no model that takes into account coverage options as C2P stations on both levels, exact synchronization, multiple trips on the second level, and grey zones simultaneously. This new problem denoted 2E-MTVRPTW-SS-GZ-CO is addressed as an innovative urban delivery scheme as presented below.

## 3 Problem Description

The proposed problem considers two homogeneous fleets of vehicles and aims to minimise the total distribution cost depicted in Eq (1). ICEVs and AFVs start and end their routes at their respective depot. ICEVs deliver the goods from the first-echelon depot, where the entire stock of goods is located at the beginning of the time horizon, to the first-level customers and the satellites. These last are also used as C2P stations where first-level customers can pick up their parcels directly. The second-echelon AFVs start from their city-center depot without any goods so they must immediately meet ICEVs at the satellites to start their deliveries. Thus, as the vehicles have to meet at a specific location at a specific time frame, spatial and temporal exact synchronization is ensured so that the transfer of the goods is as fast as possible and both vehicles can continue their deliveries. Due to the costs associated with the use of vehicles and satellites, the waiting time at satellites is limited to a specific value and is minimized in the objective function (Eq. 1) of the MILP model. Furthermore, AFVs can perform multiple trips under the condition that they must be empty every time they arrive at a satellite; a similar assumption is made in [4], [2]. Hence, once an AFV has finished its deliveries it can rejoin an ICEV at a satellite to retrieve new goods and continue a new route. Likewise, the same satellite can be used by different AFVs. In addition, customers located on the city's borders are considered grey zone customers and can be served by any vehicle.

$$\begin{aligned} min \sum_{i \in V^{all}} \sum_{j \in V^{all}} \sum_{k \in F} \left[ (Time_{i,j} + ST_i) CT_k + (Dis_{i,j} CD_k) \right] X_{i,j,k} + \\ \sum_{i \in V^{all}} \sum_{k \in F} W_{i,k} CT_k + \sum_{d \in V^D} \sum_{j \in v^{all}} \sum_{k \in F} X_{d,j,k} FC_k + \sum_{i \in C2P} Y_i CY_i \end{aligned} \tag{1}$$

Eq (1): $T_{i,k}$ defines the arrival time of vehicle $k$ to node $i$; $W_{i,k}$ accounts the waiting time of vehicle $k$ at node $i$; $U_{i,k}$ represents the load of vehicle $k$ after serving node $i$; $X_{i,j,k}$ is equal to one if vehicle $k$ travels from node $i$ to node $j$; $S_{i,l}$ is equal to one if C2P client $i$ is served by C2P station $l$; and $Y_l$ is equal to one if C2P station $l$ is opened; $FC_k$, $CD_k$, $CT_k$ and $CY_i$ denote the fixed cost of vehicles, the cost per distance, the vehicle operating cost per hour and the cost of opening satellites and $C2P$ stations respectively.$Time_{i,j}$ and $Dist_{i,j}$ defines the travel time and the distance from node $i$ to $j$; finally, $ST_i$ defines the service time needed on node $i$.

## 4 Multi-step GRASP-Based matheuristic

To solve this problem we propose a multi-step matheuristic based on a Greedy Randomized Adaptive Search Procedure (GRASP) using two types of greedy heuristics and efficient local search movements (Cross-Exchange, inter and intra-route swap, relocate and 2-opt). A learning process enables one of the two construction heuristics to be favored according to its performance on any given instance, reaching better results. To build a solution for our complex problem, this last is decomposed into different subproblems called steps within the matheuristic framework.

**Step 1: C2P Allocation:** We employ a Mixed-Integer Linear Programming (MILP) approach to determine the second-level C2P stations to open and the allocation of customers to these stations. This decision-making process takes into account various constraints regarding the stations' capacities and covering radius.

**Step 2: Creation of the second level routes:** For this, one of the two construction heuristics is selected. Subsequently, an initial stage of local search is performed.

**Step 3: Satellite Allocation and Multi-Trip:** A MILP approach is executed for the satellite allocation of first level C2P customers and second level routes. Subsequently, a fusion heuristic based on Clark and Wright's savings algorithm is used to merge different routes.

**Step 4: Creation of the first level routes:** To generate more flexibility while respecting the exact synchronization constraints, the satellites are duplicated $n$ times and the synchronization constraints are relaxed. Once the routes are created, the dummy satellites are removed with respect to their time windows. Then, the real satellites with the exact synchronization constraints are reintroduced again by an insertion heuristic. During this step multiple stages of local search are applied.

**Step 5: Solution updating:** After generating the whole solution and in order to integrate the waiting times caused by synchronization, a complete update is made on the solution data structure. To achieve this, we implement a Gantt-based algorithm that generates an ordered list of nodes to update. This list ensures that precedences among nodes are maintained.

## 5 Results and conclusions

The results of our experiments, as presented in Table (1), reveal that employing exact synchronization constraints instead of precedence constraints has only a negligible impact on the average solution cost. In our research, the utilization of exact synchronization, where satellites do not have storage, results in a 0.09% increase on the average solution cost compared to using precedence constraints without computing storage costs (SC). Nevertheless, our findings emphasize the impact of calculating these storage costs when favoring a strategy with precedence constraints over one with exact synchronization. In our results, the average storage time (383.053) significantly surpasses the average waiting time (46.035) by more than eightfold, leading to substantial storage costs that could elevate the solution cost in contrast to the distribution policy employing exact synchronization with waiting costs.

Table 1: 2E-MTVRPTW-SS-GZ-CO results, Synchronization and Grey Zone (GZ) impact

| Instance | Model Results | Prec. Syn. without SC | Gap (%) | Prec Syn. With SC | Gap (%) | All GZC to 1st | Gap (%) | All GZC to 2nd | Gap (%) |
|---|---|---|---|---|---|---|---|---|---|
| n100-c1 | 1037.23 | 993.586 | -4.21% | 1037.11 | -0.01% | 1053.45 | 1.56% | 1111.35 | 7.15% |
| n100-c2 | 1112.83 | 1092.857 | -1.79% | 1145.56 | 2.94% | 1152.80 | 3.59% | 1145.02 | 2.89% |
| n100-r1 | 1049.56 | 1060.308 | 1.02% | 1104.20 | 5.21% | 1044.75 | -0.46% | 1132.81 | 7.93% |
| n100-r2 | 975.36 | 1011.401 | 3.70% | 1038.64 | 6.49% | 979.83 | 0.46% | 1051.81 | 7.84% |
| n100-rc1 | 1000.80 | 1023.638 | 2.28% | 1074.43 | 7.36% | 1004.37 | 0.36% | 1108.05 | 10.72% |
| n100-rc2 | 1013.40 | 1012.979 | -0.04% | 1072.18 | 5.80% | 1033.26 | 1.96% | 1098.56 | 8.40% |
| n125-c1 | 1408.64 | 1369.071 | -2.81% | 1487.65 | 5.61% | 1449.75 | 2.92% | 1459.30 | 3.60% |
| n125-c2 | 1295.27 | 1286.906 | -0.65% | 1375.62 | 6.20% | 1283.28 | -0.93% | 1423.70 | 9.92% |
| n125-r1 | 1111.28 | 1127.340 | 1.44% | 1161.23 | 4.49% | 1092.14 | -1.72% | 1209.21 | 8.81% |
| n125-r2 | 1211.11 | 1213.907 | 0.23% | 1295.00 | 6.93% | 1257.68 | 3.85% | 1289.90 | 6.51% |
| n125-rc1 | 1145.31 | 1169.119 | 2.08% | 1242.35 | 8.47% | 1176.65 | 2.74% | 1264.61 | 10.42% |
| n125-rc2 | 1150.90 | 1130.739 | -1.75% | 1177.70 | 2.33% | 1186.85 | 3.12% | 1258.81 | 9.38% |
| Avg. | 1125.97 | 1124.321 | -0.04% | 1184.31 | 5.15% | 1142.90 | 1.45% | 1212.76 | 7.80% |

On the other hand, favoring a strategy that considers customers in the GZ generates a lower average distribution cost, due to the possibility of serving customers in these locations with either AFVs or ICEVs. This arises from the possibility of including customers near satellites into first-level routes, which can contribute to limiting the extent of long AFVs journeys outside urban areas. Similarly, incorporating these customers into routes at any level optimizes the capacity of AFVs or ICEVs routes. This, in turn, reduces the number of vehicles in both fleets.

Finally, our findings suggest that incorporating C2P stations can lead to a 6% to 37% reduction in the average cost in comparison to policies focused solely on home deliveries.

# References


[1] Anderluh, A., Nolz, P.C., Hemmelmayr, V.C. and Crainic, T.G.,"Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and 'grey zone' customers arising in urban logistics", *European Journal of Operational Research* , 289 (3), 940-958, 2021.

[2] Anderluh, A. ,Hemmelmayr, V.C and Nolz, P.C., " Synchronizing vans and cargo bikes in a city distribution network ", *Cent Eur J Oper Res* 25(1), 345-376, 2017.

[3] Mostafa M., Crainic T., Guajardo M. and Wallace S., " A two-echelon location-routing problem with synchronisation ", *Journal of the Operational Research Society* 72(1), 145-160, 2019.

[4] Grangier P., Gendreau M., Lehuédé F. and Rousseau L., "An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization ", *European Journal of Operational Research* 254(1), 80-91, 2016.

[5] Li, H., Wang, H., Chen, J., Bai, M., " Two-echelon vehicle routing problem with time windows and mobile satellites ", *Transportation Research Part B: Methodological* , 138(5), 179-201, 2020.

[6] Hongqi L., Yuan J., Tan Lv and Chang X.," The two-echelon time-constrained vehicle routing problem in linehaul-delivery systems considering carbon dioxide emissions ", *Transportation Research Part D: Transport and Environment* , 49, 231-245, 2016.

[7] Dumez D., Tilk C., Irnich S., Lehuédé F., Olkis K. and Péton O.," A matheuristic for a 2-echelon vehicle routing problem with capacitated satellites and reverse flows ", *European Journal of Operational Research* , 305 (1), 64-84, 2023.

[8] Dellaert N., Woensel T., Crainic T. and Saridarq F.," A multi-commodity two-Echelon capacitated vehicle routing problem with time windows: Model formulations and solution approach ", *Computers & Operations Research* , 127, 0305-0548, 2021.

[9] Marques G., Sadykov R., Deschamps J.C. and Dupas R.," A branch-cut-and-price approach for the single-trip and multi-trip two-echelon vehicle routing problem with time windows ", *Transportation Science* , 56 (6), 1598-1617, 2022.

[10] Soares R., Marques A., Amorim P. and Parragh S.," Synchronisation in vehicle routing: Classification schema, modelling framework and literature review ", *European Journal of Operational Research* , 167 (1), ISSN 0377-2217, 2023.

[11] Liu, D., Deng, Z., Zhang, W., Wang, Y. and Kaisar, E. I.," Design of sustainable urban electronic grocery distribution network ", *Alexandria Engineering Journal* , 60 (1), 145-157, 2021.

[12] dos Santos G., Viana A. and Pedroso J.," 2-echelon last-mile delivery with lockers and occasional couriers ", *Transportation Research Part E: Logistics and Transportation Review* , 162 (1), ISSN 1366-5545, 2022.

[13] Enthoven, D. L. J. U. and Jargalsaikhan, B. and Roodbergen, K. J. and uit hetBroek, M. A. J. and Schrotenboer, A. H.," The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers ", *Computers & Operations Research* , 118 (1), 104919, 2020.

[14] Ma, B., Wong, Y. D. and Teo, C.-C.," Parcel self-collection for urban last-mile deliveries : A review and research agenda with a dual operations-consumer perspective ", *Transportation Research Interdisciplinary Perspectives* , 16 (1), 100719, 2022.

# A Fragment-Based Approach for Vehicle Routing Problems

**Lucas Sippel**
School of Mathematics and Physics,
The University of Queensland,
Email: l.sippel@uq.edu.au

**Michael Forbes**
School of Mathematics and Physics,
The University of Queensland,
Email: m.forbes@uq.edu.au

## 1 Introduction

We propose a novel methodology for solving vehicle routing problems (VRPs) as an alternative to branch-and-price-and-cut (BPC) which is difficult to implement quickly and effectively. The VRPs we consider are modelled on directed graph $G = (V, E)$ where $V = V' \cup \{0, n+1\}$ for a fixed $n \in \mathbb{N}_+$. Each vertex $i \in V' = \{1, ..., n\}$ represents a request and 0 and $n+1$ represent the depot at the start and end of routes.

**Definition 1.1** *A **route** $\mathtt{r}$ is a path $(i_1 = 0, i_2, ..., i_k = n+1)$ in $G$ that satisfies the VRP's constraints.*

BPC algorithms solve a route-based formulation that includes a binary variable for each feasible route. This formulation has a strong linear relaxation, but often, route enumeration is intractable. Consequently, nodes of the branch-and-bound (BB) search tree are solved by adding negative reduced cost columns to a restricted subset. The problem of generating these is often solved with a labelling algorithm. Thus, implementing a BPC algorithm requires a bespoke BB framework because commercial solvers do not facilitate adding variables at search tree nodes. The implementation of the framework is made time consuming and technically difficult by the many design considerations and acceleration techniques that are essential for its effectiveness. There is therefor motivation to design alternative methods that have less implementation complexity, but are competitive or superior in performance.

We introduce some labelling algorithm notation that will aid in the explanation of our proposed alternative. Vehicles have a set of resources $R$ that are consumed as they traverse a path in $G$. A vector of resource consumption values called a label $\mathbf{L} = (L_r)_{r \in R}$ can be associated with a path. When a vehicle moves on an edge $(i, j) \in E$, the consumption value of resource $r \in R$ is updated by resource extension function (REF) $f_{ij}^r$ and must stay within bounds $[\alpha_j^r, \beta_j^r]$ at each vertex $j \in V$. For simplicity, we assume resources $r$ are disposable. That is, for all $L \in [\alpha_i^r, \beta_i^r]$ and $(i, j) \in E$, $f_{ij}^r(L) = \max\{L + t_{ij}^r, \alpha_j^r\}$ for some $t_{ij}^r \in \mathbb{R}$.

## 2 The Proposed Methodology

We propose to use sub-paths of routes called fragments. The fragments must be enumerable so that the resultant formulation can be solved without having to implement a custom BB framework. Fragments have been used by [1, 2, 3] on pickup-and-delivery problems (PDPs). Our methodology is an advanced, general version of these approaches that applies to a much wider variety of VRPs.

**Definition 2.1** *Given a route $\mathtt{r} = (i_1, ..., i_k)$ a **fragment** is a sub-path of $\mathtt{r}$, $\mathtt{f} = (i_l, ..., i_m)$ with $1 \leq l < m \leq k$, that satisfies rules of the practitioner's choice.*

The rules must ensure that each route is represented by the concatenation of at least, and preferably at most, one set of fragments. The rules are often chosen such that fragments encapsulate complexities such as path structural or synchronisation constraints. Let $\mathtt{f}^+$ and $\mathtt{f}^-$ be the start and end request of fragment $\mathtt{f}$ respectively. For each fragment $\mathtt{f} = (i_1, ..., i_\kappa) \in \mathtt{F}$ and resource $r \in R$ we define fragment REF $\tau^r_{\mathtt{f}}$ as a function of edge REFs $f^r_{i_1 i_2}$ to $f^r_{i_{\kappa-1} i_\kappa}$. It calculates the end consumption of $r$ after traversing $\mathtt{f}$. We let $\tau_{\mathtt{f}}(\mathbf{L})$ denote $(\tau^r_{\mathtt{f}}(L_r))_{r\in R}$ for all $\mathbf{L} = (L_r)_{r\in R} \in T_{\mathtt{f}}$, the set of starting labels for which $\mathtt{f}$ is resource feasible.

A resource expanded network (REN) is built with the fragments. It consists of resourced nodes and resourced fragments which we call *r-nodes* and *r-fragments* respectively. An *r-node* $\mathtt{v}$ is a vertex $i_{\mathtt{v}} \in V$ together with a label $\mathbf{L}_{\mathtt{v}}$. An *r-fragment* $\omega$ is a fragment $\mathtt{f}_\omega \in \mathtt{F}$ together with a label $\mathbf{L}_\omega \in T_{\mathtt{f}}$. An r-fragment $\omega$ with $\mathbf{L}_\omega = (L_r)_{r\in R}$ connects r-node $\omega^+ = (\mathtt{f}^+_\omega, \mathbf{L}_\omega)$ to $\omega^- = (\mathtt{f}^-_\omega, \mathbf{L}'_\omega)$ where $\mathbf{L}'_\omega \le \tau_{\mathtt{f}}(\mathbf{L}_\omega)$ component-wise. Holdover arcs between resourced copies of a vertex enable a vehicle to leave from an r-node with greater resource consumption. Let $\mathtt{V}$, $\Omega$ and $\mathtt{A}$ be the set of r-nodes, r-fragments, and holdover arcs respectively.

Let $c_\omega$ be the cost of a vehicle using r-fragment $\omega \in \Omega$ and $\Omega_i$ be the set of r-fragments that cover request $i \in V'$. Binary variables $x_\omega$ represent vehicle flow on r-fragments $\omega \in \Omega$ and continuous variables $y_{\mathtt{a}}$ represent vehicle flow on holdover arcs $\mathtt{a} \in \mathtt{A}$. The fragment-based formulation RFN is

$$\min \quad \sum_{\omega\in\Omega} c_\omega \cdot x_\omega \tag{1}$$

$$\text{s.t.} \quad \sum_{\omega\in\Omega_i} x_\omega = 1 \qquad \forall i \in V', \tag{2}$$

$$\sum_{\substack{\omega\in\Omega \\ \omega^+=\mathtt{v}}} x_\omega + \sum_{\substack{\mathtt{a}\in\mathtt{A} \\ \mathtt{a}^+=\mathtt{v}}} y_{\mathtt{a}} = \sum_{\substack{\omega\in\Omega \\ \omega^-=\mathtt{v}}} x_\omega + \sum_{\substack{\mathtt{a}\in\mathtt{A} \\ \mathtt{a}^-=\mathtt{v}}} y_{\mathtt{a}} \qquad \forall \mathtt{v} \in \mathtt{V}, \tag{3}$$

$$x_\omega \in \{0,1\} \qquad \forall \omega \in \Omega, \tag{4}$$

$$y_{\mathtt{a}} \ge 0 \qquad \forall \mathtt{a} \in \mathtt{A}. \tag{5}$$

Objective (1) minimises routing costs. Constraints (2) ensure each request is covered by exactly one r-fragment. Constraints (3) conserve vehicle flow at r-nodes. Constraints (4) and (5) define the domain.

Every route can be represented by a chain of connected r-fragments because the end resource consumption of each is rounded down. However, this rounding also introduces connected chains that do not correspond to routes. RFN is therefor a relaxation of the VRP and these so called *underestimating chains* must be eliminated from integer solutions. Feasibility cuts in a branch-and-cut algorithm were used for this in [1, 2, 3].

Increasing the number of r-nodes $|\mathtt{V}|$ reduces the amount of resource consumption rounding which removes underestimating chains. This improves RFN's linear relaxation and reduces the number of underestimating chain eliminations needed for convergence. On the other hand, it increases the number of constraints (3) and r-fragment variables $|\Omega|$. We call this trade-off between the relaxation quality and the formulation size the *relaxation-size trade-off*. Preliminary testing may determine a balanced discretization granularity however this has only worked for some PDPs [1, 3]. We therefor give three key enhancements for achieving this balance.

## 2.1 Enhancement 1: Dynamic Discretization Discovery

We propose to repeatedly solve RFN and add r-nodes to remove underestimating chains present in integer solutions. This process, known as *dynamic discretization discovery* [4], depends on the REN having the *longest arc property* which we extend to RENs with multiple resources: for each r-fragment $\omega$ there does not exist an end r-node candidate with label $\mathbf{L}''$ that is least as big as $\mathbf{L}'_\omega$ component-wise and $L''_r > L'_{\omega r}$ for at least one

$r \in R$. Suppose a solution has an underestimating chain $\omega_1, ..., \omega_k$ where $\omega_1, ..., \omega_{k-1}$ represents a resource feasible path. To remove the underestimating chain, we add r-nodes corresponding to the end vertices and true end labels $\tau_{\mathtt{f}_\omega}(\mathbf{L}_\omega)$ of resourced fragments $\omega_1$ to $\omega_{k-1}$. The end r-nodes of r-fragments are updated to maintain the longest arc property meaning resourced fragments $\omega_1$ to $\omega_{k-1}$ end at the newly added r-nodes. The underestimating chain is eliminated because the resource consumption of the final new r-node is not a feasible starting label for fragment $\mathtt{f}_{\omega_k}$ and so no resourced copy of it leaves this node. Letting the existence of underestimating chains in integer solutions guide the network construction allows us to obtain an REN with sufficient relaxation quality without including unnecessary r-nodes like a static discretization does.

### 2.2 Enhancement 2: Formulation Leveraging

Whilst we avoid BPC, we do solve the route-based formulation's linear relaxation by column generation in order to leverage the lower bound it provides, $z_{lb}$, which is stronger than that provided by RFN's linear relaxation. We do this with the variable fixing technique proposed in [5] for increasing the labelling algorithm efficiency of BPC approach. We use it to reduce $|\Omega|$. For each fragment $\mathtt{f} \in \mathtt{F}$ we calculate a lower bound on the reduced cost of any route containing the fragment, $\bar{c}_{\mathtt{f}}$. If $\bar{c}_{\mathtt{f}} > z_{ub} - z_{lb}$, where $z_{ub}$ is a valid upper bound of the VRP, then no route in an optimal solution contains $\mathtt{f}$. We therefor omit its resourced copies. This concept that we call *formulation leveraging* reduces $|\Omega|$ significantly more than direct variable fixing with RFN's reduced costs and optimality gap. Furthermore, it often improves RFN's relaxation quality because the r-fragment omission can remove underestimating chains.

### 2.3 Enhancement 3: Column Enumeration For Row Elimination

We also propose *column enumeration for row elimination* (CERE). For each $i \in V'$ we let $\mathtt{F}_i^+$ and $\mathtt{F}_i^-$ be the set of fragments that start and end with $i$ respectively. One can enumerate the set of all resource feasible paths $(i_1^1, ..., i_{k_1}^1, i_2^2, ..., i_{k_2}^2)$ where $(i_1^1, ..., i_{k_1}^1) \in \mathtt{F}_i^-$ and $(i_1^2, ..., i_{k_2}^2) \in \mathtt{F}_i^+$. Denote this set $\mathtt{F}_i^\cup$. Building the REN from fragments in $(\mathtt{F} \cup \mathtt{F}_i^\cup) \setminus (\mathtt{F}_i^+ \cup \mathtt{F}_i^-)$ rather than $\mathtt{F}$ maintains the ability to represent every route with an r-fragment chain, but no fragment in $(\mathtt{F} \cup \mathtt{F}_i^\cup) \setminus (\mathtt{F}_i^+ \cup \mathtt{F}_i^-)$ begins or ends at $i$. This means that no resourced copies of $i$ are needed and $|\mathtt{V}|$ is reduced. However, the number of r-fragments increases when $|\mathtt{F}| < |(\mathtt{F} \cup \mathtt{F}_i^\cup) \setminus (\mathtt{F}_i^+ \cup \mathtt{F}_i^-)|$ which is often the case. The vertices to perform CERE on are therefor selected preferentially based on how much the process increases $|\Omega|$ and formulation leveraging is used to mitigate the increase. Performing CERE on select requests can significantly reduce $|\mathtt{V}|$ without significantly increasing $|\Omega|$.

## 3 Results

We have implemented the framework on numerous VRPs. This section presents results for the truck-based drone delivery routing problem with time windows (TDDRP). Our talk will give results for more VRPs and discuss the impact of the key enhancements on algorithm performance.

Each truck has a drone in the TDDRP. As shown by the route in Figure 1 the drone can travel on the truck without using its battery or be deployed to service requests and rendezvous with the truck elsewhere. We use the truck and drone sub-path pairs between synchronization locations as fragments. Because these encapsulate the synchronization and drone battery constraints, only time and vehicle load must be included in the REN.

A BPC algorithm was introduced for the TDDRP in [6]. We were not able to obtain their benchmark instances and so we generated our own using their methodology. Table 1 compares their results to our fragment-

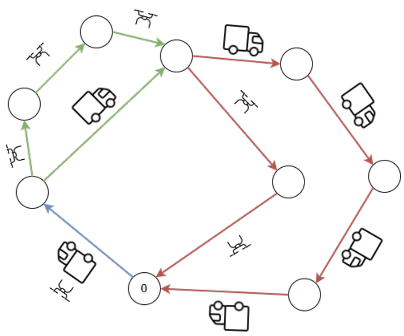


**Figure 1:** A pictorial representation of a TDDRP route made from three fragments coloured blue, green and red.

based approach. The numbers are averages over nine instances for each value of $n$. Columns $|\Omega|$ and $|\mathbb{V}|$ give the number of r-fragments and r-nodes in the final REN respectively. They exhibit that the number of variables and constraints necessary for adequate relaxation quality remains manageable because of the proposed enhancements. The fourth and fifth columns compare solve times. The last two compare the number of instances solved to optimality within the time limit. Our algorithm solves instances to optimality significantly faster than the BPC algorithm and solves instances more than twice the size within the time limit. Whilst the same instances are not solved, the results clearly show that our approach outperforms the BPC in [6]. Future research should investigate whether our approach outperforms BPC on more well known VRPs.

| | | | Solve (s) | | Optimal | |
|---|---|---|---|---|---|---|
| $n$ | $\|\Omega\|$ | $\|\mathbb{V}\|$ | Proposed | [6] | Proposed | [6] |
| 20 | 93 | 0 | 2 | 17 | 9 | 9 |
| 35 | 870 | 2 | 13 | 2694 | 9 | 7 |
| 45 | 1901 | 9 | 24 | 7067 | 9 | 6 |
| 65 | 10130 | 47 | 136 | - | 9 | - |
| 85 | 17860 | 86 | 391 | - | 9 | - |
| 105 | 98560 | 191 | 884 | - | 9 | - |
| 125 | 86526 | 234 | 2112 | - | 8 | - |

**Table 1:** Results on TDDRP Instances.

# References


[1] A. M. Alyasiry, M. Forbes, and M. Bulmer. An exact algorithm for the pickup and delivery problem with time windows and last-in-first-out loading. *Transportation science*, 53(6):1695–1705, 2019.

[2] Y. Rist and M. A. Forbes. A new formulation for the dial-a-ride problem. *Transportation science*, 55(5):1113–1135, 2021.

[3] Y. Rist and M. Forbes. A column generation and combinatorial benders decomposition algorithm for the selective dial-a-ride-problem. *Computers & operations research*, 140:105649, 2022.

[4] N. Boland, M. Hewitt, L. Marshall, and M. Savelsbergh. The continuous-time service network design problem. *Operations research*, 65(5):1303–1321, 2017.

[5] S. Irnich, G. Desaulniers, J. Desrosiers, and A. Hadjar. Path-reduced costs for eliminating arcs in routing and scheduling. *INFORMS journal on computing*, 22(2):297–313, 2010.

[6] Y. Yin, D. Li, D. Wang, J. Ignatius, T. C. E. Cheng, and S. Wang. A branch-and-price-and-cut algorithm for the truck-based drone delivery routing problem with time windows. *European journal of operational research*, 309(3):1125–1144, 2023.

# A novel formulation of the container stowage planning problem and initial results

**Agnieszka Sivertsen and Line Reinhardt**
Department of People and Technology,
Roskilde University, Denmark
Email: {avts,liner}@ruc.dk

**Rune Møller Jensen**
Computer Science Department,
IT University of Copenhagen, Denmark
Email: rmj@itu.dk

## 1 Introduction

Maritime transport, essential to international trade, shipped 165 million twenty-foot equivalent units in 2022, with a 3.5% average annual growth over two decades [1]. Even though it is the most environmentally friendly cargo transport available [2], optimizing container arrangements can significantly boost shipping network efficiency and further reduce CO2 emissions.

Despite its economic and environmental weight, the Container Stowage Planning Problem (CSPP) remains drastically under-researched, with only 200 publications since the 1970s. The principal obstacle is the industry's insularity, which necessitates years of effort to access information on problem details. The lack of understanding of its combinatorial complexities and the scarcity of public data contribute to the CSPP being over-simplified and misunderstood, as evident in mainstream literature's insufficient models (e.g.,[3, 4, 6, 5, 7]).

In collaboration with Sealytix, we have developed the Representative Container Stowage Planning Problem (RCSPP) to make the CSPP more accessible for research. Sealytix's commercial expertise in commercial stowage planning algorithms ensures our problem formulation includes essential elements without excessive complexity. We have paired the RCSPP with the most substantial benchmark suite available based on real-life data. Additionally, a Large Neighborhood Search (LNS) based algorithm was developed to evaluate the RCSPP experimentally.

## 2 The RCSPP

Container vessels follow a predetermined schedule, traveling in a circular route. At each port along the rotation, cargo is loaded onto and discharged from the vessel's stacks. The way containers are placed on the vessel is specified in a stowage plan. The goal of the CSPP is to create a stowage plan so that it is safe to sail, crane loads are considered, and the vessel's capacity is well utilized.

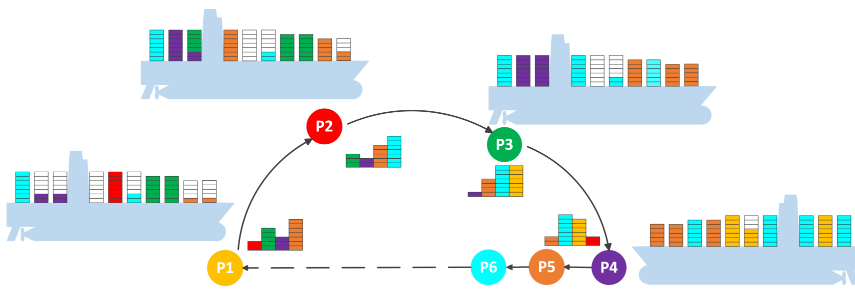



Figure 1: An example rotation.

In cooperation with the domain team from Sealytix, we propose the RCSPP with a selection of vital combinatorial elements that significantly impact the generation of an effective stowage plan. In the RCSPP, a stowage plan is created for several ports, each with its loadlist of containers to be loaded. Multi-port planning is crucial; focusing only on the current port can lead to sub-optimal stowage solutions and overall outcomes at subsequent ports. The objective is to maximize the utilization of available space on the vessel while efficiently managing ballast water and minimizing port stay by optimizing crane operations at the terminal. The constraints can be divided into stack capacities and stacking rules, constraints related to stowing special containers (refrigerated containers and dangerous goods), hydrostatics limits, and port stay constraints.

While various versions of the CSPP are found in the literature, the RCSPP assimilates the most crucial components of the CSPP and further includes the lashing and block stowage constraints that were previously overlooked. The omission of these critical constraints in the mainstream literature proves to be a blocker, as they can considerably reduce the vessel capacity and limit potential solutions. Block stowage is an indispensable practice within the shipping industry, wherein containers bound for the same discharge port are grouped. This container arrangement eliminates the need for reshuffling cargo and enhances the efficiency of managing future loadlists. In addition, lashing limitations can significantly diminish vessel capacity by subjecting containers to escalating forces based on weight, stowage height, and vessel rolling movements. The intricacies of lashing calculations, conducted through complex physical simulations, are currently excluded from the RCSPP. Therefore, we propose a simplified version of this constraint that captures all the critical factors.

## 3 LNS-based solution method

The objective of the presented solution approach is to identify viable solutions that can be used as a baseline for the attached benchmark. The implemented Large Neighborhood Search (LNS) methodology includes 16 distinct neighborhoods and 12 scores. Each score is associated with one or more neighborhoods specifically designed to address the corresponding issue. The neighborhoods are permitted to generate infeasible solutions. Therefore, scores related to the constraints are included in evaluating the solution. Depending on the specific needs, neighborhoods in the LNS can cause more significant destroy/repair moves or smaller moves involving swapping two containers. During each iteration, a neighborhood is randomly selected to address the score(s) attributed to it. Suppose a proposed solution shows improvement in the assigned scores but deterioration in others. In that case, it is permitted to use another neighborhood for further improvement before the final evaluation of the proposed solution. After the search, the post-processing phase removes containers from the vessel that cause infeasibilities.

## 4 Benchmark

The benchmark suite encompassing anonymized vessel and cargo data from top shipping liner companies is the most comprehensive real-life dataset available. This benchmark suite offers unparalleled insight into real-world scenarios, enabling a rigorous evaluation of approaches within the CSPP.

We invested substantial effort in carefully curating the presented data to ensure its quality and relevance. The dataset comprises four vessels with capacities ranging from 1080 TEUs to 18854 TEUs (twenty-foot equivalent units). The vessel data has been simplified for more straightforward calculations without sacrificing accuracy. We collected 73 instances with varying load list sizes and port call list lengths, matching the vessel's capacity. The cargo data is based on real-life situations and includes information on the initial condition of the vessel upon arrival at the first port. Through this diverse and well-crafted dataset, we aim to address various operational challenges encountered in real-world shipping scenarios.

## 5 Results and Discussion

The algorithm presented in Section 3 provided a comprehensive evaluation of the overlooked constraints in the literature, explicitly lashing limits and block stowage discussed in Section 2. Table 1 demonstrates the impact of these constraints on loaded container gaps between the basic RCSPP version (with all constraints enabled) and variants without these considerations. Tests conducted on our benchmark suite revealed that lashing

Table 1: Comparison of different variants of the RCSPP.

| Vessel | Capacity [TEU] | Loaded container points gap to the basic version of the RCSPP [%] | |
|---|---|---|---|
| | | Lashing constraint disabled | Block stowage constraint disabled |
| S | 1080 | 5.67 | 0 |
| M | 6532 | 9.03 | 8.91 |
| L | 13482 | 6.25 | 2.30 |
| XL | 18854 | 8.54 | 3.74 |

limits significantly reduce available space on the vessel, resulting in up to 9.03% fewer stowed containers when not accounted for. Similarly, the block stowage constraint restricts feasible solutions, with an average potential increase of 4.98% in stowed containers if disregarded. These findings emphasize the criticality of addressing these aspects in container stowage planning. We hope the presented results of the RCSPP and the openly accessible benchmark suite will enhance the needed research interest in this field.

## References


[1] UNCTAD, 2022. Review of maritime transport 2022, United Nations Publications, New York, NY.

[2] IMO, I.M.O., 2021. Fourth imo greenhouse study 2020, International Maritime Organization, London, England.

[3] Wilson, I.D, Roach, P.A. (1999). Principles of combinatorial optimization applied to container ship stowage planning, Journal of Heuristics, 5, pp. 403-418

[4] Kang, J.G, Kim, Y.G. (2002). Stowage planning in maritime container transportation, Journal of the Operation Research Society, 53, pp. 415-426

[5] Pacino, D., Delgado, A., Jensen, R. M., Bebbington, T. (2011). Fast generation of near-optimal plans for eco-efficient stowage of large container vessels. In Proceedings of the second international conference on computational logistics (ICCL'11): LNCS 6971. pp. 286–301.

[6] Ambrosino, D., Paolucci, M., Sciomachen, A. (2015). A MIP heuristic for Multi-Port Stowage Planning. Transportation Research Procedia, 10, 725–734

[7] Ding, D., Chou, M.C. (2015). Stowage planning for container ships: A heuristic algorithm to reduce the number of shifts. European Journal of Operational Research, Volume 246, Issue 1, pp. 242-249.

# Orienteering problem variants for pharmaceutical supply chain surveillance

**Karen Smilowitz**
Department of Industrial Engineering and Management Sciences,
Northwestern University, Evanston, Illinois
Email: ksmilowitz@northwestern.edu

**Eugene Wickett**
Industrial Engineering and Management Sciences,
Northwestern University, Evanston, Illinois

## 1 Introduction

Post-market surveillance is a crucial quality-assurance activity for medical products regulators in low- and middle-income countries. Regulators procure products from consumer-facing outlets and test these products for quality. Analysis of testing results informs the deployment of corrective actions. In this paper, we explore connections between the design of sampling plans for surveillance (e.g., where to collect samples and how many to collect) and recent advances in variants of the orienteering problem. We highlight the unique challenges in this setting and propose computationally efficient approaches to address these challenges, building on work for related orienteering problem variants.

## 2 Sampling plans for post-market surveillance

Substandard and falsified products (SFP) are a crucial challenge in global public health. Regulators in low-resource settings are tasked with ensuring that only high-quality products reach consumers. Post-market surveillance (PMS) is a principal regulatory activity where medical products from consumer-facing outlets are tested against registration specifications. A key part of PMS is the sampling plan. The sampling plan specifies which outlets to visit, the order of visits, and the number of tests to conduct at each location. Given constraints on resources, it is critical to identify sampling plans that provide the most utility, in terms of informing corrective actions.

Through a collaboration with a medical product regulatory agency, we are developing new methods to generate and evaluate sampling plans. Beginning with early work on models to infer SFP rates at various locations in a pharmaceutical supply chain, we show the value of incorporating supply chain information e.g., exploring connections between outlets with common suppliers [5], and propose an efficient means of calculating sampling plan utility that leverages this information about the supply chain [6]. We highlight two key observations from this earlier work. First, the marginal increase in utility for tests at a location is decreasing: the information gained from any test is less than the information gained from the previous test. Second, the marginal increase in utility for tests at a node depends on the tests at other nodes included in the sampling plan. This dependency reflects the connected nature of the supply chain. Consider the sampling plan depicted in Figure 1: Region 4 may have a high associated utility on its own because it sources from Manufacturer $C$ for which little information exists. The inclusion of Region 5, which also sources from Manufacturer $C$, in the sampling plan may lower the value of testing in Region 4. In this paper, we build from these observations and the proposed utility measure to address the broader question of designing sampling plans to maximize testing utility under resource constraints. The costs of dispatching personnel to distant geographic locations can be substantial: in addition to transportation, operational budgets must also include room and board for personnel spending time away from centralized regulatory agency locations. Therefore, efficient, well-designed sampling plans are essential.

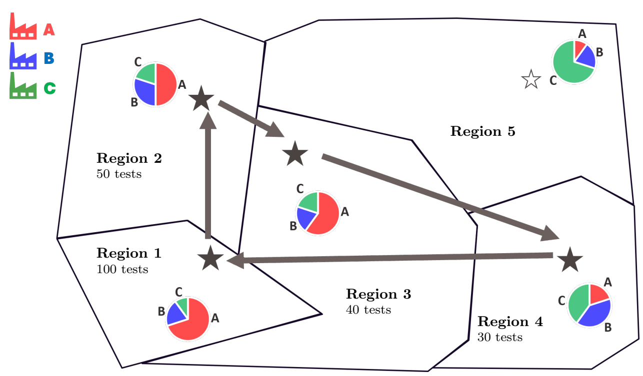



Figure 1: Example sampling plan showing regions visited and number of tests collected. Pie charts indicate sourcing distribution from each manufacturer (A, B, C).

## 3 Sampling plan design as an orienteering problem

Given the high costs of dispatching personnel to conduct PMS testing in distant locations, the problem of identifying the most useful sampling plan can be viewed as an orienteering problem, where an actor gains utility from visiting nodes in a network but is constrained by the time or cost needed to conduct visits [3]. Orienteering problems typically have additive objective functions which maximize the sum of rewards from nodes visited. However, in the PMS context, the objective of maximizing utility has two principle challenges.

First, the utility of testing at a location is dependent on the set of all test allocation decisions (e.g., the other locations and the amount of samples procured at each location). Consumer-facing locations are connected in the supply chain through manufacturers and distributors: information at one supply-chain location impacts what is known at other locations. Second, as a result of this dependence, the utility is expensive to compute. Estimating a sampling plan's utility requires considerable computational resources: estimates for problems on the scale of our case study take around five minutes on a laptop.

Fortunately, recent work in the orienteering literature has explored novel objective functions in which rewards are interdependent across nodes, particularly in areas of robotics and mobile sensing; see [4] for a recent review. Notably, [7] introduces the correlated orienteering problem which allows for spatial correlation in rewards among network nodes. This leads to the formulation of a quadratic reward function. The authors introduce an algorithm to solve this complex variant of the orienteering problem relevant in our setting. [2] presents a review of spatial coverage in routing problems, including Informative Path Planning (IPP), which most closely resembles our PMS context. In particular, the IPP does not have a closed-form reward function as IPPs rely on probabilistic models to capture the utility gained from each node visited. [2] notes that several IPP solution approaches are versions of the greedy algorithm of [1], which iterates through each node as the possible "halfway" node of the optimal path and finds distinct shortest paths from this halfway node to the origin. We explore the potential for these approaches to sample plan design.

In addition to selecting which outlets to visit and in what sequence, sampling plan design also determines the number of tests to allocate to each outlet, tying the problem to the inventory routing problem as well. Conducting too few tests at a location can lead to an inefficient use of the resources used to reach that location, while too many tests results in low marginal utility gain and lost opportunity to test more elsewhere. Testing often requires reference standards that consume significant portions of PMS budgets, thus tests are conducted in batches, which impacts the number of samples collected for a particular product. Further, PMS implementation carries fixed costs at regional and sub-regional levels. Regulators frequently coordinate with regional health authorities when visiting locations, and regional capitals often have the accommodations used by regulators taking multi-day trips. Previous orienteering works features settings where traversing one node within a larger region yields a utility from that region, but to our knowledge previous settings do not include fixed costs that incurred for visiting a region containing a set of locations.

Based on recent advances in path planning with spatial correlation, we will present a formulation to generate sample plans that maximize utility for a single sample collection team. We will also present our solution approach to generate plans. With the challenge of determining which locations to visit for testing (and the visit sequence), we also need to determine the number of tests to collect at each location. Our approach consists of two phases: an initial solution phase and an improvement phase. An initial feasible solution is found through a linear relaxation of the utility, where sampling at each location is considered independently. Utility evaluations at a minimal and maximal number of tests, coupled with the concavity requirement, provide a reasonable proxy for the utility at each location. The relaxation bounds the utility of the optimal allocation, but ignores the utility inter-dependence for allocations at different locations. The improvement phase modifies allocation amounts and/or chosen locations to propose new feasible solutions. Using de-identified data from our collaborators in medical product regulation, we highlight the potential of this approach to generating and analyzing sampling plans. This case study shows that plans constructed with this approach achieve higher utility than plans from standard approaches, translating to meaningful budgetary savings.

## References


[1] C. Chekuri and M. Pal, "A recursive greedy algorithm for walks in directed graphs", *46th Annual IEEE Symposium on Foundations of Computer Science* 245-253, 2005.

[2] K. Glock, and A. Meyer, "Spatial coverage in routing and path planning problems," *European Journal of Operational Research*, 305 (1), 1-20, 2023.

[3] B. Golden, L. Levy and R. Vohra, "The orienteering problem", *Naval Research Logistics* 34 (3), 307-318,1987.

[4] A. Gunawan, H. Lau and P. Vansteenwegen, "Orienteering Problem: A Survey of Recent Variants, Solution Approaches and Applications", *European Journal of Operational Research* 225, 2016.

[5] E. Wickett, M. Plumlee, K. Smilowitz, S. Phanouvong and V. Pribluda, "Inferring sources of substandard and falsified products in pharmaceutical supply chains", *IISE Transactions* forthcoming 2023; available at https://doi.org/10.48550/arXiv.2207.05671.

[6] E. Wickett, M. Plumlee, K. Smilowitz, S. Phanouvong and T. Nwogu, "Measuring sampling plan utility in post-market surveillance of medical product supply chains", *Manuscript available on ArXiv as of 12/12/23.*

[7] J. Yu, M. Schwager and D. Rus, "Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks", *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* 342-349, 2014.

# Integration of Rider Preferences into the Route Planning of Bicycle Courier Services

**Felix Spühler**
Institute of Business Information Systems,
Technische Universität Braunschweig, Braunschweig, Germany
Email: f.spuehler@tu-braunschweig.de

**Dirk C. Mattfeld**
Institute of Business Information Systems,
Technische Universität Braunschweig, Braunschweig, Germany

**Michael Hewitt**
Quinlan School of Business,
Loyola University Chicago, Chicago, United States

## 1 Introduction

Courier services have a long tradition in city logistics, primarily in the delivery of valuable documents or medicines. For short trips, bicycle couriers have major advantages as they are less affected by traffic volume and congestion, they do not block small streets while serving, and they can serve customers in car-free zones more easily.

Companies are slowly changing the focus towards employee satisfaction because workforce is an increasingly important and a limited resource. Thus, the main goal of an operator is not only to maximize delivery profit, but also to consider the well-being of the riders. Real-world examples for courier services with such a social focus are Cycle Logistics[1] in Berlin, Germany, and Heavy Pedals[2] in Vienna, Austria.

Looking on rider's well-being, there may be different possibilities to ensure a higher satisfaction. For example, the overall workload can be distributed evenly among all riders, as presented in [1]. However, the rider is not seen as an individual. In our research, we therefore consider the individual preferences of the riders regarding their route choice, e.g., riders may prefer a detour to take street sections with separated bicycle lanes, and incorporate these different preferences into the vehicle routing process. To do so, we model a pickup and delivery vehicle routing problem with time windows. The model is based on

[1] https://cycle-logistics.bike/
[2] https://heavypedals.at/

a multi-graph that represents the different route choices between each pair of locations. To solve the problem, we apply an Adaptive Large Neighborhood Search that is adopted to the problem specifications. Results support our assumptions and show in particular that the degree of efficiency loss depends on the characteristics of the city.

## 2 Problem description

We consider a courier service that operates locally in one city such that all customers can be served by bicycle courier riders. The main focus of the operator of this bicycle courier service is not only to fulfill all requests as efficiently as possible, but also to take the employee satisfaction into account. We assume that the riders and customer requests are known in advance for one working day. Thereby, a customer request consists of a single order with equal weight, pickup and delivery location, and a certain time window. Furthermore, the riders are paid per served request.

The fleet consists of riders that are homogeneous in terms of speed and capacity, but are divided into two preference groups: income or convenience. In each preference group, the riders are considered homogeneous regarding the preferences. On the one hand, riders with income preference want to serve as many requests as possible and thus comply with the operator's efficiency goal. On the other hand, riders with convenience preferences would accept detours to avoid unsafe or unpleasant street sections and therefore less served requests.

Overall, the goal of the operator is to serve all customers efficiently while also satisfying the preferences of the riders. Riders with income preference fit to both goals because of their efficiency-oriented preference. Riders with convenience preference contradict this efficiency principle to some extent. However, this can be balanced with the procedure that these riders can drive conveniently, if possible, but also have to drive efficiently if not. If requests cannot be served at all, an expensive third-party delivery can be used.

## 3 Model and methodology

We model the problem as a pickup and delivery vehicle routing problem with time windows based on a multi-graph, following [2] for the multi-graph modeling. Therefore, we first consider a graph representing the real underlying road network in which the real locations of the depot and the customers are defined. The road sections are described by a length attribute and attributes describing bicycle convenience. Based on this, we define the multi-graph. The vertices correspond to the depot location and the pickup and delivery locations. For each pair of vertices, we calculate two different paths in the underlying road network, a fast path and a convenient path. The fast path is the shortest path regarding the length attribute, while the convenient path is the shortest path regarding a combination of the length and the bike convenience attribute. These paths define the parallel arcs in the multi-graph with attributes for travel time and for bike convenience, which are the sums of the attributes of the paths in the real road network, respectively. In addition, we define an attribute that indicates the regret if a convenience-oriented rider has to take the fast path instead of the convenient path. It implies that a fast path between two locations has a shorter travel time and some regret, while the convenient path has a higher travel time and no regret, i.e., the regret is greater equals 0 on a fast path and 0 on a convenient path. It may happen in the real-world network that the fast path is also the most convenient one. Thus, both parallel arcs have the same travel time and the same, namely no regret.

The objective of our model is threefold. First, we want to minimize the regret over all convenience-oriented riders. Second, in order to ensure efficient routing, we also want to minimize the total travel time over all riders. The third term considers the involvement of a costly third party, that is to be minimized. This latter term ensures model feasibility in particular if riders become a scarce resource. Moreover, the general constraints from the standard pickup and delivery problem apply.

The idea of the objective is to find a solution where the convenient-oriented riders are sent on as many convenient paths as possible. At this overall regret level, the tours of all riders should be as efficiently as possible. This applies in particular to the income-oriented riders who only ride on fast paths.

To solve the problem, we apply the Adaptive Large Neighborhood Search from [2] which uses a labeling procedure for a fast evaluation of the arcs in the multi-graph. We extend this meta-heuristic by incorporating the specifications of a pickup and delivery problem and a heterogeneous fleet.

## 4 Computational results

We have conducted experiments on artificial and real instances. First, the results confirm that a larger number of requests as well as narrow time windows of the customers cause convenience-oriented riders to take some fast paths to a certain degree. Second, the comparison of the preference-oriented objective with the objective minimizing the total travel time, i.e., convenience-oriented riders taking only fast paths, yields differences in the tours of each rider. Third, using the preference-oriented objective results in the income-oriented riders serving more customers on average, while using the travel time objective results in a similar number of served customers across all riders.

A more detailed analysis of the experiments on two real-world datasets, namely Copenhagen in Denmark and Amsterdam in the Netherlands, yields the following insights: Even though both cities are considered as very bike-friendly, the two cities have different structure with regard to our utilized bike index. This also affects the computation of the fast and convenient paths. This means, for example, that the convenient paths in Copenhagen are on average about 32% longer than the fast paths, while it is just about 12% in Amsterdam. Looking at the results from Copenhagen, we find that the sum of the travel times is about 25% longer for the preference-oriented objective compared to the objective minimizing the total travel time.

## References


[1] Piotr Matl, Richard F Hartl, and Thibaut Vidal. Workload equity in vehicle routing problems: A survey and analysis. *Transportation Science*, 52(2):239–260, 2018.

[2] Hamza Ben Ticha, Nabil Absi, Dominique Feillet, and Alain Quilliot. Multigraph modeling and adaptive large neighborhood search for the vehicle routing problem with time windows. *Computers & Operations Research*, 104:113–126, 2019.

# Branch-Price-and-Cut for the Electric Vehicle Routing Problem with Multiple Recharging Technologies and Nonlinear Recharging Functions

**Magnus Stålhane and Gaute Messel Nafstad**
Department of Industrial Economics and Technology Management,
Norwegian University of Science and Technology, Trondheim, Norway
Email: magnus.staalhane@ntnu.no

**Guy Desaulniers**
Department of Mathematics and Industrial Engineering,
Ecole Polytechnique, Montreal, Canada

## 1 Introduction

The Electric vehicle routing problem (E-VRP) extends the well known VRP by considering that the vehicles have a battery capacity, which limits how far they can travel before having to either recharge or end the route. The vehicles may visit dedicated recharging stations, in addition to the depot and the customer locations, to re-charge their batteries during a route. The objective of the E-VRP is usually to either minimize the total distance traveled or the total time spent on the routes, including travel time and recharging time.

The recharging technologies available at recharging stations are represented by *recharging functions*, which describe the relationship between charging time and the increase in battery power. The battery power is often referred to as the state of charge (SoC). In the literature, recharging functions have been modelled in various ways, which leads to different versions of the E-VRP. The way the recharging is modelled varies in two important aspects, the first being whether recharging stations have the same, or different, recharging technologies. The other is whether the recharging functions are assumed to be linear or nonlinear. Having both multiple recharging technologies and nonlinear charging makes the problem considerably more complex. An example of multiple non-linear recharging functions are illustrated in Figure 1.

The purpose of this work is to present a branch-price-and-cut (BP&C) solution method for the E-VRP with time windows, multiple recharging technologies and nonlinear charging functions (E-VRPTW-NL). The novelty of the method lies in the way recharging is handled

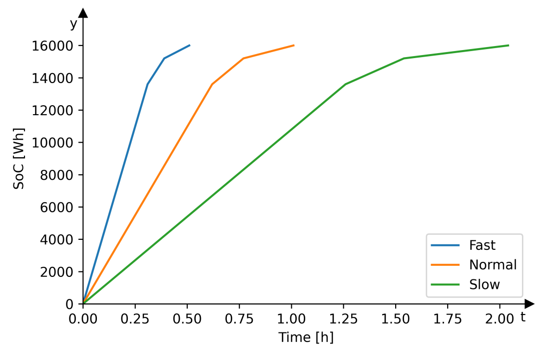



Figure 1: The recharging functions used in the Montoya benchmark instances [4].

in the column generation pricing problem, where there is a trade-off between the time spent on the path and the battery power available.

## 2 Solution method

BP&C is a method that combines branch-and-bound, column generation and cutting planes, and is arguably the most efficient method for solving many extensions of the vehicle routing problem [1]. For the E-VRPTW-NL, the master problem is just a set partitioning formulation, where a subset of vehicle routes are to be chosen so that each customer is visited by exactly one route. The subproblem is tasked with finding new negative reduced cost routes, or to prove that no such route exist, and may be formulated as a shortest path problem with resource constraints (SPPRC).

The SPPRC is solved by dynamic programming using a labeling algorithm [3] that search for the least cost path from the source node to the sink node. For the subproblem of the E-VRPTW-NL, a label representing a path from $d(s)$ to node $i$, can be described as a tuple $L_F = (i,\ d,\ \bar{c},\ t,\ y,\ \mathcal{U})$. The last node on the path is node $i$, and $d$ denotes the load onboard the vehicle at departure from the node $i$. The accumulated (reduced) cost is given by $\bar{c}$, while the resource $\mathcal{U}$ is the set of unreachable customer nodes. The resources $t$ and $y$ give the departure time, and SoC of the vehicle, respectively. Since these two resources are dependent on each other, we describe the function $f_i(y)$ as the earliest time it is possible to leave node $i$ with SoC $y$.

The main challenge of this labeling algorithm is the resource extensions of the $t$ and $y$ resources. The value of $f_i(y)$, given a SoC function $f_{i-1}(y)$ from the predecessor label, and a recharging function $r(y)$ at node $i$ is described by the following recursion:

$$f_i(y) = \min_{x \in [0,y]} \{f_{i-1}(x) + r(y) - r(x)\}. \tag{1}$$

Figure 2 illustrates an example of what such functions may look like. We have SoC

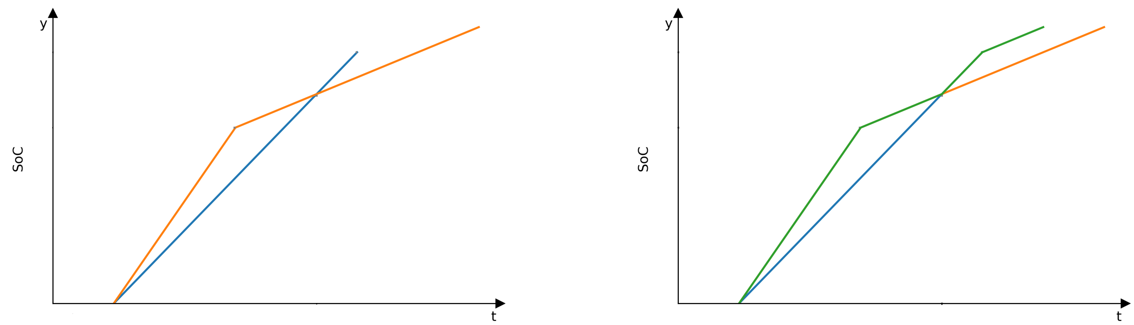



Figure 2: SoC functions $f_{i-1}(y)$ (blue) and $r(y)$ (orange) for a partial path arriving at a charging node $i$, and the resulting SoC function $f_i(y)$ (green) when leaving node $i$.

functions $f_{i-1}(y)$ (blue) for a partial path arriving at a charging node $i$, with a recharging function $r(y)$ (orange). The resulting function $f_i(y)$ is illustrated by the green line. As we can see by the figure, the resulting function does not necessarily equal one of the initial functions for all SoC values. In the presented work, we propose a linear time algorithm for computing $f_i(y)$ for all values of $y$ simultaneously.

To speed up the labeling algorithm, we use bi-directional labeling, and the NG-path relaxation of the set $\mathcal{U}$. We also solve the labeling algorithm heuristically by applying it on a reduced network the quickly find negative reduced cost columns.

Further, we add limited-memory subset row inequalities as cuts in the root node of the branch-and-bound tree to strenghten the linear programming relaxation.

# 3 Computational study

The BP&C-method has been tested on E-VRP-NL instances from [4] and compared with the current state-of-art solution method presented in [2]. The maximum computing time was set to three hours. The numerical results are presented in Table 1, and show that the BP&C method significantly improves upon the current state-of-the-art, as all except one of the 20-customer instances are solved to optimality, and is also able to solve 12 of the 40-customer instances, leading to a total of 26 new benchmark instances being solved to optimality.

Table 1: Numerical results on the E-VRP-NL instances by [4]

| | | [2] | | Our method | |
|---|---|---|---|---|---|
| #Cust. | #Inst. | #Inst. solved | Avg. time[s] | #Inst. solved | Avg. time[s] |
| 10 | 20 | 20 | 229.5 | 20 | 2.6 |
| 20 | 20 | 5 | 8222.2 | 19 | 1126.6 |
| 40 | 20 | 0 | 10800 | 12 | 5841.1 |

Furthermore, we have created new instances for the E-VRPTW-NL by combining the instances of [5] with the charging functions from [4]. The maximum computing time for these tests were set to one hour. The aggregated computational results are presented in Table 2.

The numerical results show that the algorithm is able to solve instances with up to 100 customers. The results for instances with narrow time windows, Type 1, are better than the results for the instances with wide time windows, Type 2. That is to be expected as it is well-known that wide resource windows make the exponential growth in the number of labels worse. Further, a comparison with the results presented for the E-VRPTW-L in the literature, indicates that the computing times only increases slightly when the instances contain multiple nonlinear recharging functions, compared with when they only have a single linear recharging function.

Table 2: Numerical results on the generated E-VRPTW-NL instances.

| #Cust. | Type | #Inst. | #Inst. solved | Avg. time[s] | Avg. #nodes | Avg. #SRcuts |
|---|---|---|---|---|---|---|
| 25 | 1 | 29 | 29 | 50.9 | 1.5 | 15.4 7 |
| 25 | 2 | 27 | 26 | 297.0 | 1.0 | 9.7 |
| 50 | 1 | 29 | 24 | 1023.6 | 37.9 | 51.4 |
| 50 | 2 | 27 | 12 | 2332.1 | 0.5 | 7.9 |
| 100 | 1 | 29 | 6 | 2956.4 | 27.0 | 65.6 |
| 100 | 2 | 27 | 2 | 3532.0 | 0.6 | 8.6 |

## References


[1] Costa, L., Contardo, C., Desaulniers, G., 2019. Exact branch-price-and-cut algorithms for vehicle routing. Transportation Science 53, 946–985.

[2] Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2019. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. Computers and Operations Research 104, 256–294.

[3] Irnich, S., Desaulniers, G., 2005. Shortest path problems with resource constraints, in: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), Column Generation. Springer US, Boston, MA, pp. 33–65.

[4] Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017. The electric vehicle routing problem with nonlinear charging function. Transportation Research Part B: Methodological 103, 87–110.

[5] Schneider, M., Stenger, A., Goeke, D., 2014. The electric vehicle-routing problem with time windows and recharging stations. Transportation Science 48, 500–520.

# Analysis of Locker Usage and Crowdshipping in Stochastic Pickup-and-Delivery

**Sara Stoia, Francesca Vocaturo***
Department of Economics, Statistics and Finance "Giovanni Anania"
University of Calabria, Rende (CS), Italy
Email: sara.stoia@unical.it, vocaturo@unical.it

**Demetrio Laganà**
Department of Mechanical, Energy and Management Engineering
University of Calabria, Rende (CS), Italy
Email: demetrio.lagana@unical.it

**Jeffrey W. Ohlmann**
Department of Business Analytics
University of Iowa, Iowa City, United States
Email: jeffrey-ohlmann@uiowa.edu

## 1 Motivation

We consider on a collaborative e-platform that receives requests for products from a collection of local businesses and then schedules the pick-up of the requested products from the local store and the subsequent delivery of these requested products to the e-shoppers' local locations. To facilitate the efficiency of this pickup-and-delivery operation, we consider the e-platform's use of parcel lockers and crowdsourced couriers to complement a dedicated fleet.

Born out of the sharing economy paradigm, crowdshipping refers to a delivery platform's use of couriers who are not dedicated, full-time employees. While the use of crowdshippers has the potential to reduce operating costs, the reduced managerial control of these temporary employees presents challenges. In this work, we consider various types of crowdshippers and the uncertainties involved in managing them.

*The work of Sara Stoia and Francesca Vocaturo has been supported by "ULTRA OPTYMAL - Urban Logistics and sustainable TRAnsportation: OPtimization under uncertainTY and MAchine Learning", a PRIN2020 project funded by the Italian University and Research Ministry (number: 20207C8T9M).

As an attempt to improve customer service and decrease delivery cost, we also consider the strategy of using lockers at which couriers can drop off parcels to be later picked up by the customer [1]. The motivation to employ lockers is to reduce the number of delivery locations by aggregating customer demand at locker locations. The use of a parcel locker can also ensure a secure delivery at a location that is still relatively convenient to the customer, making it an attractive option to a customer who may not be home during delivery hours and is concerned about "porch pirates."

## 2 Problem Description

Our work aims to analyze the operational impact of crowdshipping and parcel lockers for a same-day stochastic pickup-and-delivery problem facing an e-platform. We assume that an e-platform receives the customer requests randomly throughout the day. A customer request $r$ corresponds to a tuple of information: the pickup location ($o_r$), the customer location ($d_r$), and the time which the request will be ready ($e_r$), and the delivery deadline ($\ell_r$). If the vehicle handling customer request $r$ arrives at location $o_r$ before time $e_r$, the vehicle must wait until $e_r$ before executing the request pick up. To provide delivery service, we assume there is a fleet of dedicated vehicles, $\mathcal{V} = \{1, \dots, V\}$, and a set of crowdshippers, $\mathcal{G} = \{1, \dots, G\}$. Each dedicated vehicle $v \in \mathcal{V}$ is available for a daylong shift starting and ending at the depot. In contrast, when and how long each crowdshipper $g \in \mathcal{G}$ is available is unknown to the e-platform before their appearance. The objective is to minimize the cost of fulfilling customer requests over an operating day. To measure the cost of providing service, we minimize the sum of three components: (1) the total travel cost of the dedicated vehicles and crowdshippers (as function of travel time), (2) the total per-delivery fees paid to crowdshippers, and (3) the total lateness charge (as a function of the amount of time customer requests are delivered after their soft deadlines). We assume a fixed per-delivery fee of $\rho$ for all requests served by a crowdshipper and include the travel cost of the crowdshippers in the objective function to represent the variable fees paid by the e-platform to the crowdshippers to account for differences in the travel demands of requests due to their relative pickup and delivery locations.

### 2.1 Setting 1: In-store Shoppers

In our first problem setting, we consider crowdshipping via in-store shoppers. For every request that appears on the e-platform, we assume there is a probability $p$ that an in-store shopper $g$ at the request's pickup location will offer their services as a crowdshipper and accept the assignment of request deliveries as long as they can still return their specified home location $n_g$ within $w$ minutes of the current time. By generating the appearance of an in-store crowdshipper in this manner, we correlate the arrivals of crowdshippers with request demand arriving to the e-platform.

### 2.2 Setting 2: Lockers and Committed Gig Workers

In a second problem setting, we consider crowdshipping via committed gig workers. Each crowdshipper $g \in \mathcal{G}$ is available to service requests during a time window $[a_g, b_g]$ that is unknown to the dispatcher a priori. Upon appearing to the platform at time $a_g$ and declaring their location $n_g^a$, a crowdshipper $g$ communicates their availability to service requests subject to requirement that they can arrive at location $n_g^b$ by time $b_g$. For each arriving request $r$, the e-platform identifies $u_r$, the locker closest to the customer location. We determine the delivery location of a request $r$ by assuming the customer picks the closest locker location $u_r$ with probability $p(u_r)$, and picks their own location $d_r$ with probability $1 - p(u_r)$.

## 3 Solution Approach

We model our problem as Markov decision process that has a high-dimensional state, action, and outcome space. To identify a policy mapping an action to any observed state, we modify the approximate dynamic programming method in Stoia et al. [2]. In this cost function approximation, the cost of assigning a request to a vehicle is modified with a surrogate cost that depends on the residual time capacity of the vehicles weighted by a scalar parameter $\lambda$ that is calibrated through offline simulation. In this manner, we account for the relatively scarce time capacity of the crowdshippers.

## 4 Preliminary Computational Results

To assess our solution approach in two distinct problem settings, we consider a set of data instances from Stoia et al. [2] in which requests arrive according to a non-homogeneous Poisson process with an average of 225 requests over a 10-hour workday.

In our preliminary analysis of Setting 1, we compare the total cost when using a fleet of five dedicated vehicles to the total cost when using a fleet of four dedicated vehicles and crowdshipping via in-store shoppers. We vary the number of in-store crowdshippers by considering different values of $p$, their appearance probability. Further, we vary the delivery capacity of each in-store crowdshipper by considering different values of $w$, the amount of time until the crowdshipper wishes to arrive at their home location.

For each test setting, Table 1 displays $\bar{c}_{dedicated} - \bar{c}_{mixed}$, where $\bar{c}_{dedicated}$ is the average total cost when using a fleet of five dedicated vehicles and $\bar{c}_{mixed}$ is the average total cost when using a fleet of four dedicated vehicles and in-store crowdshippers with parameters $p$ and $w$. We compute average total cost over five instances. As Table 1 shows, even

Table 1: Average cost savings ($/day) when using mixed fleet of four dedicated vehicles and in-store crowdshipping versus five dedicated vehicles and no crowdshipping.

| | $w$ (minutes) | | |
|---|---|---|---|
| $p$ (%) | 20 | 30 | 60 |
| 5 | -255 | 906 | 1250 |
| 10 | 2504 | 2669 | 2829 |
| 20 | 3433 | 3366 | 3187 |
| 30 | 3626 | 3496 | 3373 |

without considering the possible cost savings from having one less dedicated vehicle versus in-store crowdshippers, the mixed fleet reduces costs for all settings except when in-store crowdshippers are rare ($p = 5\%$) and available for only a short time ($w = 20$).

In our preliminary analysis of Setting 2, we position eight lockers based on a $k$-means clustering of customer locations from sample data. We consider 3 dedicated vehicles and 28 gig workers who arrive randomly throughout the day for one- to four-hour time windows. We compare three cases: (i) $p(u_r) = 0$, each request delivered to the requesting customer location, (ii) $p(u_r) = 1$, each request delivered to the locker nearest the requesting customer's location, and (iii) $0 < p(u_r) < 1$, generated from a (reverse) sigmoid function such that the probability of locker selection decreases nonlinearly as the distance between the nearest locker $u_r$ and the customer location $d_r$ increases. Over ten instances, the average total cost in Setting 2 is $2905 for the $p(u_r) = 0$ case, $2756 for the $p(u_r) = 1$ case, and $2837 for the $0 < p(u_r) < 1$ case. This experiment suggests that universal locker use in our stochastic pickup-and-delivery problem can lead to a 5.1% decrease in the average total cost, providing an upper bound on the cost reduction possible from less aggressive locker use. Indeed, these initial tests suggests when likelihood of locker selection depends on distance between the locker and the customer, the cost reduction is 2.3%.

## References


[1] Y. Lin, Y. Wang, L.H. Lee and E.P. Chew, "Profit-maximizing parcel locker location problem under threshold Luce model," *Transportation Research Part E* 157, 2022.

[2] S. Stoia, D. Laganà and J.W. Ohlmann, "Dynamic pickup-and-delivery for collaborative platforms with time-dependent travel and crowdshipping," under first revision at *European Journal of Operational Research*, 2023.

# Reward Strategy in a Large-scale Urban Crowd-shipping System

**Patrick Stokkink**
Urban Transport Systems Laboratory (LUTS),
École Polytechnique Fédérale de Lausanne (EPFL),
Email: patrick.stokkink@epfl.ch

**Alfio Simone Mosset**
Urban Transport Systems Laboratory (LUTS),
École Polytechnique Fédérale de Lausanne (EPFL)

**Nikolas Geroliminis**
Urban Transport Systems Laboratory (LUTS),
École Polytechnique Fédérale de Lausanne (EPFL)

## 1 Introduction

Crowd-shipping as a solution to the last-mile delivery problem has gained much attention over the last few years. Traditional delivery vehicles are considered to be polluting, and thereby crowd-shipping has the potential to be more sustainable, more flexible, and less costly, if sustainable modes of transport (such as bikes or e-bikes) are used. One of the most important drivers of the success of a crowd-shipping system to obtain an adequate service level is the availability of crowd-shippers. Among the factors that affect the willingness of potential crowd-shippers to participate is the (monetary) reward they receive [1, 3].

Previous studies on reward and compensation schemes in crowd-shipping have shown the potential of such strategies [2, 4]. In the existing literature, the problem is tackled by formulating exact models to determine the optimal rewards. These works focus on small-scale instances, where these models can be solved to optimality. Contrary to the existing literature, we focus on a large-scale urban crowd-shipping system. Thereby, we take into account the high level of stochasticity and dynamic conditions in such systems.

In this work, we develop a simulation-optimization-based approach to determine the optimal reward strategy. Rewards are used to incentivize crowd-shippers to deliver a parcel, possibly with a small detour from their original path. We design a simulator that incorporates the uncertainty in the arrival of future crowd-shippers as well as the uncertainty in the acceptance of requests. The designed reward strategy incorporates the interaction between supply and demand in such a complex system, which shows to outperform constant pricing strategies.

## 2 Methodology

### 2.1 Problem description

We analyze a few-to-many last-mile delivery problem. Parcels are spread out over a small set of origin locations, from where they are picked up and delivered by crowd-shippers to their final destinations. We consider a time frame of a single day, where all parcel requests $P$ are known at the start of the day but crowd-shippers $C$ arrive dynamically over time. A penalty $\rho$ has to be paid in case the parcel is not delivered (alternatively, this can be interpreted as a lost revenue of delivery or cost of outsourcing).

The decision variables are the rewards that are offered to crowd-shippers for delivering a parcel. We determine the reward depending on the destination of a parcel. This means that for every node $n \in 1, \ldots, N$ in the network we determine a reward $r_n$. Due to the high level of stochasticity, this is more robust than determining the reward for every parcel individually. Thereby, it allows for a reduction in the number of decision variables and therefore has a computational advantage.

The number of delivered and undelivered parcels and the corresponding cost depend on the full set of rewards $r_1, \ldots, r_N$. Here, we emphasize that the reward for parcels with destination $i$ may also influence the service level of parcels with destination $j$ due to the influence on crowd-shipper behavior. To obtain an accurate approximation of these values, we use a discrete event simulator that simulates the arrivals of crowd-shippers and their response to the assigned parcel and corresponding reward (i.e., whether they accept or reject the offer). The simulation is repeated $\kappa$ times to obtain an average service level and cost. We introduce the following notation. Let $s_n(r_1, \ldots, r_N)$ be the number of delivered parcels with a final destination at node $n \in N$ when the rewards are $r_1, \ldots, r_N$. Similarly, let $u_n(r_1, \ldots, r_N)$ be the number of undelivered parcels with a final destination at node $n \in N$. We can then define the optimization problem as follows:

$$\min_{r_1,\ldots,r_N} TC(r_1, \ldots, r_N) = \sum_{n \in N} r_n s_n(r_1, \ldots, r_N) + \rho u_n(r_1, \ldots, r_N) \tag{1}$$

After crowd-shippers are offered a parcel request and are informed about a corresponding reward, they may then decide whether to accept the offer or not. We model the acceptance through a binomial logistic model. The acceptance probability decreases in the detour and increases in the reward. Determining the rewards therefore is a trade-off between offering high rewards to achieve higher acceptance but lower profits per parcel and offering low rewards to achieve lower acceptance but higher profits per parcel.

For every crowd-shipper, a central operator decides which parcel to assign to this crowd-shipper dynamically. In the full paper, we compare various existing and newly developed dynamic assignment strategies.

### 2.2 Simulation-optimization approach

We develop a simulation-optimization approach to obtain the optimal set of rewards $r_1, \ldots, r_N$. The objective $TC(r_1, \ldots, r_N)$ is evaluated through a set of $\kappa$ simulations. Typically, as $\kappa$ goes up, the objective is a more accurate approximation of the average cost. On the other hand, the computation time also increases. We heuristically optimize the set of rewards $r_1, \ldots, r_N$. First, we locally optimize the rewards by neglecting the cross-effects of rewards between nodes to initialize the rewards. This can be performed separately for every node and is therefore much faster than running a full simulation. With cross-effects, we refer to the phenomenon that increasing the reward for a node may increase its service level at the cost of deteriorating the service level in other nodes.

After the initialization, a local search procedure is used that incorporates these cross-effects. Rather than using random search directions, we determine the most promising search directions by running a reduced simulation to evaluate the potential of increasing or decreasing a specific reward. A search direction is chosen proportional to the determined potential. For the sake of computational efficiency, the potential is updated after every $\eta$ iterations. For a chosen search direction, the actual objective is evaluated through the $\kappa$ full-size simulations. A new solution is accepted if it is an improvement over the current solution.

## 3 Preliminary results

We evaluate the performance of the developed reward strategy in a case study of the city of Washington D.C. The data has been taken from [5]. Historic bike-sharing demand is used to determine the itineraries of potential crowd-shippers and population density is used to determine the demand for small parcels. We consider rewards between 1$ and 6$ and a penalty $\rho$ of 10$ for each parcel. The obtained rewards after executing our algorithm are displayed in Figure 1. Compared to offering a constant reward of 6$ for every parcel, using the reward strategy that we described reduces the costs by 50% while only having a minor effect on the service level. The results indicate that higher rewards are given in the suburbs compared to those offered in the city center. The reason for this is the asymmetry between demand and supply. Most demand for parcels occurs in the suburbs, whereas most potential crowd-shippers are active in the city center. A detailed comparison between various reward strategies is included in the full paper.

## 4 Conclusion

We conclude that differentiating rewards across parcels has a large potential to improve the performance of the system in terms of cost and service level. By differentiating the

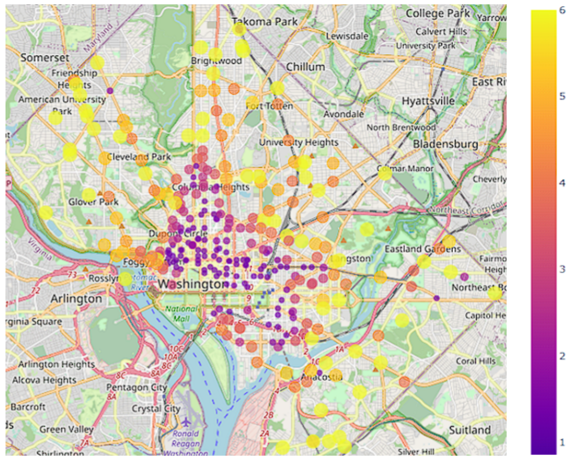



Figure 1: Rewards for every destination node. The color indicates the reward in dollars.

rewards according to the destination of a parcel, we obtain a cost reduction of 50%. The full paper focuses on extending the reward scheme based on the time of the day and the remaining number of parcels, such that the reward strategy is fully adaptive. In addition to this, detailed comparisons between the performance of existing and newly developed assignment and reward strategies in the Washington DC case study are made.

## References

[1] A. Ermagun and A. Stathopoulos. To bid or not to bid: An empirical study of the supply determinants of crowd-shipping. *Transportation Research Part A: Policy and Practice*, 116:468–483, 2018.

[2] K. Gdowska, A. Viana, and J. P. Pedroso. Stochastic last-mile delivery with crowd-shipping. *Transportation research procedia*, 30:90–100, 2018.

[3] T. V. Le and S. V. Ukkusuri. Modeling the willingness to work as crowd-shippers and travel time tolerance in emerging logistics services. *Travel Behaviour and Society*, 15:123–132, 2019.

[4] T. V. Le, S. V. Ukkusuri, J. Xue, and T. Van Woensel. Designing pricing and compensation schemes by integrating matching and routing models for crowd-shipping systems. *Transportation Research Part E: Logistics and Transportation Review*, 149:102209, 2021.

[5] P. Stokkink and N. Geroliminis. A continuum approximation approach to the depot location problem in a crowd-shipping system. *Transportation Research Part E: Logistics and Transportation Review*, 176:103207, 2023.

# The Bi-objective Electric Autonomous Dial-A-Ride Problem

**Yue Su**
CERMICS, Ecole des Ponts ParisTech, Champs-sur-Marne, France.
Email: yue.su@enpc.fr

**Sophie N. Parragh**
Institute of Production and Logistics Management,
Johannes Kepler University Linz, Linz, Austria.

**Jakob Puchinger**
Métis Laboratory, EM Normandie Business School, Clichy, France.

**Nicolas Dupin**
Univ Angers, LERIA, SFR MATHSTIC, F-49000 Angers, France.

22/11/2023

## 1 Introduction

The prevalence of ride-sharing services presents a fundamental trade-off between the operational costs and the users' convenience. While ride-sharing operations reduce operational costs, users may experience certain inconveniences, such as longer ride times when sharing their rides with others. Along with the development of ride-sharing services, the emergence of new techniques, such as electric vehicles and autonomous techniques, has drawn academic interest in operations research to apply a more eco-friendly and comfortable mode of transport. The Electric Autonomous Dial-A-Ride Problem (the E-ADARP) was first introduced by [1], which consists in designing a set of minimum-cost routes for a fleet of electric autonomous vehicles (EAVs) by scheduling them to provide ride-sharing services for users specifying their origins and destinations. In this work, we emphasize the conflicting interests of service providers and users in the objective function of the E-ADARP and investigate the Bi-objective E-ADARP (hereafter BO-EADARP), where the two objectives are the total travel time of all vehicles and the total excess user ride time of all users. We generalize a single objective branch-and-price (B&P) algorithm to the bi-objective case, relying on ideas of [2], to solve it. Numerical results and the managerial insights that we observe from the obtained efficient solutions are summarized.

## 2 The BO-EADARP Description

The problem is defined on a complete directed graph $G = (V, A)$, where $V$ represents the set of vertices and $A = \{(i, j) : i, j \in V, i \neq j\}$ the set of arcs. $V$ can be further partitioned into several subsets, i.e., $V = P \cup D \cup S \cup O \cup F$. $P$ and $D$ represent the set of all pickup and drop-off vertices, $S$ is the set of recharging stations, and $O$ and $F$ denote the set of origin depots and destination depots, respectively. Each user request is a pair $(i, n+i)$ for $i \in P$ and has a maximum user ride time of $m_i$. The travel time on each arc $(i, j) \in A$ is denoted as $t_{i,j}$. Detailed mixed-integer-linear program (MILP) of the E-ADARP can be found in [1]. We replace the weighted-sum objective function in [1] to separate objective functions, as follows:

$$\min \sum_{i,j \in V} t_{i,j} x_{i,j}^k \tag{1}$$

$$\min \sum_{i \in P} R_i \tag{2}$$

where $x_{i,j}^k$ is a binary decision variable which denotes whether vehicle $k$ travels from node $i$ to $j$. $R_i$ denotes the excess user ride time of request $i \in P$ and is formulated as the difference between the actual ride time and direct travel time from $i$ to $n+i$.

## 3 Methodologies

In this section, we first present the $\epsilon$-constraint method to solve the BO-EADARP, which is used to generate benchmark results. Then, we present the framework of the bi-objective branch-and-price (BOBP) algorithm.

### 3.1 Epsilon-constraint method

The $\epsilon$-constraint method starts by solving two objectives in lexicographical order with the single-objective B&P. To facilitate reading, we denote $z_1(x)$ as the value of the total travel time and $z_2(x)$ the value of the total excess user ride time for the solution $x$. In other words, we first solve $lex \min_{x \in \mathcal{X}}\{z_1(x), z_2(x)\}$ and then solve $lex \min_{x \in \mathcal{X}}\{z_2(x), z_1(x)\}$, with $\mathcal{X}$ representing the set of all feasible solutions. We use the term $lex \min_{x \in \mathcal{X}}\{z_1(x), z_2(x)\}$ to describe the process in which we find solutions with the smallest values for $z_2(x)$ among solutions in $\mathcal{X}$ that have the smallest values for $z_1(x)$, and similar for $lex \min_{x \in \mathcal{X}}\{z_2(x), z_1(x)\}$. The obtained non-dominated points $z^T$ and $z^B$ define the search area where other non-dominated points are included. The $\epsilon$-constraint method always optimizes one objective (e.g., $z_1(x)$) while the other is bounded by an $\epsilon$ value (i.e., $z_2(x) \leq \epsilon$). In each iteration, the $\epsilon$ value is updated with the $z_2(x')$, where $x'$ is the newly-found non-dominated solution.

By using the value of the other objective function to restrict the search iteratively, all the non-dominated points are obtained. The $\epsilon$-constraint method finishes when $z^B$ is reached.

### 3.2 The BOBP algorithm

The principle of the BOBP algorithm is extended from the single-objective B&P introduced in [3], which aims to divide the original problem into easier subproblems and store them in the form of *"nodes"*. We denote each subproblem of the BO-EADARP as $P(\eta)$, where $\eta$ represents the associated node. However, the BOBP algorithm is different from the single-objective case as lower bound and upper bound sets (instead of single numerical values) are used to decide whether to fathom a node. The main ingredients of the BOBP algorithm are presented as follows:

- **Calculate lower bound set and update upper bound set:** On each branch-and-bound node, we calculate the lower bound set with the dichotomic method. To solve each weighted-sum objective problem, the CG algorithm presented in [3] is applied. Once the lower bound set of the analyzed node $\eta$ (denoted as $\mathcal{L}(\eta)$) is calculated, we first check if new non-dominated points are obtained. If this is the case, the upper bound set $\mathcal{U}$ is updated.

- **Lower bound filtering and node fathoming:** Then, the lower bounds in the set are filtered with the current upper bound set $\mathcal{U}$, which stores each candidate point that corresponds to the integer solution that is not dominated by other points in the set. The *filtering* process compares the current $\mathcal{L}(\eta)$ with $\mathcal{U}$ and returns a set of non-dominated portions. If no portion is generated after the filtering process, then the analyzed node $\eta$ can be fathomed, as it is fully dominated by the current upper bound set $\mathcal{U}$.

- **Branching procedure:** If the analyzed node cannot be fathomed, branching is applied to generate child nodes. We consider three kinds of branching strategies and apply them to each disjoint non-dominated portion. After branching, a set of child nodes is added to the unprocessed node set $\mathcal{T}$.

The tree search terminates when there is no unprocessed node remaining in $\mathcal{T}$, and we have the set of non-dominated points $\mathcal{Y}_N$ equals to $\mathcal{U}$.

## 4 Numerical Experiments and Discussion

In this work, we solve the BO-EADARP, where the total travel time and the total excess user ride time are considered as two separate objectives. The BO-EADARP is more difficult to solve than the E-ADARP, as one must fully explore the bi-dimension search area in order to demonstrate the completeness of the Pareto front. To tackle the BO-EADARP, we introduce one criterion space search algorithm (i.e., the $\epsilon$-constraint) and a decision space search algorithm (i.e., the BOBP algorithm). The BOBP algorithm is based on the generalized branch-and-bound algorithm proposed in [2], where the lower bound set is calculated by the CG algorithm ([3]). In the computational experiments, we solve the BO-EADARP with two different algorithms on small-to-medium-sized instances and we compare the generated efficient solutions and their average computational time from different algorithms. Compared with the classic $\epsilon$-constraint method, the BOBP algorithm seems to be more efficient and generates more efficient solutions in a less average computational time. Then, we analyze the obtained efficient solutions, which offer the following managerial insights for different service providers: (1) for profitable service providers, it is possible to significantly improve service quality while keeping near-optimal operational costs; (2) for non-profitable service providers, there exist efficient solutions of high service quality while at lower operational costs. These efficient solutions are very interesting for this kind of service provider. To sum up, the obtained efficient solutions can help decision-makers select Pareto-optimal transportation plans according to their priorities and preferences.

## References


[1] C. Bongiovanni, M. Kaspi and N. Geroliminis,“The electric autonomous dial-a-ride problem”, *Transportation Research Part B : Methodological* 122, 436-456, 2019.

[2] S. N. Parragh and F. Tricoire, “Branch-and-bound for bi-objective integer programming”, *INFORMS Journal on Computing* 31(4), 805-822, 2019.

[3] Y. Su, N. Dupin, S. N. Parragh and J. Puchinger, “A Column Generation Approach for the Electric Autonomous Dial-a-Ride Problem”, arXiv preprint arXiv:2206.13496, 2022.

# Sustainable hub location under uncertainty

**Gita Taherkhani**
Quinlan School of Business, Loyola University Chicago, USA
Email: gtaherkhani@luc.edu

**Mojtaba Hosseini**
Tippie College of Business, University of Iowa, Iowa, USA
Email: hosseini@uiowa.edu

**Sibel A.Alumur**
Department of Management Science and Engineering, University of Waterloo, Canada
Email: sibel.alumur@uwaterloo.ca

## 1 Introduction and Problem Definition

The trucking industry is the lifeblood of the economy since almost every good that is consumed is put on a truck at some point. As a result, the trucking industry hauled 72.2% of all freight transported in the United States in 2021 alone, equating to 10.93 billion tons (ATA 2022). The industry makes this possible by widely employing a consolidation-based hub network structure. In a hub network, the demand between origin-destination pairs is routed through hubs instead of using point-to-point connections. The aim is to save the cost of building a network connecting many origins to many destinations with a fewer number of links and also to exploit economies of scale by consolidating flows at hubs.

In the era of climate change action, transportation is clearly one of the prime targets for reducing greenhouse gas (GHG) emissions. The United States Environmental Protection Agency (EPA) reports that GHG emissions from transportation account for about 27% of total GHG emissions, making it the largest contributor to U.S. GHG emissions. Hence, improving the environmental footprint in the transportation and logistics sector is a necessity. In the transportation sector, carbon dioxide (CO2) emissions are the primary GHG emitted by vehicles. Accordingly, carbon pricing is a widely used tool for deriving such improvements aiming to reduce the amount of GHG released into the atmosphere. Options for pricing carbon in the transportation sector include a carbon tax and a cap on carbon dioxide emissions ([2] and [4]). As can be understood from their names, the carbon tax puts a direct fee on CO2 emissions, and the carbon cap places a limit on the estimated amount of CO2 emissions for the entire transportation of the shipment.

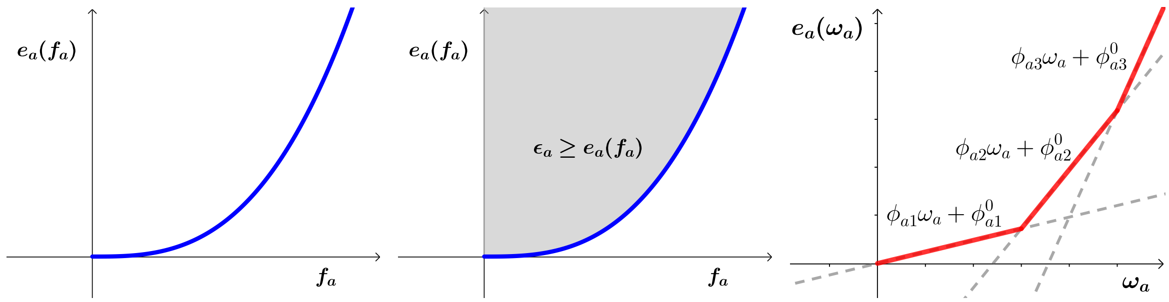
According to [5], several countries and sub-national jurisdictions, including regions or states within the United States, have adopted carbon pricing policies as a means to mitigate

GHG emissions. In particular, carbon cap programs are widely used, with significant examples found in the European Union, the state of California, and the Northeast states of the USA. On the other hand, carbon taxes are gaining popularity and have been implemented in various regions, including several Canadian provinces, Sweden, and the United Kingdom.

The amount of carbon emissions in a transportation network depends on the total demand routed on each connection which is in turn directly influenced by the design of the network. Accordingly, in this study, we address the design of sustainable hub networks incorporating factors related to carbon pricing. In particular, we focus on the design of hub networks for less-than-truckload (LTL) transportation considering both a carbon tax and a carbon cap on emissions. Our focus is directed toward the intricate task of designing hub networks for common carriers. Within the operations of an LTL common carrier, hubs assume a pivotal role in the process of consolidating and deconsolidating freight of various commodities. Those common carriers offer a set of routes and rates to accommodate the demand for commodities to be transported from multiple origins to multiple destinations. Accordingly, the design problem addresses the determination of the locations of hubs, allocation of demand nodes to these hubs, and choosing optimal routes of flow between origin-destination (O-D) pairs regarding the available capacities of hubs while not exceeding the carbon cap set for the network. We study a multiple allocation setting where there is no limit on the maximum number of hubs to which a node can be allocated. We consider a profit-maximizing setting in which a portion of the demand of O-D pairs can remain unserved depending on the trade-off between profit, cost, and carbon emissions.

We develop a model in which, in addition to transportation and hub installation costs, the carbon tax is also explicitly included in the objective function. Moreover, to ensure that the total amount of carbon emissions emitted by trucks does not exceed a predefined cap, we incorporate a constraint limiting the emissions on the entire transportation network. We model the total carbon emissions on each arc with a generic convex function of the total demand routed on that arc. We then assume that these functions can be approximated as piecewise linear functions (see Figure 1).). This allows us to accurately model and analyze the relationship between demand and carbon emissions while maintaining computational tractability.

In practice, there is often a significant delay between the design of the network and its implementation. During this delay, various sources of uncertainty can affect the model's parameters. Particularly, demands may deviate from the expected value, either being higher or lower. Thus, incorporating uncertainty into the decision process is a necessity which is especially important in our setting since the amount of carbon emissions is a function of the satisfied demand. In order to address this issue and offer a more reliable solution framework for this problem, we take the demand values under uncertainty. To effectively handle uncertainty, we assume demand is stochastic where there is a known probability distribution that describes its behavior. We model the problem as a two-stage stochastic program. In the first stage, we consider the strategic location decisions, as demand variations will not influ-




(a) Smooth $e_a$ (b) Epigraph of $e_a$ (c) Piecewise linear $e_a$

Figure 1: A convex smooth emission function, its epigraph, and piecewise linear approximation.

ence these long-term decisions. While, the tactical decisions, which involve allocations and determining the optimal routes of flows through the network, are made in the second stage.

## 2 Solution Methodology and Computational Results

Our two-stage optimization problem is formulated as follows: $\max_{\mathbf{y}\in Y} \mathbf{g}^\top \mathbf{y} + \mathbb{E}_\psi[Q(\mathbf{y},\psi)]$, where $\mathbf{y}$ represents the selected hubs, $Y \subseteq \{0,1\}^n$ denotes the domain of $\mathbf{y}$, and $Q(\mathbf{y},\psi)$ stands for the revenue accrued under demand scenario $\psi$. Since evaluating $\mathbb{E}_\psi[Q(\mathbf{y},\psi)]$ involves assessing a large number of demand scenarios, we approximate it using a Monte Carlo simulation-based algorithm known as the Sample Average Approximation (SAA) scheme ([3, 1, 6], and [7]). In particular, we generate a sufficiently large set of random demand scenarios $\{\psi_1,\psi_2,\ldots,\psi_N\}$ and solve a MILP of the form $\max_{\mathbf{y}\in Y} \mathbf{g}^\top \mathbf{y} + \frac{1}{N}\sum_{n=1}^N Q(\mathbf{y},\psi_n)$. This procedure is repeated for a predetermined number of replications. Although solving each MILP in every replication is necessary, they are large-scale, and off-the-shelf solvers may struggle to handle them within a reasonable timeframe. Therefore, to address this challenge and exploit their combinatorial substructures, we propose a Benders decomposition (BD) algorithm accelerated with several enhancement techniques.

We conduct comprehensive computational experiments to assess the efficiency of our algorithm. The algorithm was able to solve stochastic instances with up to 1600 commodities optimally using a sample size of 25 with 20 replications. This implies that the algorithm is capable of providing optimal solutions for practical-sized instances with reasonable sample size and replications demonstrating its effectiveness. Moreover, the implemented acceleration methodologies achieved an average computation time reduction of up to three times confirming their efficiency.

We analyze the impact of carbon pricing factors on the resulting hub networks as well as on total profit across various parameter settings. The results and the analysis highlighted the importance of incorporating carbon pricing factors into the problem setting in the design of optimal hub networks to maximize profit while also effectively controlling environmental footprints. In our computational experiments, the carbon cap turned out to be the most influential metric in determining the optimal solutions, particularly the percentage of satisfied demand and total profit, compared to carbon tax and the emission function. In all of our test instances, the total emission reached the maximum allowed carbon emissions for the network.

We further compare the optimal solutions obtained using two emission functions with different policies and the results show that there is a slight increase in profit and the percentage of satisfied demand when utilizing a function with a looser policy in contrast to a function with a tighter policy. We also observe that the optimal solutions yield denser hub networks with increased allocation connections under tight emission functions compared to those with loose emission functions to mitigate congestion and avoid traffic concentration on fewer links.

## References

[1] Y. Adulyasak, J.-F. Cordeau, and R. Jans. Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867, 2015.

[2] G. Andersen. Capping carbon. *National Conference of State Legislatures, 2009*, 2009.

[3] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[4] B. Leard, J. Linn, and K. Cleary. Pricing carbon in the transportation sector. *Resources for the future*, 2020.

[5] Resources for the Future. `https://www.rff.org/topics/carbon-pricing/`, 2023.

[6] G. Taherkhani, S. A. Alumur, and M. Hosseini. Benders decomposition for the profit maximizing capacitated hub location problem with multiple demand classes. *Transportation Science*, 54(6):1446–1470, 2020.

[7] G. Taherkhani, S. A. Alumur, and M. Hosseini. Robust stochastic models for profit-maximizing hub location problems. *Transportation Science*, 55(6):1322–1350, 2021.

# Exact solution methods for an integrated multi-stakeholder freight transportation system with stochastic demand

**Gita Taherkhani[*], Mojtaba Hosseini[†], Ali Hassanzadeh[‡]**

## 1 Introduction and Problem Definition

Freight transportation is a preeminent part of all supply chain and logistics systems and plays an important role in the social, environmental, and economic developments of our society. Before the COVID–19 pandemic, United States transportation system moved approximately 51 million tons of goods worth $51.8 billion every day, which translates to 56.9 tons of freight for every resident of the United States [6]. Hence, improving the Transportation and Logistics (TL) sector and stakeholder efficiency and profitability along with environmental footprint is a necessity.

In this study, we explore the Many-to-One-to-Many (M1M) system, an integrated decision-making structure engaging multiple stakeholders.Shippers - producers, wholesalers, and distributors - seek efficient transportation for their goods. On the other hand, carriers, including service providers and terminal operators, aim for profitable loads within their capacity. To simultaneously address both shippers' and carriers' requests, we introduce an Intelligent Decision Support Platform (IDSP) aimed at optimizing the overall platform profit. The IDSP is tasked with coordinating transportation activities with an approach that addresses the dynamic nature of shipper requests and carrier offers, facilitating optimal planning and execution across both time and space dimensions.

We explore revenue management concepts in this application by focusing on network capacity allocation, assigning service capacities to shipper requests, and distinguishing between contract-based and non-contracted shippers. Contract-based shippers have demands that must be fully met, whereas non-contracted shippers' demands may be partially fulfilled based on profitability and service availability. We also consider the flexibility of the IDSP to satisfy demands outside the preferred time windows of shippers, subject to penalties.

Our work aims to develop an effective tool for tactical-level transportation planning within the M1M system, focusing on operations over a specific time period known as the schedule length (e.g., a week), which then can be frequently executed for a longer planning horizon

[*]Quinlan School of Business, Loyola University Chicago, IL. Email: gtaherkhani@luc.edu
[†]Tippie College of Business, University of Iowa, IA. Email: mojtaba-hosseini@uiowa.edu
[‡]Alliance Manchester Business School, University of Manchester, UK. Email: ali.h@manchester.ac.uk

(e.g., a season). There is a broad stream of research on the tactical planning performed by consolidation–based transportation carriers. For comprehensive insights into tactical planning, see the survey papers by [1], and [3] on the subject.

In our research, we aim to optimize the M1M system through tactical planning by aligning demand and supply with the goal of maximizing its profitability. On the demand side, we focus on selecting and fulfilling profitable requests from non-contract shippers, alongside all requests from contract-based shippers. On the supply side, our goal is to utilize carrier services efficiently to meet these demands cost-effectively. Tactical planning also involves designing itineraries for shipping requests, detailing their origin, destination, pick-up and delivery times, and the logistics of movement, including transfers and consolidations, within the service network.

While the majority of studies assume perfect information readily availability for tactical decisions [5], the reality often involves a delay between planning and execution, introducing uncertainty in parameters, in particular, shipper demands. Recognizing this, our approach incorporates uncertainty by assuming a known probability distribution for demand variability. This approach aims to enhance the M1M system's tactical planning reliability by comparing solutions under uncertain demand with those based on deterministic demand, thus acknowledging and addressing the real-world challenge of information imperfection and its impact on operational planning.

We develop a Scheduled Service Network Design (SSND) platform as the methodology of choice for the M1M system's tactical planning. This platform addresses time-dependent shipper demands and carrier services, utilizing a time-space network for modeling events in time and location dimensions. Our model maximizes the profit over the horizon while minimizing the expected cost of adjusting the plan at each realization of uncertain parameters. We propose a two–stage stochastic program with recourse. The first stage focuses on selecting shipper demands and carrier services to define the network structure, paths, and capacities. The second stage deals with demand distribution and planning shippers' itineraries across the network, enhancing operational efficiency and strategic flexibility.

## 2 Solution Methodology

Our two-stage optimization problem can be stated as $\max_{\mathbf{y}\in Y} \mathbf{f}^\top \mathbf{y} + \mathbb{E}_\xi[Q(\mathbf{y},\xi)]$ where $\mathbf{y}$ determines the selected services, $Y \subseteq \{0,1\}^n$ is the domain of $\mathbf{y}$, and $Q(\mathbf{y},\xi)$ is the revenue accrued under demand scenario $\xi$. As measuring $\mathbb{E}_\xi[Q(\mathbf{y},\xi)]$ requires assessing a large number of demand scenarios, we approximate $\mathbb{E}_\xi[Q(\mathbf{y},\xi)]$ via a Monte-Carlo simulation-based algorithm, known as Sample Average Approximation (SAA) scheme. Briefly, we generate a sufficiently large set of random demand scenarios $\{\xi_1, \xi_2, \dots, \xi_S\}$ and solve a MILP of the form $\max_{\mathbf{y}\in Y} \mathbf{f}^\top \mathbf{y} + \frac{1}{S}\sum_{s=1}^{S} Q(\mathbf{y},\xi_s)$. We then replicate this procedure by producing new sets of demand scenarios for a predetermined number of replications. While at each replication, we need to solve a MILP, they are still large-scale and an off-the-shelf solver may not be able to handle them in a reasonable time. Therefore, to solve each of these MILPs and exploit their combinatorial substructures, we propose a Benders decomposition (BD) algorithm accelerated with several enhancement techniques.

Our implementation of BD consists of three phases: (i) preprocessing, (ii) root node processing, (iii) branch-and-cut. In the preprocessing phase, we propose a novel preprocessing and bounding scheme that allow us to eliminate several infeasible services. In addition, we produce combinatorial cuts based on cut-set inequalities as well as network connectivity inequalities to help the master problem produce feasible solutions. Given that some of the arcs in our network structure are holding arcs and the O-D pairs do not have pre-specified times, these inequalities are different than the cuts introduced in the literature [7]. In the second phase, we process the root node through an in-out scheme [4] which allows us to find the root node solution in very few iterations. Finally, in the B&C phase, we employ partial decomposition [2] by retaining a few scenarios in the master problem. This proved effective in solving the subproblems for more promising solutions. In addition, we introduce a novel unified cut lifting procedure for strengthening both feasibility and optimality cuts. Furthermore, exploiting the repetitive nature of the SAA scheme coupled with our Benders decomposition algorithm, we further develop acceleration techniques to improve the CPU times by enhancing the convergence of the algorithms.

# 3 Computational Experiments

To perform numerical experiments, we generate several problem instances based on real freight transportation networks. In generating these instances, we follow a systematic approach by making sure we have a balanced set of instances in terms of the complexity of the underlying network, the number of services, and requests among other aspects. We generate a base instance with a fixed network structure and a minimal number of offered services and generated requests. Then we expand these base instances by increasing one or more of the following parameters: i) number of requests, ii) number of services, and iii) capacity of the services.

We perform extensive computational experiments to evaluate the effectiveness of our proposed solution method in terms of running time as well as providing a proof-of-concept of the model. For instance, as illustrated in Figure 1, CPLEX quickly hits the time/memory limit as the number of demand scenarios increases, while our BD implementation scales almost linearly with the number of scenarios. We also highlight the impact of uncertainty on the structure and behavior of the optimal solutions, including selected shippers and the design of the service network (amount of satisfied demand, consolidation and the net platform profit).

# References

[1] T. G. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288, 2000.

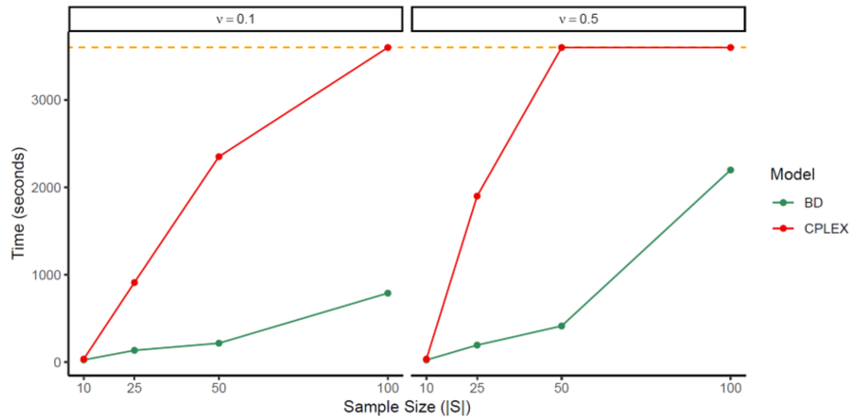



Figure 1: Performance of BD and CPLEX on solving stochastic instances with different sample sizes ($|S|$) and coefficients of variation ($\nu$). The orange horizontal line shows the time limit or hitting the memory limit.


[2] T. G. Crainic, M. Hewitt, F. Maggioni, and W. Rei. Partial Benders decomposition: general methodology and application to stochastic network design. *Transportation Science*, 55(2):414–435, 2021.

[3] T.G. Crainic and M. Hewitt. Service Network Design. In T.G. Crainic, M. Gendreau, and B. Gendron, editors, *Network Design with Applications in Transportation and Logistics*, chapter 12, pages 347–382. Springer, Boston, 2021.

[4] M. Fischetti, I. Ljubić, and M. Sinnl. Redesigning Benders decomposition for large-scale facility location. *Management Science*, 63(7):2146–2162, 2017.

[5] M. Hewitt, W. Rei, and S. Wallace. Stochastic Network Design. In T.G. Crainic, M. Gendreau, and B. Gendron, editors, *Network Design with Applications in Transportation and Logistics*, chapter 12, pages 283–315. Springer, Boston, 2021.

[6] P. Hu, R. R. Schmitt, J. Schwarzer, W. H. Moore, et al. Transportation statistics annual report 2021. *United States. Department of Transportation. Bureau of Transportation Statistics*, 2021.

[7] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. Accelerating the Benders decomposition method: Application to stochastic network design problems. *SIAM Journal on Optimization*, 28(1):875–903, 2018.

# A column generation approach to solve the Joint Order Batching and Picker Routing Problem with picker congestion

**Pablo Torrealba-González**
Univ. Lille, CNRS, Inria, Centrale Lille,
UMR 9189 CRIStAL, F-59000 Lille, France
Email:pablo.torrealba-gonzalez@inria.fr

**Dominique Feillet**
Mines Saint-Etienne, Univ. Clermont Auvergne,
LIMOS UMR CNRS 6158, Centre CMP, Gardanne F-13541, France

**Maxime Ogier, Frédéric Semet**
Univ. Lille, CNRS, Inria, Centrale Lille,
UMR 9189 CRIStAL, F-59000 Lille, France

## 1 Introduction and motivation

In warehouse operations, order picking stands out as one of the most critical processes, with its primary objective being the preparation of customer orders. It represents the majority of total operational costs. Warehouses can operate in different ways depending on the level of technology. Operations may involve humans, robots, or both, as presented in [3]. This work focuses on a classic picker-to-parts system, where pickers push a trolley around the warehouse to collect different items.

Despite the increasing popularity of robotic operations in recent years, many companies still employ human operators for several reasons. Firstly, humans offer flexibility by easily adapting to changes in operations such as highly fluctuating demand. Additionally, the high initial investment required to purchase infrastructure discourages the adoption of automated systems.

From an operational point of view, the main decisions are order batching and picker routing. The Order Batching Problem (OBP) deals with assigning different customer orders to a given set of pickers, and the Picker Routing Problem (PRP) is defined as determining, for a single picker, the sequence in which the products to be picked are collected. Both problems have as objective the minimization of the traveled distance or time. When solving the OBP with an optimal routing policy, the focus shifts to integrating both OBP and PRP. This integrated problem, known as the Joint Order Batching and Picker Routing Problem (JOBPRP), is usually solved using a column generation based approach as proposed in [2] or [4].

In warehouses, avoiding congestion is challenging and carries significant consequences for cost, performance, and safety [1]. The existing literature about congestion in warehouses mainly focuses on evaluating congestion by simulation in the context of narrow aisles or picker blocking. When solving the JOBPRP, works in the literature usually assume a no congestion situation. However, in practice, congestion is a very well known phenomenon in warehouses with human pickers. In this work, we propose a solution method for the JOBPRP that considers congestion aspects.

## 2 Modeling congestion and problem formulation

Given the complexities of human behavior, precise coordination of pickers does not seem relevant. Therefore, we propose a rough estimation of congestion levels by introducing time discretization, and dividing the planning horizon into time intervals. In our analysis, we examine a rectangular warehouse layout featuring parallel cross and vertical aisles. Vertical aisles contain the products, and cross-aisles serve as navigation paths for pickers throughout the warehouse. In each time interval, an extra delay is determined by using a function based on the number of pickers in each sub-aisle (the intersection of the space between two consecutive cross-aisles and a vertical aisles). The delay is applied to all pickers in that sub-aisle. It is crucial to note that congestion computation is typically nonlinear, as delay in one period leads to additional delay in the future. Moreover, including congestion in time minimization can result in undesirable situations. For instance, an optimal solution could consider picker waiting times, or a picker walking a distance longer than the shortest path between two consecutive picking locations. Such situations are not realistic in practice and should be avoided in a feasible solution of JOBPRP when considering congestion.

In particular, we consider a rectangular warehouse with $v$ vertical aisles and $b$ blocks, a set $\mathcal{K}$ of pickers with a trolley capacity $C$. To tackle the problem, we proposed an exponential Mixed Integer Programming (MIP) formulation. In this formulation, variables (or picker routes) are depicted as a sequence of locations followed by the picker, along with the time each location is visited and the level of congestion in each visited sub-aisle and time intervals. As the timing on the routes depends on the levels of congestion, the mathematical formulations guarantee to select routes that are compatible (have same levels of congestion), ensuring that each order is collected without using more than the available pickers.

## 3 A column generation heuristic

To solve the JOBPRP with picker congestion and provide a performance guarantee (lower and upper bounds), we propose a heuristic based on column generation. The linear relaxation of the extended MIP formulation is solved through a column generation approach. In each iteration, the pricing problem is tackled using a labeling algorithm. Note that column generation approaches for the JOBPRP without congestion do not use labeling algorithm techniques to solve the pricing problem since the order requirement (all products of an order must be collected in the same route) is challenging to consider. However, when considering congestion, the timing aspect does not permit easily adapting the existing pricing solvers in the literature. Using a labeling algorithm permits easily incorporating timing aspects and non-linearity of congestion.

The labeling algorithm relies on representing partial solutions as a set of attributes denoted as a label. Labels are propagated from a picking location to the next one until the represented solution reach the depot. To speed up the process and avoid evaluating every potential label, two main components have been introduced. A completion bound, derived from optimistic dual values, and a time bound obtained from the analysis of an optimal solution of the JOBPRP are proposed.

As mentioned earlier, congestion involves a nonlinear component that can lead to undesirable situations. As a main advantage, the labeling algorithm enables us to address these complexities by precisely computing the congestion value and discarding the exploration of labels with undesirable situations. The column generation algorithm starts with an initial set of columns derived from the optimal solution of the classic JOBPRP. At the end of the column generation process, we obtain a lower bound, which can be derived from a lower bound of the solution of the JOBPRP, a Lagrangian bound, or the optimal solution of the linear relaxation of the extended formulation. Finally, an upper bound is obtained by solving the integer problem with the columns generated so far.

## 4 Computational experiments

We generated a new benchmark of instances, considering scenarios of interest in terms of congestion. Table 1 presents preliminary results obtained by running our Column Generation Heuristic (CGH) with a time limit of 40 minutes. The first six columns report information about the instances: *ID*: an identifier, $v$: the number of vertical aisles, $b$: the number of blocks, $|\mathcal{K}|$: the number of pickers, $C$: the capacity of each trolley, $|\mathcal{OLN}|$: total number of order lines (for a given order an order line indicates a product to be collected). The next two columns report information about an initial solution, obtained by solving the JOBPRP with no congestion. $Z^*_J$ is the optimal objective, and $UB_J$ is an upper bound obtained by computing the congestion in the JOBPRP solution. The last columns report

| Instance | | | | | | Initial Solution | | Column generation heuristic | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | $v$ | $b$ | $\lvert\mathcal{K}\rvert$ | $C$ | $\lvert\mathcal{OLN}\rvert$ | $Z^*_J$ | $\text{UB}_J$ | $\text{LB}_{CG}$ | $\text{UB}_{CG}$ | opt gap | imp |
| 1 | 12 | 1 | 6 | 9 | 30 | 594 | 628 | 594 | 596.8 | 0.5% | 5.0% |
| 2 | 6 | 3 | 4 | 20 | 35 | 656 | 714.4 | 656 | 691.4 | 5.1% | 3.2% |
| 3 | 12 | 1 | 8 | 9 | 37 | 834 | 878.7 | 834 | 838.4 | 0.5% | 4.6% |
| 4 | 6 | 3 | 6 | 14 | 43 | 814 | 853.1 | 814 | 817.6 | 0.4% | 4.2% |
| 5 | 12 | 2 | 10 | 9 | 49 | 1430 | 1505 | 1430 | 1484.9 | 3.7% | 1.3% |
| 6 | 16 | 2 | 10 | 9 | 49 | 1954 | 2084.4 | 1965 | 1979.6 | 0.7% | 5.0% |
| 7 | 6 | 3 | 10 | 9 | 49 | 1328 | 1431.5 | 1328 | 1402.5 | 5.3% | 2.0% |
| 8 | 12 | 1 | 6 | 20 | 57 | 914 | 964.3 | 914 | 927.7 | 1.5% | 3.8% |
| 9 | 6 | 3 | 6 | 20 | 57 | 1054 | 1128.5 | 1054 | 1102.9 | 4.4% | 2.3% |
| 10 | 16 | 2 | 6 | 20 | 59 | 1744 | 1838.5 | 1744 | 1746.3 | 0.1% | 5.0% |
| 11 | 12 | 1 | 5 | 30 | 92 | 994 | 1045.4 | 994 | 1004.4 | 1.0% | 3.9% |
| 12 | 12 | 1 | 5 | 68 | 219 | 1644 | 1769.3 | 1644 | 1690.4 | 2.7% | 4.5% |

Table 1: Preliminary results

information about CGH: $LB_{CG}$ and $UB_{CG}$ report the lower and upper bounds, *gap* is the optimality gap and *imp* reports the improvement of the solution w.r.t. the initial solution.

The results show the ability of the algorithm to improve the initially provided solution and reduce the initial optimality gap. The algorithm is strongly dependent on the number of order lines, limiting its performance in large-sized instances. Although there are instances where the initial lower bound of the JOBPRP can be improved, its computation requires solving the optimality pricing problem, representing a significant challenge. Future research aims to evaluate the approach on larger instances and enhance the labeling prioritization to accelerate the production of promising columns.

## References


[1] F. Ciarallo B. Heath and R. Hill. An agent-based modeling approach to analyze the impact of warehouse congestion on cost and performance. *The International Journal of Advanced Manufacturing Technology*, 67(1-4):563–574, October 2012.

[2] O. Briant, N. Catusse A. Ladier H. Cambazard, D. Cattaruzza, and M. Ogier. An efficient and general approach for the joint order batching and picker routing problem. *European Journal of Operational Research*, 285(2):497–512, September 2020.

[3] M. Melacini G. Marchet and S. Perotti. Investigating order picking system adoption: a case-study-based approach. *International Journal of Logistics Research and Applications*, 18(1):82–98, August 2014.

[4] J. Wahlen and T. Gschwind. Branch-price-and-cut-based solution of order batching problems. *Transportation Science*, 57(3):756–777, 2023.

# Machine-Learning-Based Prediction of Multi-Compartment Vehicle Fleet Performance

**Christian Truden**

*Department of Operations, Energy, and Environmental Management,*
*University of Klagenfurt, Klagenfurt, Austria*
*Email: christian.truden@aau.at*

**Mike Hewitt**

*Quinlan School of Business,*
*Loyola University Chicago, Chicago, IL, USA*

## 1 Introduction

In this work, we present a method to perform a comprehensive analysis of the fleet composition problem [1] that is suitable for most variants of the vehicle routing problem. Its basic principle is to estimate a fleet's performance by using the company's delivery planning tools in a black-box fashion. In a case study, we analyze the fleet size and mix for a fictional grocery home delivery service. A fleet comprising multi-compartment vehicles is employed, where each compartment is designated for storing groceries at specific temperature zones tailored to their storage requirements.

In general, the stakeholders are interested in finding a fleet configuration that enables good performance regarding defined key performance indicators (KPIs). Seasonal demand changes occur in nearly all types of routing applications. Therefore, we aim to identify fleet configurations that ensure consistent and satisfactory performance across all seasons. We do not propose a methodology for choosing a fleet. This is because stakeholders may consider multiple KPIs when making fleet composition decisions, and these KPIs may be conflicting and vary by scenario. Thus, we focus on a method for predicting the values of multiple KPIs for a given fleet.

## 2 Regression-Based Approximation of the Fleet Performance

Establishing a functional dependency between i) expected demand, ii) fleet configuration, and iii) KPIs is a well-established approach when interested in finding *high-quality* policies concerning delivery problems on the operational level, e.g., [5, 3]. Such a relationship, formally expressed by a regression model, can also provide decision support for tactical considerations like the fleet size and mix.

For any considered KPI we fit a (regression) model that explains the relationship between the seasonal parameters and the fleet configurations. A linear regression model, for example, to explain a KPI can be written in the following form (coefficients and error terms are omitted here)

$$\text{KPI} \sim \underbrace{\alpha_1 + \ldots + \alpha_L}_{\textit{seasonal parameters}} + \underbrace{n_1 + \ldots + n_V}_{\textit{fleet configuration}}$$

Here, the seasonal parameters shall describe how the demand deviates from the annual average encompassing factors such as the quantity, size, and composition of orders. On the other hand, the fleet configuration is characterized solely by the count of vehicles across different types considered for the fleet.

In the absence of historical observations for all the fleet configurations and seasons under consideration, we propose a method to generate vehicle routing instances based on the available historical customer order data. Initially, we sample orders from the historical data set and then adjust these sampled orders to align with the defined seasons. Subsequently, we construct a collection of plausible fleet configurations. Combining the created orders with these fleet configurations forms a collection of routing problems, each of which is then solved using the decision maker's preferred delivery planning tool. This process is repeated for all possible combinations, enabling us to calculate the KPIs based on the solutions obtained. This process is the foundational data source for subsequent analyses or evaluations.

## 3 Case Study

In a case study centered around a fictional grocery home delivery service, we aim to demonstrate how our approach can function as an effective decision-support tool. Within this scenario, we have identified four KPIs for evaluation: i) service level (SL): no. accepted order / no. all orders placed; ii) cost per item (CI): no. delivered grocery items / total delivery cost; iii) capacity utilization (CU): storage capacity required for the accepted orders / total storage capacity of the fleet; and iv) annual fleet ownership cost (AC): annual cost of registration, maintenance, and lease of the fleet. We apply a variant of the well-established adaptive large neighborhood search (ALNS) to solve the occurring vehicle routing problems with time windows and three-dimensional capacity constraints.

The locations in the ORTEC VRPTW instances [4], derived from a real US-based grocery delivery service, will form the location library of our fictional delivery service. The orders, conceptualized as three-dimensional shopping baskets, are derived from a grocery shopping dataset comprising 3 million grocery orders [2]. We consider three types of vehicles in this study: small, medium, and large vans.

The resulting data set comprises five seasons (baseline, spring, summer vacations, fall, and holidays) and 593 fleet configurations, which were selected after eliminating obviously impractical configurations in terms of loading capacity. This results in a total of 2965 combinations for which we create 100 samples each.

Our initial analysis of created samples examines how fleet capacity influences SL. We aggregate capacities and AC of the three considered vehicle types (small, medium, and large). The scatter plots (Figure 1) illustrate the relationship between SL and the aggregated capacity and AC of the fleet compositions. A positive correlation between fleet size and service level is evident. We find that similar ownership costs (AC) or total capacity can lead to markedly different SL values, even within the same season. This emphasizes the need for a thorough analysis of fleet composition to achieve optimal performance.

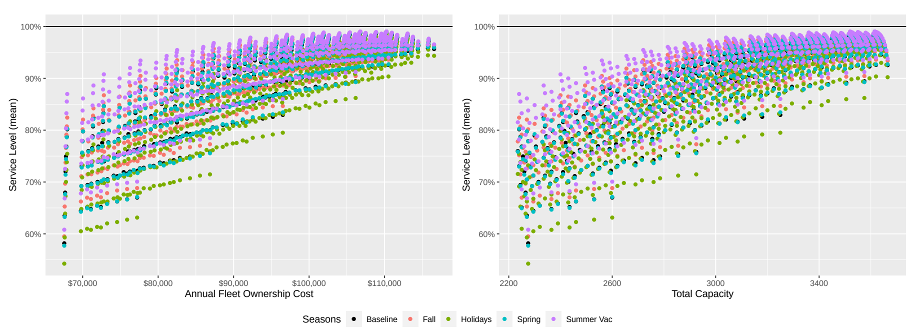



Figure 1: Scatter-plot showing the service level (SL) (y-axis) in respect to the annual fleet ownership cost (AC) and the total capacity (x-axis).

We find that a quadratic regression model works best for SL and CU using the Akaike information criterion (AIC). These models include the number of vehicles of each type, $n_{\text{small}}$, $n_{\text{medium}}$, and $n_{\text{large}}$. Additionally, the parameter $\alpha$ signifies the seasonal variation in order volume, whereas $\beta$ characterizes the fluctuations in the size of the orders. In Table 1, we present the complete model specifications. The models demonstrate that incorporating the seasonal multipliers $\alpha$ and $\beta$ is indeed a suitable and justified approach. As expected, the number of vehicles of each type explains the resulting SL and CU values quite well. While the linear terms have the largest coefficients, the quadratic terms give a sufficiently good correction to compensate for the ceiling effects that occur for SL and CU.

Table 1: Summary of the regression models explaining service level (SL) and capacity utilization (CU). Reported are the coefficients, standard errors (in brackets), significance levels, and goodness-of-fit statistics of statistical models.

| | SL | CU | | SL | CU |
|---|---|---|---|---|---|
| $n_{\text{small}}$ | $0.1463^{***}$ | $0.0671^{***}$ | $\alpha^2$ | $-0.4061^{***}$ | $-11.6804^{***}$ |
| | (0.0003) | (0.0004) | | (0.0224) | (0.2714) |
| $n_{\text{medium}}$ | $0.1990^{***}$ | $0.0764^{***}$ | $\beta^2$ | $0.5031^{***}$ | $-9.1052^{***}$ |
| | (0.0004) | (0.0006) | | (0.0237) | (0.1241) |
| $n_{\text{large}}$ | $0.2148^{***}$ | $0.0590^{***}$ | $n_{\text{small}}{:}n_{\text{medium}}$ | $-0.0096^{***}$ | $-0.0065^{***}$ |
| | (0.0006) | (0.0007) | | (0.0000) | (0.0000) |
| $\alpha$ | $0.3841^{***}$ | $2.6847^{***}$ | $n_{\text{small}}{:}n_{\text{large}}$ | $-0.0101^{***}$ | $-0.0062^{***}$ |
| | (0.0450) | (0.1598) | | (0.0000) | (0.0000) |
| $\beta$ | $-1.0135^{***}$ | $-2.3020^{***}$ | $n_{\text{medium}}{:}n_{\text{large}}$ | $-0.0142^{***}$ | $-0.0080^{***}$ |
| | (0.0461) | (0.1625) | | (0.0000) | (0.0001) |
| $(n_{\text{small}})^2$ | $-0.0036^{***}$ | $-0.0025^{***}$ | $\alpha{:}\beta$ | | $20.8872^{***}$ |
| | (0.0000) | (0.0000) | | | (0.3919) |
| $(n_{\text{medium}})^2$ | $-0.0066^{***}$ | $-0.0042^{***}$ | | | |
| | (0.0000) | (0.0000) | | | |
| $(n_{\text{large}})^2$ | $-0.0078^{***}$ | $-0.0038^{***}$ | $R^2$ | 0.9994 | 0.9985 |
| | (0.0000) | (0.0000) | Adj. $R^2$ | 0.9994 | 0.9985 |

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

# References


[1] A. Hoff, H. Andersson, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37(12):2041–2061, 2010. ISSN 0305-0548. doi:10.1016/j.cor.2010.03.015.

[2] Kaggle. Instacart market basket analysis, 2017. URL `https://www.kaggle.com/c/instacart-market-basket-analysis/data`. Online, accessed: 2020-10-12.

[3] D. Nicola, R. Vetschera, and A. G. Dragomir. Total distance approximations for routing solutions. *Computers & Operations Research*, 102:67–74, 2019. ISSN 0305-0548. doi:10.1016/j.cor.2018.10.008.

[4] Rutgers University. DIMACS Challenge - Capacitated VRP, 2021. URL `http://dimacs.rutgers.edu/programs/challenge/vrp/cvrp/`.

[5] M. Ulmer, A. Erera, and M. Savelsbergh. Dynamic service area sizing in urban delivery. *OR Spectrum*, 44:763–793, 2022. doi:10.1007/s00291-022-00666-z.

# Deep Reinforcement Learning for Master Stowage Planning

**Jaike van Twiller, Djordje Grbic, and Rune Møller Jensen**
IT University of Copenhagen, Copenhagen, Denmark
Email: {jaiv,djgr,rmj}@itu.dk

## 1 Introduction

Maritime transport has become the foundation of global trade and modern consumerism in the last century. Container vessels of liner shipping companies handle around 45% of annual transported goods, valued at $8.1 trillion in global trade [6]. Despite emitting far less CO2 per cargo ton kilometre than other modes of transportation [2], this industry has a significant impact on worldwide emissions. Liner companies create stowage plans to assign containers to vessel slots, aiming to optimize revenue and operational costs as profit margins are low [2]. However, publications in stowage planning are scarce, especially compared to vehicle routing [2, 7], despite the large economic and environmental impact.

Our study proposes a novel application of deep reinforcement learning to a non-trivial master planning problem for realistic-size vessels. The objective is optimizing vessel utilization and hatch-overstowage for equal-sized cargo in weight classes while satisfying deterministic voyage demand, location capacity, and longitudinal and vertical stability.

## 2 Background and Related Work

In order to understand the stowage planning domain, we will provide background information. Liner companies employ a fleet of container vessels to serve ports with fixed schedules on a closed-loop route, also called voyages. Container vessels have a cellular structure that is divided into *bays* with *rows* and *tiers*, which contain *cells* able to hold one 40-foot or two 20-foot container(s) in *slots*. *Stacks* are vertical arrangements of cells, whereas bays are horizontally separated by *hatch covers* into general *locations* that are either *on-deck* or *below-deck*. Note that hatch covers and on-deck containers need to be removed to access the below-deck cells. Cargo is loaded onto and discharged from the vessel by *quay cranes* from the top of the vessel. Each container has a *port of load (POL)* and a *port of discharge (POD)*, which defines the length of the transport. Containers come in various heights, weights, and types which affect the capacity of the ship in *Twenty-foot Equivalent Units (TEU)*. Furthermore, cargo must be evenly distributed over the ship to maintain safe levels of stability and stress forces.

To ensure on-time cargo delivery and resource-efficient operations, liner shipping uses stowage plans to allocate containers to vessel slots at each port of the voyage. The objective is to optimize revenue and operational costs during voyages, achieved by robust plans that often maximize cargo uptake and minimize time spent at ports. A fundamental combinatorial aspect is minimizing (hatch-)overstowage, which is an NP-hard task to reduce unnecessary crane moves [2]. A myriad of other aspects interact in this combinatorial optimization (CO) problem, such as cargo dimensions, seaworthiness requirements, stowage rules, demand uncertainty and planning best practices [2]. Moreover, modern vessels have over 20,000 TEU capacity and visit around 10 ports in voyages. This results in a complex multi-port problem, for which (near-)optimal solutions are yet to be found [7].

When dealing with complexity, hierarchical decomposition can be used to create two sequential subproblems: master planning (MPP) and slot planning (SPP). The former allocates groups of similar containers to on-deck and below-deck locations to meet global constraints, such as hatch-overstowage, location capacity, and stability. The latter, on the other hand, takes the MPP as input and assigns individual containers to slots, thereby satisfying local constraints, such as stack overstowage, maximum stack heights and weights. Although this decomposition has been successful in finding scalable algorithms for SPP, the search for scalable solutions to the MPP is still ongoing [7].

Most MPP contributions use exact methods or traditional heuristics, but these have not produced satisfactory results so far [7]. To broaden our horizon, we recognize that machine learning has in recent years proven to be an effective tool for solving CO problems, at times outperforming traditional methods [1]. Specifically, deep reinforcement learning (DRL) has emerged as a promising method to construct solutions to hard CO problems [3]. Stowage research has seen few ML initiatives, with most focus on SPPs or smalle-scale problems that relax local constraints [7]. However, a challenge arises when stowing 20,000 containers, each subject to many local constraints. We suggest DRL is better suited for developing approximate plans, such as the decomposed MPP, which abstracts away individual slots and considers global constraints during decision-making. A similar decomposition approach is used in chip design, where proximal policy optimization (PPO) finds a model that approximates chip floorplans, accelerating the process significantly [4].

# 3 Model and Algorithm

Our MPP optimizes vessel utilization and hatch-overstowage for equal-sized cargo in weight classes while satisfying deterministic voyage demand, location capacity, and longitudinal and vertical stability. To apply DRL, problems are generally defined as episodic Markov decision processes (MDPs). Our MDP represents episodes as voyages with $n$ ports. The state $s_t \in \mathcal{S}$ at time step $t$ is a pair $s_t = (u(t), q)$ and is fully observable, consisting of vessel utilization $u(t)$ and demand quantity $q$. For example, $u^{b,d}_{(i,j),k}(t) = 0.05$ means that

5% of the vessel capacity is cargo stowed in bay $b$ at deck $d$ with POL $i$, POD $j$ and class $k$ at step $t$. Similarly, $q_{(i,j),k} = 0.13$ represents a cargo demand of 13% of the capacity with POL $i$, POD $j$ and class $k$. The action is defined by pair $a_t = (l(t), \tau_t)$, where $l(t)$ is the fraction of vessel capacity to load in bay $b$ at deck $d$ of class $k$ on transport $\tau_t = (p, j')$ with current port $p$ and future port $j'$. For instance, $l_k^{b,d}(t) = 0.10$ indicates 10% of the capacity will be stowed in bay $b$ at deck $d$ with class $k$. Any state $s_t$ transitions to future state $s_{t+1}$ based on $a_t$. If cargo remains on board, then $u_{(i,j),k}^{b,d}(t)$ is unchanged. If cargo should be loaded, the utilization becomes $l_k^{b,d}(t)$. Otherwise, cargo that should not be on board is set to zero. The reward function $r(s_t, a_t)$ computes the degree to which vessel utilization satisfies voyage demand, avoids hatch overstowage and is not violating capacity and stability constraints. Figure 1 shows $n-p$ transitions at port $p$, where $r(s_t, a_t)$ evaluates demand at each step and other objectives between ports, e.g., $t+n-p+1$. Upon reaching port $p+1$, cargo destined for that port is removed from the observed state.

A proximal policy optimization (PPO) framework is proposed with three networks: an encoder to extract features, an actor decoder to find policy $\pi_\theta$, and a critic decoder to evaluate policies $V_\omega$. All networks are multilayer perceptrons with ReLU activation. Let $N$ parallel actors run policy $\pi_{\theta_{old}}$ with parameter $\theta_{old}$ for $H$ time steps, and compute the estimated advantage $\hat{A}_t$ of state-action pairs $(s_t, a_t)$ for each $t \in \{1, ..., H\}$. Using a minibatch of $M$ steps, the actor and critic loss are optimized with respect to parameters $\theta$ and $\omega$ with the Adam solver for $E$ epochs. With sufficient training, we can obtain a near-optimal policy $\pi_\theta \approx \pi^*$ to generate master plans. We refer to [5] for details on PPO.

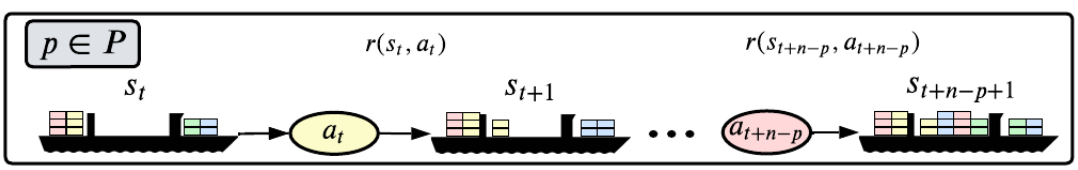



Figure 1: Visualization of MDP at port $p$ with colours representing PODs.

# 4 Results and Discussion

Our PPO framework is a function approximator of a heuristic that generates solutions to a non-trivial MPP. During training episodes, instances are generated by sampling $q$ from a Gaussian distribution $q \sim \mathcal{N}(\mu, \sigma)$ with expected value $\mu$ and standard deviation $\sigma$. Through Bayesian optimization, we determine the hyperparameter configuration that maximizes the episodic return on an in-distribution validation set, after which we obtain policy $\pi_\theta$ to generate solutions in test instances. Table 1 evaluates the performance of PPO against a benchmark MIP for various realistic vessel sizes. During in-distribution testing, PPO finds near-optimal solutions for most cases, which is not true for out-distribution tests. This highlights the importance of simulators that generate realistic instances. As policy $\pi_\theta$ does not guarantee feasibility, this will be a focal point in future work. Additionally, PPO demonstrates significantly shorter runtimes compared to MIP, which becomes advantageous for larger and more complex instances.

Table 1: Evaluation of PPO and MIP on 100 instances of two test sets, i.e., in-distribution Gaussian and out-of-distribution uniform instances, for voyages with $P$ ports and vessels with $B$ bays and $\sum c$ capacity in TEU. Average results over instances are given for the objective value (Obj.), the optimality gap (Gap) for PPO is relative to the MIP objective, and the MIP gap (Gap) refers to the duality gap. The average runtime in milliseconds (Time) and the number of feasible solutions (Feas.) are also given.

| | | | | **In-distribution test set** | | | | **Out-of-distribution test set** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | $\boldsymbol{P}$ | $\boldsymbol{B}$ | $\sum \boldsymbol{c}$ | **Obj.** | **Gap** | **Time** | **Feas.** | **Obj.** | **Gap** | **Time** | **Feas.** |
| PPO | 4 | 4 | 400 | 1.95 | 1.1% | 0.65 | 100 | 1.37 | 50.2% | 0.65 | 20 |
| PPO | 4 | 10 | 10,000 | 1.97 | 3.2% | 0.67 | 98 | 1.42 | 49.4% | 0.67 | 41 |
| PPO | 4 | 20 | 20,000 | 2.20 | 4.3% | 1.26 | 100 | 1.41 | 49.8% | 1.27 | 67 |
| MIP | 4 | 4 | 400 | 1.97 | 0.0% | 14.99 | 100 | 2.75 | 0.0% | 17.83 | 100 |
| MIP | 4 | 10 | 10,000 | 2.03 | 0.0% | 17.61 | 100 | 2.81 | 0.0% | 22.86 | 100 |
| MIP | 4 | 20 | 20,000 | 2.30 | 0.0% | 36.22 | 100 | 2.81 | 0.0% | 37.29 | 100 |

# 5 Conclusion

This novel application of PPO to a non-trivial MPP finds near-optimal solutions for realistic vessel sizes. The preliminary results are encouraging and show that DRL can be useful in our search for scalable algorithms. In the near future, we will enhance performance by extensive hyperparameter tuning and training, perform additional experiments with respect to voyage length $P$, and improve solution feasibility and problem complexity.

# References


[1] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 4 2021.

[2] Rune Møller Jensen, Dario Pacino, Mai Lise Ajspur, and Claus Vesterdal. *Container Vessel Stowage Planning*. Weilbach, 2018.

[3] Nina Mazyavkina, Sergey Sviridov, Sergei Ivanov, and Evgeny Burnaev. Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research*, 134:105400, 10 2021.

[4] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, Jiwoo Pak, Andy Tong, Kavya Srinivasa, William Hang, Emre Tuncer, Quoc V Le, James Laudon, Richard Ho, Roger Carpenter, and Jeff Dean. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.

[5] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *Computing Research Repository*, 7 2017.

[6] UNCTAD. Review of Maritime Transport 2021. Technical report, United nations conference on trade and development UNCTAD, 2021.

[7] Jaike Van Twiller, Agnieszka Sivertsen, Dario Pacino, and Rune Møller Jensen. Literature survey on the container stowage planning problem. *European Journal of Operational Research*, 12 2023.

# The bilevel vehicle routing problem with private fleet and external drivers

**Juan G. Villegas**
Department of Industrial Engineering,
Universidad de Antioquia, Medellín, Colombia
Email: juan.villegas@udea.edu.co

**José Fernando Camacho-Vallejo**
Tecnologico de Monterrey, Escuela de Ingenieria y Ciencias, Monterrey, México
Email: fernando.camacho@tec.mx

**Alejandro Montoya**
Department of Computer Science and Analytics Universidad EAFIT,
and Coordinadora Mercantil S.A. Medellín, Colombia
Email: jamontoya@coordinadora.com

## 1 Introduction

This work presents an extension of the vehicle routing problem arising in the operation of a Colombian courier company that uses its private heterogeneous fleet of vehicles to serve its customers and outsources part of its deliveries with a heterogeneous set of external drivers. The proposed model extends the vehicle routing problem with private fleet and common carriers (VRPPC [1]) by considering multiple common carriers in a bilevel fashion using a profitable tour problem (PTP) to model their explicit routing decisions. The proposed problem is also closely related to a problem arising in peer-to-peer transportation platforms studied in [2]. In this work, the authors also model carriers' decisions in a bilevel fashion using a PTP at the lower level. However, they model the problem from the perspective of a digital platform offering deliveries to carriers. Therefore, their model does not include routing decisions at the upper level. Moreover, we consider an additional feature arising in the company's operation. As in [3], the objective function of the carrier company includes a penalty term for missed deliveries. In our case, this term appears when the external drivers do not serve some deliveries because they are unprofitable for them.

## 2 Problem formulation

In the bilevel vehicle routing problem with private fleet and external drivers, a set of heterogeneous vehicles $\mathcal{V}$ and a set of external drives $\mathcal{E}$ serve a set of customers $\mathcal{C} = \{1, \ldots, n\}$ departing from a depot. The extended set of nodes $\mathcal{N} = \{0\} \cup \mathcal{C}$ includes the depot as node 0. Each customer $i \in \mathcal{C}$ has a demand $q_i$, a service time $s_i$, a cost for being assigned to the external drivers $r_i$, and a penalty for not being served $p_i$. Each vehicle $k \in \mathcal{V}$ in the fleet has capacity $Q_k$, a fixed cost $f_k$, a maximum route duration $D_k$, and a per distance variable cost $v_k$. Similarly, the external driver $e \in \mathcal{E}$ has a capacity $\psi_e$ and a per distance variable cost $\nu_e$. Finally, parameters $d_{ij}$ and $t_{ij}$ represent the distance and travel times between nodes, respectively.

In the upper level, the courier company solves a VRPPC with the following decision variables: $Y_{ik}$ is a binary variable indicating if customer $i \in \mathcal{C}$ is assigned to vehicle $k \in \mathcal{V}$, $(Y_{ik} = 1)$ or not, $(Y_{ik} = 0)$; $X_{ijk}$ is a binary variable indicating if vehicle $k \in \mathcal{V}$ travels from node $i, j \in \mathcal{N} : i \neq j$, $(X_{ijk} = 1)$ or not, $(X_{ijk} = 0)$; $W_{ie}$ is a binary variable indicating if customer $i \in \mathcal{C}$ is assigned to the external driver $e \in \mathcal{E}$, $(W_{ie} = 1)$ or not, $(W_{ie} = 0)$; and $T_{ik}$ is a non-negative variable representing the arrival time of vehicle $k \in \mathcal{V}$ to node $i \in \mathcal{C}$,

$$\begin{aligned} minimize \quad & \sum_{k\in\mathcal{V}}\sum_{i\in\mathcal{C}} f_k X_{0ik} + \sum_{k\in\mathcal{V}} v_k \left( \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}:j\neq i} d_{ij} X_{ijk} \right) \\ & + \sum_{e\in\mathcal{E}} \left( \sum_{i\in\mathcal{C}} \left( r_i \sigma_{ie} + p_i \mu_{ie} \right) \right) \end{aligned} \tag{1}$$

Subject to

$$\sum_{k\in\mathcal{V}} Y_{ik} + \sum_{e\in\mathcal{E}} W_{ie} = 1 \qquad \forall i \in \mathcal{C} \tag{2}$$

$$\sum_{j\in\mathcal{N}} X_{jik} = Y_{ik} \qquad \forall i \in \mathcal{C}, \forall k \in \mathcal{V} \tag{3}$$

$$\sum_{j\in\mathcal{N}} X_{ijk} = Y_{ik} \qquad \forall i \in \mathcal{C}, \forall k \in \mathcal{V} \tag{4}$$

$$\sum_{i\in\mathcal{C}} q_i Y_{ik} \leq Q_k \qquad \forall k \in \mathcal{V} \tag{5}$$

$$T_{ik} + s_i Y_{ik} + t_{ij} + D_k(1 - X_{ijk}) \geq T_{jk} \qquad \forall i \in \mathcal{N}, j \in \mathcal{C} : i \neq j, \forall k \in \mathcal{V} \tag{6}$$

$$T_{ik} + s_i Y_{ik} + t_{ij} - D_k(1 - X_{ijk}) \leq T_{jk} \qquad \forall i \in \mathcal{N}, j \in \mathcal{C} : i \neq j, \forall k \in \mathcal{V} \tag{7}$$

$$T_{ik} + s_i Y_{ik} + t_{ij} \leq D_k \qquad \forall i \in \mathcal{C}, \forall k \in \mathcal{V} \tag{8}$$

$$\sum_{i\in\mathcal{C}} q_i W_{ie} \leq \psi_e \qquad \forall e \in \mathcal{E} \tag{9}$$

The objective function of the company (upper-level objective) comprises four terms defined in equation (1): the fixed cost of the fleet, the variable cost of their routes, the outsourcing cost with the external drivers, and the penalty term for unserved customers. Upper-level constraints state that customers must be served by the private fleet or assigned to external drivers (2). If a customer is assigned to a given fleet vehicle, its route must enter (3) and leave that customer (4). The capacity of the vehicles in the fleet must be kept (5). Constraints (6)-(8) ensure the route duration of the private fleet routes and also act as subtour elimination constraints. Finally, the load assigned to a given external driver must not exceed his/her (known) vehicle capacity (9).

In the lower level, each external driver solves a profitable tour problem with conditional openness using the following variables: $\sigma_{ie}$, is a binary variable indicating if external driver $e \in \mathcal{E}$ serves customer $i \in \mathcal{C}$ ($\sigma_{ie} = 1$) or not ($\sigma_{ie} = 0$); $\chi_{ije}$ is a binary (routing) variable indicating if external driver $e \in \mathcal{E}$, goes from node $i, j \in \mathcal{N} : i \neq j$ ($\chi_{ije} = 1$) or not ($\chi_{ije} = 0$); $\tau_{ie}$ is a non-negative variable indicating the arrival time of the external driver $e \in \mathcal{E}$ to customer $i \in \mathcal{C}$; and, finally, $\mu_{ie}$ is an auxiliary binary variable indicating if customer $i \in \mathcal{C}$ is not served after being assigned to the external driver $e \in \mathcal{E}$, ($\mu_{ie} = 1$). This last variable connects the decisions of the external drivers with the penalty term in the upper-level objective function.

An important feature of the lower-level problem is that the external driver has to return to the depot if there are unserved customers to give back their packages. Therefore, two additional variables are needed at the lower level: a binary variable $\beta_e$ indicating if the external driver $e \in \mathcal{E}$ has to return to the depot after finishing the route ($\beta_e = 1$ in that case); and a binary variable ($\lambda_{ie}$) indicating the last customer visited by the external driver $e \in \mathcal{E}$ ($\lambda_{ie} = 1$ for this customer) that is needed to model open routes when all customers are served. Using this notation, the extended PTP (EPTP) solved for each external driver in the lower level is as follows:

$$maximize \sum_{e \in \mathcal{E}} \left( \sum_{i \in \mathcal{C}} r_i \cdot \sigma_{ie} - \nu_e \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}: i \neq j} d_{ij} \chi_{ije} \right) \tag{10}$$

subject to:

$$\sigma_{ie} + \mu_{ie} = W_{ie} \qquad \forall e \in \mathcal{E}, \forall i \in \mathcal{C} \tag{11}$$

$$\sum_{j \in \mathcal{N}: j \neq i} \chi_{jie} = \sigma_{ie} \qquad \forall e \in \mathcal{E}, \forall i \in \mathcal{C} \tag{12}$$

$$\sum_{j \in \mathcal{N}: j \neq i} \chi_{ije} = \sigma_{ie} - \lambda_{ie} \qquad \forall e \in \mathcal{E}, \forall i \in \mathcal{C} \tag{13}$$

$$\tau_{ie} + s_i \sigma_{ie} + t_{ij} \chi_{ije} \leq \tau_{je} + M(1 - \chi_{ije}) \qquad , \forall e \in \mathcal{E}, \forall i, j \in \mathcal{N} : i \neq j \tag{14}$$

$$\mu_{ie} \leq \beta_e \qquad \forall i \in \mathcal{C}, \forall e \in \mathcal{E} \tag{15}$$

$$\sum_{i \in \mathcal{C}} \chi_{0,i,e} \leq \sum_{i \in \mathcal{C}} \chi_{i,0,e} + \sum_{i \in \mathcal{C}} \lambda_{ie} \qquad \forall e \in \mathcal{E} \tag{16}$$

$$\sum_{i \in \mathcal{C}} \lambda_{ie} \leq 1 - \beta_e \qquad \forall e \in \mathcal{E}, \tag{17}$$

In the lower level, we adopt an EPTP that seeks, in the objective function (10), the maximization of the profit minus travel costs for each driver. The lower-level constraints include the selection of the customers assigned to external drivers to be served or not (11). The in- and out-degree of the routes in the customers selected by a given external driver, equations (12) and (13), respectively. The right-hand side of the out-degree constraints include variables $\lambda_{ie}$ to allow open routes when all customers are served. The arrival time of the external driver increases if two nodes are visited consecutively (14). Finally, constraints (15)-(17) ensure the conditions return of the external drivers to the depot.

## 3 Partial results and conclusions

We implemented the upper and lower levels hierarchically. That is, the company solves the upper-level problem (1)-(9) ignoring the optimal reaction of the external drivers. With this solution, we solve the EPTP for each external driver (10)-(17) based on the values of variables $W^*_{ie}$ of the upper level. To assess the impact of the interaction between the firm and external drivers, in an alternative scenario, we assume that all customers assigned to each driver must be served. In this case, the profit term of their objective function (10) is fixed, and the lower level reduces to a Hamiltonian path problem.

A preliminary experiment with public VRPPC instances [1] confirms the importance of considering the bilevel structure of the problem. If the external driver cost for each customer is very low, the company myopically ignores the possible rejection of the customer deliveries by the external drivers, resulting in a high percentage of unserved customers, ranging from 53 to 97% of the outsourced customers. Middle-value outsourcing costs can decrease external drivers' profits by up to 40% if they serve all assigned customers, and may even result in total losses. Finally, high external driver cost per customer leads to having no unserved customers but with high outsourcing costs for the company.

## References


[1] M. C. Bolduc, J. Renaud, F. Boctor and G. Laporte, G. A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of the Operational Research Society* 59, 776-787,2008.

[2] M. Cerulli, C. Archetti, E. Fernandez and I. Ljubic. A bilevel approach for compensation and routing decisions in last-mile delivery. arXiv preprint arXiv:2304.09170. 2023.

[3] L.D.P. Pugliese, D. Ferone, G. Macrina, P. Festa, P and F. Guerriero. The crowd-shipping with penalty cost function and uncertain travel times. *Omega* 115, 102776,2023.

# Last mile delivery routing problem with some-day option

**Stefan Voigt, Markus Frank, Heinrich Kuhn**
Catholic University of Eichstätt-Ingolstadt
Ingolstadt School of Management, Germany
Email: stefan.voigt@ku.de

## 1 Motivation and problem setting

Parcel deliveries are at an all-time high, amounting for 161 billion parcels worldwide in 2022. This is accompanied by a growing demand for even faster deliveries in the business-to-consumer (B2C) sector. However, short delivery times put tremendous pressure on transportation networks and often lead to less efficient distribution processes. We explore a delivery concept that deliberately slows down the logistics processes involved in parcel delivery, thereby allowing for the consolidation of more shipments over an extended time period. For instance, Amazon already provides a "free no-rush" delivery option in many U.S. regions. Customers who select this option accept a longer delivery time frame and, in return, e.g., receive a discount. Alternatively, customers may be nudged by information highlighting the reduced emissions associated with slower delivery, which could encourage them to accept the potential delay [2]. We assume the point of view of an e-commerce retailer who operates its own delivery fleet and offers a range of delivery options, including a notably slower *some-day* alternative. Each customer's order is subject to specific delivery date constraints, comprising both an earliest and a latest possible delivery date. The retailer must determine the most cost-effective delivery day for each customer, as well as the clustering and routing. We term this problem setting *some-day delivery problem* (SDDP). Our modeling assumptions align with the broader category of multiple period VRPs (MPVRP) [1]. In practice, customer orders arrive continuously throughout the planning period, creating a dynamic setting. While a similar dynamic MPVRP, has been proposed by [8], our situation differs in that not all relevant information is known in advance but may be available in a stochastic manner.

## 2 Static and deterministic model formulation

In the following, we formulate a static and deterministic definition of the SDDP. The model is inspired by an MPVRP proposed by [1]. The problem is defined on a directed graph $G(N, A)$ with node set $N$ and arc set $A$. Node set $N$ consists of the depot 0 and customers $j \in C$. The arc set is defined as $A = \{(i, j) : i \neq j, i, j \in N\}$. The parameter $c_{ij}^{\text{trans}}$ denotes the associated costs with each arc, $c_i^{\text{inv}}$ denotes the inventory/waiting costs per period and customer, and $c_i^{\text{back}}$ denotes the costs when the customer is served by a backup option. A fleet $K$ of homogeneous vehicles with capacity $Q$ is available to perform routes starting and ending at the depot in each period. The vehicle has to arrive at the depot within the deadline $D$. A customer's demand $q_i$ can be delivered on any day within its allowed delivery interval $[e_i, l_i]$, where $e_i \leq l_i$ and $l_i \leq T$. On the selected delivery day the service which takes service time $S_i$ must begin and end within the customer's designated time window $[etw_i, ltw_i]$. If multiple orders from the same customer are considered, this can be modeled by co-locating customers. The decision variables are as follows.

- $x_{ijkt}$: Indicates whether arc $(i, j)$ is traversed by vehicle $k$ in period $t$; $i, j \in N, k \in K, t \in T$
- $z_{ikt}$: Indicates whether customer $i$ is served by vehicle $k$ in period $t$; $i \in C, k \in K, t \in T$
- $y_{it}$: Indicates whether customer $i$ is served by a backup option in period $t$; $i \in C, t \in T$
- $s_{ikt}$: Start time of service of vehicle $k$ at node $i$ in period $t$; $i \in N, k \in K, t \in T$

$$Minimize \quad \sum_{k \in K} \sum_{i,j \in N} \sum_{t \in [e_i, l_i]} c_{ij}^{\text{trans}} \cdot x_{ijkt} + \sum_{i \in C} c_i^{\text{inv}} \cdot \sum_{t \in [e_i+1, l_i]} (t - e_i) \cdot \left( y_{it} + \sum_{k \in K} z_{ikt} \right) + \sum_{i \in C} c_i^{\text{back}} \cdot \sum_{t \in [e_i, l_i]} y_{it} \tag{1}$$

s.t.

$$\sum_{k \in K} \sum_{j \in N} \sum_{t \in [e_i, l_i]} x_{ijkt} + \sum_{t \in [e_i, l_i]} y_{it} = 1 \qquad \forall i \in C \tag{2}$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijkt} \leq Q \qquad \forall k \in K, t \in T \tag{3}$$

$$\sum_{j \in N} x_{ijkt} = \sum_{j \in N} x_{jikt} \qquad \forall i \in N, k \in K, t \in T \tag{4}$$

$$z_{ikt} = \sum_{j \in N} x_{ijkt} \qquad \forall i \in C, k \in K, t \in T \tag{5}$$

$$\sum_{j \in N} x_{0jkt} \leq 1 \qquad \forall k \in K, t \in T \tag{6}$$

$$\sum_{j \in N} x_{j0kt} \leq 1 \qquad \forall k \in K, t \in T \tag{7}$$

$$s_{jkt} + S_j + c_{j0} \leq D \qquad \forall j \in C, k \in K, t \in T \tag{8}$$

$$s_{jkt} - s_{ikt} \geq (c_{ij} + S_i) x_{ijkt} - D(1 - x_{ijkt}) \qquad \forall i, j \in C,\ i \neq j, k \in K, t \in T \tag{9}$$

$$etw_j \leq s_{jkt} \leq ltw_j \qquad \forall j \in C, k \in K, t \in T \tag{10}$$

$$x_{ijkt} \in \{0, 1\} \qquad \forall i, j \in N, k \in K, t \in T \tag{11}$$

$$z_{ikt} \in \{0, 1\} \qquad \forall i \in C, k \in K, t \in T \tag{12}$$

$$y_{it} \in \{0, 1\} \qquad \forall i \in C, t \in T \tag{13}$$

$$s_{ikt} \in \mathbb{R}_0^+ \qquad \forall i \in N, k \in K, t \in T \tag{14}$$

The objective (1) minimizes transportation costs, inventory/waiting costs of customers served, and costs for customers served via the backup option. Constraints (2) ensure that every customer is delivered within the requested delivery interval. Constraints (3) prohibit the vehicle capacities to be exceeded. Constraints (4) are flow-conserving constraints. Constraints (5) link the assignment variables $z_{ikt}$ with the flow variables $x_{ijkt}$. Constraints (6) and (7) ensure that only one tour is performed per vehicle and day. Constraints (8–10) guarantee that the vehicle arrives back at the depot in time, subtours are eliminated, and customer time windows are respected.

## 3 Stochastic and dynamic solution approach

We use the deterministic model as the basis for our stochastic-dynamic modeling approach. In this dynamic setting, only a subset of customer orders is known at planning instant, i.e., only customers revealing their demand in the previous period. The customer demands within the following periods are uncertain and stochastic. New orders arrive at the end of each period. A periodic re-planning is carried out each period to account for the newly arrived orders and updated demand forecasts. We define a benefit measure for each known customer indicating the value of serving the customer in the current period rather than postponing. We henceforth call this benefit measure "*prize*." The *prize* combines positive and negative effects on the advantageousness of servicing an order in the current period, including the following aspects: urgency of the order, future capacity utilization, the probabilities of emerging nearby customers in future periods, as well as inventory costs. As intuition, the prize should be high if the order is urgent or we expect a high demand in future periods. Contrary, the prize should be low if we expect nearby customers emerging in future periods. This prize is then used to solve an auxiliary prize-collecting VRP with time windows (PCVRPTW), that decides which customers to deliver in that period and the corresponding routing. The PCVRPTW is solved heuristically with a hybrid adaptive large neighborhood search with granular insertion operators (HALNS-G). The HALNS-G extends the original version [6, 7] with problem-specific operators, in particular granular insertion operators. The concept of granular insertion operators is inspired by the granular tabu search [4]. In the granular insertion operators, the insertion positions are confined to customers located in close proximity to the customer being inserted.

## 4 Sample results and contribution

**Performance evaluation of HALNS-G** We evaluate the performance of the original HALNS and HALNS-G on PCVRPTW instances with up to 1,000 customers. We compare the solutions obtained against PyVRP [3] by running all algorithms on the same machine and time limits. PyVRP is a powerful version of the open-source hybrid genetic search by [5]. Figure 1 shows that the HALNS-G tends to outperform PyVRP for smaller instance sizes. HALNS-G clearly outperforms HALNS, and the effect becomes more pronounced as the number of customers increases.

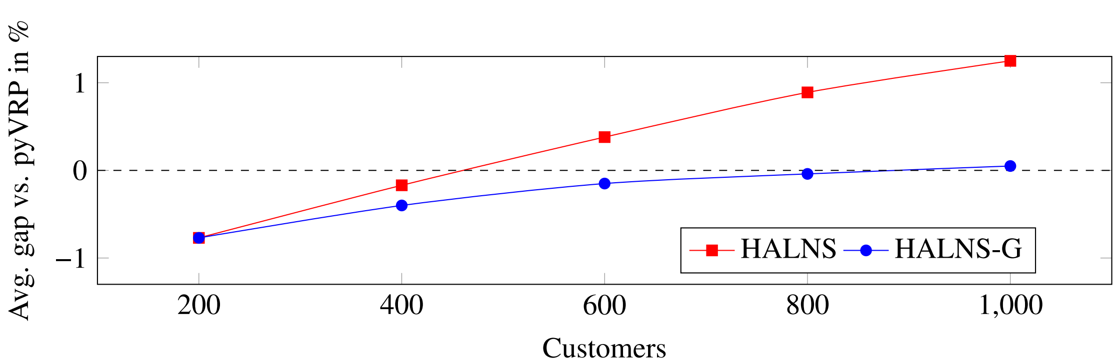



Figure 1: Performance of HALNS and HALNS-G against PyVRP

**Experiments** In our experiments we introduce a simulator based on VRPTW instances with 1,000 customers. In each period, the simulator randomly samples a set of 100 customers from the respective instance. This set is then revealed as the pending customer set for the current planning period. For each instance, we generate 30 periods and calculate several performance measures (e.g., the average costs per customer, the average and maximum number of vehicles used per period). We conduct several experiments to generate managerial insights such as the cost reduction against several benchmark policies, the impact of the length of the delivery interval, and the impact of the share of customers selecting the some-day option. As an exemplary experiment, Table 1 shows the results on varying the length of the delivery interval of the some-day option $(1, 2, \ldots,$ or 5 days). We compare the average costs against the costs achieved with an earliest policy. In this policy, we serve each customer as quickly as possible, i.e., within the period following the order. Compared to the earliest policy, we can reduce costs to 78.9 % with a delivery interval of just 1 day. The costs savings increase with longer delivery intervals but become increasingly marginal, indicating that a moderate interval length of 3 days seems sufficient.

Table 1: Average costs for various delivery interval lengths of the some-day option

| **Delivery interval [days]** | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **Average costs vs. earliest policy [%]** | 78.9 | 68.1 | 61.3 | 56.8 | 54.0 |

**Contribution** The paper contributes by (1) describing a novel slow logistics concept for B2C parcel delivery, (2) reviewing and categorizing MPVRPs with delivery dates, (3) introducing a solution approach for a dynamic MPVRP with stochastic information, (4) implementing a hybrid adaptive large neighborhood search with granular insertion operators for solving prize-collecting VRPTWs, and (5) showing by simulation that a slow delivery option significantly improves costs.

## References


[1] C. Archetti, O. Jabali, and M. G. Speranza. Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 61:122–134, 2015.

[2] M. Dietl, S. Voigt, and H. Kuhn. From rush to responsibility - evaluating economic and ecological incentives on customers' willingness to wait for online orders. *SSRN Electronic Journal*, 2023.

[3] PyVRP. PyVRP - a python library for vehicle routing problems. `https://github.com/PyVRP/PyVRP`, 2023.

[4] P. Toth and D. Vigo. The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, 15(4):333–346, 2003.

[5] T. Vidal. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Computers & Operations Research*, 140:105643, 2022.

[6] S. Voigt, M. Frank, P. Fontaine, and H. Kuhn. Hybrid adaptive large neighborhood search for vehicle routing problems with depot location decisions. *Computers & Operations Research*, 146:105856, 2022.

[7] S. Voigt, M. Frank, P. Fontaine, and H. Kuhn. The vehicle routing problem with availability profiles. *Transportation Science*, 57(2):531–551, 2023.

[8] M. Wen, J.-F. Cordeau, G. Laporte, and J. Larsen. The dynamic multi-period vehicle routing problem. *Computers & Operations Research*, 37(9):1615–1623, 2010.

# Practical Vehicle Routing in an Urban Road Network: Is Stochastic or Time-dependent Speed Important?

**Stein W. Wallace**
Department of Business and Management Science,
NHH Norwegian School of Economics, Bergen, Norway
Email: stein.wallace@nhh.no

**Feng Guo**
Business School of Sichuan University, Chengdu, China

**Zhaoxia Guo**
Business School of Sichuan University, Chengdu, China

**Michal Kaut**
SINTEF Technology and Science, Norway

Vehicle routing, in all its variants, is one of the most studied problems in logistics. But for historical as well as numerical reasons, the vast majority of papers are on deterministic problems. Stochastic versions have started to occur. The most studied stochastic phenomenon is demand, followed by travel time (speed), service time and finally, random occurrence of customers.

We analyze a variety of vehicle routing problems where travel speed is stochastic across time and space. Using real maps and real speed data for the mega-city Chengdu, China, we ask two questions. 1) Is time dependence of speed important and 2) Is stochastics important? We study the questions with respect to both decisions and out-of-sample objective values in order to understand the actual behavior of solutions. While time dependence is important most of the time, stochastics is not. Hence, we analyze and characterize which VRP formulations are sensitive to stochastic speeds and which are not. We solve with an accuracy of about 1% VRPs with 6000 links, 240 time periods and 100 customers, resulting in over 1.4 million dependent random variables.

We show that very large problems, in terms of stochastics, can be solved, and we want to understand when stochastics is important, and when not. We proceed as follows:

- We define a number of different VRPs based on models that appear in the literature.

- We solve all of them using deterministic speeds, so that a certain link has the same speed all day, set as the average speed for that link across all our data points. This is very close to the standard of using distance rather than speed.

- We then solve all of them using deterministic speeds, but where averages are taken time period by time period. This refers to what is called time-depenent travel times in the literature.

- We compare the solutions to these two problem settings using the full data set to calculate an out-of-sample value for all solutions. So we do not compare the solutions and objective functions as such, but we compare how the solutions function in a stochastic environment.

- If for some problem settings, time-dependence is found not to be important, we do not use them in the further tests.

- Being careful about how scenarios are generated and solution quality tested, we now solve two-stage stochastic versions of the problems. Again testing using the full data set, we check if stochastics adds anything substantial to the quality of the solutions. We end up with some problems where stochastics is very important, and some where it barely matters.

- We characterize what makes a problem sensitive to stochastics, and we discuss in more detail how to generate scenarios for the important cases.

Our tests are based on maps and speed data ([1]) for Chengdu, a city in western China with about 21 million people. In our previous paper ([2]) we used a real map of Beijing, but the speed data were fake. But in this paper, all data are real.

All the models are in a two-stage setting. In the first stage each vehicle is assigned a sequence (not just a set) of customers. This could be because it is a delivery problem, and the vehicle must be packed based on first-in, last-out, so that we are facing an operational problem where deterministic demand is natural. This could also be a tactical problem where the same routes will be used for a while, and all customers must always be visited, and have a deterministic demand on the day of operations, be that pickup or delivery. We shall not consider stochastic speed and demand together, and to the best of our knowledge there is no literature doing so. Purely operational multi-stage problems (when related to stochastic speeds), such as “where to go next” seem better fit to methods such as Approximate Dynamic Programming, and are not covered here.

The second-stage problem is to choose routes between pairs of customers, based on the present speeds at the time the vehicle leaves the first customer in the pair. So our stochastic speeds mimic what real-time speeds may be faced, depending on when the vehicle leaves a given customer, and the optimization in the second stage mimics what a vehicle navigation system would do. We assume that once a path is chosen between a pair of customers, it is followed. But we could easily handle a case where the vehicle navigation system "changed its mind" as the driving progressed, node by node. However, we are not able to anticipate these changes. That would make this a multistage problem.

We showed in [2] that for this problem class it is necessary to operate on the map, not on the customer-node network. On the customer-node network stochastic dependencies become impossible to handle. Also, data are most likely on the map (as is the case for us), not on our customer node pairs.

**How to measure quality of scenarios and importance of stochastics.**

When looking at the literature, it is normal to measure the importance of stochastics using VSS (the Value of the Stochastic Solution), an absolute measure of the expected gain of using a stochastic model rather than deterministic. On the other hand, most methods looking at the quality of a scenario tree, use relative measures, such as relative error. However, importance can be measured using relative numbers and scenario tree quality can be measured in absolute terms. We shall see that care must be taken when choosing between relative and absolute measures when asking if stochastics is important.

The following extreme case offers a warning when relative measures are used. Assume we have a VRP where the number of vehicles is given as input. If we, for some reason, add fixed costs for the vehicles to the objective function, absolute measures for the importance of stochastics and the necessary number of scenarios will not change, as the optimal solution will not change. But relative measures will change, apparently reducing the necessary number of scenarios and the importance of stochastics. There is nothing right or wrong here, but care must be taken.

**Cases for this presentation**

In the presentation we will focus on two cases.

1. There is a penalty (overtime pay) if a route duration exceeds a normal working day, but no gain if we finish early. The number of vehicles is given.

2. There are soft time windows at each customer, with a penalty for early or late arrival. The penalty for late arrival is largest. The number of vehicles is given and we minimize travel costs plus penalty costs.

For both these cases, time dependence is important, and stochastics matters. The talk will present numbers and focus on why this is the case for our Chengdu data set, and try to understand, more generally, what characterizes problems where stochastics matter (or not).

**Generating scenarios**

Most scenario generation methods do not work in these dimension. We see only two options, namely sampling and a copula-based method ([3]) which we also used in [2]. We find that the accuracy from the copula-based method is about twice as high as what we get for sampling. Details will be given in the presentation.

## References


[1] Feng Guo, Xin Gu, Zhaoxia Guo, Yucheng Dong, and Stein W. Wallace. Understanding the marginal distributions and correlations of link travel speeds in road networks. *Scientific Reports*, 10:1–8, 2020.

[2] Zhaoxia Guo, Stein W. Wallace, and Michal Kaut. Vehicle routing with space- and time-correlated stochastic travel times: Evaluating the objective function. *INFORMS Journal on Computing*, 31(4):654–670, 2019.

[3] Michal Kaut. A copula-based heuristic for scenario generation. *Computational Management Science*, 11(4):503–516, 2014.

# Addressing water level uncertainty for inland waterway transportation: a partially joint chance-constrained programming approach

**Chenghao WANG, Ioana C. BILEGAN**
Univ. Polytechnique Hauts-de-France, LAMIH, CNRS, UMR 8201
F-59313 Valenciennes, France
{chenghao.wang, ioana.bilegan}@uphf.fr

**Walter REI**
CIRRELT and School of Management Sciences
Université du Québec à Montréal, Montréal, Canada

**David DUVIVIER**
Univ. Polytechnique Hauts-de-France, LAMIH, CNRS, UMR 8201
F-59313 Valenciennes, France

## 1 Introduction

Intermodal freight transportation is regarded as a transportation strategy to enhance environmental sustainability and operational efficiency, seamlessly integrating various modes such as trucks, rail, ships, airplanes, and cargo bikes. Furthermore, authors in [1] have highlighted that intermodal transport, supported by the European Commission as well as by national and regional governments, is actively promoting a modal shift from road transport to inland waterways and railways. Consolidation-based carriers, including rail and navigation companies, play a crucial role in intermodal freight transportation by efficiently achieving economies of scale, optimizing service networks, and meeting diverse shipper requirements. Their primary objective is to maximize net profits while satisfying the needs and specifications of shippers, involving the establishment of a resource- and cost-efficient service network and schedule informed by forecasted demand.

Concentrating on specific aspects of the consolidation-based intermodal freight transportation problem, incorporating Inland Waterway Transportation (IWT), we first turn our attention to [2], where Tawfik and Limbourg examined this problem using a service network design model and a scenario-based analysis approach. For an in-depth exploration of tactical planning in intermodal freight transportation for IWT carriers, we refer to [3], where Bilegan *et al.* addressed this aspect and proposed a new service network design model integrating both resource and revenue management considerations.

The significance of IWT, often underutilized in intermodal freight logistics, has been underscored in recent decades thanks to its cost-effectiveness, energy efficiency, and overall environmental sustainability. Nevertheless, the integration of inland waterways into the intermodal transport network presents both opportunities and challenges. This is primarily attributed to the rather difficult predictability of water levels, which can impact transit times and vessels capacities, potentially affecting the efficiency and reliability of IWT. In this context, our focus is on a critical aspect of IWT, namely the tactical planning problem for consolidation-based intermodal freight transportation carriers. We introduce novel considerations on this decision-making type of problems and, more broadly, on addressing uncertainty arising from water level variations within mathematical programming models. This work is advancing the studies initiated in [3], by explicitly introducing an important source of uncertainty that affects the tactical planning performed in IWT and impacts the design of the decision-making framework, models and solution techniques.

## 2 Uncertainties of water level and induced vessel capacity

Water level, a critical factor in IWT, undergoes significant fluctuations influenced by seasonal weather conditions. It plays a crucial role in determining the navigability of rivers and canals, directly impacting the cargo capacity of vessels operating in these waterways. However, as highlighted in [4], most of the existing research work focuses on uncertainties related to demand, transit time, and cost, with limited attention given to capacity, particularly in the context of capacity influenced by water levels. Insufficient water levels may compel vessels to reduce their carrying capacity to prevent grounding or hull damage, resulting in prolonged transit times and increased transportation costs. This situation might require the use of container trucks, incurring additional expenses to transport cargo beyond the vessel's capacity in specific sections of the river or canal affected by low water levels. Conversely, excessive water levels may hinder vessels from passing under bridges or through locks, leading to delays and disruptions in supply chains. Consequently, variations in water levels directly impact the loading capacity of vessels within inland waterways.

## 3 Handling the uncertainty of water levels

We assume known a piece-wise linear function [5], denoted $g(l,i,j)$, establishing the relationship between the water level on the physical link $(i,j)$ and the corresponding carrying capacity $cap(l,i,j)$ when sailing between terminals $i$ and $j$ for each vessel type $l$. Let us consider the general upper bound constraint on the capacity of a ship navigating in a specific river section, expressed as: $\sum_{d\in D} x(d,s,k) \leq \text{cap}(l(s), i_k, i_{k+1}) y(s), \forall k \in K(s), s \in S$.

Here, the continuous decision variable $x(d, s, k)$ represents the volume of demand $d \in D$ transported by the leg $k \in K(s)$ of a service $s \in S$, the binary decision variable $y(s)$ takes the value of 1 if service $s$ is activated, $l(s)$ specifies the vessel type used by a service $s$, $i_k$ and $i_{k+1}$ denote respectively the origin and destination terminals of leg $k$, and $\text{cap}(l(s), i_k, i_{k+1})$ indicates the transportation capacity of a service $s$ on leg $k$.

When considering the practical settings of inland waterway transportation systems, one often finds that the fluctuations in water levels exhibit a non-uniform distribution along the river or canal. Rather than being evenly spread, these changes persist over extended distances and time, affecting specific sections and highlighting the uncertainty in water levels. Therefore, the modeling approaches relying on classical individual or joint chance constraints may prove either too lax or excessively restrictive when performing the tactical planning. These approaches might fail to properly formulate the desired feasibility guarantees for the tactical transportation plan, i.e., the planned freight itineraries being feasible on certain zones of the network that are viewed as strategic.

To address this limitation, we propose a partially joint chance-constrained variant that incorporates a designated partially joint partition set, denoted as $\Omega$. This set comprises subsets of indexes corresponding to the capacity constraints that are used to define the probabilistic constraints. These subsets, denoted as $\Omega_\xi$, enable the grouping of interconnected sections of the river or canal. Each such subset, individually constrained, maintains the associated probabilistic constraints jointly constrained within this subset. By employing this approach, one may appropriately consider and group together diverse geographical or infrastructure characteristics within different sections of the physical waterway network, thereby enhancing the overall modeling consistency. Moreover, from a managerial perspective, we suggest introducing a so-called tolerated volume $\text{Tol}_{\Omega_\xi}$ (i.e., an upper bound on available capacities of other non-IWT modes needed to handle the exceeding volumes of cargo not able to be transported by waterway) as a proactive measure to mitigate the adverse effects of fluctuating water levels within each subset $\Omega_\xi$. This approach is designed to offer a flexible and adaptable solution for effective system management under varying conditions.

To numerically implement this approach, we introduce the continuous variable $\tilde{w}(s, k)$, which represents the required capacity adjustment that would be necessary to satisfy the probabilistic capacity constraint, when the activated service lacks sufficient capacity $\tilde{\text{cap}}(l(s), i_k, i_{k+1})$ to accommodate the demands on its $k^{th}$ service leg due to the uncertain water level. By configuring the specific tolerated volume $\text{Tol}_{\Omega_\xi}$ for each subset $\Omega_\xi$, we can effectively address the impact of fluctuating water levels and ensure the resilience of the system within a threshold of infeasibility, denoted by $\epsilon_{\Omega_\xi}$. A tolerance-based partially joint chance-constrained paradigm is then introduced to address the uncertainty of water levels along the river or canal:

$$\sum_{d \in D} x(d,s,k) \leq \tilde{\text{cap}}(l(s), i_k, i_{k+1}) y(s) + \tilde{w}(s,k), \quad \forall s \in S, k \in K' \tag{1}$$

$$\mathcal{P}\left( \sum_{(s,k) \in \Omega_\xi} \tilde{w}(s,k) \leq \text{Tol}_{\Omega_\xi} \right) \geq 1 - \epsilon_{\Omega_\xi}, \quad \forall \Omega_\xi \in \Omega \tag{2}$$

$$\tilde{w}(s,k) \geq 0, \quad \forall s \in S, k \in K' \tag{3}$$

The aforementioned probabilistic constraints (1)-(3) present challenges in resolution, even when the probability distributions for water levels are formulated using classical random distributions. The present study tackles this issue by developing a MILP reformulation aligned with constraints (1)-(3). It incorporates a finite set of water level scenarios across a relevant inland waterway physical network and time horizon, along with an efficient approach to solve the large-scale instances.

In this study, we aim to tackle water level uncertainties within consolidation-based intermodal freight transportation, focusing on inland waterway networks. The MILP reformulation, incorporating constraints (1)-(3) in accordance with the partially joint chance-constrained paradigm and the concise representation of the investigated problem will be presented during the Odysseus conference, as well as elements of the experimental setting used, numerical results and analyses performed.

## References


[1] T. Ambra, K. M. Mommens, and C. Macharis, "Intermodal and synchromodal freight transport", In *International Encyclopedia of Transportation*, pages 456–462, 2021.

[2] C. Tawfik and S. Limbourg, "Scenario-based analysis for intermodal transport in the context of service network design models", *Transportation research interdisciplinary perspectives*, 2,100036, 2019.

[3] I. C. Bilegan, T. G. Crainic, and Y. Wang, "Scheduled service network design with revenue management considerations and an intermodal barge transportation illustration", *European Journal of Operational Research*, 300(1),164–177, 2022.

[4] T. Delbart, Y. Molenbruch, K. Braekers, and A. Caris, "Uncertainty in intermodal and synchromodal transport: Review and future research directions", *Sustainability*, 13(7),3980, 2021.

[5] C. van Dorsser, F. Vinke, R. Hekkenberg, and M. van Koningsveld, "The effect of low water on loading capacity of inland ships", *European Journal of Transport and Infrastructure Research*, 20(3),47–70, 2020.

# Planning of Truck Platoons: An Exact Solution for The Multi-trip Pickup and Delivery Problem


**WANG YUAN JESSICA**

School of Business,

Singapore University of Social Science, Singapore

Email: wangyuan@suss.edu.sg

**JINTAO YOU**

Department of Industrial Systems Engineering and Management,

National University of Singapore, Singapore

Email: youjintao@163.com


## 1 Background

Recently, driven by the breakthrough of autonomous driving technologies, truck platooning has emerged as an innovative concept that crafts a new paradigm for container truck management. It is composed of a lead truck with human driver and a convoy of following trucks with automated longitudinal control. The new mode of human-machine cooperation with collective intelligence produces significant savings in fuel consumption and driver workload, and enables better traffic flow efficiency and safety. These advantages have attracted increasing attention from governments and industries to deploy truck platooning in practice. For instances, Singapore government has cooperated with Scania and Toyota to host the autonomous truck platooning trials between container ports, with the goals of alleviating the shortage of manpower and easing traffic congestion (MOT-PSA 2017).

In this paper, we aim to provide an exact solution for truck platnoons planning. We demonstrate the concept of Multi-trip Pickup and Delivery Problem (MTPDP) with load dependent cost to model the truck platooning problem. The MTPDP is a combination of the MTVRP and PDP, factoring in both multi-trip and delivery and pickup requirements. In this paper, we study MTPDP with load dependent cost, which considers the special situation where the travel costs of vehicles between task nodes are dependent on the load onboard. Load dependent cost requires special attention in VRPs due to several emerging applications. For example, ride sharing allows customers to share taxis from certain origins to destinations, and load dependent cost is a critical decision factor for optimal scheduling. The total cost of a single trip of one taxi is dependent on the number or sequence of customers onboard, and the taxis have to perform multiple trips in one day [2]. Besides, some VRP variants require the consideration of fuel consumption/ gas emissions [1], or battery consumption of electric vehicles [5], which make energy consumption dependent on the load on vehicles.

Truck platooning problem can be presented as MTPDP with load dependent cost. Under the truck platooning mode, truck drivers are allowed to leave the depot to deliver and pickup container trucks several times within their planning horizons, and each driver is able to operate several trucks at a time using automation technologies. Besides, the travel costs between task nodes are dependent on the number of trucks in the platoons due to fuel cons umption and the air-drag reduction.

## 2 Problem Description

Figure 1 is a simple example to illustrate the advantages and features of the MTPDP with load dependent cost based on the assumptions of truck platooning studied by [3]. Under the truck platooning mode, a driver is able to operate several trucks at a time and only the leading truck is human driven while the ensuing trucks can follow it using automation technologies. The labor force and fuel consumption can be potentially saved through the coordination of truck platoons and air-drag reduction. In this example of Figure 1, four containers are to be delivered to four customer sites 1,2,3,4 from the depot respectively; the drivers often leave containers at customer sites because the (un)packing times are relatively long, therefore the containers are left at the customer sites for (un)packing, and after the (un)packing tasks are fulfilled, the containers are picked up by the same driver and sent back to the depot, where the pickup sites are represented by nodes 5,6,7,8. The pickup sites 5,6,7,8 have the same geographical locations with delivery sites 1,2,3,4, respectively. Drivers can perform multiple trips within their working days, and the maximum number of trucks handled concurrently is taken to be 3 due to practical and law reasons. One typical feasible operation illustrated in Figure 1 is to deliver three containers to 1,2,3 together and drop them at the customers' sites respectively. After the containers' duties are all fulfilled, the driver picks them up in the order 6,7,5 and return back to the depot. Next, the driver continues his/her second trip to finish the service of delivering one container to the customer site 4 and waits until the (un)packing task is finished. Then the driver picks the container up from 8 and returns back to the depot.

In Figure 1, solid lines represent the delivery actions and the dashed lines represent the pickup activities. The definitions of routes and schedule are consistent with [4]. In this simple example, three containers are delivered in one single trip, therefore reducing total service time, and making it feasible to service all four customers within the planning horizon T. Besides, since the driver is only able to handle 3 trucks concurrently, he/she will have to perform two routes to service all 4 customers within his/her working time. In addition to the above mentioned advantages, the air-drag reduction can lead to the variations of the cost on each arc.

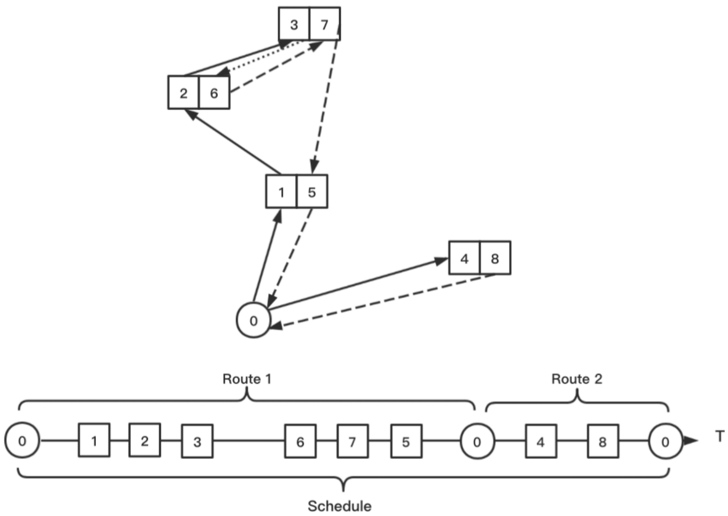



**Figure 1 An illustrative example of 4 customers for the Truck Platooning problem**

As illustrated by the above demonstrative example in Figure 1, the MTPDP with load dependent cost studied in this paper includes the following characteristics: (1) The goal is to minimize the total routing cost; (2) all customer sites must be visited twice, and each customer is split into two task nodes, which are serviced by the same driver, resulting in the precedence relations between the deliveries and pickups (delayed paired precedence relationships); (3) the travel cost on each arc is dependent on the current load onboard (load dependent cost); (4) each driver is allowed to leave and return to the depot for multiple times (MTVRP); (6) multiple homogeneous drivers are available, but are restricted within their planning horizons (limited maximum working time).

## 3 Solution and Contribution

The simultaneous consideration of the above characteristics makes the MTPDP with load dependent cost complicated to solve. We propose a series of set partitioning models and relaxations, and attempt to analyse their relationships to simplify the computation process. At the same time, selected valid inequalities are proposed to tighten the final relaxed set partitioning model without increasing the computation complexity of the pricing problem, which together are incorporated into the exact Branch-and-Price-and-Cut (BPC) algorithm. We propose several novel bounding and pruning procedures based on the analyses of the dual information from the master problem and the pricing problem structures, and design a tailored pulse algorithm to solve the pricing problem.

We attempt to close the research gap in the field of VRP and make contributions to BPC algorithm from the following aspects:

• We study the MTPDP with load dependent cost, which simultaneously considers multi-trip, PDP and load dependent cost, to solve truck platooning planning problem.

• We propose a series of set partitioning models and relaxations and study their relationships, and then we enhance the final relaxed model with selected valid inequalities that will only yield non-positive dual values.

• We solve the complicated pricing problem through a tailored pulse algorithm, and propose novel bounding and pruning procedures based on dual information and problem structure to improve the efficiency of the algorithm.

• We provide an exact BPC algorithm based on the aforementioned analyses for the studied MTPDP with load dependent cost.

• We present extensive numerical experiments based on instances generated based on several well studied problems. Despite the computational complexity of the problem, numerical experiments show that the proposed methods and algorithms are able to solve instances with up to 100 task nodes efficiently, and the addressed MT-CDP-TP can help to save labor costs and reduce total working time without increasing fuel costs significantly.

## References


[1] Bektaş T, Demir E, Laporte G, Green vehicle routing. Green transportation logistics, 243–265 (Springer), 2016.

[2] Chen W, Mes M, Schutten M, Quint J, A ride-sharing problem with meeting points and return restrictions. Transportation science 53(2):401–426, 2019.

[3] Larsson EG, Sennton G, Larson J, The vehicle platooning problem: Computational complexity and heuristics. Transportation Research Part C: Emerging Technologies 60:258–277, 2015.

[4] Mingozzi, A., Roberti, R., Toth, P., 2013. An exact algorithm for the multitrip vehicle routing problem. INFORMS J. Comput. 25 (2), 193–207.

[5] Pelletier S, Jabali O, Laporte G, 50th anniversary invited articlegoods distribution with electric vehicles: review and research perspectives. Transportation Science 50(1):3–22, 2016.

# An evaluation model for time window templates in online grocery delivery

**Katrin Waßmuth**
Chair of Supply Chain Management,
University of Mannheim, Mannheim, Germany
Email: katrin.wassmuth@uni-mannheim.de

**Niels Agatz**
Department of Technology and Operations Management,
Erasmus University Rotterdam, Rotterdam, Netherlands

**Moritz Fleischmann**
Chair of Supply Chain Management,
University of Mannheim, Mannheim, Germany

March 19, 2024

## 1 Introduction

The online grocery business continues to grow rapidly. We see a diversification of offered services, with new business models entering the market and established players rethinking their delivery service offering. Groceries and other products that are perishable or require additional installation services, such as large appliances, require the customer to be at home at the time of delivery. The respective home delivery concept is commonly referred to as *attended home delivery*. To reduce the number of missed deliveries and the amount of time spent waiting for customers to be present, service providers typically ask customers to select a delivery time period from a (subset of a) fixed menu of time windows which we refer to as the initial *time window template*. This research examines the impact of such time window templates on demand and operational performance, including costs, fleet size, and travel distance.

Being profitable is a challenge in grocery home delivery, even more so due to low margins and because of the time constraints on customer order fulfillment. To address these challenges, service providers can take action on both the supply and demand sides. The supply-side levers seek the most efficient fulfillment of demand as, for example, in capacity planning and vehicle routing. Demand management, on the other hand, focuses on shaping customer demand to enhance profitability for given supply capabilities. Typical decisions address the specific time windows offered and the delivery fees charged to customers. Most of the demand management literature to date has focused on operational demand management for a given set of time windows [1, 2]. To our knowledge, there are no modeling efforts for determining these sets. We aim to bridge this gap by answering the following research questions: What are the key trade-offs underlying decisions about the length and number of time windows, including overlapping time windows? How do key problem parameters affect these decisions and their performance?

To answer these questions, we study the impact of the length and number of time windows, including offering overlapping time windows, on expected demand and operational performance. To this end, we develop a time window template evaluation model that captures essential demand and fulfillment interactions by approximating the components of the delivery system through tractable functional expressions based on continuous approximation [3]. We derive structural properties for important classes of demand functions and evaluate relevant time window templates under realistic conditions in a complementary numerical study.

Our contribution to the literature includes (i) developing an evaluation model for time window templates, (ii) evaluating fundamental template modifications; deriving structural properties and identifying various interacting effects, and (iii) providing managerial insights to inform time window template design and capacity planning decisions. Furthermore, our findings contribute to understanding existing practice and help practitioners to review their strategies and adapt to changing business environments.

## 2 Model Formulation

We assume a homogeneous delivery region of fixed size $R$. We consider a single fulfillment shift, which is defined as a contiguous time period during which delivery tours fulfill a set of customer orders. The service is offered according to a time window template $T$, which consists of time windows $i \in T$ of length $l_i$. The total time spanned by the time windows defines the length of the fulfillment shift $L$. We assume a time window template to be valid for the entire delivery region. The expected demand $N_i(T)$ in time window $i$ is measured in the number of orders and we assume a constant order size. Demand in time window $i$ yields a revenue per order of $r_i$, which we assume to be exogenous.

We assume homogeneous delivery tours, meaning that each tour covers the entire fulfillment shift and meets the same amount of demand. We consider fixed and variable fulfillment costs for each delivery tour. Fixed costs are vehicle-related costs $f$ and the cost to cover the stem distance to and from customers $d_0$. Variable costs are determined with a cost per minute $c$ and relate to all operations performed during the fulfillment shift, including travel, service, and any waiting time, which may occur due to the assumption of homogeneous delivery tours. Delivery tours are constrained by time windows, and we take vehicle capacity as endogenous since our model is intended to inform capacity decisions. Lastly, we assume a constant vehicle speed $\alpha$ and a constant service time per order $\tau$.

We evaluate the expected profit of a time window template based on a demand model and a fulfillment model. The fulfillment costs result from the sum of the fixed and variable costs per delivery tour times the number of tours required due to the time window constraints. Equation (1) provides the expression of the expected total profit for consecutive time windows.

$$P(T) = \sum_{i \in T} r_i N_i - \left( f + c\alpha d_0 + c \sum_{i \in T} l_i \right) \max_{i \in T} \left( \frac{\tau N_i + \alpha k \sqrt{R N_i}}{l_i} \right) \tag{1}$$

To evaluate overlapping time windows, we extend the profit function to account for flexibility in redistributing demand. To this end, we introduce decisions $\omega_i \in [0, 1]$ to allocate demand from overlapping time windows $T$ to a corresponding set of non-overlapping time windows $\hat{T}$ of length $\hat{l}_i$, on the basis of which we evaluate fulfillment costs. Accordingly, we express the total fulfillment costs by

$$\begin{aligned} \min_{\omega \in [0,1]} \quad & \left( f + c\alpha d_0 + c \sum_{i \in \hat{T}} \hat{l}_i \right) \max_{i \in \hat{T}} \left( \frac{\tau \theta_i(\omega) + \alpha k \sqrt{R \theta_i(\omega)}}{\hat{l}_i} \right) \\ \text{s.t.} \quad & \theta_i(\omega) = (1 - \omega_{i-1}) N_{i-1} + \omega_i N_i \quad \forall i \in \hat{T} \end{aligned} \tag{2}$$

## 3 Preliminary Results

We derive general structural properties of the time window template evaluation model under stylized conditions. This allows us to isolate the individual effects of template modifications on demand and fulfillment, and to identify the key trade-offs underlying template design decisions. We focus on modifying the length of time windows and modifying the number of time windows, both consecutive and overlapping, and apply functional analysis or numerical experimentation where appropriate.

We show that while offering shorter time windows increases the total demand volume, there is a trade-off between increasing the profit margin by achieving more densely populated time windows and increasing the number of delivery tours and thus, the fixed costs.

Increasing the number of time windows can affect demand in two ways. First, it can increase the total demand volume (*demand attraction*) and second, it can lead to a substitution of demand between the offered options (*demand cannibalization*). To isolate the individual effects, we analyze the two special cases separately, assuming that either only attraction effects or only cannibalization effects occur. We show that under demand attraction and given that the profit margin is positive, there is no trade-off in terms of economic objectives, as offering more time windows always leads to an increase in profits. This is because the demand volume increases for a constant profit margin and constant fixed costs. Under demand cannibalization, the main trade-off is between clustering demand into more densely populated time windows to achieve a higher profit margin, and spreading demand to have fewer delivery tours and thus lower fixed costs.

Adding an overlapping time window can attract new customers without having to extend the length of the fulfillment shift. Furthermore, overlapping time windows provide flexibility to redistribute demand. Since fulfillment efficiency decreases as the demand imbalance between different time windows increases (see Equation 1), we show that the unique optimal strategy is to use this flexibility to balance demand as much as possible.

## 4 Conclusion

This study addresses a critical gap in the existing literature by focusing on time window template design decisions in online grocery delivery. By capturing the fulfillment system through tractable functional expressions and analyzing various template modification operations, the research sheds light on essential trade-offs in deciding the number and length of time windows and to offer overlapping time windows. With this, the research provides a nuanced understanding of the complex relationships between template design decisions, operational performance, and profitability. This knowledge is critical for service providers looking to optimize their delivery operations and meet the challenges of this rapidly expanding market.

## References


[1] D. Fleckenstein, R. Klein, C. Steinhardt, "Recent Advances in Integrating Demand Management and Vehicle Routing: A Methodological Review", *European Journal of Operational Research* 306(2), 499-518, 2023.

[2] K. Waßmuth, C. Köhler, N. Agatz, M. Fleischmann, "Demand management for attended home delivery - A literature review", *European Journal of Operational Research* 311(3), 801-815, 2023.

[3] C. F. Daganzo, "Modeling Distribution Problems with Time Windows: Part I", *Transportation Science* 21(3), 171-179, 1987.

# Stochastic Single-Allocation Hub Location Routing Problem for the Design of Intra-City Express Systems

**Yuehui Wu**

College of Transportation Engineering

Dalian Maritime University, Dalian, China

Email: wu_yuehui_61b@dlmu.edu.cn

**Hui Fang**

College of Transportation Engineering

Dalian Maritime University, Dalian, China

## 1 Introduction

With the development of e-commerce, intra-city express has become an increasingly essential segment in urban logistics systems. As a result, various cargo companies are offering “next day delivery service”, or “delivery within 24 hours service”, e.g., SF Express, Yamato Transport, Japan Post, and so on. For these companies, how to satisfy the intra-city express requests in a practical environment via a cost-efficient way arises as an important issue.

In this study, we focus on the design of an intra-city express system in a practical environment. Parcels are transported from the origin branch offices to the destination branch offices, resulting a many-to-many distribution system. As the parcel flows are usually less-than-truckload (LTL), it is very costly to link them directly [1]. Instead, the hubs and branch offices are connected by local tours instead of direct links. The parcels are picked up at the origin branch offices, sorted in the first hub, possibly transported to the second hub, and delivered to the destination branch offices. Moreover, the collection and distribution processes are conducted at the same time along local tours.

Branch offices usually do not own enough sorting resources (e.g., labour, spaces, and so on). Therefore, parcels are collected in a mixed status and have to be sorted based on their destinations in hubs for further delivery. More specifically, in the current service cycle (e.g., this morning), each vehicle leaves its corresponding hub and traverses a subset of branch offices, while distributing the parcels collected in the previous service cycle (e.g., yesterday morning) and collecting the parcels to be distributed in the next service cycle (e.g., tomorrow morning), i.e., the service system is a warmed-up transportation system [2]. Furthermore, inter-hub transportation is conducted after the vehicles return to the hubs (e.g., at night). Based on the above descriptions, one can find that the main decisions of the planning problem for the referred system include hub location, allocation between branch offices and hubs, and vehicle routing. Moreover, the following three practical conditions are considered:

(i) Capacity. Capacitated hubs and vehicles should be employed due to the limitation of land resources and the limitation of the use of large-volume trucks in urban areas.

(ii) Single-allocation. In practical applications, each branch office is usually served by precisely one hub, as branch offices generally do not have enough sort capacities.

(iii) Stochastic demand. The company might not know the parcel flows beforehand, i.e., the intra-city express demands are stochastic. Please find more details in Section 2.

With these considerations, we propose the planning problem for the intra-city express system, named capacitated single-allocation hub location routing problem with stochastic demand (CSAHLRPSD), belonging to the field of the hub location routing problem (HLRP).

## 2 Mathematical Formulation and Conclusions

The CSAHLRPSD is defined on a complete graph $G = (V, A)$, in which $V$ and $A$ are vertex set and edge set, respectively. Vertex set $V$ consists of potential hub set $H$ and client (branch office) set $C$, while edge set $A$ consists of edges between all vertices. For each pair of clients $i \in C$ and $j \in C$, $d_{ij}$ represents the flow to be transported from $i$ to $j$, which is assumed to be a random variable with known and independent distribution. Without loss of the generality, we assume that all realizations of $d_{ij}$ are greater than 0 and they do not exceed the vehicle capacity.

Each potential hub has a capacity $Q_k$ and a fixed cost $F_k$. It is assumed that only receiving flows from clients consumes hub capacity. Local tours are operated by an unlimited fleet of identical vehicles, and each vehicle is associated with a capacity $q$ and a fixed cost $f$. Furthermore, inter-hub transportation is assumed to be realized by an unlimited fleet of identical trucks, and there is no capacity limitation and fixed cost of the trucks.

Each edge $(i, j) \in A$ is addressed with a nonnegative travel distance $c_{ij}$, satisfying the triangle inequality. Local tour cost depends on the sum of travel distances of the travelled edges, while inter-hub transportation cost is calculated based on travel distances and transferred flows [3]. The unit inter-hub transportation cost and unit local tour cost are denoted as $\alpha$ and $\beta$, respectively. The CSAHLRPSD belongs to the field of stochastic programming. We model the CSAHLRPSD via a multi-stage recourse model as follows:

i) In the first stage, the hub location and the allocation between clients and hubs (long-term decisions) are determined before the random variables $(d_{ij} | i, j \in C)$ are realized, since changing these decisions are costly for a warmed-up system.

ii) Then, in the second stage, the flows to be delivered to each client $i \in C$ $(d_{ji}|j \in C)$ are revealed first (since these parcels have been collected in the previous service cycle, as shown in Section 1), forming the distribution demands $(D_i|i \in C)$. After the distribution demands are known, the vehicles are routed to link the hubs and clients (short-term decisions) before knowing the collection demands $(O_i|i \in C)$.

iii) In the third stage, the collection demands are revealed, and a predetermined recourse policy is applied when a failure occurs. The classical recourse policy is employed, in which the vehicles return to the hub, drop off the collected parcels, and continue their planned route at the point of failure. Furthermore, if the total collection demand assigned to a hub exceeds its capacity due to uncertainty, a penalty cost must be paid, representing the overwork cost. The unit overwork cost is expressed as $\omega$. Note that inter-hub transportation costs are also calculated in this stage.

For each edge $(i,j) \in A$, $x_{ij}$ is a binary variable equal to 1 if there is a vehicle travelling directly from node $i$ to node $j$. $z_{ik}(i \in C, k \in H)$ is a binary variable equal to 1 if client $i$ is allocated to hub $k$. For each node $i \in V$, let $v_i$ be the delivery load on the vehicle just after having served node $i$. $b_k$ is a binary variable equal to 1 if potential hub $k \in H$ is open. Moreover, $y_{ijkl}$ denotes the fraction flow from client $i \in C$ to client $j \in C$ passing hub $k \in H$ and hub $l \in H$. Finally, $e_k$ denotes the overwork load of hub $k \in H$. The CSAHLRPSD is modelled as (1)-(21), in which $Q_1(\boldsymbol{b}, \boldsymbol{z}, \xi)$ and $Q_2(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{z}, \xi)$ are the optimal values of the second-stage problem and third-stage problem, respectively.

$$Stage\ 1\ min \sum_{k \in H} F_k b_k + E[Q_1(\boldsymbol{b}, \boldsymbol{z}, \boldsymbol{\xi})] \tag{1}$$

$$s.t. \sum_{k \in H} z_{ik} = 1\ \forall i \in C \tag{2}$$

$$z_{ik} \leq b_k\ \forall i \in C, k \in H \tag{3}$$

$$z_{ik} \in \{0, 1\}\ \forall i \in C, k \in H \tag{4}$$

$$b_k \in \{0, 1\}\ \forall k \in H \tag{5}$$

Objective function (1) minimizes the operation cost, including the hub fixed cost and expected recourse cost. Constraint (2) is single-allocation constraint. Only open hubs can serve clients, which is ensured by Constraint (3). Constraints (4) and (5) are variable domains.

$$Stage\ 2\ Q_1(\boldsymbol{b}, \boldsymbol{z}, \xi) = \min \sum_{k \in H} \sum_{j \in V} f x_{kj} + \sum_{i \in V} \sum_{j \in V} \beta c_{ij} x_{ij} + E[Q_2(\boldsymbol{b}, \boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\xi})] \tag{6}$$

$$s.t. \sum_{j \neq i \in V} x_{ij} = 1\ \forall i \in C \tag{7}$$

$$\sum_{j \in V} x_{ji} = \sum_{j \in V} x_{ij}\ \ \forall i \in V \tag{8}$$

$$x_{ik} \leq z_{ik}\ \forall i \in C, k \in H \tag{9}$$

$$x_{ki} \leq z_{ik}\ \forall i \in C, k \in H \tag{10}$$

$$x_{ij} + z_{ik} + \sum_{l \neq k \in H} z_{jl} \leq 2 \ \forall i \in C, j \neq i \in C, k \in H \tag{11}$$
$$v_i - D_j + q(1 - x_{ij}) \geq v_j \ \forall i \in V, j \neq i \in C \tag{12}$$
$$v_i \leq q \ \forall i \in V \tag{13}$$
$$x_{ij} \in \{0,1\} \ \forall i \in V, j \in V \tag{14}$$
$$v_i \geq 0 \ \ \forall i \in V \tag{15}$$

Objective function (6) minimizes the vehicle fixed cost, local tour cost, and expected recourse cost. Each client should be visited by exactly one vehicle, which is guaranteed by Constraint (7). Constraint (8) balances the vehicle flow at each node. Constraints (9)-(11) link the allocation variables with routing variables. Constraint (12) describes the delivery load on vehicles. Vehicle capacity constraints are imposed via Constraint (13). Decision variables are defined by Constraints (14)-(15).

$$Stage\ 3\ Q_2(\boldsymbol{b}, \boldsymbol{z}, \boldsymbol{x}, \xi) = min\ R(\boldsymbol{x}, \xi) + \sum_{k \in H} \omega e_k + \sum_{i \in C} \sum_{j \in C} \sum_{k \in H} \sum_{l \in H} \alpha d_{ij} c_{kl} y_{ijkl} \tag{16}$$
$$s.t. \sum_{l \in H} y_{ijkl} = z_{ik} \ \forall i \in C, j \in C, k \in H \tag{17}$$
$$\sum_{k \in H} y_{ijkl} = z_{jl} \ \forall i \in C, j \in C, l \in H \tag{18}$$
$$e_k \geq \sum_{i \in C} \sum_{j \in C} \sum_{l \in H} d_{ij} y_{ijkl} - Q_k \ \ \forall k \in H \tag{19}$$
$$0 \leq y_{ijkl} \leq 1 \ \forall i \in C, j \in C, k \in H, l \in H \tag{20}$$
$$e_k \geq 0 \ \forall k \in H \tag{21}$$

Objective function (16) optimizes the realized recourse cost and overwork cost. Also, the inter-hub transportation cost is calculated via the third term of it. Constraints (17)-(18) correlate the flow variables and allocation variables. Overwork cost for each hub $k \in H$ is calculated via Constraint (19). Constraints (20)-(21) are variable domains. We solve the problem based on the sample average approximation (SAA) scheme used, in which the SAA problem is solved by the algorithm used in [4]. The model and algorithm have been tested on the real instances, in which our method reduce the operating cost by 9.43% on average.

**References**


[1] S. Gelareh & Nickel, "Hub location problems in transportation networks", Transportation Research Part E: Logistics and Transportation Review 47(6),1092-1111, 2011.

[2] H. Karimi, "The capacitated hub covering location-routing problem for simultaneous pickup and delivery systems", Computers & Industrial Engineering 116, 47-58, 2018.

[3] S. Çetiner, C. Sepil, & H. Süral, "Hubbing and routing in postal delivery systems", Annals of Operations research, 181(1), 109-124,2010

[4] Y. Wu, A. G. Qureshi, & T. Yamada, "Adaptive large neighborhood decomposition search algorithm for multi-allocation hub location routing problem", European Journal of Operational Research, 302(3), 1113-1127, 2022.

# A Branch and Cut algorithm for a skip pick-up and delivery problem

**Sanne Wøhlk**
Department of Economics and Business Economics,
Aarhus University, Aarhus, Denmark
Email: sanw@econ.au.dk

**José M. Belenguer**
Department of Statistics and Operations Research
University of Valencia, Burjassot, Valencia, Spain

**Maximiliano Cubillos**
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Milano, Italy

## 1 Introduction

We study a skip pickup and delivery problem which was first presented by [3]. In this problem, $n$ skips must be picked up from their origin by a vehicle that can carry two skips. They are then transported to their predefined treatment facility depending on the content, where they are emptied, after which the empty skips are returned to their origins. The problem origins from the transport of skips from recycling stations to recycling treatment facilities, and back. A sufficiently large fleet of vehicles is available to perform the service of the requests for having the skips emptied. A vehicle route starts from the first pickup, performs a sequence of pickups, treatments, and deliveries, and ends at the last delivery. From the vehicle perspective, the three actions of pickup, treatment, and delivery can be performed in any order that respects the vehicle capacity of two and the route duration constraint, but for the single request, the three actions must be performed in the stated order. The objective of the problem is to identify a set of open routes such that every skip is picked up, emptied, and returned while respecting the vehicle capacity of two and the route duration limit. The total cost to be minimized includes both the transportation cost and the fixed cost for each vehicle used in the solution.

The problem is naturally modeled on a directed graph in which each skip $i$ is associated with three nodes $i \in N_P, i+n \in N_T$, and $i+2n \in N_D$, representing pickup, emptying, and delivery of the skip, respectively. Furthermore, the graph contains dummy nodes for origin and destination of the vehicles. Compared to a complete graph, some edges are removed from the graph due to the logic of the actions. This is illustrated by dashed arcs in Figure 1. For instance, the dashed arc from $N_T$ to $N_P$ represents that any skip can be picked up after an emptying *except* the skip that was just emptied. On the other hand, any skip can be emptied after any pickup, assuming that it is on the vehicle, which is represented by the solid arc from $N_P$ to $N_T$. Based on this graph, [3] provides a mathematical formulation and a number of valid inequalities for the problem.

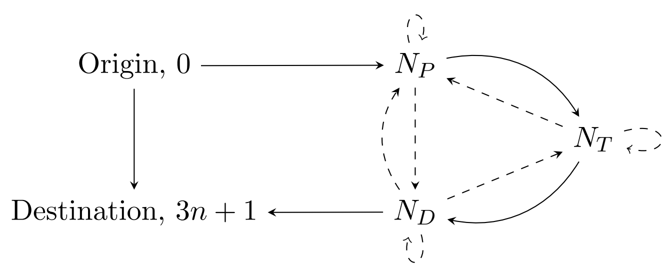



Figure 1: Illustration of the graph structure.

## 2 Solution method

We present a new tighter model, which is based on the same graph and the same variables, and we present several classes of valid inequalities and use them in a Branch and Cut algorithm. In the following, we give a flavor of two classes of our valid inequalities. Further details can be found in [1].

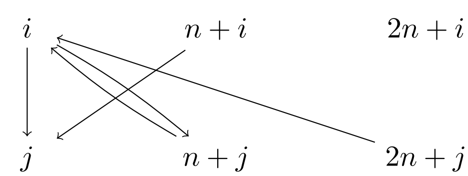



Figure 2: Illustration of an asymmetric cross inequality.

In Figure 2, we illustrate one of our *asymmetric cross inequalities*. In the figure, we show the pickup, emptying, and delivery nodes for two skips: $i$ and $j$. Consider the arcs illustrated in the figure. Our asymmetric cross inequality states that only one of these arcs may be used in any solution. Suppose, for instance, that we use the arc $(i, j)$ which represents that a vehicle will pick up skip $i$ and then immediately after pick up skip $j$. Since each node can only be visited once, the only two arcs that need to be checked are $(n + j, i)$ and $(2n + j, i)$. However, if we also use one of those arcs, it would mean that the vehicle would first empty (or deliver) skip $j$ and then later pick it up, which is not possible. Similarly, it can be argued that any other combination of two of the arcs would lead to conflicts with the order of activities.

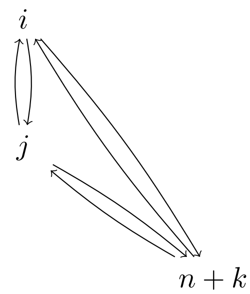



Figure 3: Illustration of an capacity for triplets inequality.

In Figure 3, we illustrate one of our *capacity for triplets inequalities.* They are an extension of the lifted capacity inequalities for the classical pickup and delivery problem presented by [2]. In the figure, we show nodes corresponding to pickup of skips $i$ and $j$ as well as node $n+k$ corresponding to emptying of skip $k$ together with all arcs between these three nodes. Because the vehicle can only carry two skips at a time and because the corresponding skip will be on the vehicle after visiting the node, it is only possible to use one of these arcs. On the other hand, if we instead consider the three nodes corresponding to pickup of skips $i$ and $j$ as well as node $2n+k$ corresponding to delivery of skip $k$, we can use two of the arcs simultaneously because after visiting node $2n+k$, the skip is no longer on the vehicle.

## 3 Results

We have implemented our Branch and Cut in C++ in MS Visual Studio Professional 2015 and executed it on an Intel Xeon CPU with 12 cores running at 3.5 GHz and 64 GBs RAM, using cplex 12.8. All experiments are performed on a single thread, and we have used one hour computation limit.

In order to compare the lower bounds obtained by our tighter model and by our Branch and Cut algorithm to those obtained by the model in [3], we use 17 instances of [3] and consider the optimality gaps computed as $100\frac{UB-LB}{UB}$, where the upper bounds (UB) are obtained by the heuristic presented in [3].

In order to compare the strength of the base models, we use cplex default settings and turn off all cplex cuts. In the original model, the LP-relaxation is solved in 175 seconds on average with an average gap of 85.7. For our model, it takes on average 3.7 seconds to reach an average gap of 83.1 for the LP relaxation. After one hour of computation, our model reaches a gap of 78.7, while the original model obtains 83.7. This clearly indicates that our model is both tighter and easier to solve.

In order to compare the full power of our algorithm, we run the model from [3] with the cplex settings and valid inequalities from that paper and our Branch and Cut algorithm with our cplex settings, valid inequalities from that paper, as well as our new valid inequalities. We now allow cplex default cut generation. After one hour of computation, we reach a gap of 12.4 with our Branch and Cut algorithm, compared to a gap of 15.3 for our rerun of the model from [3]. This supports the strength of our valid inequalities.

## References


[1] J.M. Belenguer, M. Cubillos and S. Wøhlk, "A branch-and-cut algorithm for a skip pick-up and delivery problem", *To be submitted*, 2023.

[2] S. Ropke, J.-F. Cordeau and G. Laporte, "Models and branch-and-cut algorithms for pickup and delivery problems with time windows", *Networks* 49(4), 258-272, 2007.

[3] S. Wøhlk and G. Laporte, "Transport of skips between recycling centers and treatment facilities", *Computers & Operations Research* 145, 105879, 2022.

# A Real-Life Stochastic and Dynamic Pickup-and-Delivery Problem in Megacities

**M. Arya Zamal*, Albert H. Schrotenboer, Tom Van Woensel**
Department of Industrial Engineering and Innovation Sciences,
Eindhoven University of Technology, Eindhoven, the Netherlands
Email*: m.a.zamal@tue.nl

## 1 Introduction

Modern city logistics operations are increasingly centered around two-echelon distribution systems. This is fueled by the need for sustainability, leading to an increase in innovative solutions such as (movable) parcel lockers, electric vehicles, and cargo bikes. At the same time, transportation volumes are only increasing, leading to the opportunity for consolidating freight streams within a city. A megacity such as Jakarta, Indonesia, only exacerbates these developments - bringing the need for optimizing logistics problems of complexity and scale yet unexplored in the current scientific literature.

Inspired by the close collaboration with our industry partner, we study a logistics platform that offers same- and next-day delivery between local shops and consumers. Due to the scale of operations and the need for consolidation, the platform organizes its logistics in geographically separated two-echelon distribution systems. Customers and shops are often located in separate two-echelon distribution systems. Thus, goods need to be consolidated and transported between the two-echelon distribution systems. In the case of megacities, this transport between the two-echelons systems is done via scheduled linehauls (e.g., public transport). The costs and capacity of the linehauls differ throughout the day, and the allocation of orders to the linehauls directly affects the efficiency of the first and last-mile distribution within the two-echelon systems. The joint optimization of the allocation of orders to linehauls and the first- and last-mile distribution is crucial for saving costs.

This paper introduces the stochastic and dynamic order allocation and dispatching problem (SDOA-DP). Its goal is to find a cost-minimizing policy that determines (i) the allocation of orders to linehauls *between* the two-echelon distribution systems and (ii) a dispatching strategy for two-echelon vehicle routes with pickups from shops and deliveries to customers, which happens *within* the two-echelon distribution systems.

We make the following scientific contributions: First, our work enhances existing work on corridor-based logistics (see, e.g., [2]) by explicitly modeling the first and last-mile vehicle routing, linking efficient linehaul planning and efficient first and last-mile routing. Second, we propose a cost-function approximation within a two-stage stochastic programming formulation to dynamically assign orders to linehauls. The cost-function approximation considers the slack for the operations within the two-echelon systems. Embedding within a two-stage stochastic programming formulation allows for incorporating real-time state information in our approximation. Third, we combine this cost-function approximation for the linehaul assignment with a parameterized Adaptive Large Neighborhood Search algorithm as a policy for the first and last mile routing, based on [1]. Fourth, on real-life data from our industry partner, we show the effectiveness of our approach.

The remainder of this abstract is as follows. In Section 2, we present our model and solution approach. In Section 3, we present preliminary results.

## 2 Model and Solution Approach

We model the SDOA-DP on a finite discrete time horizon $\mathcal{T}$. We consider $n$ distinct geographical zones that are each organized as a two-echelon distribution system. That is, each zone has a hub, multiple satellite locations, customers, and shops. For the sake of readability in this extended abstract, we let $n = 2$.

We consider a stochastic and dynamic set of orders $\mathcal{O}$. An order $o \in \mathcal{O}$ is defined by its pickup ($p_o$) and delivery ($d_o$) locations and associated zones, a required service type (i.e., same-day or next-day), hard time windows at the pickup location and the delivery location ($[e_p, l_p]$,$[e_d, l_d]$), and a demand size $q_o$. To transport orders from the pickup to the delivery location, we consider short-haul and long-haul transport. The short-haul transport utilizes vehicle sets $K_1$ and $K_2$ representing the transportation on the first- and second-echelon *within* each zone. They have capacities $\bar{Q}_{k_1}$ and $\bar{Q}_{k_2}$. Each first-echelon vehicle, $k_1 \in K_1$, is stationed at the hub and undertakes tours collecting and distributing parcels between the hub and satellites. Each second-echelon vehicle, $k_2 \in K_2$, is assigned a particular satellite at which it starts and ends its routes. For the long-haul transport between the different zones, we consider two linehaul sets. Slow but relatively cheap linehauls ($\mathcal{L}_1$) or fast but expensive linehauls ($\mathcal{L}_2$). Each type has a given capacity $\bar{Q}_l$, travel time $\tau_l$, fixed costs $F_l$ and variable costs per unit weight $U_l$, and a schedule $S_l \subset \mathcal{T}$ of departure from hub $O^l$ to hub $D^l$, for each $l \in \mathcal{L}_1 \cup \mathcal{L}_2$.

We assume that transportation within each zone cannot be dynamically re-routed, i.e., we can only dispatch vehicles and wait for their return. That is, as soon as we start transporting an order in a two-echelon zone, we cannot alter the associated transportation moves. Our problem can then be formulated as a Markov Decision Process (MDP). For the sake of readability in this abstract, we only introduce the essential mathematical notation.

A state $s \in \mathcal{S}$ describes for each order a status. The status is 'unassigned' (i.e., just known in the system), 'at hub origin' (it is picked up at the shop and is at the hub of the shop's zone), or 'at hub destination' (it is transported via a linehaul to the customer's zone and ready for final dispatch). Additionally, the status includes a time stamp indicating when the order is or will be, at that associated location. Furthermore, it details the number of used vehicles and the time of the day. Note $\mathcal{T}$ comprises multiple days. The action $a \in \mathcal{A}(s)$ describes (i) for orders with status 'unassigned' and 'at origin hub' an allocation to the linehauls, and (ii) a potential dispatch decision in each two-echelon zone. A dispatch decision consists of selecting a subset of orders with the status being 'unassigned' or 'at destination hub' and creating the associated two-echelon vehicle routes. Once dispatched, these routes cannot be rerouted or revoked.

The transition function models the operations to the next decision epoch. This is the change in order status for all dispatched orders and a change for all allocated orders of which the linehaul departs before the next decision epoch. It also incorporated new orders that arrive as expressed by the incoming exogenous information.

The goal is then to minimize the sum of expected travel costs associated with the two-echelon dispatching decisions and linehaul allocation decisions.

## 3 Solution Approach and Preliminary Results

The MDP is clearly impossible to solve to optimality due to the explosion in states, actions, and transitions. To solve the problem, we provide a two-step approach in each decision epoch where we first determine an order allocation decision and subsequently determine a dispatching decision. For both problems, we propose a Cost-Function approximation approach with a limited number of tuneable parameters.

At each decision epoch, the order-allocation problem consists of a set of eligible customer orders $\tilde{O}$, a set of available linehauls $\tilde{L}$, and a set of feasible linehauls $\tilde{L_o} \subseteq \tilde{L}$ that ensures each order $o$ to arrive before their deadline at their destination. A myopic allocation is then given by:

$$\begin{aligned}
\min \quad & \sum_{l\in\tilde{L}} F_l y_l + \sum_{l\in\tilde{L}} \sum_{o\in\tilde{O}_k} c_{lo} x_{lo} && (1)\\
\text{s.t.} \quad & \sum_{l\in\tilde{L}_o} x_{lo} = 1 && \forall o \in \tilde{O}_k \quad (2)\\
& \sum_{o\in\tilde{O}_k} q_o x_{lo} \leq \bar{Q}_l y_l && \forall l \in \tilde{L} \quad (3)\\
& y_l, x_{lo} \in \{0,1\} && \forall o \in \tilde{O},\ \forall l \in \tilde{L} \quad (4)
\end{aligned}$$

Here $x_{lo}$ equals 1 if order $o$ is assigned to linehaul $l$, and $y_l$ equals 1 if linehaul 1 is used. The cost $c_{lo}$ contains a cost-function approximation; it consists of the linehaul unit costs and $\alpha$ times the slack time for further delivery or pickup at each hub. This MIP, we augment in two ways.

First, we consider a consensus approach, by enhancing $\tilde{O}$ with sampled future orders. We then solve $m$ of such MIPs at each decision epoch and select the solution that is, after removing sampled orders, most similar to the other solutions. Second, we again consider sampled future order sets but now create a two-stage stochastic program where we ensure actual orders are scheduled at the same linehauls among the scenarios.

For the *dispatching decision*, we develop a tailored adaptive large neighborhood search, which we parameterize with $\beta$ and $\omega$. Here, we dispatch (i.e., execute the ALNS) if there exists an order with less than $\beta$ time remaining to be transported, i.e., to still arrive on time at its destination. If there is such an order, then we route all orders in that zone having less than $\omega$ time to perform their transportation.

We conduct a preliminary experiment for a 1500-order, 7-day system based on our real-life data. The results are provided in Table 1. Here, MIPSC is our developed solution approach, and three intuitive policies are used as a benchmark. We optimized for each approach $\beta$ and $\omega$, and for MIPSC we also optimized $\alpha$.

Table 1: Result comparison with the heuristics policy

| Policy | Allocation Cost | Routing Cost | Total Cost |
|---|---|---|---|
| Earliest Departure | 47,050 | 103,986 | 151,036 |
| Latest Departure | 48,180 | 136,597 | 184,777 |
| Highest Utilization | 42,345 | 144,923 | 187,268 |
| MIPSC | 41,100 | 100,356 | 141,456 |

At this moment, all our methods are implemented, and at the time of the conference we present, on real-life data, detailed results on the efficiency of combining stochastic programming and cost-function approximation.

## References

[1] Muchammad Arya Zamal, Albert H. Schrotenboer, and Tom Van Woensel, "The Two-Echelon Vehicle Routing Problem with Pickups, Deliveries, and Deadlines", SSRN: 4565559, 2023.

[2] Teodor Gabriel Crainic, Franklin Djeumou Fomeni, Walter Rei, "Multi-period bin packing model and effective constructive heuristics for corridor-based logistics capacity planning", *Computers & Operations Research* 132, 105308, 2021.

# A branch-and-price algorithm for for heterogeneous multi-compartments vehicles routing problem

**Simone Zanda**
Department of Mathematics and Computer Science
University of Cagliari, Cagliari, Italy
Email: simone.zanda@unica.it
**Massimo Di Francesco**
Department of Mathematics and Computer Science
University of Cagliari, Cagliari, Italy
**Roberto Wolfler Calvo**
LIPN, Université Sorbonne Paris Nord Villetaneuse, France
Department of Mathematics and Computer Science
University of Cagliari, Cagliari, Italy

## 1 Introduction

Multi-compartment vehicle routing problems (MCVRPs) are variants of the classical capacitated vehicle routing problem [1] in which several product types must be transported separately and the vehicle capacity is split or can be split into several compartments. The transportation of products in separated compartments is necessary for various real-world problems, e.g., the transportation of dangerous goods, liquid or bulk products, as well as the transportation of food products in different temperature zones. Instead of using one type of vehicle for each product type, it is often beneficial to collect or deliver several product types combined in one vehicle [2]. Various multi-compartment vehicle configurations can be adopted, e.g., the size of separated zones can be fixed or flexible, the assignment of product types to compartments can be preset or arbitrary, and there can exist different (in)compatibilities between different product types or compartments and product types [3]. This article addresses a special type of vehicle routing problem arising in the context of fuel replenishment for gas stations in Sardinia (Italy). The problem tackled reflects the intricate logistical challenges faced by companies involved in the distribution of petroleum products. These problems can be described as a variant of the Unsplit-delivery vehicle routing problem (UD-VRP) considering multiple compartments, multiple trips, multiple products and a heterogeneous fleet of vehicles. The main scientific contribution of this research consists of developing a Branch-and-price ($B\&P$) approach for solving real-life instances to optimality.

## 2 The problem

In the context of petroleum distribution gas stations are supplied by a heterogeneous fleet of vehicles. They load fuel from a central depot and, once completing the deliveries, the vehicles return to the central depot for refueling and continue serving other customers, i.e. this is a multi-trip problem. At the end of the day, the vehicles return to the main depot. The vehicles used for gas station replenishment are divided into one or more compartments, each containing a single petroleum product due to product incompatibility. The compartments have fixed capacities to prevent leakage and contamination. By utilizing flow meters, the content of each compartment can be allocated among multiple customers. Each customer demands one or more (petroleum) products within a specified time window. However, each customer can be served by a vehicle at most and, according to [5], our problem can be classified as an unsplit delivery problem. Therefore, we can call this problem as Multi-Compartment Multi-Trip Multi-Product Vehicle Routing Problem with time windows (MCMTMPVRPs). The challenge of the MCMTMPVRPs lies in minimizing the overall routing costs with capacity, demand and time-window constraints.
Two similar problems to the MCMTMPVRPs were addressed in the literature. In [6] each compartment is supposed to have identical capacity and a single customer may be served by multiple vehicles. The main differences with respect to our problem lie in the fact that each customer is served by a single vehicle and the capacities of the compartments are heterogeneous. The second similar problem is proposed by [7]. The difference with respect to our problem is the flexibility in the size of the compartments. In this paper, instances with up to 15 customers, 9 products, 9 compartments were solved to optimality by a branch-and-price ($B\&P$) algorithm.

## 3 Mathematical model and proposed approach

In this section we present the mathematical formulation for the MCMTMPVRPs. Although one can construct an arc-flow mathematical model of the problem, where variables are associated with arcs in the physical graph, the continuous relaxation of such a formulation is usually rather weak. Path-based formulations are attractive, because a truck trip cannot visit many customers in the real instances motivating this research.

Let $\mathcal{V}$ be the set of customers, $\mathcal{K}$ the set of vehicles, $\mathcal{T}_k$ the set of all potential trips of vehicle $k \in \mathcal{K}$. Each trip starts and ends at the main depot and may visit one or more customers. Let $\delta_t$ be the duration of trip $t$, $T_{max}$ the maximum workload of vehicle $k \in \mathcal{K}$ and $u_{i,t}$ a parameter equal 1 if customer $i$ is served in route $t$, 0 otherwise. To make sure that two trips, assigned to the same vehicle do not overlap, we propose to discretize time and to define a boolean indicator $\alpha_{\tau,t,k}$ that would state if trip $t \in \mathcal{T}_k$ of vehicle $k$ includes instant $\tau$ in its schedule. For each vehicle at each time instant, the sum $\sum_{t\in\mathcal{T}_k} \alpha_{\tau,t,k}$ should than not be greater than one and $nb$ is the number of decimal places defining the instance precision. By using binary variables $W_t$ which select trip $t \in \mathcal{T}_k$ assigned to vehicle $k \in \mathcal{K}$, we formulate the Master problem of MCMTMPVRPs as follows:

$$
\begin{aligned}
(MP) \quad \min \quad & \sum_{t \in \mathcal{T}_k} \sum_{k \in \mathcal{K}} \delta_t W_t && (1)\\
& \sum_{t \in \mathcal{T}_k} \delta_t W_t \leq T_{MAX} \quad \forall k \in \mathcal{K} && (2)\\
& \sum_{t \in \mathcal{T}_k} \sum_{k \in \mathcal{K}} u_{i,t} W_t = 1 \quad \forall i \in \mathcal{V} && (3)\\
& \sum_{t \in \mathcal{T}_k} \alpha_{\tau,t,k} W_t \leq 1 \quad \forall \tau \in \{0, \ldots, T_{MAX} * 10^{nb}\} \quad \forall k \in \mathcal{K} && (4)\\
& W_t \in \{0,1\} \quad \forall k \in \mathcal{K}, t \in \mathcal{T}_k && (5)
\end{aligned}
$$

Consider the Linear Master Problem $(LMP)$ of $(MP)$. The generation of all feasible trips becomes prohibitively expensive in scenarios where the number of trips (variables) is exceedingly large. Following a column generation $(CG)$ procedure ([9]), $(LMP)$ is solved for a small subset of trips for all vehicles. At each iteration of the $CG$ procedure, we solve a pricing problem determining feasible trips for each possible packing scheme of vehicles. The pricing problems are solved by the algorithm PULSE [10]. The column generation is repeated at all nodes of the enumeration tree and results in a full-blown Branch-and-price $(B\&P)$ algorithm with branching on arcs.

## 4 Discussion of results and conclusions

Here, we summarize the performance of the proposed $(B\&P)$ algorithm and present concise results that prove its effectiveness in solving the MCMTMPVRPs. The analysis includes a preliminary analysis of computational times and optimality gaps or real data. They consist of daily orders placed over a period of two months. We took the daily demand for each customer and the data of vehicles, including their compartment capacities. Moreover, since we know the customer locations, we determined the geographic coordinates for each customer, enabling the creation of a time matrix. We propose preliminary results in Table 1 using this notation: $ist$ = number of the instance, $|\mathcal{V}|$ = the number of customers, $|\mathcal{K}|$ = the number of vehicles, $|\mathcal{C}|$ = the number of compartments of the vehicle, $LB_{afm}$ = lower bound of arc-flow mathematical model, $UB_{afm}$ = upper bound of arc-flow mathematical model, $GAP_{afm}$ = the gap between the optimal solution and the lower bound of arc-flow mathematical model, $LB_{cg}$ = lower bound of CG, $UB_{cg}$ = upper bound of CG, $GAP_{cg}$ = the gap between the optimal solution and the lower bound at the end the $CG$, $T_{cg}$ = the time (in seconds) at the end of the column generation, $GAP_B$ = the gap between the optimal solution and the lower bound at the end the of Branching phase. $T_B$ = the time (in seconds) to obtain the optimal solution in the branching phase.

In the presentation, we will also present an arc-flow mathematical model of the problem. Moreover, we extrapolate the implications of our findings, discussing the potential impact on the operational strategies of petroleum distribution companies. We will also highlight the strengths and limitations of our study, providing a grounded perspective on the MCMTVRP's place within the broader field of logistics. We will conclude with a brief synthesis of the research's main points, emphasizing the innovative aspects of our algorithmic solution and its practical significance.

| ist | $\mathcal{V}$ | $\mathcal{K}$ | $\mathcal{C}$ | $LB_{afm}$ | $UB_{afm}$ | $GAP_{afm}$ | $T_{afm}$ | $LB_{cg}$ | $UB_{cg}$ | $GAP_{cg}$ | $T_{cg}$ | $GAP_B$ | $T_B$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 7 | 2 | 5-2 | 9459 | 9459 | 0 | 0.2 | 9459 | 9459 | 0 | 1.3 | | |
| 2 | 7 | 3 | 1-2-2 | 12194.2 | 12194.2 | 0 | 0.17 | 12194.2 | 12194.2 | 0 | 1.5 | | |
| 3 | 10 | 3 | 3-4-2 | 13707.2 | 13707.2 | 0 | 3.3 | 13707.2 | 13707.2 | 0 | 12.4 | | |
| 5 | 13 | 2 | 3-2 | 8025.5 | 8025.5 | 0 | 12.3 | 8025.5 | 8025.5 | 0.19 | 35.18 | 0 | 101.8 |
| 6 | 15 | 3 | 5-2-4 | 16967.2 | 16967.2 | 0 | 14.82 | 16967.2 | 16967.2 | 0 | 39.9 | | |
| 7 | 16 | 2 | 2-2 | 13579.4 | 13579.4 | 0 | 7.5 | 13579.4 | 13579.4 | 0 | 120.3 | | |
| 8 | 18 | 3 | 3-2-4 | 17505.1 | 17505.1 | 0 | 20.3 | 17505.1 | 17505.1 | 0 | 52.4 | | |
| 9 | 20 | 6 | 3-3-2-2-4-4 | 27209,3 | 30715,3 | 11.41 | TL | 30126.6 | 30831.6 | 2.2 | 95.4 | 0 | 2559.0 |
| 10 | 28 | 6 | 3-3-3-2-2-4 | 32435.1 | 45783,2 | 29.15 | TL | 44477.0 | 45329.8 | 1.6 | 261.8 | 0 | 1445.5 |
| 12 | 36 | 9 | 3-2-4-3-2-4-3-2-4 | 39701.9 | 109875.4 | 63.87 | TL | 101144.1 | 105086.7 | 3.75 | 1554.8 | - | TL |
| 13 | 38 | 8 | 3-2-4-3-2-4-3-2 | - | - | - | TL | 75740.4 | 81329.9 | 6.87 | 893.7 | - | TL |
| 14 | 40 | 8 | 3-2-4-3-2-4-3-2 | 46799.6 | 125010.3 | 62.56 | TL | 102207.6 | 102677.0 | 0.4 | 662.5 | 0.4 | TL |
| 15 | 45 | 8 | 3-2-4-3-2-4-3-2 | 38300.7 | 89770.7 | 57.33 | TL | 77558.6 | 82717.0 | 6.23 | TL | - | TL |

Table 1: results with compartments of different sizes

# References


[1] P.Toth, D.Vigo (Eds.) (2014). Vehicle routing. SIAM series on optimization: 18. Philadelphia, PA: Society for Industrial and Applied Mathematics.

[2] Muyldermans, L., & Pang, G. (2010). On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. *European Journal of Operational Research*, 206(1), 93–103.

[3] Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., & Limbourg, S. (2014). Vehicle routing problems with loading constraints: state-of-the-art and future direc- tions. OR Spectrum, 37(2), 297–330.

[4] Koch, H., Henke, T., & Wäscher, G. (2016). A genetic algorithm for the multi- compartment vehicle routing problem with flexible compartment sizes. Femm working papers 160004 Otto-von-Guericke University Magdeburg, Faculty of Economics and Management.

[5] Coelho, L.C., Laporte, G., 2015. Classification, models and exact algorithms for multi-compartment delivery problems. *Eur. J. Oper. Res. 242 (3), 854–864.*

[6] L. Wang, J. Kinable, T. van Woensel.(2020). The fuel replenishment problem: A split-delivery multi-compartment vehicle routing problem with multiple trips, *Computers & Operations Research, Volume 118.*

[7] Katrin Heßler. (2021). Exact algorithms for the multi-compartment vehicle routing problem with flexible compartment sizes, *European Journal of Operational Research, Volume 294*

[8] Claudia Archetti, Nicola Bianchessi, M. Grazia Speranza. (2015). A branch-price-and-cut algorithm for the commodity constrained split delivery vehicle routing problem, *Computers & Operations Research, Volume 64.*

[9] G. Desaulniers, 2010. Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. Oper. Res. 58 (1), 179–192.

[10] Leonardo Lozano, Daniel Duque, Andrés L. Medaglia. (2015) An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints. Transportation Science.

# The Impact of Vehicular Technologies on Curbside Recyclable Waste Collection

**Hani Zbib**
Department of Analytics, Operations, and Information Technologies,
University of Quebec in Montreal, Montrea, Canada
Email: zbib.hani@uqam.ca

**Gilbert Laporte**
Department of Decision Sciences, HEC Montréal, Canada
School of Management, University of Bath, United Kingdom

## 1 Introduction

Traditional curbside waste collection consists of collecting one unsorted waste stream from households using single compartment vehicles and transporting it for disposal. Such collection services have been adequately modeled in the past as a Capacitated Arc Routing Problem (CARP) [1]. However, with recycling gaining traction in the context of circular economies, collection services have become more complex. Current collection systems targeted towards waste disposal are ineffective for recycling, hence requiring the rethinking of waste infrastructure, technologies, and collection policies [2].

Given a curbside recyclable collection service based on source separation of the waste and co-collection, the existence of many technological features on collection vehicles and all their possible configurations complexifies the collection service. The proper selection of a technology can significantly increase the efficiency of the collection process by optimizing the design of collection routes and their costs, optimizing the types and numbers of vehicles needed, as well as the composition of the collection crew. To effectively model curbside recyclable collection under a source separation and co-collection policy, the problem needs to factor in all the above mentioned elements that impact the efficiency of collection.

The aim of this work is therefore to derive a comprehensive taxonomy of the technological features on collection vehicles that affect the collection service, demonstrate how these can be modeled taking into account the collection path and the nature of the street network, develop a matheuristic-based solution strategy, and test the strategy on real-life instances from waste collection operations of six municipalities in Denmark [3]. The algorithm enables municipalities and waste collection companies to determine the best fleet mix to adopt, and can also help recycling trucks manufacturers to determine which vehicle types and configurations are more attractive to offer to their customers.

## 2 Curbside waste collection technologies

We derive a taxonomy of technological features on collection vehicles that affect the recyclable collection service. To this end, we analyze the catalogue of vehicle products of 18 of the largest waste collection vehicle manufacturers based on their webpage, product brochures, and technical data sheets. The taxonomy contains five technological specifications: the collected waste bins, the body, the loading mechanism, the compaction mechanism, and the collection crew. Each specification, its attributes, and the description of the attribute or the most common configuration are presented in Table 1.

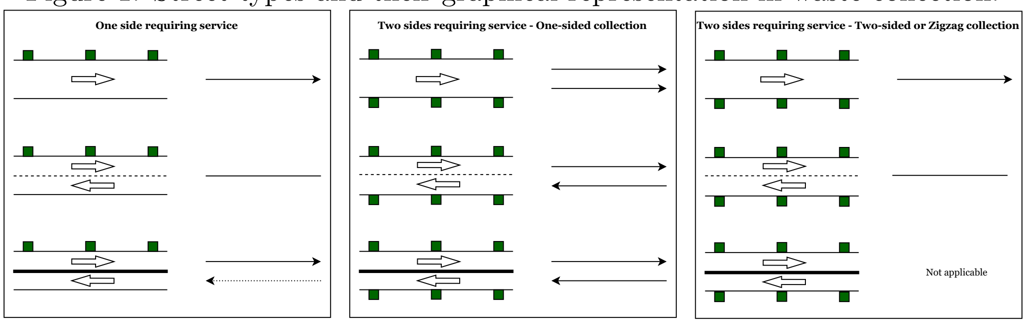
Table 1: Technological specifications and attributes that affect the collection service.

| Technological specification | Attribute | Description and common configurations |
|---|---|---|
| Waste bin | Type | Residential, commercial |
| | Size | Expressed in volume. |
| Body | Size | Expressed in volume. |
| | Number of compartments | Varies between 1 and 4. |
| | Size split among compartments | Expressed in percentage of total volume, e.g., 40/60, 50/25/25. |
| | Loading side | Front, side, rear. |
| Loading mechanism | Arm lift | Semi-automated, automated gripping claw. |
| | Number of bin tippers | Typically 1 or 2 per semi-automated arm lift. |
| | Cycle time | Expressed in seconds. |
| Compaction mechanism | Presence of a mechanism | Packer. |
| | Compaction force | Expressed in mass per volume. Can vary based on the collected waste stream. |
| | Automation | Semi-automated, automated. |
| | Cycle time | Expressed in seconds. |
| Collection crew | Driver | Required for any vehicle. |
| | Collector | 1 or 2 members might be required based on the loading side and arm lift automation. |

## 3 Modeling curbside recyclable waste collection

To model the Capacitated-Arc Routing Problem for Recyclable Waste Collection (CARP-RWC), we first need to translate each street type and the number of sides requiring service into a mixed graph based on the collection path followed by the waste collectors (one-sided, two-sided, or zigzag) as in Figure 1. Given a set of waste streams, with each required link in the mixed graph is associated a traversal cost and time, and the link contains a number of bins and a total demand for each waste stream. We consider a fleet of heterogeneous vehicle based at a depot node, where the heterogeneity stems from the different technological specifications and attributes in Table 1. Each vehicle is characterized by the capacity of its body, the number of compartments, the size split among compartments, the existence of a compaction mechanism, the compaction factor for each waste stream, the automation of the compaction mechanism, the compaction cycle time, the loading side, the type of arm lift, the number of bin tippers, the loading cycle time, and the composition of the collection crew required to operate the vehicle and its technologies.

Figure 1: Street types and their graphical representation in waste collection.




The total working time of a vehicle is divided into the total service time and the total deadheading time. The deadheading time is independent of the vehicle type, while the service is highly dependent on the vehicle's attributes. The service time is dictated by the loading mechanism, the collection crew, the street type, and the collection path. We model the service time of each link, waste stream, and vehicle as a function of the loading cycle time to service all bins on the link, and the traversal time of the link while servicing it. To calculate the loading cycle time, we divide it into its three operations: bin wheeling, bin lifting, and bin emptying, which vary from vehicle type to another. We assume that the total working time of any vehicle cannot exceed a maximal workday duration.

The objective of the CARP-RWC is to find a set of least-cost routes that start and end at the depot node, such that all waste bins on a link for each waste stream are collected exactly once by the compartments of one vehicle collecting that waste stream in at least one of its compartments, without violating the capacity of any compartment and the total workday time of each vehicle. The total cost has three components: the service cost, the deadhead cost, and a daily vehicle operational cost.

## 4 Solution strategy

The CARP-RWC is made up of four decision levels: 1) selecting the configurations of vehicles to include in the fleet mix, 2) selecting the number of vehicles needed for each selected configuration, 3) assigning waste streams to the compartments of the selected vehicles, 4) designing a capacity- and time-feasible route for each vehicle. Our solution strategy consists of a two-phase sequential matheuristic, preceded with a pre-phase where the service times are calculated for all vehicle configurations. The first phase tackles decision levels (1) and (3). It determines which vehicle configurations to include in the fleet mix, and which assignments of waste streams to the compartments of the selected configurations are more attractive from a cost perspective. The second phase tackles decision levels (2) and (4), and is given as input the chosen vehicle configurations and a subset of attractive waste stream to compartment assignments for each configuration. It consists of determining a set of least-cost routes that respect capacity and time constraints, while at the same time assigning a vehicle configuration and waste stream assignment to the selected routes. The routing phase iteratively orders all required links in a giant tour, and then uses an extension of the commodity-split tour splitting algorithm (CSTA) of [4] to split the giant tour into feasible routes. Our version of the CSTA, in addition to including multi-compartments and waste stream-dependent compartment capacities, considers a mixed graph, service times, a work day limit, and an explicit choice of the best vehicle configuration-assignment pair to service each route. This sequential solution strategy highly reduces the scope of the CARP-RWC solution space.

## 5 Preliminary results and discussion

We test the solution strategy on the instances of [4] for the Commodity-Split Multi-Compartment CARP extended to the CARP-RWC by transforming the undirected graph into a mixed graph, and generating values for the number of bins and time related parameters for each link. The graphs contain up to 7,110 edges, 3,797 required edges, and sorting in three, four, and six waste streams. We generate new vehicle files for each collection path (one-sided, two-sided, and zigzag) resulting in a fleet of 25, 40, and 30 vehicles respectively with between one and three compartments. The results show that the first phase of the solution strategy is effective at choosing the most attractive vehicle configurations, and the routing phase finds good solutions for the problem within relatively fast times.

## References


[1] B.L. Golden and R.T. Wong, "Capacitated arc routing problems", *Networks* 11(3),305-315, 1981.

[2] G. Ghiani, D. Laganá, E. Manni, R. Musmanno, and D. Vigo, "Operations research in solid waste management: A survey of strategic and tactical issues", *Computers & Operations Research* 44,22–32, 2014.

[3] L. Kiilerich and S. Wøhlk, "New large-scale data instances for CARP and new variations of CARPs", *INFOR: Information Systems and Operational Research* 56(1),1–32,2018.

[4] H. Zbib and G. Laporte, "The commodity-split multi-compartment capacitated arc routing problem", *Computers & Operations Research* 122:104994, 2020.

# Consistent Time Window Assignments for Stochastic Multi-Depot Multi-Commodity Pickup and Delivery

**Shohre Zehtabian**
Faculty of Economics and Management,
Otto-von-Guericke Universität Magdeburg, Germany
Email: shohre.zehtabian@ovgu.de

**Marlin W Ulmer**
Faculty of Economics and Management,
Otto-von-Guericke Universität Magdeburg, Germany

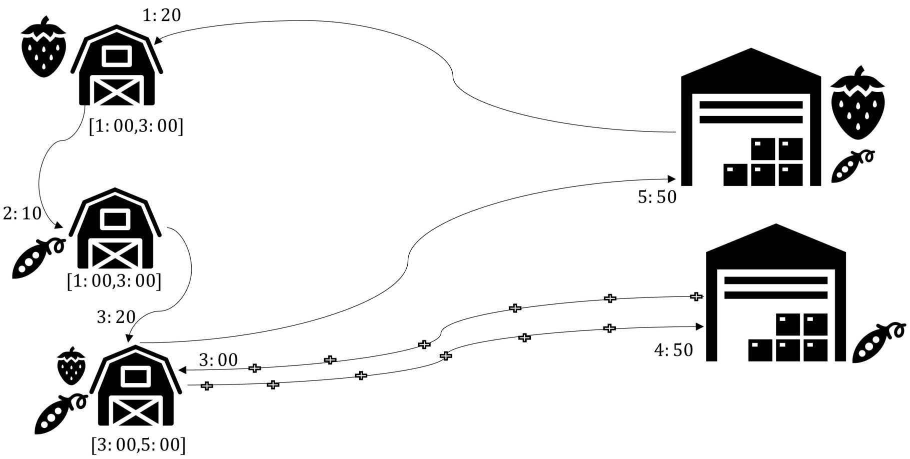
**Introduction.** There is an increasing trend worldwide for schools, hospitals, and company canteens to source their groceries locally, if possible. Products are picked up at farmers and transported to local distribution centers or *food hubs*. Food hubs are a special type of food distribution infrastructure where the collected food products from local/regional farmers are consolidated and then distributed to businesses such as grocery retailers and supermarkets, food and catering services, or institutional kitchens, e.g. school canteens [4].

With this new trend, new challenges occur. The supply chain becomes shorter, but at the same time more fragmented, as individual farmers may offer only a subset of groceries and in limited quantities. Thus, the regular collection and transportation process of products from farmers to the local distribution centers is not trivial and becomes an important cost factor, given the relatively high salaries for truck drivers. Furthermore, the smaller local farmers are often responsible for the entire process of farming and handling the shipping. Therefore, a seamless and reliable operation is vital, to ensure that the farmers can work effectively. Here, timing is of particular importance. Ideally, the time of the day when groceries are picked up does not vary because the farmers need to ensure that the products are ready for pickup and at the same time, they need to follow their daily routines without many additional interruptions. The importance of this time consistency is amplified when considering that farmers may provide products for several distribution centers, all operating their own vehicles, resulting in multiple pickups per day. The goal of a company operating the food hubs is therefore to determine a consistent time window (TW) for each farmer that will allow efficient transportation while keeping TW-violations at a minimum.

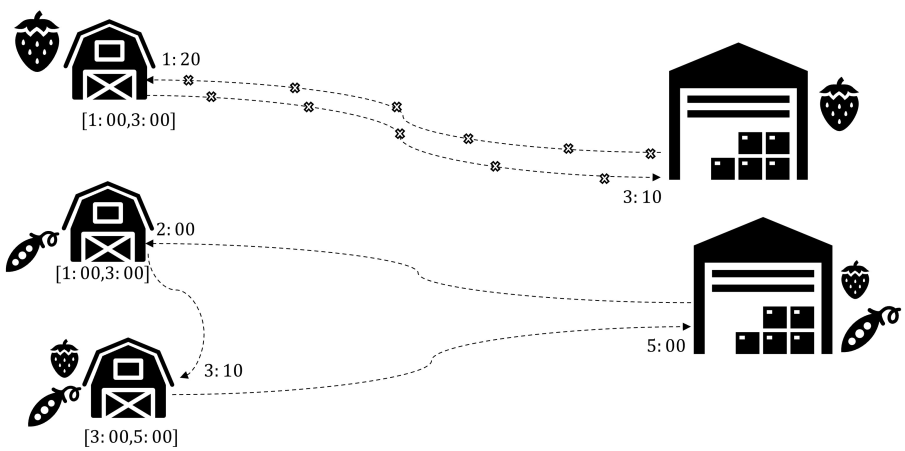
Setting TWs is challenging for a variety of reasons. Vehicles often visit several farmers per trip to collect different products. Furthermore, vehicles from different distribution centers may visit the same farmer on the same day. Thus, when setting TWs for farmers, the routing of the fleet for all distribution centers has to be considered. While this is already a challenging optimization problem, the difficulty is increased by demand uncertainty at distribution centers. There might be days with low demand while on other days the demand might be higher than expected. Figure 1 illustrates an example of a distribution network consisting of three farms (on the left) and two distribution centers (on the right). The available supplies of two fresh products and the assigned TW for each farmer are denoted next to and below each farm. The demands for fresh produce are presented next to each distribution center for two days, i.e. two demand scenarios, in Figures 1a and 1b. A subset of routes visiting farmers to collect and transport fresh foods from farms to distribution centers are provided for each demand scenario with the arrival (return) times of the truck at each farm (distribution center) denoted by the arrow entering the location. This figure highlights the complexity of assigning consistent TWs to farmers in the presence of demand uncertainty. Therefore, TW-decisions have to account for varying demand scenarios and consequently, for different daily routing solutions. The resulting decision process is a combination of decisions made at the tactical and operational levels of the planning.

**Problem.** In this paper, we present the problem of assigning consistent time windows for the collection of multiple fresh products from local farmers and delivering them to distribution centers for consolidation and further distribution in a short agri-food supply chain with stochastic demand. We formulate the problem as a two-stage stochastic program. In the first stage, the time windows are assigned from a set of discrete time windows to farmers, and in the second stage, after the demand is realized the collection routes are planned by solving yet a newly introduced multi-depot multi-commodity team orienteering problem with soft time windows. The objective is to minimize the routing cost while ensuring TW-consistency for serving farmers. Our problem has similarities with the TW-assignment vehicle routing problem introduced in [2] and its discrete version presented in [3].

**Methodology.** As solving the full deterministic equivalent problem for several scenarios of realistic sizes is computationally intractable, we design a heuristic solution approach based on a scenario decomposition technique; namely the progressive hedging algorithm (PHA) [1]. Over a number of iterations, our (heuristic) PHA solves the individual scenarios and derives a consensus first-stage solution that is fed in the next iteration of PHA. Over time, the weight of the current consensus solution is increased, likely leading to convergence to a common solution for all scenarios. As the second-stage decisions are by themselves very




(a) Day 1.




(b) Day 2.

Figure 1: Illustrative example of a network of farmers (on the left) with their available supply and assigned TWs, distribution centers (on the right) with two demand scenarios, and routes associated with each scenario collecting and transporting fresh produce from farmers with the arrival times of the trucks at farmers.

challenging, we rely on a matheuristic. Our approach solves a route-based formulation of the second-stage multi-depot multi-commodity orienteering problem with soft TWs via a commercial solver by heuristically generating a pool of routes based on demand scenarios.
**Results.** We test our method for a variety of instance settings to analyze both the methodology and the problem. We derive the following main insights: (i) for small instances, our PHA provides solutions very close to optimality. For instances of real-world size, it outperforms other scenario-based methods and heuristic policies for all instances, (ii) compared to optimizing on expected demands, our policy reduces both travel time and farmer inconvenience at the same time, (iii) "soft" TW-consistency with rare and minor

violations can be achieved at a cost increase of about 3%. Guaranteed TW-consistency increases overall routing cost by about 7% in our setting, (iv) wider TWs can keep the cost of consistency reasonable while narrower TWs can become relatively costly.

In the case of soft TW-consistency versus guaranteed TW-consistency, for example, Figure 2 depicts the expected routing cost, when the penalty on TW-violations varies, for a group of instances. The routing costs increase (decrease), and the more (less) critical the TWs become, on average. When no TW is enforced, the expected routing cost has the smallest value, meaning the trucks save the most time. In case of the minor penalties on TW-violations, the expected routing cost increases by 3%. Imposing severe penalties on TW-violations will increase the routing costs by 7%, meaning the trucks will lose more time, on average.


Figure 2: Evolution of the expected routing cost for varying levels of penalty on TW-violations.

## References

[1] R. T. Rockafellar and R. J.-B. Wets, "Scenario and Policy Aggregation in Optimization Under Uncertainty", *Mathematics of Operations Research* 16(4), 119-147, 1991.

[2] R. Spliet and A. F. Gabor, "The Time Window Assignment Vehicle Routing Problem", *Transportation Science* 49(4), 721-731, 2015.

[3] R. Spliet and G. Desaulniers, "The discrete time window assignment vehicle routing problem", *European Journal of Operational Research* 244(2), 379-391, 2015.

[4] L. Palacios-Argüello, I. Sanchez-Diaz, J. Gonzalez-Feliu, and N. Gondran, "The Role of Food Hubs in Enabling Local Sourcing for School Canteen", in *Food Supply Chains in Cities: Modern Tools for Circularity and Sustainability*, E. Aktas and M. Bourlakis (eds), 233-263, Springer: Palgrave Macmillan Cham, Switzerland, 2020.

# 2–Index formulation for Multiple Allocation Hub Location Problems

**Nicolás Zerega**
Departamento de Estadística e Investigación Operativa
Universidad de Cádiz, Cádiz, Spain
Email: nicolas.zerega@uca.es

**Elena Fernández**
Departamento de Estadística e Investigación Operativa
Universidad de Cádiz, Cádiz, Spain

## 1 Introduction

Hubs serve as central nodes within hub-and-spoke networks. One of the main characteristics of hub-and-spoke networks is that direct connections between origin/destination pairs are be replaced by fewer, indirect but privileged connections by using the hub facilities as transshipment, consolidation, or sorting points.

Hub location problems (HLPs) consider strategic decisions (hub location and link activation) and operational decisions (demand routing) guided by some criteria, including, but not limited to, cost minimization, profit maximization or travel time minimization.

Our research considers a Multiple Allocation HLP (MA–HLP) with hub setup costs. Allowing non-hub nodes to be allocated to multiple hubs enables different routing options for demands with a common origin.

Commonly used Mixed Integer Linear Programming (MILP) formulations for MA–HLPs often resort to 4-index or 3-index variables. Specifically, 4-index formulations, known as path-based formulations, connect each commodity via a dedicated path where all intermediate nodes are activated as hubs. While these formulations usually provide very tight lower bounds, one of their main drawbacks is the increase of the number of decision variables as the amount of nodes in the network rises, thus limiting the size of the instances that can be solved efficiently.

The 3-index formulations address the previous issues by modeling HLPs as multi-commodity flow problems. In this type of formulations, decision variables are associated with the flows, aggregated over all origins, traversing each link. This implies that decision variables increase in the order of O$(n^3)$, generally producing smaller models. They have been deeply studied, even when they generally produce worse bounds than their 4-index counterparts.

## 2 Our Contribution

We introduce a formulation for MA–HLPs by using only 2-index variables. Previous studies have used 2-index variables for the Single Allocation HLP [2, 1], but the proposed models do not apply to the Multiple Allocation case that we study. To the best of our knowledge, the formulation that we propose is the first one that considers only 2-index variables for the Multiple Allocation case.

Our formulation is developed over a graph $G(V, E)$ defined by node set $V$ and edge set $E$. Demand is expressed by a set of commodities indexed in set $R$. Each commodity $r \in R$ is defined by the triplet $(o_r, d_r, w^r)$, with both origin $(o_r)$ and destination $(d_r)$ nodes in $V$ where $w^r$ denotes the demand that must be sent from $o_r$ to $d_r$.

We define binary and continuous variables associated with strategic and operational decisions, respectively. In addition to the usual binary hub location variables $z$, we categorize connections into three types: access (from non-hub to hub), distribution (from hub to non-hub), and interhub. Hence, we also define link activation variables $(x^1, x^2, y)$ to address access, distribution and interhub connections. Furthermore, we define continuous flow variables $(h^1, h^2, f)$ for the flows through access, distribution and interhub arcs, respectively.

The formulation includes constraints on the binary variables $(x^1, x^2, y, z)$ that guarantee the feasibility of the obtained solution network, following the hub-and-spoke structure, as well as flow balance constraints relating the flows $(h^1, h^2, f)$ through the different types of connections.

The validation of the proposed formulation is guaranteed by the introduction of *logic based* constraints, which ensure the correct routing of commodities throughout the network while also enforcing that the flows through the different connection types are consistent with such routes. These feasibility constraints identify the best consistent paths on an auxiliary backbone network, together with the total amount of flow that must be routed through each activated connection.

We also introduce several families of valid inequalities that reinforce the formulation and improve the associated LP bound. These include a family of *aggregated demand* inequalities as well as another family denoted as *residual capacity* inequalities.

The *aggregated demand* inequalities impose that the minimum amount of flow through the dicut associated to a given set of nodes must, at least, satisfy the total demand of commodities whose origin/destination pairs lie in different nodes set of the dicut.

The *residual capacity* inequalities impose that the total amount of "extra" capacity in the solution network, i.e. the flow through all active connections minus the demand of directly connected commodities, must be sufficient to satisfy the demands of all non-directly connected commodities.

## 3 Separation procedures

Since the families of feasibility constraints and reinforcing inequalities introduced are of exponential size in the number of nodes, we develop exact and heuristic separation procedures for them. These separations are embedded within a branch-and-cut algorithm in which violated constraints and inequalities are added dynamically.

We separate the *logic based* constraints as lazy constraints at the nodes of the enumeration tree with integer solutions. An auxiliary problem is solved where the hub-and-spoke network induced by the binary variables is fixed, and shortest paths are identified for all commodities. The information derived from the shortest paths allows to identify violated feasibility constraints, when they exist.

All other inequalities can be separated for fractional solutions as well. For *aggregated demand* inequalities we introduce an exact separation, consisting in solving a max–flow problem, and a heuristic separation by identifying minimum cut trees. The *residual capacity* inequalities separation requires solving an auxiliary MILP problem.

## 4 Results

We benchmark, compare and analyze several branch-and-cut strategies to solve the formulation. These strategies involve different combinations regarding the inequalities used and their separation.

Numerical results on well-known instances from the literature, with up to 80 nodes, are presented, analyzed, and compared to the state of the art [3] providing insights on the competitiveness of our formulation and allowing for the exploration of further developments in the future.

## References

[1] I. Espejo, A. Marín, J. M. Muñoz-Ocaña, and A. M. Rodríguez-Chía. A new formulation and branch-and-cut method for single-allocation hub location problems. *Computers & Operations Research*, 155:106241, July 2023.

[2] M. Labbé and H. Yaman. Projecting the flow variables for hub location problems. *Networks*, 44(2):84–93, July 2004.

[3] S. Wandelt, W. Dai, J. Zhang, and X. Sun. Toward a reference experimental benchmark for solving hub location problems. *Transportation Science*, 56(2):543–564, March 2022.

# Dynamic Capacity Management for Crowdsourced Delivery

**Shu Zhang**
School of Economics and Business Administration,
Chongqing University, Chongqing, China
Email: zhangshu@cqu.edu.cn

**Zhiwei Zhang**
School of Economics and Business Administration,
Chongqing University, Chongqing, China

**Justin W. Goodson**
School of Business,
Saint Louis University, Saint Louis, United States

## 1 Motivation and Problem Formulation

With the continue growth in E-retailing, online retailers are facing increasing pressure in making fast delivery. The growing popularization of crowdsourced delivery serves as a way to alleviate the challenges faced by the retailers. For instance, Amazon flex hired crowdsourced drivers to deliver online orders. Drivers can either use their own vehicles or rent vehicles from car-sharing companies to provide delivery services. While more and more car-sharing companies are employing electric vehicles in their business, we investigate a crowdsourced delivery problem in which crowdsourced drivers rent electric vehicles to deliver orders for an online retailer. We assume that crowdsourced drivers will send out their work applications when they are available to provide delivery services. The lengths of time the crowdsourced drivers are available to work are specified in their work application. The system needs to decide whether or not to hire a newly arrived crowdsourced driver. If hiring a crowdsourced driver, the system will pay for the driver's whole working duration. After hiring the driver, the system will then assign the driver to a nearby rental station, where the driver can pick up the electric vehicle reserved for her. The battery level of the vehicle that can be reserved for a driver is uncertain. We assume that both crowdsourced drivers and customer orders arrive stochastically into the system. Each order needs to be scheduled for delivery by the end of the system's operation time horizon. If not enough crowdsourced drivers are hired, all the unscheduled drivers will be outsourced to third party drivers at higher costs. We assume that after finishing delivering all orders in her route, each driver will return the electric vehicle to the rental station closest to the last customer in the route and then go off work.

We formulate the problem as a Markov decision process (MDP) model. A decision epoch is triggered by the arrival of a new crowdsourced driver. The decision to make at a decision epoch is whether to hire the new crowdsourced driver. Between two decision epochs, we observe the information on newly arrived orders, the availability of the new crowdsourced driver, the location and the battery level of the EV that can be rented by the new driver. The objective of the MDP model is to minimize the total expected costs over the entire horizon.

## 2 Solution Approach and Analytical Results

Due to the curse of dimensionality presented in our model, we are not able to find optimal policies via backward dynamic programming approach. Instead, we propose a cost function approximation method to solve the problem. Our cost function approximation is motivated by the fact that with the uncertainties in the availability of crowdsourced drivers, the energy level of available electric vehicles, and customer demands, we may not be able to hire the "correct" number of drivers when making driver recruitment decisions. We may either hire too many or too few drivers. Hiring too many crowdsourced drivers may result in paying too much salary to the drivers whose delivery capacity may not be used up, while hiring too few drivers will result in a shortage in delivery capacity and the platform needs to recruit third-party drivers at greater costs. In this study, we focus on developing policies that balance the hiring cost paid to crowdsourced drivers and the shortage cost paid to third-party dedicated drivers.

Let $H(S_k, a_k)$ represent the hiring costs (salary paid to crowdsourced drivers) in decision epoch $k$ with state $S_k$ and action $a_k$. We have $H(S_k, a_k) = c_h a_k \tilde{q}$. Let $W(S_k, a_k)$ represent the minimum of the extra capacity that could be brought in by hiring the driver at epoch $k$ and the actual capacity shortage at the end of the horizon. Let $L(S_k, a_k)$ represent the estimated shortage cost induced by action $a_k$ taken at state $S_k$. We have $L(S_k, a_k) = c_p W(S_k, a_k)$. we develop policies that balance the hiring cost $H(S_k, a_k)$ and the shortage cost $L(S_k, a_k)$ that are associated with action $a_k$ at state $S_k$. Specifically, we consider a cost function approximation

$$Z(S_k, a_k) = \left| H(S_k, a_k) - L(S_k, a_k) \right|, \tag{1}$$

with which we develop the CFA policy. Let $\pi_b$ represent the CFA policy. Based on the above discussions, the CFA policy is obtained by solving the following equation at each decision epoch

$$\delta_k^{\pi_b}(S_k) = \arg \min_{a_k \in \{0,1\}} \mathbb{E}\left[ Z(S_k, a_k) \middle| S_k \right], \tag{2}$$

While the CFA policy was developed to balance the driver hiring cost and the capacity shortage cost, we name the CFA policy as the cost-balancing policy in the following discussion. We can show that with Assumption 1, the cost-balancing policy is an optimal policy. We also investigate the conditions under which Assumption 1 is satisfied.

**Assumption 1 (Shortage Cost)** *The total expected delivery shortage cost is equal to the expected cost paid to hire the third-party dedicated drivers at the end of the horizon if following the cost-balancing policy or the best policy, i.e.,*

$$\mathbb{E}\left[\sum_{k=0}^{K} L\left(S_k, \delta_k^{\pi}(S_k)\right) \middle| S_0\right] = \mathbb{E}\left[c_p|\tilde{\Theta}_K^{\pi}| \middle| S_0\right], \forall \pi \in \{\pi_b, \pi_o\}, \tag{3}$$

*where $\tilde{\Theta}_K^{\pi}$ is the set of customer orders that have not been scheduled for delivery by the end of the time horizon if following policy $\pi \in \{\pi_b, \pi_o\}$.*

**Theorem 1 (Optimality)** *The cost-balancing policy $\pi_b$ is the best policy in set $\Pi$.*

**Theorem 2 (Sufficiency)** *Under the following conditions, Assumption 1 is satisfied: There exists a decision epoch $k^* \in [0, K]$ such that*

$$\mathbb{E}\big[c_p W(S_i, a_i)|S_0\big] = 0, \forall i \in [0, k^*-1], a_i \in \{0, 1\}, S_i \in \mathcal{S}, \tag{4}$$

$$\mathbb{E}\big[c_p W(S_{k^*}, a_{k^*} = 0)|S_0\big] < \mathbb{E}\left[c_h \tilde{q}_{k^*} \middle| S_0\right], \tag{5}$$

$$\mathbb{E}\left[|\Theta_{k^*}| + \sum_{k'=k^*+1}^{K} |\hat{\Theta}_{k'}| \middle| S_0\right] \leq \mathbb{E}\left[\sum_{k'=k^*}^{K} R\big(S_{k'}, \delta_{k'}^{\pi_{\mathcal{H}}}(S_{k'})\big) \middle| S_0\right], \tag{6}$$

*and*

$$\mathbb{E}\big[c_p W(S_j, a_j = 0)|S_0\big] > \mathbb{E}\left[c_h \tilde{q}_j \middle| S_0\right], \forall j \in [k^*+1, K], S_j \in \mathcal{S}. \tag{7}$$

*where $\tilde{q}_{k^*}$ and $\tilde{q}_j$ are the available working duration from the new crowdsourced drivers arriving at decision epoch $k^*$ and $j$, respectively.*

## 3 Preliminary Results

We consider four types of policies. The first is the CFA or cost-balancing policy. The second is the *chase demand* policy such that at each decision epoch the crowdsourced driver is hired only if the current hired crowdsourced drivers cannot serve all the orders received. The third policy is the CFA-based one-step rollout policy, in which we make decisions by estimating the cost-to-go obtained by following policy $\pi_b$ from the next decision epoch till the end of the horizon. The fourth policy is the *chase-demand* based one-step rollout

| | CFA | One-Step CFA | Chase-Demand | One-Step Chase-Demand |
|---|---|---|---|---|
| Objective | 126.41 | 126.41 | 185.00 | 138.87 |
| Gap on objective | - | 0% | 46.3% | 9.9% |
| Hired crowdsourced drivers | 2 | 2 | 2.93 | 2.11 |
| Costs for hiring crowdsourced driver | 125.15 | 125.15 | 184.32 | 131.4 |
| Orders served by third party drivers | 0.08 | 0.08 | 0 | 0.50 |

Table 1: Small Scale Results with Assumption 1 satisfied

| | CFA | One-Step CFA | Chase-Demand | One-Step Chase-Demand |
|---|---|---|---|---|
| Objective | 482.46 | 409.07 | 572.39 | 428.81 |
| Gap on objective | 17.9% | - | 39.9% | 4.8% |
| Hired crowdsourced drivers | 6.41 | 7.08 | 9.05 | 6.89 |
| Costs for hiring crowdsourced driver | 401.78 | 399.62 | 570.83 | 409.64 |
| Orders served by third party drivers | 5.38 | 0.63 | 0.10 | 1.28 |

Table 2: Large Scale Results with Assumption 1 unsatisfied

policy. In this policy, we make decisions by estimating the cost-to-go obtained by following the *chase-demand* policy from the next decision epoch till the end of the horizon.

In Table 1 and 2, we present preliminary results for two instances. One is a small instance with 5 drivers and 84 orders on average, while the other is a bigger instance with 20 crowdsourced drivers and 262 orders on average. The results presented in the two tables are the average values over 1000 samples. We note that for the small instance, the condition in Assumption 1 is satisfied, while for the bigger instance, the condition is not satisfied. As shown in Table 1, when the condition specified in Assumption 1 is satisfied, the objectives of the policies obtained via the CFA and the CFA-based one-step rollout are the same and both policies are the optimal policy (they obtain the same objective as the optimal policy). The *chase-demand* and *chase-demand* based one-step rollout policies perform worse than the CFA and CFA-based one-step rollout policies. When the condition in Assumption 1 is not satisfied in the bigger instance, the CFA-based one-step rollout policy is the best, followed by the *chase-demand* based one-step rollout, the CFA policy and the *chase-demand* policy. The preliminary results demonstrate the effectiveness of our solution approach. As the next step, we will conduct more experiments with larger problem instances.

# A hybrid genetic algorithm for the inventory routing problem

**Jingyi Zhao**
Department of Mathematics and Industrial Engineering,
Polytechnique Montreal, Canada

**Tuan Anh Pham**
Anvita.,JSC, Vietnam

**Thibaut Vidal**
CIRRELT & SCALE-AI Chair in Data-Driven Supply Chains,
Department of Mathematical and Industrial Engineering,
Polytechnique Montreal, Canada Email: thibaut.vidal@polymtl.ca

## 1 Introduction

In this study, we address the Multi-Vehicle Inventory Routing Problem (MIRP), focusing on optimizing both inventory management and routing decisions for distributing products from a supplier to retailers. This involves integrated decisions on (i) routing, (ii) delivery days and quantities. Despite the importance of Large Neighborhood Search (LNS) in advancing solutions for vehicle routing problems, its exploration in MIRP has been limited. Traditional neighborhood strategies often become inefficient when applied to MIRP, indicating a significant gap in optimization techniques and the need for innovative solutions. Thus, we introduce a novel LNS operator to address this complex problem. This operator is specifically designed for MIRP, relying on efficient pre-processing and dynamic programming routines, integrated into a hybrid genetic search, producing high-quality solutions previously unattainable. Our method significantly enhances the heuristic's ability to navigate and assess large neighborhoods efficiently, leading to effective solutions. Moreover, it is able to address MIRP allowing out-of-stocks (with a penalty in objective). This problem is solved for the first time and we provide benchmark results with different penalty settings. Extensive experiments on MIRP confirm the superiority of our approach, demonstrating a remarkable number of new benchmark solutions across diverse datasets, marking a notable advancement in the field.

## 2 Solution Approach

Some solution methods exist for the IRP, but they typically do not allow stock-out penalties, beyond this significant methodological improvements may still permit better solutions for this problem. Thus, we explore the MIRP using the Hybrid Genetic Search with Adaptive Diversity Control (HGSADC) algorithm [2], which is proposed for periodic vehicle routing problems and successful at solving a wide range of other problems[3, 4]. HGSADC, like the classical genetic algorithm [1], evolves a population of solutions using selection and crossover followed by a local search on the solutions generated by the crossover to improve their objective values as well as to restore their feasibility. In our study, we propose a novel large neighborhood search operator (PI) that can be used in HGSADC to solve the MIRPs. Our PI operator is an iterative procedure and works as fol-

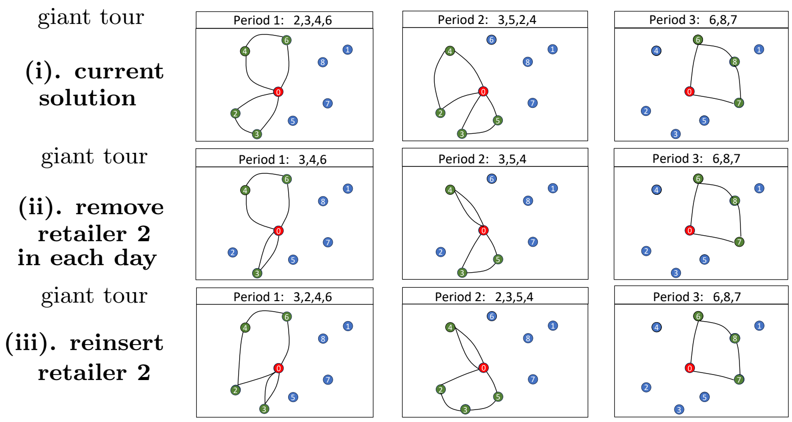



Figure 1: In this example of the PI operator, where customer 2 is selected from a scenario with 8 customers, 3 periods, and 2 vehicles, the process is illustrated in three steps: (i). the first line displays the current solution; (ii). the second line removes all trips that include customer 2; and (iii). the third line reintroduces the improved result that was identified.

lows: We randomly sort all retailers and replan the number of deliveries for each customer in turn within $T$ days and insert them into existing routes. Meanwhile, the decisions of the other retailers remain unchanged. After re-optimizing the decisions, the current solution will be updated and we move to the next customer. This process continues until no retailer can improve further.

Once a specific retailer $i$ is selected, all its scheduled visits are removed from the current solution. Considering the number of deliveries $q_i^t$ allocated to retailer $i$ during period $t$. When incorporating customer $i$ into daily routes with multiple vehicles, we first determine the optimal insertion point in each route to minimize the detour and keep track of the remaining load (freeload) on that route. Routes are then arranged by increasing detour length and the routes that are not dominant in terms of both freeload and detour cost will be discarded. Subsequently, we introduce two preprocessing piecewise-linear functions, for each day $t$ that relate carried quantities $q_i^t$ to their costs: $F_1(q_i^t)$ for sufficient replenishment and $F_2(q_i^t)$ for insufficient inventory. The calculation encompasses the detours, the stock-out costs for [daily demand minus $q_i^t$] goods as calculated in $F_1(q_i^t)$, the holding costs for [$q_i^t$ minus daily demand] goods as calculated in $F_2(q_i^t)$, and penalties for exceeding vehicle capacity, which are charged at $\omega$ per unit. For clarity, refer to Figure 2 illustrating these two functions.

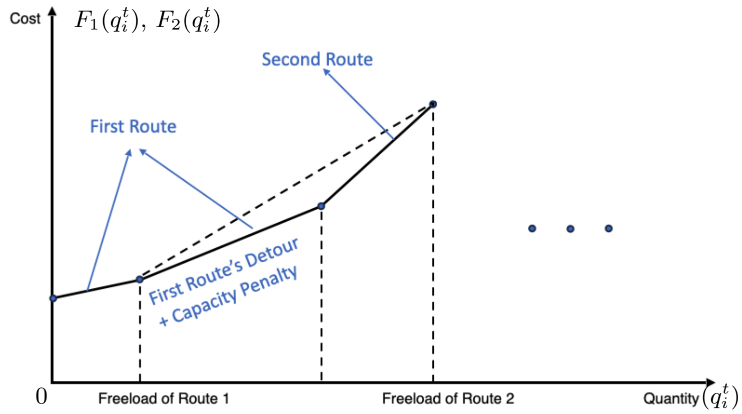



Figure 2: Illustration of the Cost Functions' Profiles, where freeload refers to the residual capacities of the routes.

Following this, re-planning the replenishment strategy for $i$ can be solved as a lot-sizing problem, with the goal of minimizing the total cost over the entire time horizon. We introduce a Dynamic Programming (DP) recursive approach to solve this specific type of lot sizing problem efficiently and achieve the minimum target value $C_t(I_i^t)$. This involves considering various inventory levels $I_i^t$ at the conclusion of each day, transitioning from $C_{t-1}(I_i^{t-1})$ to $C_t(I_i^t)$. These transitions within DP encompass four scenarios: (1) Inventory sufficiency without delivery, i.e., $q_i^t = 0$, which indicates that there is sufficient remaining inventory from the previous day to satisfy the retailer $i$'s demand during $t$; (2) Inventory insufficiency without delivery, i.e., $q_i^t = 0$ but there is insufficient remaining inventory from the previous day to satisfy the retailer $i$'s demand during $t$; (3) Inventory insufficiency for delivery involving $q_i^t \neq 0$, i.e., the sum of the remaining inventory and the quantity delivered is insufficient to satisfy the retailer $i$'s demand during the $t$ period; and (4) Deliveries are made in the quantities required, i.e., $q_t^i \neq 0$, where the sum of the remaining inventory and the quantity delivered is insufficient to satisfy the retailer i's demand during the *t-th* period. It is worth noting that all transfer computations are handled very efficiently, e.g., the storing functions operate in the dimension of the linear piece. Therefore, $C_t(I_i^t)$ can be updated to the minimum value among these four scenarios. Finally, an optimal reinsertion plan for $i$ is determined when $t$ reaches the total time horizon and we update the current best solution.

## 3 Numerical experiments

We evaluated our algorithm by adjusting the out-of-stock penalty to 1,000,000, thus addressing the classical MIRP problem. Our approach's efficacy, particularly the HGSADC's performance, was benchmarked against notable heuristics in the field. In the analysis of small-scale instances, we observed notable results: for the 3-day data, comprising 360 instances, our method successfully identified 98 new best solutions, as shown in Table 3. This demonstrates the efficiency and effectiveness of our method, particularly in dealing

Table 1: The benchmarks can be found on the OR-brescia website at `https://or-brescia.unibs.it/instances`, which hosts an updated compilation of IRP benchmark results from literature. These include the ABS heuristic by [5], the CCJ approach by [8], along with the ILS and SA methods described by [7], and the KS heuristic as outlined by [9]. Comprehensive computational results from these comparisons are accessible online. The "Gap* " values are defined as $= (\frac{\text{OurSolution}}{\text{BEST*}} - 1) \times 100\%$.

| | | | *ABS* | | *CCJ* | | *ILS* | | *SA* | | *KS* | | HGSADC | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *CO* | *N* | *BSHeu* | *sABS* | tABS | *sCCJ* | tCCJ | *sILS* | tILS | *sSA* | tSA | *sKS* | tKS | *sHGS* | *Gap* | *tHGS* | *BST* | *Avg* | *Gap* |
| | 10 | 3053.13 | 3053.13 | *369.70* | 3221.39 | *4.16* | 3091.48 | *150.00* | 3079.21 | *104.97* | 3053.18 | *89.25* | 3053.13 | 0.0 | *37.0* | 5.8 | 3053.13 | 0.0 |
| | 15 | 3118.34 | 3120.69 | *871.80* | 3200.77 | *10.33* | 3180.22 | *150.00* | 3131.15 | *116.05* | 3121.36 | *118.45* | **3117.85** | **-0.0** | *47.2* | 6.8 | 3119.88 | 0.1 |
| | 20 | 3682.93 | 3718.31 | *1499.10* | 3771.04 | *20.50* | 3766.86 | *150.00* | 3694.52 | *127.92* | 3702.66 | *225.80* | 3687.48 | 0.09 | *60.4* | 14.2 | 3694.77 | 0.38 |
| | 25 | 3939.81 | 3970.44 | *1685* | 3994.91 | *60.58* | 4069.50 | *150.00* | 3953.31 | *133.76* | 3951.99 | *337.65* | **3936.56** | **-0.08** | *65.3* | 16.0 | 3951.72 | 0.33 |
| LC | 30 | 3952.87 | 4041.60 | *2110.40* | 3989.68 | *44.14* | 4034.72 | *150.01* | 3972.08 | *144.64* | 3978.59 | *482.05* | **3947.63** | **-0.11** | *94.4* | 28.3 | 3969.83 | 0.52 |
| | 35 | 4203.43 | 4303.03 | *2492.70* | 4237.04 | *58.07* | 4282.66 | *150.01* | 4216.67 | *150.01* | 4224.87 | *556.90* | **4197.62** | **-0.1** | *110.7* | 40.9 | 4203.77 | 0.01 |
| | 40 | 4339.84 | 4404.74 | *3235.60* | 4397.24 | *94.44* | 4422.83 | *150.01* | 4386.57 | *150.04* | 4377.20 | *619.85* | **4315.39** | **-0.52** | *130.9* | 58.3 | 4328.69 | **-0.22** |
| | 45 | 4480.24 | 4584.04 | *3756.90* | 4517.73 | *107.36* | 4652.46 | *150.01* | 4609.12 | *150.05* | 4576.23 | *698.30* | **4471.97** | **-0.17** | *143.4* | 60.2 | 4501.42 | 0.43 |
| | 50 | 5248.82 | 5358.77 | *5288.70* | 5296.47 | *152.66* | 5341.09 | *150.02* | 5316.94 | *150.06* | 5445.02 | *1098.15* | **5232.80** | **-0.30** | *207.2* | 118.2 | 5268.34 | 0.43 |
| | 10 | 4856.47 | 4857.22 | *415.90* | 5013.81 | *4.32* | 4895.47 | *150.00* | 4876.67 | *107.76* | 4856.89 | *91.30* | 4856.50 | 0.0 | *38.2* | 7.3 | 4856.73 | 0.00 |
| | 15 | 5532.56 | 5534.34 | *803.50* | 5620.86 | *11.29* | 5553.87 | *150.00* | 5540.95 | *116.30* | 5534.01 | *114.90* | **5530.878** | **-0.03** | *54.3* | 15.6 | 5531.11 | **-0.02** |
| | 20 | 7089.31 | 7132.28 | *1319.70* | 7191.44 | *25.30* | 7115.82 | *150.00* | 7102.96 | *127.99* | 7105.34 | *202.60* | 7137.27 | 0.04 | *57.1* | **10.2** | 7140.22 | 0.09 |
| | 25 | 8430.10 | 8455.88 | *1506.20* | 8501.63 | *37.56* | 8469.70 | *150.01* | 8435.09 | *135.55* | 8438.19 | *324.30* | **8429.69** | **-0.01** | *79.0* | **29.8** | 8438.88 | 0.09 |
| HC | 30 | 9673.44 | 9743.89 | *1962.00* | 9699.83 | *64.09* | 9705.07 | *150.00* | 9683.37 | *145.56* | 9721.92 | *504.80* | **9660.65** | **-0.13** | *106.4* | **47.0** | 9663.74 | **-0.10** |
| | 35 | 10174.68 | 10285.07 | *2605.10* | 10210.21 | *85.21* | 10240.00 | *150.01* | 10187.86 | *150.04* | 10205.92 | *544.20* | **10170.79** | **-0.04** | *128.1* | **66.1** | 10177.53 | 0.02 |
| | 40 | 10960.14 | 11032.03 | *3232.70* | 10986.12 | *141.90* | 11024.37 | *150.01* | 10995.18 | *150.04* | 11012.14 | *614.70* | **10936.74** | **-0.21** | *141.5* | **76.9** | 10975.26 | 0.13 |
| | 45 | 11972.65 | 12063.12 | *3699.40* | 11998.55 | *179.86* | 12092.75 | *150.01* | 12080.79 | *150.04* | 12077.40 | *663.10* | **11966.4** | **-0.05** | *159.3* | **95.0** | 11992.49 | 0.17 |
| | 50 | 13545.97 | 13676.21 | *5138.70* | 13571.12 | *232.10* | 13624.86 | *150.02* | 13631.41 | *150.06* | 13692.80 | *1173.80* | **13526.2** | **-0.15** | *204.7* | **116.5** | 13554.73 | 0.06 |

with larger and more complex datasets.

# References


[1] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem", *Computers & Operations Research*, vol. 31, no. 12, pp. 1985–2002, 2004.

[2] T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei, "A hybrid genetic algorithm for multidepot and periodic vehicle routing problems", *Operations Research*, vol. 60, no. 3, pp. 611–624, 2012.

[3] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows", *Computers & Operations Research*, vol. 40, no. 1, pp. 475–489, 2013.

[4] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A unified solution framework for multi-attribute vehicle routing problems", *European Journal of Operational Research*, vol. 234, no. 3, pp. 658–673, 2014.

[5] Archetti, Claudia, Natashia Boland, and M. Grazia Speranza. "A matheuristic for the multivehicle inventory routing problem." INFORMS Journal on Computing 29.3 (2017): 377-387.

[6] Chitsaz, Masoud, Jean-François Cordeau, and Raf Jans. "A unified decomposition matheuristic for assembly, production, and inventory routing." INFORMS Journal on Computing 31.1 (2019): 134-152.

[7] Alvarez, Aldair, Pedro Munari, and Reinaldo Morabito. "Iterated local search and simulated annealing algorithms for the inventory routing problem." International Transactions in Operational Research 25.6 (2018): 1785-1809.

[8] Archetti, Claudia, et al. "A kernel search heuristic for the multivehicle inventory routing problem." International Transactions in Operational Research 28.6 (2021): 2984-3013.

[9] Archetti, Claudia, et al. "A kernel search heuristic for the multivehicle inventory routing problem." International Transactions in Operational Research 28.6 (2021): 2984-3013.