Aalto University
School of Electrical Engineering

Classification of weighing matrices

Pekka Lampio
Joint work with Mikhail Ganzhinov

8th Hadamard Workshop
May 26-30, 2025 Sevilla

Weighing matrix

Definition

A weighing matrix of order n and weight k, denoted by $W(n, k)$ is an $n \times n$ matrix with entries $0, 1, -1$, such that,

$$\mathbf{W}\mathbf{W}^T = k\,\mathbf{I}_n \quad ,$$

where $1 \leq k \leq n$.

Weighing matrix

Definition

A weighing matrix of order n and weight k, denoted by $W(n, k)$ is an $n \times n$ matrix with entries $0, 1, -1$, such that,

$$\mathbf{W}\mathbf{W}^T = k\,\mathbf{I}_n \quad ,$$

where $1 \leq k \leq n$.

Example: a $W(8, 4)$

0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	-1
0	1	0	1	0	1	-1	0
0	1	-1	0	-1	0	1	0
1	0	0	1	0	-1	0	1
1	0	-1	0	1	0	0	-1
1	1	1	-1	0	0	0	0
1	-1	0	0	-1	1	0	0

Weighing matrix

Definition

A weighing matrix of order n and weight k, denoted by $W(n,k)$ is an $n \times n$ matrix with entries $0, 1, -1$, such that,

$$\mathbf{W}\mathbf{W}^T = k\,\mathbf{I}_n \quad ,$$

where $1 \leq k \leq n$.

Example: a $W(8,4)$

0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	-1
0	1	0	1	0	1	-1	0
0	1	-1	0	-1	0	1	0
1	0	0	1	0	-1	0	1
1	0	-1	0	1	0	0	-1
1	1	1	-1	0	0	0	0
1	-1	0	0	-1	1	0	0

- rows (columns) are mutually orthogonal

Weighing matrix

Definition

A weighing matrix of order n and weight k, denoted by $W(n, k)$ is an $n \times n$ matrix with entries $0, 1, -1$, such that,

$$\mathbf{W}\mathbf{W}^T = k\,\mathbf{I}_n \quad ,$$

where $1 \le k \le n$.

Example: a $W(8, 4)$

0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	-1
0	1	0	1	0	1	-1	0
0	1	-1	0	-1	0	1	0
1	0	0	1	0	-1	0	1
1	0	-1	0	1	0	0	-1
1	1	1	-1	0	0	0	0
1	-1	0	0	-1	1	0	0

- rows (columns) are mutually orthogonal
- when $k = n$, the matrix is a Hadamard matrix.

Weighing matrix

Definition

A weighing matrix of order n and weight k, denoted by $W(n, k)$ is an $n \times n$ matrix with entries $0, 1, -1$, such that,

$$\mathbf{W}\mathbf{W}^T = k\,\mathbf{I}_n \quad ,$$

where $1 \leq k \leq n$.

Example: a $W(8, 4)$

$$\begin{matrix}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & -1 \\
0 & 1 & 0 & 1 & 0 & 1 & -1 & 0 \\
0 & 1 & -1 & 0 & -1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & -1 & 0 & 1 \\
1 & 0 & -1 & 0 & 1 & 0 & 0 & -1 \\
1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & -1 & 1 & 0 & 0
\end{matrix}$$

- rows (columns) are mutually orthogonal
- when $k = n$, the matrix is a Hadamard matrix.
- when $k = n - 1$, the matrix is equivalent to a conference matrix.

Equivalence of weighing matrices

These operations produce a weighing matrix when applied to any weighing matrix:

1. Permuting the order of rows,
2. Permuting the order of columns,
3. Multiplying a row by -1,
4. Multiplying a column -1.

Equivalence of weighing matrices

These operations produce a weighing matrix when applied to any weighing matrix:

1. Permuting the order of rows,
2. Permuting the order of columns,
3. Multiplying a row by -1,
4. Multiplying a column -1.

Weighing matrices are equivalent if they are essentially the same in the following sense:

Definition

Weighing matrices A and B are equivalent, denoted by, $A \cong B$, if B can be generated from A by applying Operations 1, 2, 3, and 4.

Classification of weighing matrices

Classification of weighing matrices

1. Existence
   - Known for weights $<=5$.
   - Smallest unknown case is $W(35, 25)$.

Classification of weighing matrices

1. Existence
   - Known for weights $<= 5$.
   - Smallest unknown case is $W(35, 25)$.
2. Enumeration of all inequivalent matrices
   - Smallest order with unknown cases is 16.
   - In this work we give complete classification of orders 16,18,19,20 and 21.
   - Also we give complete classification of some cases in orders 22,24,28.

Classification of weighing matrices

1. Existence
   - Known for weights $<= 5$.
   - Smallest unknown case is $W(35, 25)$.
2. Enumeration of all inequivalent matrices
   - Smallest order with unknown cases is 16.
   - In this work we give complete classification of orders 16,18,19,20 and 21.
   - Also we give complete classification of some cases in orders 22,24,28.
3. Exhaustive computer search
   - A new computational method: the criss-cross search.

Computational methods

1. Orderly generation of matrices

Computational methods

1. Orderly generation of matrices
2. Larger matrices are constructed from smaller by adding rows (or columns).

Computational methods

1. Orderly generation of matrices
2. Larger matrices are constructed from smaller by adding rows (or columns).
3. Pruning of isomorphic matrices and substructures

Computational methods

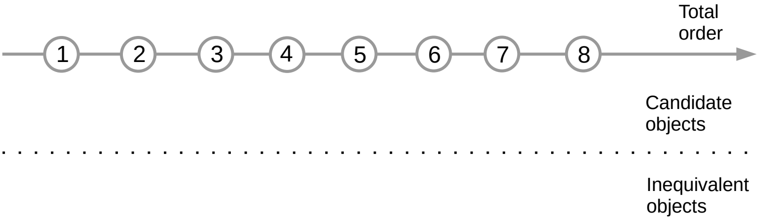
1. Orderly generation of matrices
2. Larger matrices are constructed from smaller by adding rows (or columns).
3. Pruning of isomorphic matrices and substructures
4. Criss-cross search (NEW!)

Computational methods

1. Orderly generation of matrices
2. Larger matrices are constructed from smaller by adding rows (or columns).
3. Pruning of isomorphic matrices and substructures
4. Criss-cross search (NEW!)
5. Cliquer search

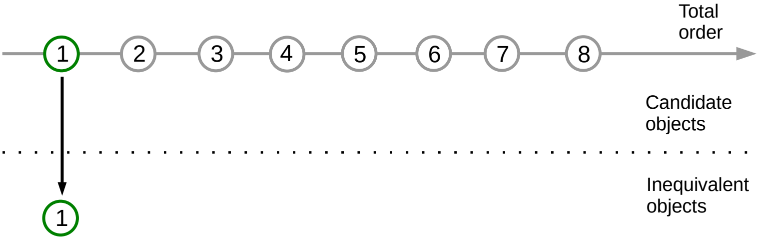
Orderly generation

- Objects are generated in some predefined total order.
- Example:
  - colors depict equivalence classes



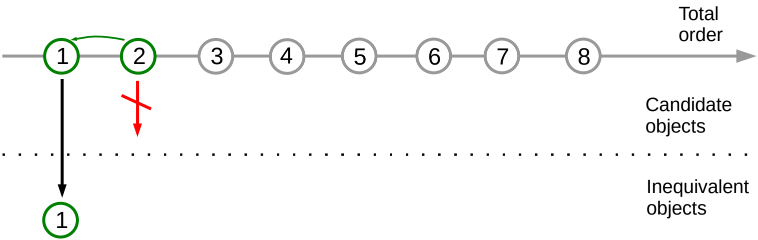
Orderly generation

- Objects are generated in some predefined total order.
- Example:
  - colors depict equivalence classes



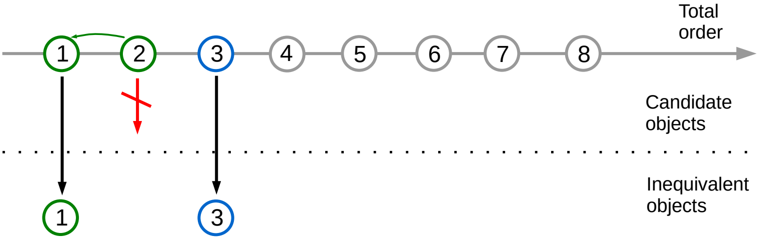
Orderly generation

- Objects are generated in some predefined total order.
- Example:
  - colors depict equivalence classes



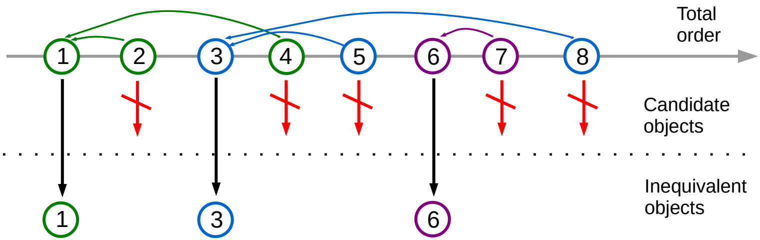
Orderly generation

- Objects are generated in some predefined total order.
- Example:
  - colors depict equivalence classes



Orderly generation

- Objects are generated in some predefined total order.
- Example:
  - colors depict equivalence classes



Computational methods: basic idea

Append rows (or columns) to a rectangular weighing matrix until it becomes a square matrix. Example: $W(8,4)$:

$$\begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & -1 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \end{matrix} \quad \in \mathrm{R2}$$

Computational methods: basic idea

Append rows (or columns) to a rectangular weighing matrix until it becomes a square matrix. Example: $W(8,4)$:

$$\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 1 & 0 & 1 & -1 & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \end{array} \quad \in \mathrm{R3}$$

Computational methods: basic idea

Append rows (or columns) to a rectangular weighing matrix until it becomes a square matrix. Example: $W(8,4)$:

$$\begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 1 & 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 1 & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \end{matrix} \quad \in \text{R4}$$

Computational methods: basic idea

Append rows (or columns) to a rectangular weighing matrix until it becomes a square matrix. Example: $W(8,4)$:

$$\begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 1 & 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 & 1 & 0 & 0 & -1 \\ 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \end{matrix} \quad \in \mathrm{R8}$$

Computational methods: basic idea

Append rows (or columns) to a rectangular weighing matrix until it becomes a square matrix. Example: $W(8,4)$:

$$
\begin{matrix}
0 & 0 & . & . & . & . & . & . \\
0 & 0 & . & . & . & . & . & . \\
0 & 1 & . & . & . & . & . & . \\
0 & 1 & . & . & . & . & . & . \\
1 & 0 & . & . & . & . & . & . \\
1 & 0 & . & . & . & . & . & . \\
1 & 1 & . & . & . & . & . & . \\
1 & -1 & . & . & . & . & . & .
\end{matrix}
\quad \in \mathrm{C2}
$$

Computational methods: basic idea

Append rows (or columns) to a rectangular weighing matrix until it becomes a square matrix. Example: $W(8,4)$:

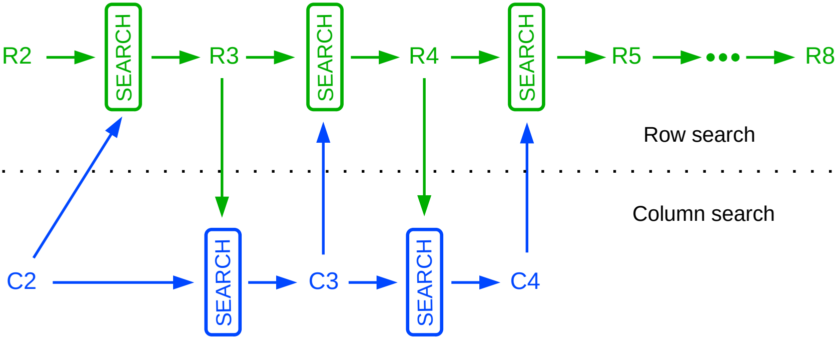
$$\begin{array}{cccccccc} 0 & 0 & 0 & . & . & . & . & . \\ 0 & 0 & 1 & . & . & . & . & . \\ 0 & 1 & 0 & . & . & . & . & . \\ 0 & 1 & -1 & . & . & . & . & . \\ 1 & 0 & 0 & . & . & . & . & . \\ 1 & 0 & -1 & . & . & . & . & . \\ 1 & 1 & 1 & . & . & . & . & . \\ 1 & -1 & 0 & . & . & . & . & . \end{array} \quad \in \mathrm{C3}$$

Computational methods: basic idea

Append rows (or columns) to a rectangular weighing matrix until it becomes a square matrix. Example: $W(8, 4)$:

$$\begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 1 & 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 & 1 & 0 & 0 & -1 \\ 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \end{matrix} \qquad \in \mathrm{C8}$$

Criss-cross search



Criss-cross search (cont.)

Let C be an $n \times k$ and R $m \times n$ rectangular weighing matrices:

C/R	C/R	R	R	R	R	R	R
C/R	C/R	R	R	R	R	R	R
C/R	C/R	R	R	R	R	R	R
C	C	.	.	.	.	.	.
C	C	.	.	.	.	.	.
C	C	.	.	.	.	.	.
C	C	.	.	.	.	.	.
C	C	.	.	.	.	.	.

C and R are matching row and column search matrices.

Criss-cross search (cont.)

Let C be an $n \times k$ and R $m \times n$ rectangular weighing matrices:

C/R	C/R	R	R	R	R	R	R
C/R	C/R	R	R	R	R	R	R
C/R	C/R	R	R	R	R	R	R
C	C	.	.	.	.	.	.
C	C	.	.	.	.	.	.
C	C	.	.	.	.	.	.
C	C	.	.	.	.	.	.
C	C	.	.	.	.	.	.

C and R are matching row and column search matrices.

An equivalence defining operation f (row/column permutation, row/column multiplication) is compatible with C if

$$f(R) \geq R \quad , \text{for all } R.$$

Computational methods: clique search

Check if a $k \times n$ rectangular matrix $\mathbf{A}$ could be extended to a full weighing matrix $\mathbf{W}$ of order n.

Computational methods: clique search

Check if a $k \times n$ rectangular matrix **A** could be extended to a full weighing matrix **W** of order n.

- Find set C of all those rows that are orthogonal with all rows of A.

Computational methods: clique search

Check if a $k \times n$ rectangular matrix **A** could be extended to a full weighing matrix **W** of order n.

- Find set C of all those rows that are orthogonal with all rows of A.
- Form a graph G where each row in C is a node and there is an edge between the rows if they are orthogonal.

Computational methods: clique search

Check if a $k \times n$ rectangular matrix $\mathbf{A}$ could be extended to a full weighing matrix $\mathbf{W}$ of order n.

- Find set C of all those rows that are orthogonal with all rows of A.
- Form a graph G where each row in C is a node and there is an edge between the rows if they are orthogonal.
- Solve maximum clique problem in G with Cliquer software

Computational methods: clique search

Check if a $k \times n$ rectangular matrix **A** could be extended to a full weighing matrix **W** of order n.

- Find set C of all those rows that are orthogonal with all rows of A.
- Form a graph G where each row in C is a node and there is an edge between the rows if they are orthogonal.
- Solve maximum clique problem in G with Cliquer software (Niskanen, Östergård)
- If the size of the maximum clique is $n - k$, we have found a weighing matrix.

Computational methods: example

How we computed $W(20, 12)$.

1. Created seeds with Criss-cross search.

Computational methods: example

How we computed $W(20, 12)$.

1. Created seeds with Criss-cross search.
   - seeds are 5×20 rectangular weighing matrices
   - single-threaded computation
   - CPU-time 4.36 hours
   - Real-time 4.38 hours
   - Found $4,055,445$ seeds

Computational methods: example

How we computed $W(20, 12)$.

1. Created seeds with Criss-cross search.
   - seeds are 5×20 rectangular weighing matrices
   - single-threaded computation
   - CPU-time 4.36 hours
   - Real-time 4.38 hours
   - Found $4, 055, 445$ seeds
2. Extended seeds to full matrices with Clique search.

Computational methods: example

How we computed $W(20, 12)$.

1. Created seeds with Criss-cross search.
   - seeds are 5×20 rectangular weighing matrices
   - single-threaded computation
   - CPU-time 4.36 hours
   - Real-time 4.38 hours
   - Found $4, 055, 445$ seeds
2. Extended seeds to full matrices with Clique search.
   - Parallel computation on a computing cluster using 640 cores
   - CPU-time 38.3 years
   - Real-time 23.4 days
   - $3, 503, 212$ inequivalent matrices found

Results

Order	Weight	#	Ref.
16	1,2,3	1	[1]
	4	10	[1]
	5	4	[1]
	6	30	[3]
	7	55	
	8	4631	
	9	704	[3]
	10	743	670 in [3]
	11	43	
	12	279	[3]
	13	14	
	14	17	
	15	1	[1]
	16	5	[2]

Results

Order	Weight	#	Ref.
16	1,2,3	1	[1]
	4	10	[1]
	5	4	[1]
	6	30	[3]
	7	55	
	8	4631	
	9	704	[3]
	10	743	670 in [3]
	11	43	
	12	279	[3]
	13	14	
	14	17	
	15	1	[1]
	16	5	[2]

Order	Weight	#	Ref.
17	1	1	[1]
	4	2	[1]
	9	2360	[3]

Results

Order	Weight	#	Ref.
16	1,2,3	1	[1]
	4	10	[1]
	5	4	[1]
	6	30	[3]
	7	55	
	8	4631	
	9	704	[3]
	10	743	670 in [3]
	11	43	
	12	279	[3]
	13	14	
	14	17	
	15	1	[1]
	16	5	[2]

Order	Weight	#	Ref.
17	1	1	[1]
	4	2	[1]
	9	2360	[3]
18	1	1	[1]
	2	1	[1]
	4	11	[1]
	5	5	[3]
	8	4817	
	9	11891	[3]
	10	70133	
	13	53	
	16	4	
	17	1	[1]

Results

Order	Weight	#	Ref.
16	1,2,3	1	[1]
	4	10	[1]
	5	4	[1]
	6	30	[3]
	7	55	
	8	4631	
	9	704	[3]
	10	743	670 in [3]
	11	43	
	12	279	[3]
	13	14	
	14	17	
	15	1	[1]
	16	5	[2]

Order	Weight	#	Ref.
17	1	1	[1]
	4	2	[1]
	9	2360	[3]
18	1	1	[1]
	2	1	[1]
	4	11	[1]
	5	5	[3]
	8	4817	
	9	11891	[3]
	10	70133	
	13	53	
	16	4	
	17	1	[1]
19	1	1	[1]
	4	5	[1]
	9	189076	

Order	Weight	#	Ref.
20	1,2,3	1	[1]
	4	18	[1]
	5	7	[3]
	6	49	[3]
	7	159	
	8	18294	
	9	2924696	
	10	73418583	
	11	13506863	
	12	3503212	
	13	34312	
	14	6254	
	15	1351	
	16	2164	
	17	58	
	18	53	[3]
	19	2	[2]
	20	3	[2]

Order	Weight	#	Ref.
20	1,2,3	1	[1]
	4	18	[1]
	5	7	[3]
	6	49	[3]
	7	159	
	8	18294	
	9	2924696	
	10	73418583	
	11	13506863	
	12	3503212	
	13	34312	
	14	6254	
	15	1351	
	16	2164	
	17	58	
	18	53	[3]
	19	2	[2]
	20	3	[2]

Order	Weight	#	Ref.
21	1	1	[1]
	4	1	[1]
	9	5756272	
	16	40	

Order	Weight	#	Ref.
22	1,2	1	[1]
	4	22	[1]
	5	9	[3]
	8	196986	
	9,10,13,16	?	
	17	1588	
	18	1621	
	20	6	

Order	Weight	#	Ref.
22	1,2	1	[1]
	4	22	[1]
	5	9	[3]
	8	196986	
	9,10,13,16	?	
	17	1588	
	18	1621	
	20	6	
23	1	1	[1]
	4	10	[1]
	9,16	?	

Order	Weight	#	Ref.
22	1,2	1	[1]
	4	22	[1]
	5	9	[3]
	8	196986	
	9,10,13,16	?	
	17	1588	
	18	1621	
	20	6	
23	1	1	[1]
	4	10	[1]
	9,16	?	

Order	Weight	#	Ref.
24	1,2,3	1	[1]
	4	33	[1]
	5	14	[3]
	6	190	[3]
	7	336	
	8-20	?	
	21	2493	
	22	1440	
	23	9	[5]
	24	60	[5]

Order	Weight	#	Ref.
22	1,2	1	[1]
	4	22	[1]
	5	9	[3]
	8	196986	
	9,10,13,16	?	
	17	1588	
	18	1621	
	20	6	
23	1	1	[1]
	4	10	[1]
	9,16	?	

Order	Weight	#	Ref.
24	1,2,3	1	[1]
	4	33	[1]
	5	14	[3]
	6	190	[3]
	7	336	
	8-20	?	
	21	2493	
	22	1440	
	23	9	[5]
	24	60	[5]
25	1	1	[1]
	4	11	[1]
	9,16	?	

Order	Weight	#	Ref.
26	1,2	1	[1]
	4	39	[1]
	5	16	[3]
	8,9,10,13,16,17,18,20	?	
	25	4	[4]

Order	Weight	#	Ref.
26	1,2	1	[1]
	4	39	[1]
	5	16	[3]
	8,9,10,13,16,17,18,20	?	
	25	4	[4]
27	1	1	[1]
	4	18	[1]
	9,16	?	

Order	Weight	#	Ref.
26	1,2	1	[1]
	4	39	[1]
	5	16	[3]
	8,9,10,13,16,17,18,20	?	
	25	4	[4]
27	1	1	[1]
	4	18	[1]
	9,16	?	
28	1,2,3	1	[1]
	4	57	[1]
	5	22	[3]
	6	684	
	7-26	?	
	27	41	[5]
	28	487	[5]

Summary

Summary

- We have classified 32 cases.

Summary

- We have classified 32 cases.
- These results are preliminary.

Summary

- We have classified 32 cases.
- These results are preliminary.
- Further performance improvements are possible.

Summary

- We have classified 32 cases.
- These results are preliminary.
- Further performance improvements are possible.
- The work continues.

Thank you!

References

[1] H. C. Chan, C. A. Rodger, and Jennifer Seberry. On inequivalent weighing matrices. volume 21, pages 299–333. 1986. Thirteenth Australasian conference on combinatorial mathematics and computing (Sydney, 1985).

[2] Marshall Hall, Jr. *Combinatorial theory*. Blaisdell Publishing Co. [Ginn and Co.], Waltham, Mass.-Toronto, Ont.-London, 1967.

[3] Masaaki Harada and Akihiro Munemasa. On the classification of weighing matrices and self-orthogonal codes. *J. Combin. Des.*, 20(1):40–57, 2012.

[4] A. J. L. Paulus. Conference matrices and graphs of order 26. Report WSK 73/06, Technische Hogeschool Eindhoven, Eindhoven, 1973.

[5] Edward Spence. Classification of hadamard matrices of order 24 and 28. *Discret. Math.*, 140(1-3):185–243, 1995.