Computing Automorphism Groups of Designs - a Way to construct New Symmetric Weighing Matrices

Assaf Goldberger[1] Yossi Strassler[2] Giora Dula[3]

[1]Tel-aviv University assafg@post.tau.ac.il

[2] Dan Yishay danyishay@gmail.com

[3] Netanya College giora@netanya.ac.il

February 17, 2018

content

- introduction.
- the code invariant ci, mci
- finding unsigned permutatuion
- finding a signed permutation
- the orbits of the automorphism group

introduction and examples

- A weighing matrix $W(n,k)$ is a $n \times n$ matrix W whose elements are $0, \pm 1$ such that $WW^t = kI_n$. $W(n,k)$ denotes both a single matrix and the class of all $W(n,k)$.
- The following are $W(2,k), 1 \le k \le 2$

$$\left(\begin{array}{c|cc} I_2 & 1 & 1 \\ & 1 & - \end{array}\right)$$

- The following are $W(3,k), 1 \le k \le 3$

$$\left(I_3 \;\middle|\; None \;\middle|\; None\right)$$

- The following are $W(4,k), 1 \le k \le 4$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & - & 0 & 0 & 1 & 0 & 1 & - & 1 & - & 1 & - \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & - & 0 & 1 & 1 & 1 & - & - \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & - & - & - & 1 & 0 & 1 & - & - & 1 \end{array}\right)$$

Questions

- Applications: Chemistry, Spectroscopy, Quantum Computing and Coding Theory.
- For which n and k $W(n,k) \neq \emptyset$ is an open question.
- Hadamard conjecture: $W(n,n) \neq \emptyset$ for every $n = 4k, k \in \mathbb{N}$.
- The main mathematical interest is to exhibit a concrete $W(n,k)$ or to prove that it does not exist.
- To date the smallest Hadamard matrix whose existence is unknown is $H(668)$.
- Given $W(n,k)$ it is a mathematical interest to find if an (anti)symmetric $W(n,k)$ exists.
- In this note we present a concrete symmetric $W(23,16)$ derived from $W(23,16)$ found recently.

Isomorphic (Hadamard equivalent) weighing matrices

- A monomial matrix (a signed permutation) P is a permutation matrix whose non zero elements are ± 1.
- Two matrices U, V are isomorphic (Hadamard equivalent) if there exists two monomial matrices P, Q with $PUQ^t = V$.
- The following exhibits an Hadamard equivalence between the Kronecker matrix $H2 \otimes H2$ and the circulant matrix $CH4$.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & - \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & - & 1 & - \\ 1 & 1 & - & - \\ 1 & - & - & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ - & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} - & 1 & 1 & 1 \\ 1 & - & 1 & 1 \\ 1 & 1 & - & 1 \\ 1 & 1 & 1 & - \end{bmatrix}$$

The code vector cv of a rectangular matrix *sm*

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

$$\text{Code} = (13, 38, 1, -30, -18, 23, -33, 17, -11, -6, -19, 34, -31)$$

.

- the code is the multiplication of the weight vector $(1, 3, 9, 27)$ with the matrix.
- Multiplying with a weight vector is a bijection $M_{4,n}(0, \pm 1) \longrightarrow [-\frac{3^4-1}{2}, \frac{3^4-1}{2}]^n$. Thus cv determines the matrix.
- There are $2^4 4! \times 2^{13} 13!$ matrices isomorphic to the original matrix and each has its own code vector.

The code invariant ci of a rectangular matrix

Stage 1: Normalize columns

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

$$\text{Code} = (13, -38, 1, 30, -18, -23, -33, -17, -11, -6, 19, 34, 31)$$

.

The code invariant ci of a rectangular matrix

Stage 2: Permute Columns

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

$$\text{Code} = (-38, -33, -23, -18, -17, -11, -6, 1, 13, 19, 30, 31, 34)$$

$$\text{Permutation} = (1, 2, 7, 10, 11, 4, 5, 8, 3, 6, 9)(12, 13)$$

Start Over

Start Over:

$$sm = \begin{bmatrix} 1 & -1 & 1 & 0 & 0 & -1 & 0 & -1 & 1 & 0 & -1 & 1 & -1 \\ 1 & 1 & 0 & -1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & 0 & 1 & 0 & -1 & -1 & -1 & -1 & 1 & 1 & 0 \\ 0 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & -1 & 1 & -1 \end{bmatrix}$$

Start Over

Permute Rows of SM (possibly with signs):

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

$$\text{Permutation} = (2, 3, 4)$$

Start Over

Repeat Stages 1–2: Normalize and Permute Columns

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

Code$=(-38, -35, -18, -15, -14, -6, -5, 1, 7, 25, 30, 31, 37)$.

This is (lexicographically) smaller, but not the smallest.

ci:=The Smallest Code cv for *SM*

- We permute the rows of *sm* by $(1, 3, 4, 2)$.
- Then we multiply rows 2 and 3 by -1.
- We apply stages 1-2 to the columns.

ci:=The Smallest Code cv for *SM*

- We permute the rows of *sm* by $(1,3,4,2)$.
- Then we multiply rows 2 and 3 by -1.
- We apply stages 1-2 to the columns.
- We obtain the matrix

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

with the minimal code vector

$$cv := (-38, -35, -33, -26, -18, -15, 3, 7, 10, 13, 22, 25, 31).$$

ci:=The Smallest Code cv for SM

- We permute the rows of sm by $(1,3,4,2)$.
- Then we multiply rows 2 and 3 by -1.
- We apply stages 1-2 to the columns.
- We obtain the matrix

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

  with the minimal code vector

$$cv := (-38,-35,-33,-26,-18,-15,3,7,10,13,22,25,31).$$

- ci smallest increasing cv over signed permutations on the rows.

The multiple code invariant mci of a square matrix

- Given a matrix $A \in M_n(0, \pm 1)$, one chooses $m \leq n$, e.g $m = 4, n = 13$ as before. There are $\binom{n}{m}$ submatrices obtained by choosing m rows out of n.
- Each such rectangular submatrix has its code invariant ci.
- The mci(A) is a multiset of $\binom{n}{m}$ ci.
- If mci(A)$\neq$mci(B) then A and B are not isomorphic.
- A similar procedure appears in Fang & Ge, with a different ci.
- In all the examples found recently of $W(n, 16)$ for $n = 25, 27,$ 29, $mci(W) \neq mci(W^t)$ and therefore were not isomorphic.

unsigned permutation

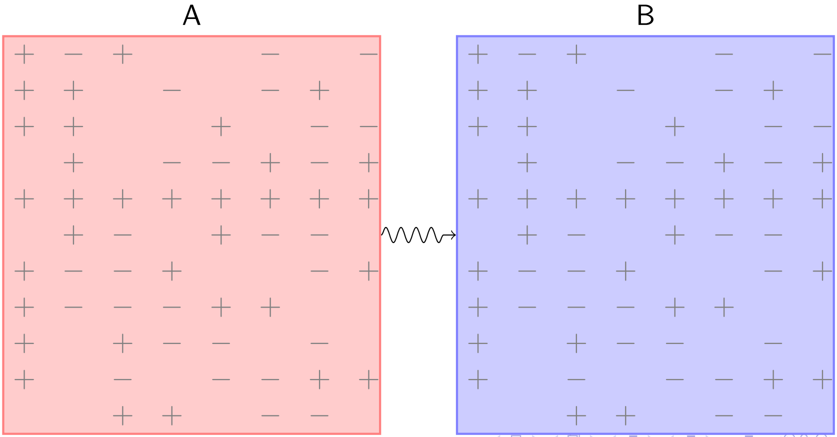
- Given A, B square matrices so that there exist unsigned permutations α, β such that $\beta A\alpha = B$ it follows that $\beta AA^t\beta^{-1} = BB^t$.
- if A is a weighing matrix then the previous equality becomes $kI = kI$ and carries no information.
- In this case we can use componentwise a monic function $f : \{0, \pm 1\} \to \mathbb{R}$ so that $f(A), f(B)$ are not weighing matrices.
- The equality used now is $\beta f(A)f(A)^t\beta^t = f(B)f(B)^t$.
- β acts on the gram matrix $f(A)f(A)^t$ and all the eigenvectors. It can be recoved from the eigenvectors of multuiplicity one.
- Once β is found then $\alpha = (\beta A)^{-1}B$.

from code invariant mci to an isomophism

- Suppose $mci(A) = mci(B)$ in $M_n(\{0, \pm 1\})$. Then $smA \subset A$ and $smB \subset B$ are paired (not uniqly) by cv.
- Given that $cv(smA) = cv(smB)$, there are σ of length m and τ of length n such that $smB = \sigma smA\tau$.
- τ acts on A, and one needs to find α so that $B = \alpha A\tau$.
- One can normalize the columns of $A\tau$ as done for rows, and find a scalar matrix δ as was done for the columns, and find a nonsigned permutation ϵ as before and set $\alpha = \epsilon\delta$.
- Some tricks exist to reduce the enumeration in the last part.

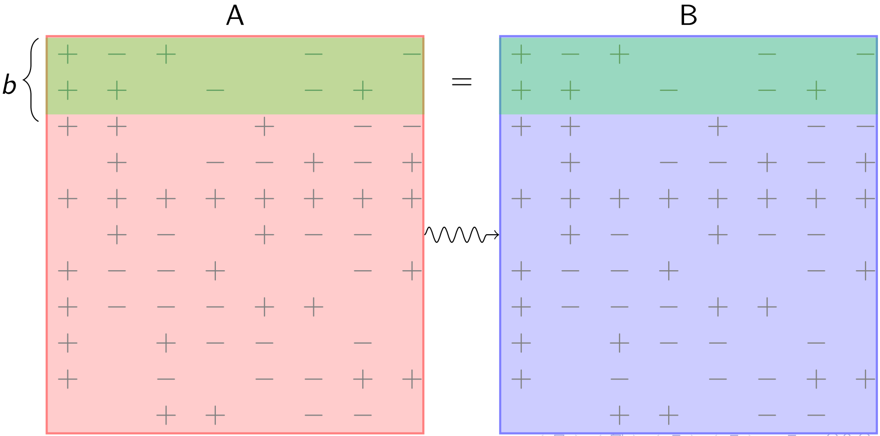
Reducing the enumeration

Need to find isomorphism between A and B.



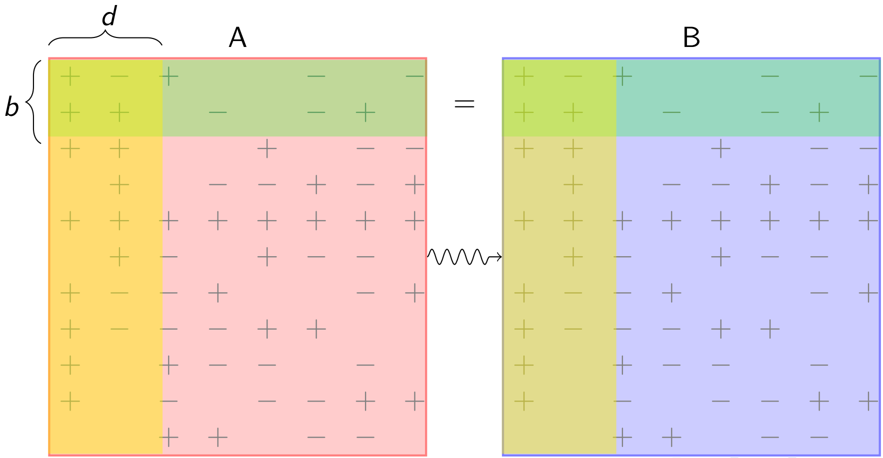
Reducing the enumeration

Assume the first b lines are normalized and equal.



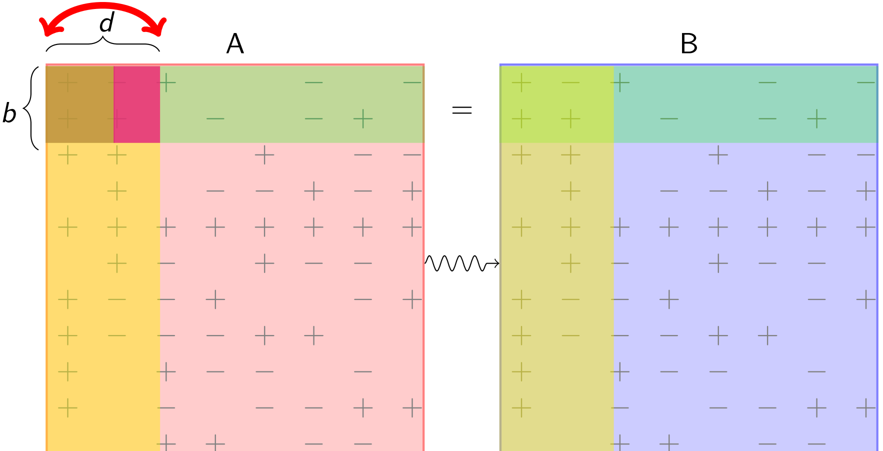
Reducing the enumeration

Take the 1st d columns.



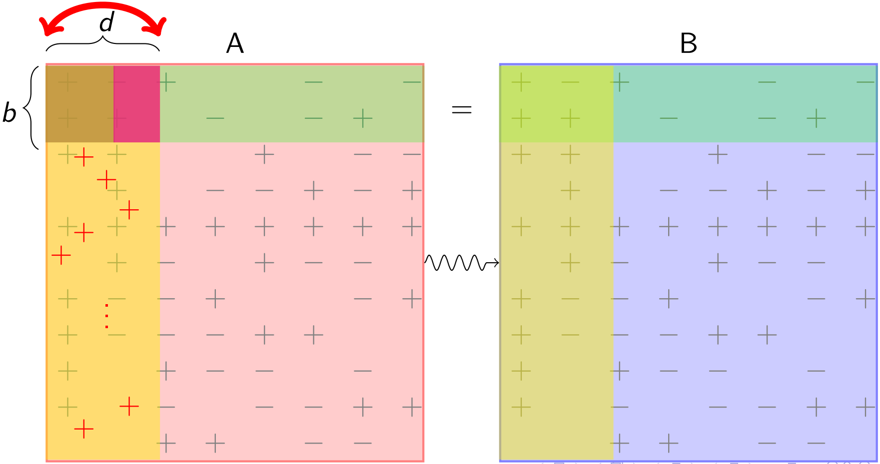
Reducing the enumeration

Permute the 1st d columns if necessary. Since we know the b× d matrix we may need less than $d!$ permutations.



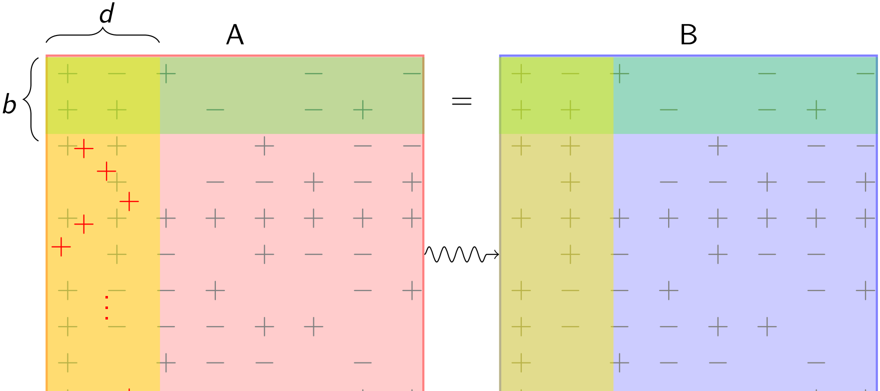
Reducing the enumeration

Normalize row fronts to '+' signs



Reducing the enumeration

- We got rid of signs!
- $B = PAQ$ for unsigned permutations. Finish by applying (the fast) eigenvector method.



The (anti)symmetric representative

- There is a totally different way to find α above if A is invertible. $\alpha A\tau = B$ implies $\alpha = B(A\tau)^{-1}$. Compute the above α and if it is a signed permutation we are done.

Theorem

There exists an (anti)symmetric representative in the class of W $\Leftrightarrow$ there exists an isomorphism $LWR = W^t$ such that $L = (-)R$.

- We find all isomorphisms $W \approx W^t$ and look for those with $R = \pm L$. If no such isomorphism exists then there is not an (anti)symmetric representative in the class of W.

The automorphism group

- The automorphisms of A form a group with composition $(L, R)(K, S) = (LK, RS)$ This group acts on $Iso(A, B)$ from the left and $Iso(C, A)$ from the right.
- $Aut(W)$ divides W to orbits. We can reproduce another W by changing in each orbit one element. Some orbits must be assigned the value 0.
- For n odd $W \approx V$ where V is circulant $\Leftrightarrow (\mathbb{Z}_n \subset Aut(W))$.

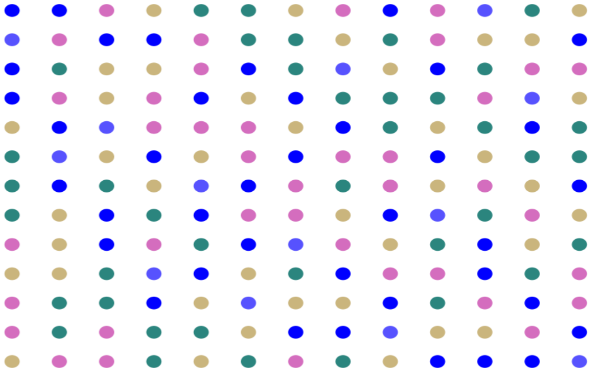
Example

This is a $W(13, 9)$:

$$W = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & -1 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & -1 & -1 & -1 & 0 \\
1 & -1 & -1 & 1 & 0 & 0 & -1 & 1 & 0 & 1 & 1 & 0 & -1 \\
1 & -1 & 0 & 0 & -1 & -1 & 1 & 0 & 1 & -1 & 1 & 0 & 1 \\
1 & 1 & -1 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & -1 & 0 & -1 & 0 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & 0 & 1 & 1 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & -1 & 1 \\
1 & 0 & 1 & 0 & -1 & 1 & -1 & -1 & 0 & -1 & 0 & 1 & -1 \\
1 & 0 & 0 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & 1 & 0 \\
0 & 1 & -1 & 1 & -1 & 0 & 1 & 0 & -1 & -1 & 0 & -1 & -1 \\
0 & 1 & -1 & 0 & 0 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 0 \\
0 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 1 & -1 & 1 \\
0 & 0 & 1 & -1 & 1 & -1 & 0 & 1 & -1 & -1 & 1 & 0 & -1
\end{pmatrix}$$

Example

Here is how $Aut(W)$ breaks W into orbits:



- There are 5 orbits of orders $39, 39, 39, 39, 13$.
- Each orbit is determined via $Aut(W)$ from a single entry.
- $\Longrightarrow$ Can recover W from at most 3^5 candidates.

This is $W(23,16)$ found 2.5 years ago

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

Here is an L matrix

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

This is a $SW(23, 16)$

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

Thank you for your attension