Simple folding of array-based VLSI structures

Liudmila Cheremisinova

The United Institute of Informatics Problems of National Academy of Sciences of Belarus

The problem under consideration is:
to **reduce the area of the layout** of regular VLSI structures as Programmable Logic Array (PLA) by means of their folding.

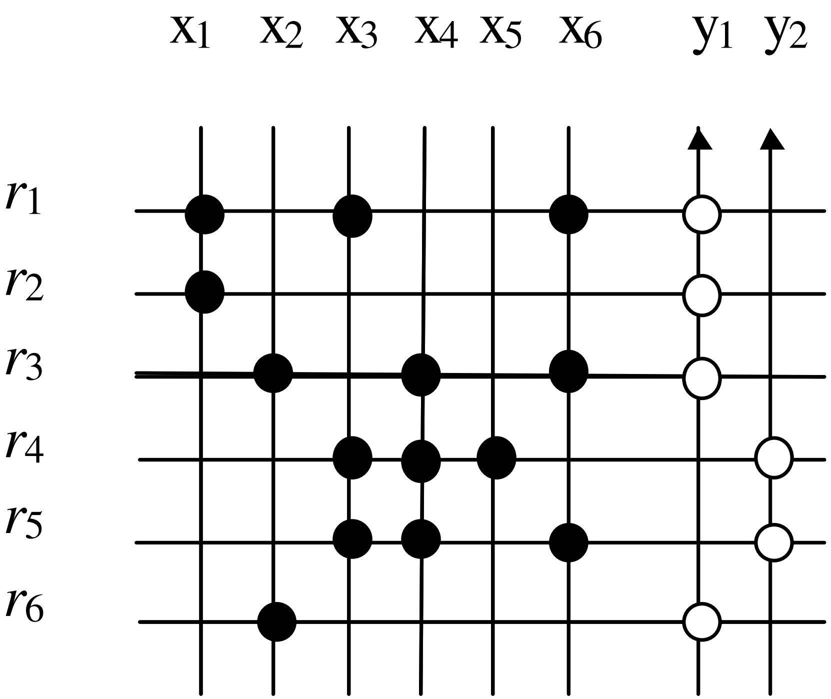
Two approaches are usually used:

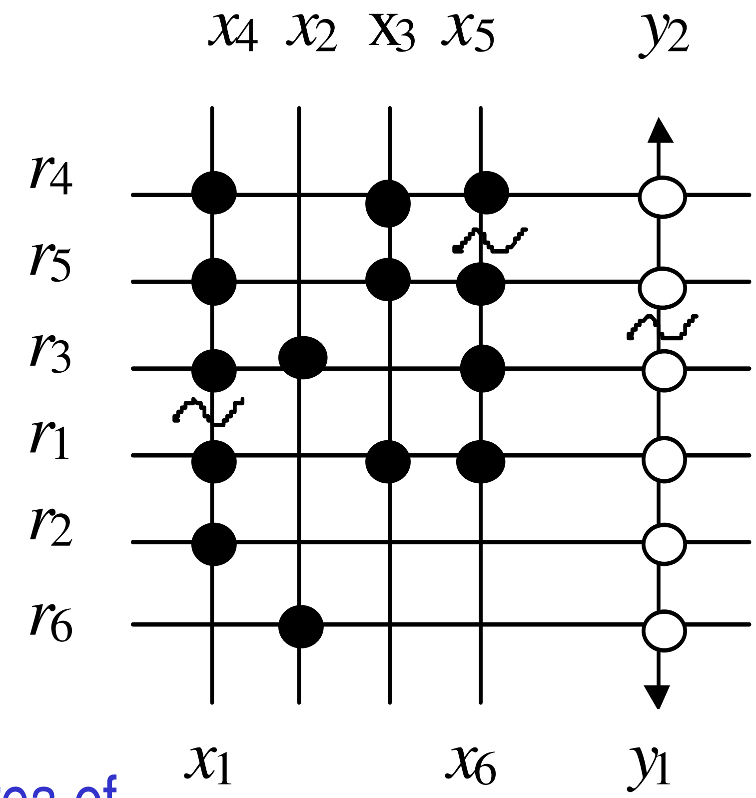
- logic minimization;
- topological minimization reclaiming unused space.

The problem of PLA topological optimizing by means of its **folding** is considered.

Simple folding of array-based VLSI structures

PLA (Programmable Logic Array) is a standard two level VLSI structure.
PLA consists of AND and OR planes

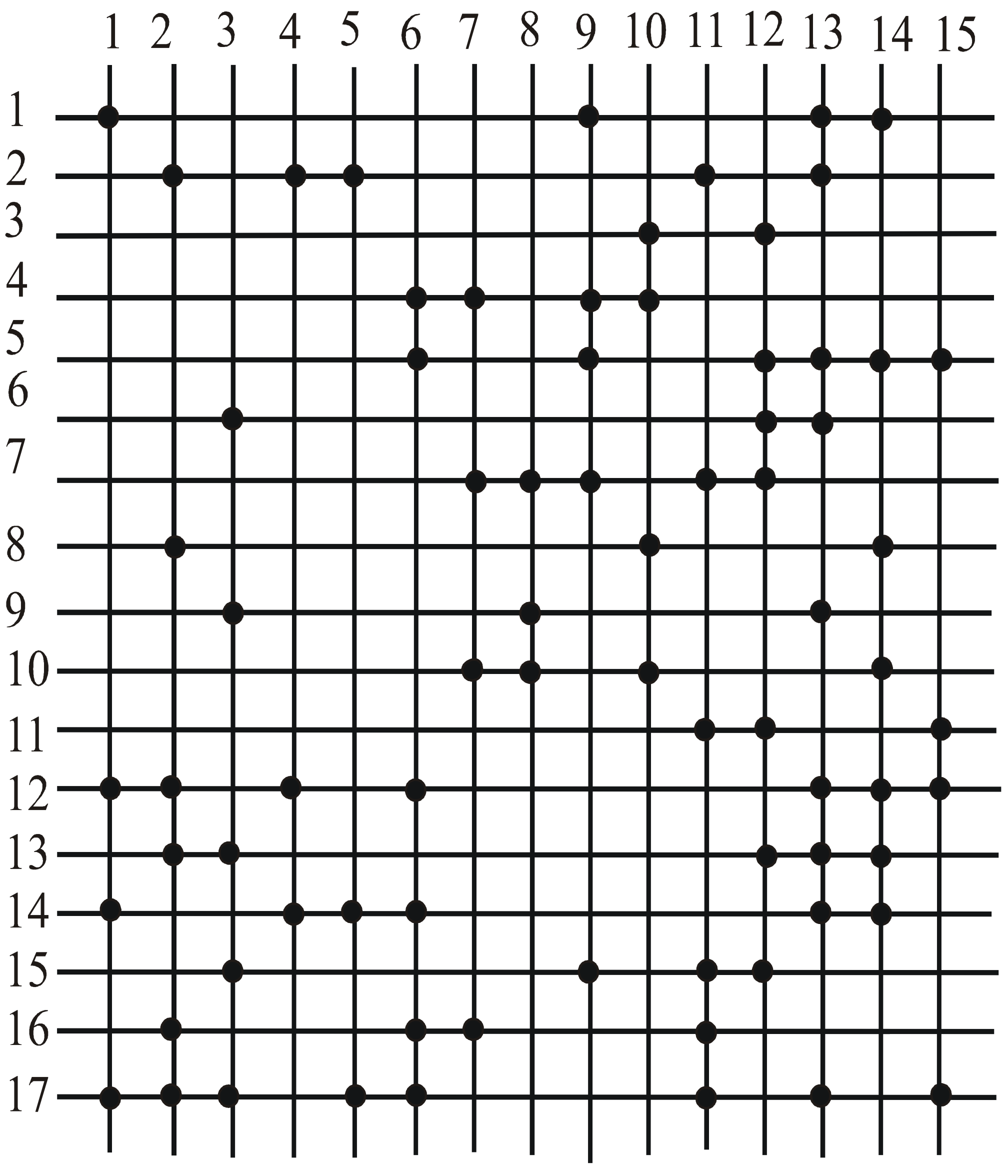




The area of

the initial PLA is 48 = 8 x 6 the column folded PLA is 30 = 5 x^{2}6

Regular structure and its symbolic form



Symbolic form is a Boolean matrix $\boldsymbol{B}$ having the sets $C(\boldsymbol{B})$ and $R(\boldsymbol{B})$ of columns c_j and rows r_i .

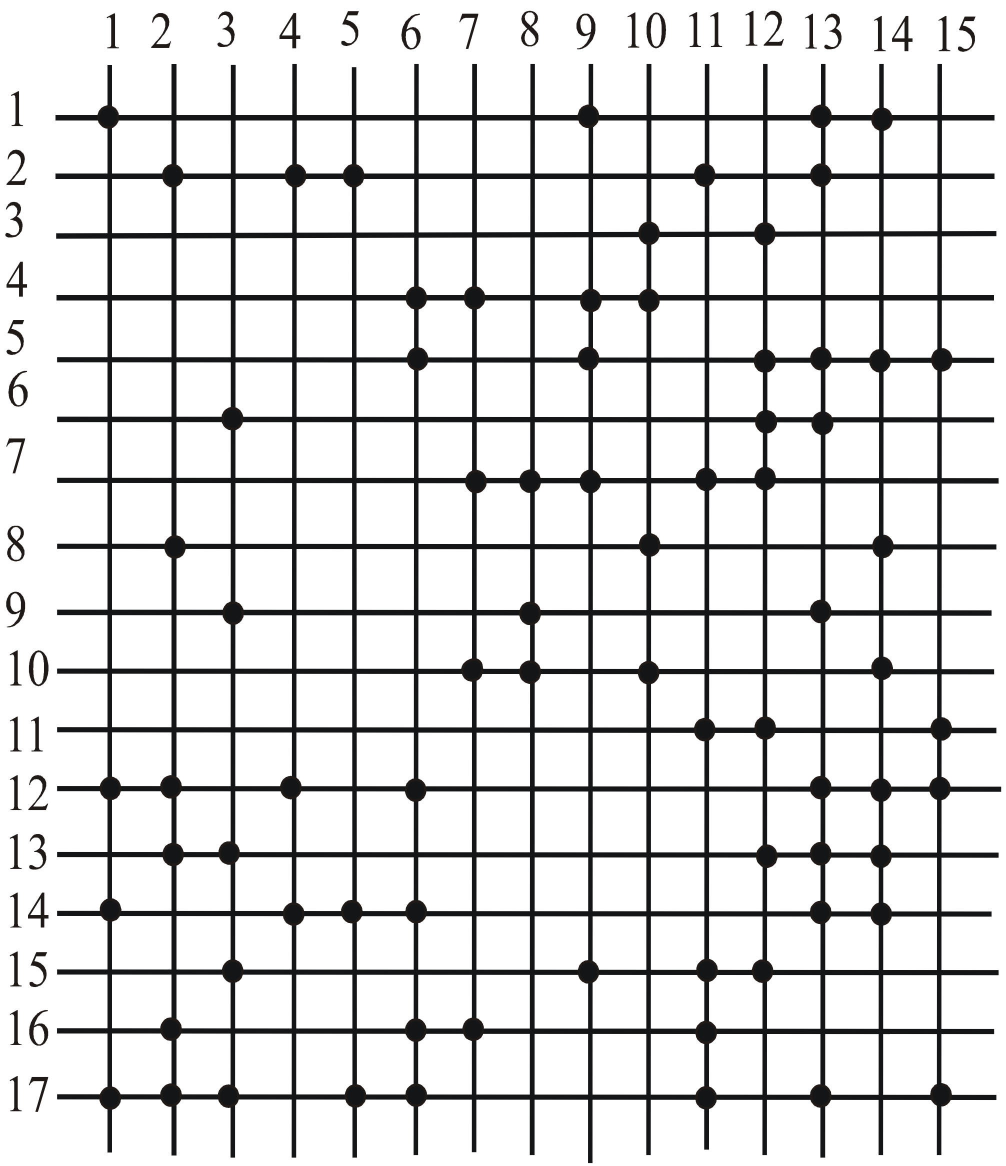
A column c_j^B implies a set $R(c_j^B)$ of rows: $r_i^B \in R(c_j^B) \leftrightarrow b_i^j = 1$.

c_i and c_j are ***compatible*** and form a ***foldable pair*** if they are **disjoint** : $R(c_i) \cap R(c_j) = \varnothing$.

(c_{10},c_{11}) is a folding pair:
$R(c_{10}) \cap R(c_{11}) = \{r_3, r_4, r_8, r_{10}, r_3\} \cap \{r_2, r_7, r_{11}, r_{15}, r_{16}, r_{17}\} = \varnothing$

(c_1,c_2) is not a folding pair:
$R(c_1) \cap R(c_2) = \{r_{12}, r_{17}\} \neq \varnothing$

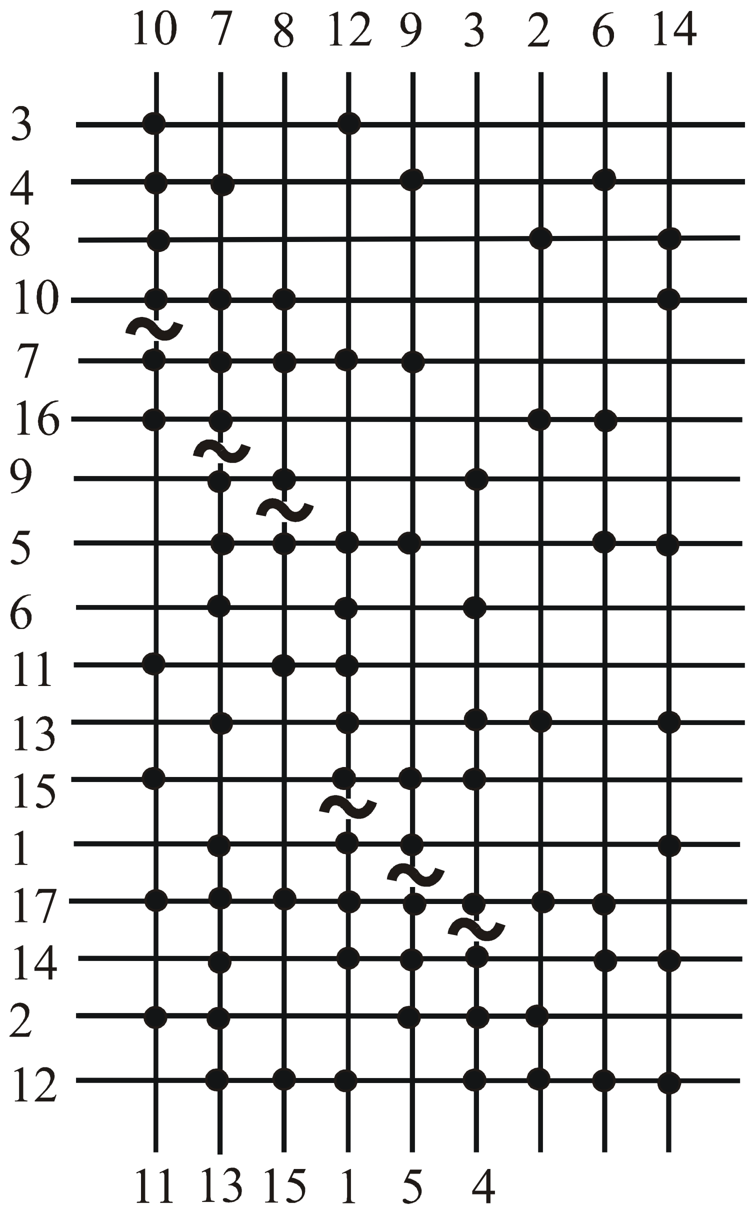
Column compatibility matrix



Symmetric and irreflexive relation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0
2	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
3	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0
4	0	0	1	0	0	0	1	1	1	1	0	1	0	0	0
5	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0
6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	1	0	1	1	1	0	0	0	0	0	0	0	1	0	1
8	1	1	0	1	1	1	0	0	0	0	0	0	0	0	1
9	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
10	1	0	1	1	1	0	0	0	0	0	1	0	1	0	1
11	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
12	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0

Simple folding of an array-based VLSI structure



	10	7	8	12	9	3	
	1	0	0	0	0	0	11
	1	1	0	0	0	0	13
F =	1	1	1	0	0	0	15
	1	1	1	1	0	0	1
	1	1	1	1	1	1	5
	1	1	1	1	1	1	4

A **foldable compatibility matrix** (**FCM**)

F is submatrix of the column compatibility matrix.

F is square.

F has all 1s lower triangle about the leading diagonal.

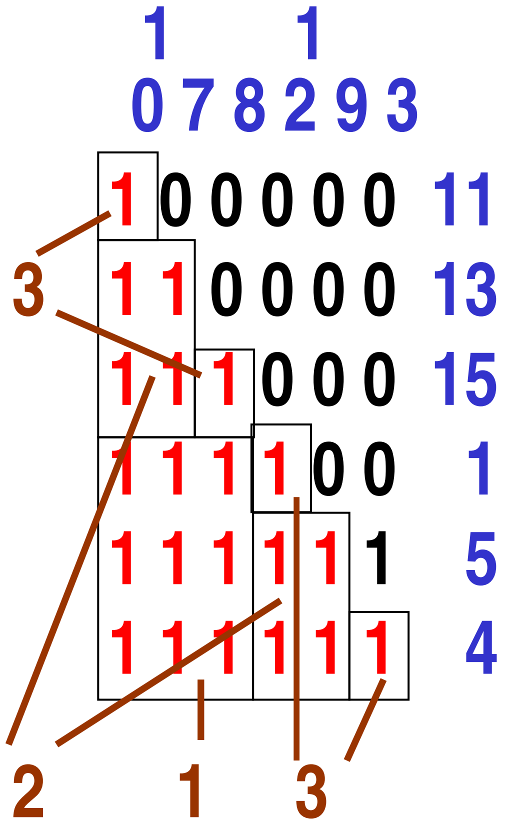
(c_i^F, r_i^F) is a foldable pair.

Column compatibility matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0
2	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
3	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0
4	0	0	1	0	0	0	1	1	1	1	0	1	0	0	0
5	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0
6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	1	0	1	1	1	0	0	0	0	0	0	0	1	0	1
8	1	1	0	1	1	1	0	0	0	0	0	0	0	0	1
9	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
10	1	0	1	1	1	0	0	0	0	0	1	0	1	0	1
11	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
12	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0

	10	7	8	12	9	13	1	2	4	5	6	11	3	14	15
11	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
13	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
2	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0
6	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	1	0	1	1	0	0	1	0	1
8	0	0	0	0	0	0	1	1	1	1	1	0	0	0	1
9	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
10	0	0	0	0	0	1	1	0	1	1	0	1	1	0	1
12	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

The idea of the method of searching for a foldable compatibility matrix of the greatest size



FCM of size *m* exists if there exist

1-level: **square unit minor of size**

$$p^1 =]\, m/2\, [: \qquad p^1 =]\, 6/2\, [= 3$$

2-level: **companion unit minors of size**

$$p^2 =]\, (m - p^1)/2\, [: \quad p^2 =]\, (6 - 3)/2\, [= 2$$

3-level: **companion unit minors of size**

$$p^3 =]\, (m - (p^1 + p^2)/2\, [:$$

$$p^3 =]\, (6 - 3 - 2)/2\, [= 1$$

$$p^j =]\, (m - \sum_{i<j} p^i)/2\, [$$

Derivation of the foldable compatibility matrix (FCM)

	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0
2	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
3	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0
4	0	0	1	0	0	0	1	1	1	1	0	1	0	0	0
5	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0
6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	1	0	1	1	1	0	0	0	0	0	0	0	1	0	1
8	1	1	0	1	1	1	0	0	0	0	0	0	0	0	1
9	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
10	1	0	1	1	1	0	0	0	0	0	1	0	1	0	1
11	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
12	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0
w =	4	2	3	6	5	1	6	6	3	7	2	3	2	1	3

FCM is upper bounded by]n / 2[=]15/2[= 7 pairs of foldable columns exist

Theorem 2. If for an array structure there exists a FCM of size m then at least $2([m/2]+1)$ columns in its compatibility matrix will have weights greater than or equal to $]m/2[$.

For $m = 7$ 8 columns with w > 3 must be
we have 4 only
For $m = 6$ 8 columns with w > 2 must be
we have 10

Theorem 3. The necessary condition for $m \times m$ Boolean matrix to become a FCM of size m is the existence of a square unit minor of size $]m/2[\times]m/2[$ in it.

Derivation of the foldable compatibility matrix (FCM)

	1	3	4	5	7	8	9	0	2	5
1	0	0	0	0	1	1	0	1	1	0
3	0	0	1	0	1	0	0	1	0	0
4	0	1	0	0	1	1	1	1	1	0
5	0	0	0	0	1	1	1	1	1	0
7	1	1	1	1	0	0	0	0	0	1
8	1	0	1	1	0	0	0	0	0	1
9	0	0	1	1	0	0	0	0	0	0
10	1	1	1	1	0	0	0	0	0	1
12	1	0	1	1	0	0	0	0	0	0
15	0	0	0	0	1	1	0	1	0	0
w =	4	3	6	5	6	6	3	7	3	3

	7	8	10	w
3	1	0	1	2
13	1	0	1	2
15	1	1	1	3

	9	12
1	0	1
4	1	1
5	1	1
w =	2	3

	7	10	w
3	1	1	2
11	0	1	2

	8
15	1

	10
13	1

	3
4	1
w	1

10	7	8	12	9	13	
1	0	0	0	0	0	11
1	1	0	0	0	0	13
1	1	1	0	0	0	15
1	1	1	1	0	0	1
1	1	1	1	1	1	5
1	1	1	1	1	1	4

$$(C^1, R^1) = (\{7, 8, 10\}, \{1, 4, 5\})$$

$$C_1^2 \subseteq \{7, 8, 10\}, R_2^2 \subseteq \{1, 4, 5\}, R_1^2, C_2^2 \subseteq \{2,3,6,9,11,12,13,14,15\}$$

Conclusion

1. A new simple folding technique is presented.
2. The problem of the simple folding is reduced to a search for a maximal unit minors of a Boolean matrices.
3. The method helps not to lose the way to the optimal solution of the folding problem at the first step.