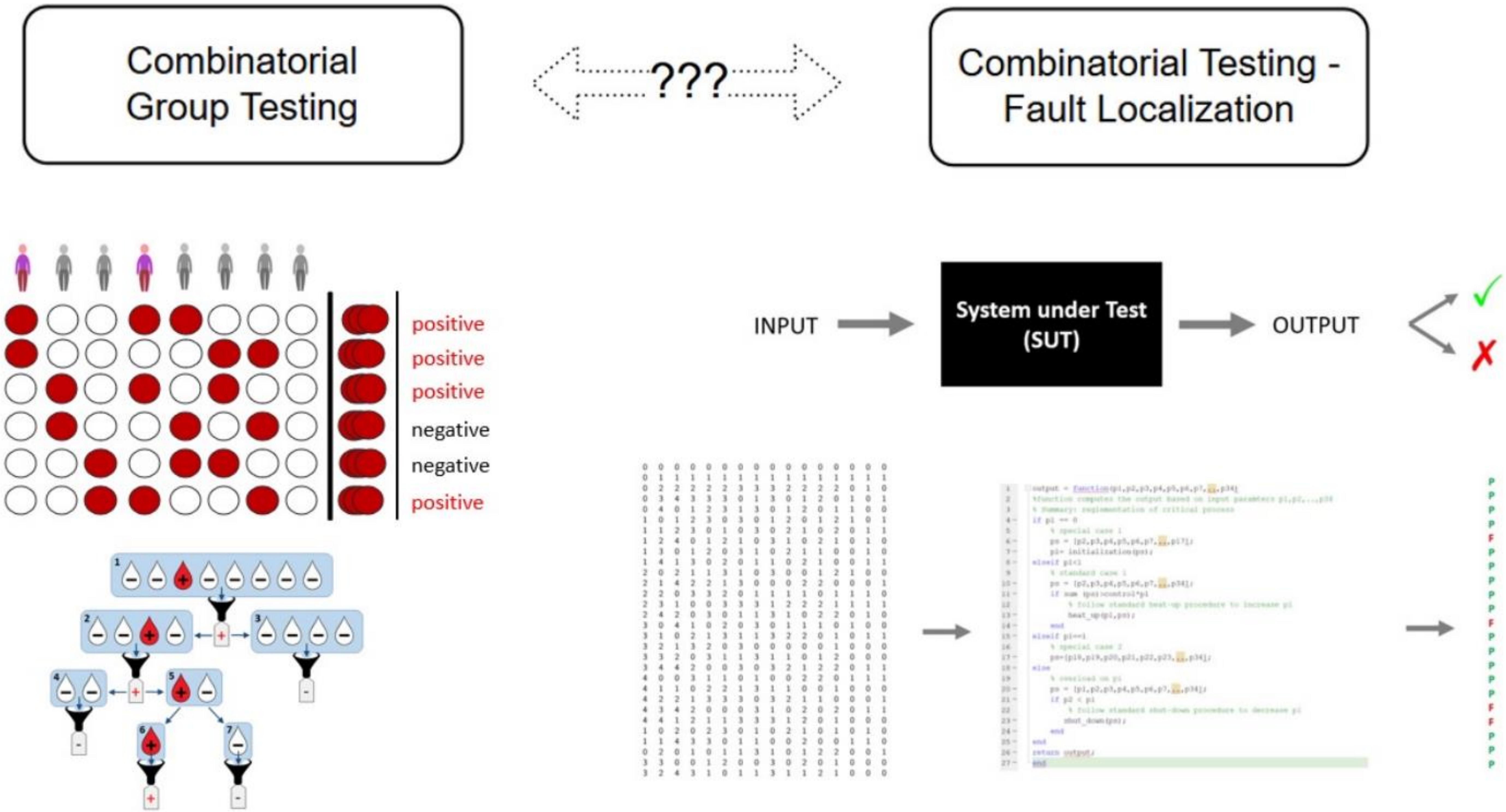
Locating Single Failure Inducing t-way Interactions with 0^t-Locating Arrays
or
Locating Failure Inducing t-way Interactions using d-separable Matrices

Ludwig Kampel, Irene Hiess, Michael Wagner, Marlene Koelbing, Dimitris E. Simos

AAAM Research, MATRIS Research Group, SBA Research

5th Pythagorean Conference, June 4, 2025
Kalamata, Greece

Outline


Combinatorial Group Testing
???
Combinatorial Testing - Fault Localization
positive
positive
positive
negative
negative
positive
INPUT
System under Test (SUT)
OUTPUT

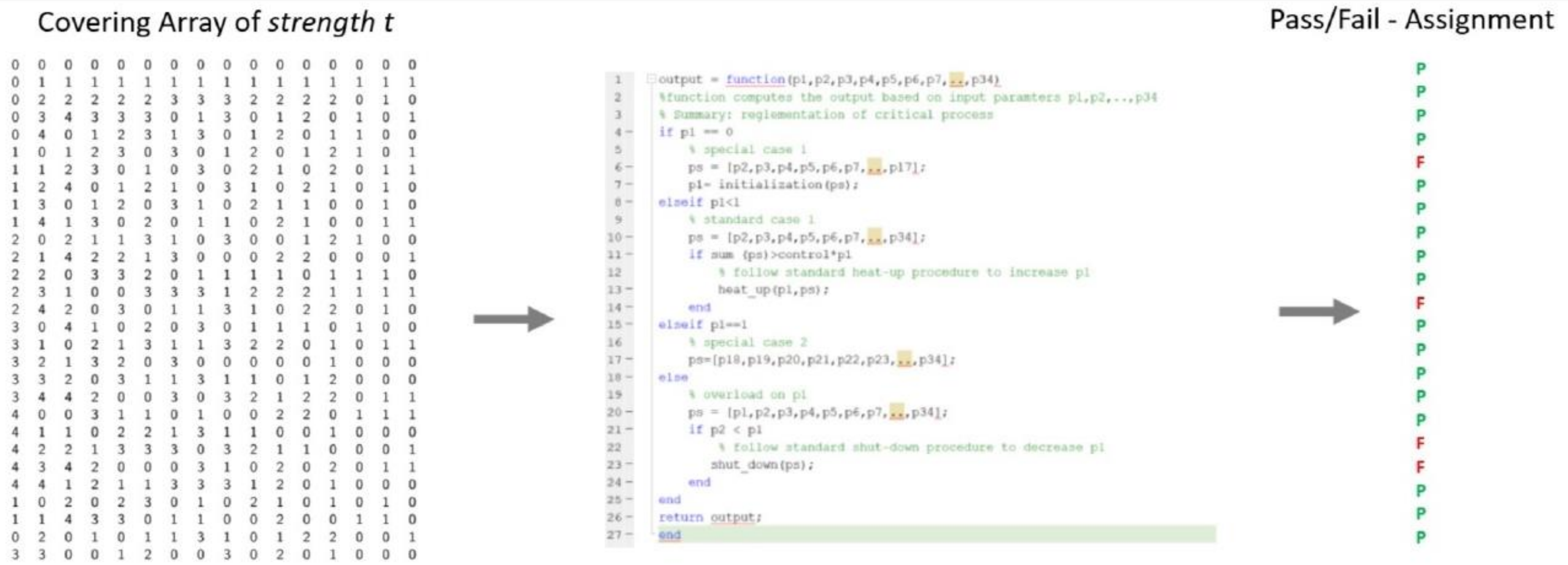
Outline of the talk

Goal of Combinatorial Testing (CT)

Detect (verify presence) of t-way failure inducing interactions (t-way FITs)

Combinatorial Designs Applied

- Covering Arrays (CAs), also known as surjective arrays, qualitatively independent partitions (resp. family of sets)
- Mixed-Level Covering Arrays (MCAs)
- Variable-strength covering arrays (VCAs)



The Problem of Combinatorial Testing - Fault Localization

Detecting failure inducing t-way interactions

Goal of Combinatorial Testing - Fault Localization (CT-FLA)

Locate t-way FITs (i.e. identify the positions & values in the input)

We are given

- Test set executed on system under test
- Pass/Fail - assignment from execution

Given Information

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	oracle
$c_1 =$	0	1	0	1	1	0	0	P
$n_1 =$	1	0	1	0	1	1	0	F
$c_2 =$	1	0	1	1	1	0	1	P
$c_3 =$	0	0	1	1	1	0	1	P
$c_4 =$	1	0	0	0	1	0	1	P
$n_2 =$	0	1	1	0	1	1	1	F
$\vdots$								$\vdots$

The Problem of Combinatorial Testing - Fault Localization

Detecting failure inducing t-way interactions

Goal of Combinatorial Testing - Fault Localization (CT-FLA)

Locate t-way FITs (i.e. identify the positions & values in the input)

We are given

- Test set executed on system under test
- Pass/Fail - assignment from execution

Given Information

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	oracle
$c_1 =$	0	1	0	1	1	0	0	P
$n_1 =$	1	0	1	0	1	1	0	F
$c_2 =$	1	0	1	1	1	0	1	P
$c_3 =$	0	0	1	1	1	0	1	P
$c_4 =$	1	0	0	0	1	0	1	P
$n_2 =$	0	1	1	0	1	1	1	F
$\vdots$								$\vdots$

Goal of Combinatorial Testing - Fault Localization (CT-FLA)

Locate t-way FITs (i.e. identify the positions & values in the input)

We are given

- Test set executed on system under test
- Pass/Fail - assignment from execution

Given Information

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	oracle
$c_1 =$	0	1	0	1	1	0	0	P
$n_1 =$	1	0	1	0	1	1	0	F
$c_2 =$	1	0	1	1	1	0	1	P
$c_3 =$	0	0	1	1	1	0	1	P
$c_4 =$	1	0	0	0	1	0	1	P
$n_2 =$	0	1	1	0	1	1	1	F
⋮								⋮

The Problem of Combinatorial Testing - Fault Localization

Detecting failure inducing t-way interactions

Goal of Combinatorial Testing - Fault Localization (CT-FLA)

Locate t-way FITs (i.e. identify the positions & values in the input)

We are given

- Test set executed on system under test
- Pass/Fail - assignment from execution

Given Information

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	oracle
$c_1=$	0	1	0	1	1	0	0	P
$n_1 =$	1	0	1	0	1	1	0	F
$c_2=$	1	0	1	1	1	0	1	P
$c_3=$	0	0	1	1	1	0	1	P
$c_4=$	1	0	0	0	1	0	1	P
$n_2=$	0	1	1	0	1	1	1	F
$\vdots$								$\vdots$

The Problem of Combinatorial Testing - Fault Localization

Detecting failure inducing t-way interactions

Goal of Combinatorial Testing - Fault Localization (CT-FLA)

Locate t-way FITs (i.e. identify the positions & values in the input)

We are given

- Test set executed on system under test
- Pass/Fail - assignment from execution

Given Information

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	oracle
$c_1 =$	0	1	0	1	1	0	0	P
$n_1 =$	1	0	1	0	1	1	0	F
$c_2 =$	1	0	1	1	1	0	1	P
$c_3 =$	0	0	1	1	1	0	1	P
$c_4 =$	1	0	0	0	1	0	1	P
$n_2 =$	0	1	1	0	1	1	1	F
$\vdots$								$\vdots$

Parallels, Related Concepts and Existing Connections

Detecting failure inducing t-way interactions

J Comb Optim (2008) 15: 17–48 33

are somewhat different. Pools are always formed by the selection, for each factor, of exactly one level. While this severely limits the applicability of specific results from the literature on combinatorial group testing to our problem, the basic framework remains quite similar. Hence we believe that there is much value in pursuing the parallels between locating arrays and combinatorial group testing, especially that for complexes.

C. J. Colbourn, D. W. McClary, Locating and detecting arrays for interaction faults, JCO, 2008.

J Comb Optim (2008) 15: 17–48 33

are somewhat different. Pools are always formed by the selection, for each factor, of exactly one level. While this severe[…] the literature on combinatorial gro[…] remains quite similar. Hence we b[…] parallels between locating arrays an[…] complexes.

C. J. Colbourn, D. W. McCla[…]

The testing problem considered here is related to combinatorial group testing, where, in its simplest form, a set of up to d defective items are to be found among a set of m items. Combinatorial group testing has been extensively studied (see [13]). Most relevant to us are the problems of *searching for edges in a graph* [1] where defects may come from sets of two items (rather than single items) and tests are done on sets of two items (see also section 12.5 in [13]), and of *group testing for complexes* where subsets of size t of the items (rather than single items) are responsible for errors [24] (see also section 5 in [10]). However, there are significant differences between these variations of group testing and our testing problem: in our case, the structure of the pools for testing is very restricted (one must select k items corresponding to selecting one value for each of the k factors), and subsets of size up to t of the items may be responsible for errors.

Covering arrays are combinatorial designs that correspond to test suites that cover

C. Martinez, L. Moura, D. Panario, B. Stevens, Locating Errors Using ELAs, Covering Arrays, and Adaptive Testing Algorithms SIAM J. DM, 2009.

Parallels, Related Concepts and Existing Connections

Detecting failure inducing t-way interactions

J Comb Optim (2008) 15: 17–48 33

are somewhat different. Pools are always formed by the selection, for each factor, of exactly one level. While this severe... the literature on combinatorial gro... remains quite similar. Hence we b... parallels between locating arrays an... complexes.

C. J. Colbourn, D. W. McCla... ns, Locating Errors Using ELAs, ...rithms SIAM J. DM, 2009.

The testing problem considered here is related to combinatorial group testing, where, in its simplest form, a set of up to d defective items are to be found among a set of m items. Combinatorial group testing has been extensively studied (see [13]). Most relevant to us are the problems of *searching for edges in a graph* [1] where defects may come from sets of two items (rather than single items) and tests are done on sets of two items (see also section 12.5 in [13]), and of *group testing for complexes* where subsets of size t of the items (rather than single items) are responsible for errors [24] (see also section 5 in [10]). However, there are significant differences between these variations of group testing and our testing problem: in our case, the structure of the pools for testing is very restricted (one must select k items corresponding to selecting ... to t of the items may be ... nd to test suites that cover

5.2. Software interaction testing

Martínez et al. [15, Section 5] give an adaptive algorithm for software interaction testing when tests can fail due to interactions of up to $r=2$ parameter values; their algorithm requires an additional hypothesis that the testing problem has the so-called "safe values". Their algorithm relies in two parts. First, it creates a list of tests that covers all possible faulty interation of parameter values. Second, for each of these tests, it applies an algorithm that is equivalent of solving a group testing for complexes for $r=2$.

The algorithm we presented in Section 2, can be used to generalize the algorithm by Martínez et al. to solve the problem of software interaction testing for failing interactions of up to $r \geq 2$ parameter values under the hypothesis of "safe values". For fixed r, this yields a generalization of the method of Martínez et al. that performs at most $O(d^r(\log n)^2+\log n)$ tests.

J. Chodoriwsky, L. Moura, An adaptive algorithm for group testing for complexes, TCS (2015) .

- t-way interaction: $\{(p_1, u_1), (p_2, u_2), \ldots, (p_t, u_t)\}$
  Pairs of a *position/index* and a *value*, where $1 \leq p_1 < p_2 < \ldots < p_t \leq k$, and $u_i \in [0, v-1] \cap \mathbb{Z}$.
- A row of an array $A = (A_{r,1}, \ldots, A_{r,k})$ covers a t-way interaction $\tau = \{(p_1, v_1), \ldots, (p_t, v_t)\}$ iff $A_{r,p_i} = v_i$ for $1 \leq i \leq t$. $\rho_A(\tau)$ denotes the set of rows of A that cover the interaction τ.

Definition: (d, t)-Locating Array (LA) - Colbourn, McClary (2008)

An $N \times k$ array A over a v-ary alphabet is a (d, t)-Locating Array iff:
for all d-sets $\mathcal{T}$, $\mathcal{T}'$ of t-way interactions it holds that:

$$\rho_A(\mathcal{T}) = \rho_A(\mathcal{T}') \Rightarrow \mathcal{T} = \mathcal{T}', \text{ where } \rho_A(\mathcal{T}) := \bigcup_{\tau \in \mathcal{T}} \rho_A(\tau).$$

Failure-inducing t-way interaction (FIT) localization via row-signatures

Definition: (d, t)-Detecting Array (DA) - Colbourn, McClary (2008)

An $N \times k$ array A over a v-ary alphabet is a (d, t)-Detecting Array iff:
for any d-set $\mathcal{T}$ of t-way interactions and any t-way interaction τ it holds that:

$$\rho_A(\tau) \subseteq \rho_A(\mathcal{T}) \Rightarrow \tau \in \mathcal{T}.$$

FIT localization via marking t-way interactions as non-FIT

Non-adaptive CT-FLA Approaches

- No communication to SUT required
- Fault localization relies solely on test set and PASS/FAIL-assignment
- $\Rightarrow$ Test set needs a-priori sufficient structure
- Fault localization guaranteed if preconditions are met
- SUT agnostic $\Rightarrow$ reusable structures/test sets
- "Rigid "structures/test sets

001221	112002
010122	121200
021012	102120
022101	100212
012210	120021
102112	010220
120211	001022
112021	020102
111202	022010
121120	002201

A (1,2)-LA(20;2,6,3)

Adaptive CT-FLA Approaches

- Require communication to SUT
- Fault localization through re-sampling of tests
- Ranking of "suspicious" t-way interactions
- Exploration of neighbourhood of failing tests
- SUT specific $\Rightarrow$ versatile usable
- Non-reusable test sets

Inspection of Individual Failing Tests

Combinatorial Testing in Short
Motivation
0^t-LA & 0^t-DA
CGT and CT-FLA

Failing test:

(1,	5,	1,	1.39,	0,	5,	10,	20)	Fail

Additionally derived tests:

(2,	5,	1,	1.79,	1,	5,	10,	20)	Fail
(2,	5,	1,	1.39,	0,	1,	15,	20)	Pass
(1,	6,	1,	1.79,	0,	1,	10,	20)	Pass
(1,	6,	1,	1.39,	1,	5,	15,	20)	Fail
(1,	5,	3,	1.39,	1,	1,	10,	20)	Pass
(1,	5,	3,	1.79,	0,	5,	15,	20)	Pass
(*,	*,	1,	*,	*,	5,	*,	*)	⇒ Fail

Inspection of Individual Failing Tests

Failing test:

(1,	5,	1,	1.39,	0,	5,	10,	20)	Fail

Binary Matrix

Additionally derived tests:

(2,	5,	1,	1.79,	1,	5,	10,	20)	Fail
(2,	5,	1,	1.39,	0,	1,	15,	20)	Pass
(1,	6,	1,	1.79,	0,	1,	10,	20)	Pass
(1,	6,	1,	1.39,	1,	5,	15,	20)	Fail
(1,	5,	3,	1.39,	1,	1,	10,	20)	Pass
(1,	5,	3,	1.79,	0,	5,	15,	20)	Pass
(*,	*,	1,	*,	*,	5,	*,	*)	$\Rightarrow$ Fail

Inspection of Individual Failing Tests

Failing test:

(1,	5,	1,	1.39,	0,	5,	10,	20)	Fail

0^t-Locating Array

Additionally derived tests:

(2,	5,	1,	1.79,	1,	5,	10,	20)	Fail
(2,	5,	1,	1.39,	0,	1,	15,	20)	Pass
(1,	6,	1,	1.79,	0,	1,	10,	20)	Pass
(1,	6,	1,	1.39,	1,	5,	15,	20)	Fail
(1,	5,	3,	1.39,	1,	1,	10,	20)	Pass
(1,	5,	3,	1.79,	0,	5,	15,	20)	Pass
(*,	*,	1,	*,	*,	5,	*,	*)	⇒ Fail

Definition of 0^t-locating arrays and 0^t-detecting arrays

0^t-way interaction: $\tau = \{p_1, \ldots, p_t\}$ is a set of t positions, corr. to a t-way interaction with only 0 values: $\{(p_1, 0), (p_2, 0), \ldots, (p_t, 0)\}$

Definition: $(d, 0^t)$-Locating Array ($(d, 0^t)$-LA)

A binary $N \times k$ array A is a $(d, 0^t)$-Locating Array, denoted as $(d, 0^t)$-$LA(N; t, k)$, iff:
for any two d-sets of 0^t-way interactions $\mathcal{T} = \{\tau_1, \ldots, \tau_d\}$, $\mathcal{T}' = \{\tau'_1, \ldots, \tau'_d\}$ it holds that:

$$\rho_A(\mathcal{T}) = \rho_A(\mathcal{T}') \Rightarrow \mathcal{T} = \mathcal{T}'.$$

Note: For a 0^t-LA only a single 0^t-way interaction τ can have $\rho_A(\tau) = \emptyset$.

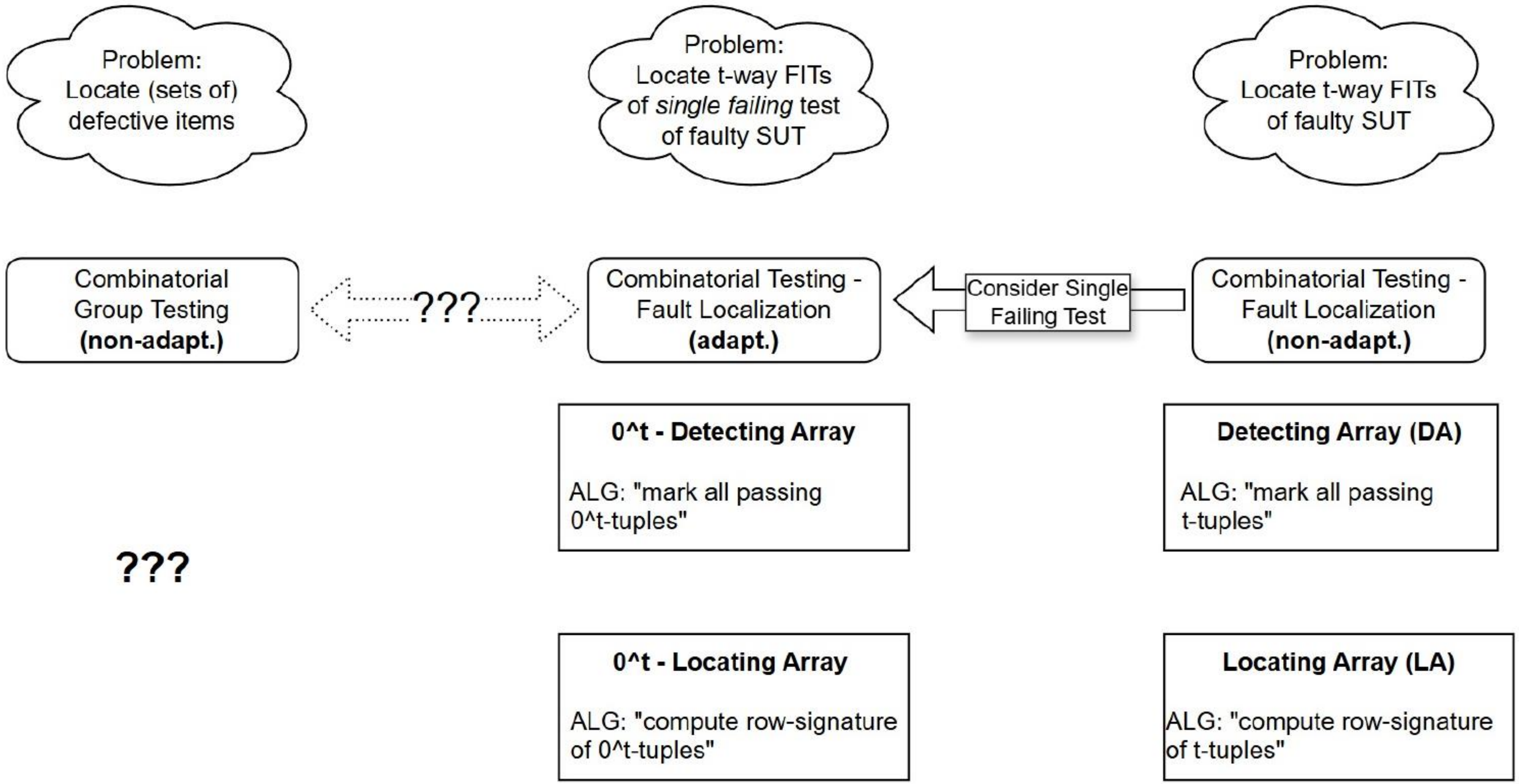
Definition: $(d, 0^t)$-Detecting Array (DA)

A binary $N \times k$ array A is a $(d, 0^t)$-Detecting Array, denoted as $(d, 0^t)$-$DA(N; t, k)$, iff:
for any d-set $\mathcal{T}$of 0^t-way interactions and every 0^t-way interaction τ it holds that:

$$\rho_A(\tau) \subseteq \rho_A(\mathcal{T}) \Rightarrow \tau \in \mathcal{T}.$$

Note: A 0^t-DA covers all 0^t-way interactions, since $\rho_A(\tau) = \emptyset$ is not possible.

Combinatorial Group Testing and CT-FLA



Examples

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

An example of a $(1, 0^2)$-LA$(3; 2, 4)$.

τ	$\rho_A(\tau)$
{(0,1),(0,2)}	{1,2}
{(0,1),(0,3)}	{1}
{(0,2),(0,3)}	{1,3}
{(0,2),(0,4)}	{2,3}
{(0,3),(0,4)}	{3}
{(0,1),(0,4)}	{2}

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

An example of a $(1, 0^2)$-DA$(6; 2, 4)$.

τ	$\rho_A(\tau)$
{(0,1),(0,2)}	{5,6}
{(0,1),(0,3)}	{4,6}
{(0,2),(0,3)}	{1,2,6}
{(0,2),(0,4)}	{2,5}
{(0,3),(0,4)}	{2,4}
{(0,1),(0,4)}	{3,4,5}

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

An example of a CA$(5; 2, 4, 2)$.

τ	$\rho_A(\tau)$
{(0,1),(0,2)}	{1,2}
{(0,1),(0,3)}	{1,3}
{(0,2),(0,3)}	{1,4}
{(0,2),(0,4)}	{1}
{(0,3),(0,4)}	{1}
{(0,1),(0,4)}	{1}

Disjunct Matrices

Definition: $(w, r; d)$-Cover-Free Family[1] resp. $(w, r; d)$-Disjunct Matrix

A $(w, r; d)$-*cover-free family* or a $(w, r; d)$-*cover-free family* is a binary matrix where in any sub-matrix, composed of $w + r$ columns, each binary vector with weight w appears at least d times.

$$
\text{d}\left\{\begin{matrix} \\ \\ \\ \\ \end{matrix}\right.
\begin{pmatrix}
0/1 & 0/1 & 0/1 & 0/1 & 0/1 \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
0/1 & 0/1 & 0/1 & 0/1 & 0/1 \\
\cdots & \overbrace{0\;0\cdots0}^{r} & \cdots & \overbrace{1\;1\cdots1}^{w} & \cdots \\
\cdots & \overbrace{0\;0\cdots0}^{r} & \cdots & \overbrace{1\;1\cdots1}^{w} & \cdots \\
 & \vdots & & \vdots & \\
\cdots & \overbrace{0\;0\cdots0}^{r} & \cdots & \overbrace{1\;1\cdots1}^{w} & \cdots \\
0/1 & 0/1 & 0/1 & 0/1 & 0/1 \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
0/1 & 0/1 & 0/1 & 0/1 & 0/1
\end{pmatrix}
$$

[1] Stinson, Wei, "Generalized cover-free families", Discrete Mathematics, 2004

Proposition

A a binary matrix. A is a $(w, r, 1)$-disjunct matrix $\Leftrightarrow$ A is a $(w, 0^r)$-DA.

Proof: Let τ be an arbitrary 0^r-tuple.

$$A = \begin{pmatrix} 0/1 & 0/1 & \cdots & 0/1 & 0/1 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0/1 & 0/1 & \cdots & 0/1 & 0/1 \\ \cdots & \overbrace{1\ 1 \cdots 1}^{w} & \cdots & \overbrace{0\ 0 \cdots 0}^{r} & \cdots \end{pmatrix}$$

$$\begin{array}{lll} 0 & 0 \cdots 0 & =: \tau_1 \\ \ \ 0 & 0 \cdots 0 & =: \tau_2 \\ \ \ \vdots & \vdots & \\ \quad 0 & 0 \cdots 0 & =: \tau_w \end{array}$$

Equivalence of 0^t-Locating Arrays and Separable Matrices

d-Separable Matrix

A d-separable matrix is a binary matrix where no two sub-matrices composed of d columns have the same row-wise Boolean sum (/superposition sum/ supremum).

A a d-separable $n \times k$ matrix, D_1, D_2 $n \times d$ sub-matrices, then $\bigvee D_1 = s_1 \neq s_2 = \bigvee D_2$

Equivalence of d-Separable Matrices and 0^d-Locating Arrays

A binary array A is a d-separable matrix iff it is 0^d locating.

For a sub-matrix D composed of d columns $\{i_1, \ldots, i_d\}$, we have

$$\bigvee D = \overline{\rho_D(\vec{0})} = \overline{\rho_A(\{i_1, \ldots, i_d\})}^1.$$

[1] $\overline{B} = \vec{1} - B = (B)^C$, the (bit-wise) complement of B.

Inspection of Individual Failing Tests

Failing test:

(1,	5,	1,	1.39,	0,	5,	10,	20)	|	Fail

Inspection of Individual Failing Tests

Failing test:

(1,	5,	1,	1.39,	0,	5,	10,	20)	|	Fail

2-separable matrix, resp. a $(1, 0^2)$-$LA(6; 2, 8)$:

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

Inspection of Individual Failing Tests

Failing test:

(1,	5,	1,	1.39,	0,	5,	10,	20)	Fail

2-separable matrix, resp. a $(1, 0^2)$-$LA(6; 2, 8)$:

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

Additionally derived tests:

(2,	5,	1,	1.79,	1,	5,	10,	20)
(2,	5,	1,	1.39,	0,	1,	15,	20)
(1,	6,	1,	1.79,	0,	1,	10,	20)
(1,	6,	1,	1.39,	1,	5,	15,	20)
(1,	5,	3,	1.39,	1,	1,	10,	20)
(1,	5,	3,	1.79,	0,	5,	15,	20)

Inspection of Individual Failing Tests

Failing test:

(1,	5,	1,	1.39,	0,	5,	10,	20)	Fail

2-separable matrix, resp. a $(1, 0^2)$-$LA(6; 2, 8)$:

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

Additionally derived tests:

(2,	5,	1,	1.79,	1,	5,	10,	20)	Fail
(2,	5,	1,	1.39,	0,	1,	15,	20)	Pass
(1,	6,	1,	1.79,	0,	1,	10,	20)	Pass
(1,	6,	1,	1.39,	1,	5,	15,	20)	Fail
(1,	5,	3,	1.39,	1,	1,	10,	20)	Pass
(1,	5,	3,	1.79,	0,	5,	15,	20)	Pass

Inspection of Individual Failing Tests

Failing test:

(1,	5,	1,	1.39,	0,	5,	10,	20)	Fail

2-separable matrix, resp. a $(1, 0^2)$-$LA(6; 2, 8)$:

$$\begin{pmatrix} 1, & 0, & 0, & 1, & 1, & 0, & 0, & 0 \\ 1, & 0, & 0, & 0, & 0, & 1, & 1, & 0 \\ 0, & 1, & 0, & 1, & 0, & 1, & 0, & 0 \\ 0, & 1, & 0, & 0, & 1, & 0, & 1, & 0 \\ 0, & 0, & 1, & 0, & 1, & 1, & 0, & 0 \\ 0, & 0, & 1, & 1, & 0, & 0, & 1, & 0 \end{pmatrix} \begin{array}{|c} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{array}$$

Additionally derived tests:

(2,	5,	1,	1.79,	1,	5,	10,	20)	Fail
(2,	5,	1,	1.39,	0,	1,	15,	20)	Pass
(1,	6,	1,	1.79,	0,	1,	10,	20)	Pass
(1,	6,	1,	1.39,	1,	5,	15,	20)	Fail
(1,	5,	3,	1.39,	1,	1,	10,	20)	Pass
(1,	5,	3,	1.79,	0,	5,	15,	20)	Pass

Inspection of Individual Failing Tests

Failing test:

(1,	5,	1,	1.39,	0,	5,	10,	20)	Fail

2-separable matrix, resp. a $(1, 0^2)$-$LA(6; 2, 8)$:

$$\begin{pmatrix} 1, & 0, & 0, & 1, & 1, & 0, & 0, & 0 \\ 1, & 0, & 0, & 0, & 0, & 1, & 1, & 0 \\ 0, & 1, & 0, & 1, & 0, & 1, & 0, & 0 \\ 0, & 1, & 0, & 0, & 1, & 0, & 1, & 0 \\ 0, & 0, & 1, & 0, & 1, & 1, & 0, & 0 \\ 0, & 0, & 1, & 1, & 0, & 0, & 1, & 0 \end{pmatrix} \begin{vmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{vmatrix}$$

Additionally derived tests:

(2,	5,	1,	1.79,	1,	5,	10,	20)	Fail
(2,	5,	1,	1.39,	0,	1,	15,	20)	Pass
(1,	6,	1,	1.79,	0,	1,	10,	20)	Pass
(1,	6,	1,	1.39,	1,	5,	15,	20)	Fail
(1,	5,	3,	1.39,	1,	1,	10,	20)	Pass
(1,	5,	3,	1.79,	0,	5,	15,	20)	Pass

Inspection of Individual Failing Tests

Failing test:

(1,	5,	1,	1.39,	0,	5,	10,	20)	Fail

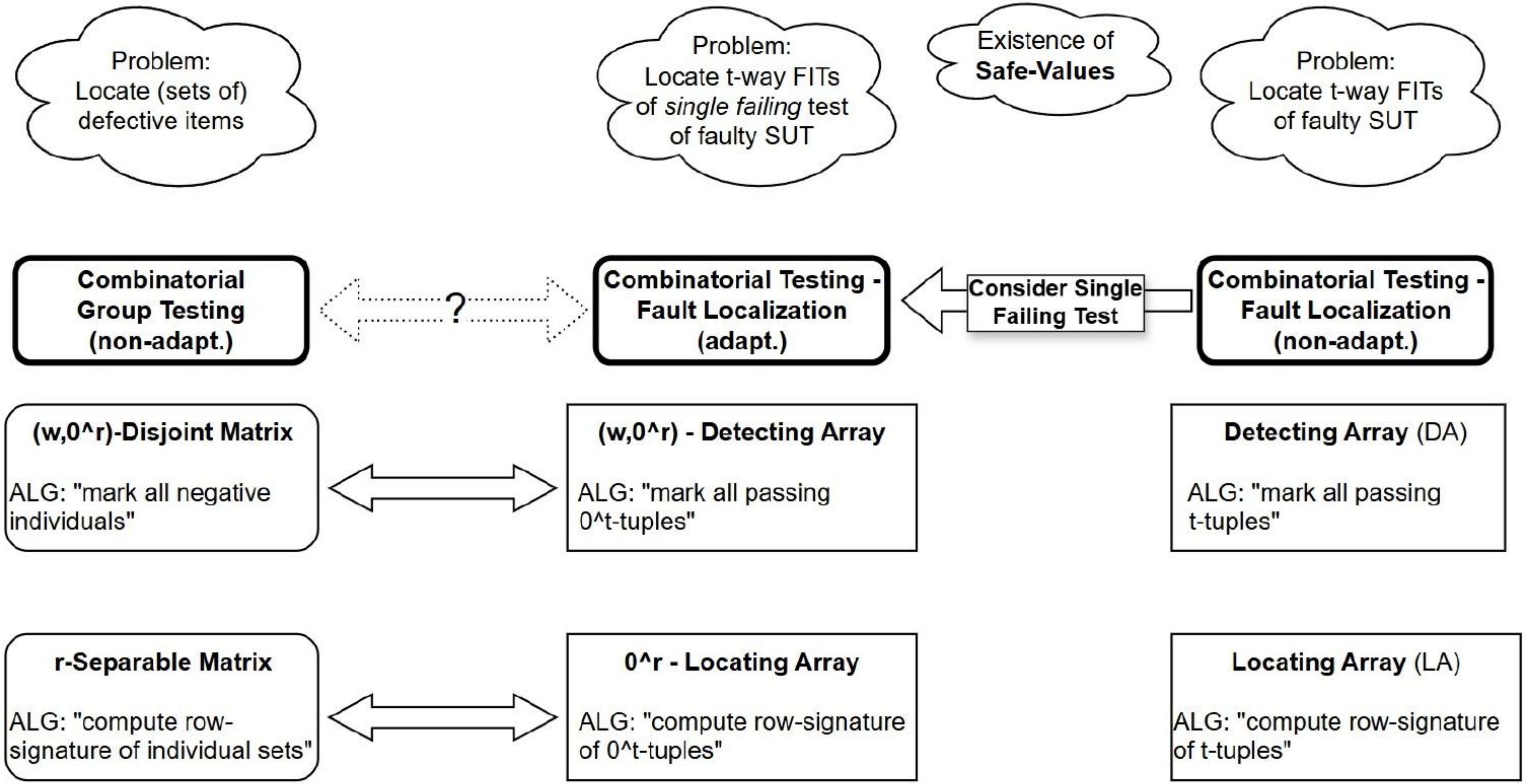
2-separable matrix, resp. a $(1,0^2)$-$LA(6;2,8)$:

$$\begin{pmatrix} 1, & 0, & {\color{red}0}, & 1, & 1, & {\color{red}0}, & 0, & 0 \\ 1, & 0, & 0, & 0, & 0, & 1, & 1, & 0 \\ 0, & 1, & 0, & 1, & 0, & 1, & 0, & 0 \\ 0, & 1, & {\color{red}0}, & 0, & 1, & {\color{red}0}, & 1, & 0 \\ 0, & 0, & 1, & 0, & 1, & 1, & 0, & 0 \\ 0, & 0, & 1, & 1, & 0, & 0, & 1, & 0 \end{pmatrix} \left| \begin{matrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{matrix} \right.$$

Additionally derived tests:

(2,	5,	1,	1.79,	1,	5,	10,	20)	Fail
(2,	5,	1,	1.39,	0,	1,	15,	20)	Pass
(1,	6,	1,	1.79,	0,	1,	10,	20)	Pass
(1,	6,	1,	1.39,	1,	5,	15,	20)	Fail
(1,	5,	3,	1.39,	1,	1,	10,	20)	Pass
(1,	5,	3,	1.79,	0,	5,	15,	20)	Pass
(*,	*,	1,	*,	*,	5,	*,	*)	$\Rightarrow$ Fail

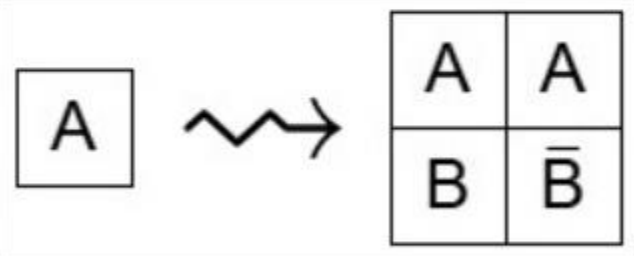
Combinatorial Group Testing and CT-FLA



A Roux-Type (Doubling) Construction for 0^t-LAs

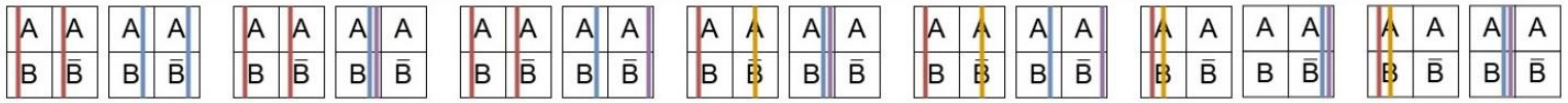
Construction

- **Input:** A $(1, 0^2)$-$LA(N; 2, k)$ A with N rows, strength 2 and k columns and a covering array $CA(M; 2, k, 2)$ B with M rows, all 2-way interactions covered and k columns
- **Output:** A $(1, 0^2)$-$LA(N + M; 2, 2k)$ with $N + M$ rows, strength 2 and $2k$ columns



Proof sketch

- Take two different arbitrary 0^2-way interactions τ, τ'
- Distinguish cases:

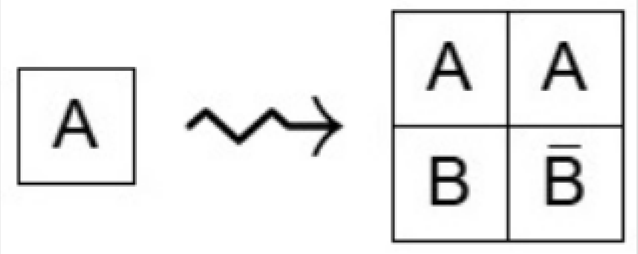


- Show for each case that τ and τ' cannot have the same row signature

A Roux-Type (Doubling) Construction for 0^t-LAs

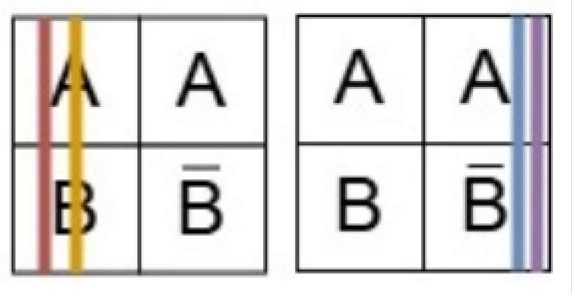
Construction

- **Input:** A $(1, 0^t)$-$LA(N; 2, k)$ A with N rows, strength 2 and k columns and a covering array $CA(M; 2, k, 2)$ B with M rows, all 2-way interactions covered and k columns
- **Output:** A $(1, 0^t)$-$LA(N + M; 2, 2k)$ with $N + M$ rows, strength 2 and $2k$ columns



Proof sketch for one case

- Let $\tau = \{p_1, p_2\}$ with $1 \leq p_1 < p_2 \leq k$ and $\tau' = \{p'_1, p'_2\}$ with $k + 1 \leq p_1 < p_2 \leq 2k$
- Assume τ and τ' are covered by exactly the same rows
- Then $\{p_1, p_2\} = \{p'_1 - k, p'_2 - k\}$ because A is a $(1, 0^2) - LA$.
- $\{(p_1, 0), (p_2, 0)\}$ is covered in some row r of B
- The row $r + N$ in the constructed array covers τ but not τ'



Conclusion and Outlook

Concluding Remarks

- Inspecting single failing tests in CT-FLA (with safe values!) coincides with combinatorial group testing
- d-separable matrices can be used for FIT localization via row-signatures
- d-disjunct matrices can be used for FIT localization marking t-way interactions as non-FIT

Outlook

- Generalized Roux-type constructions for d-separable and d-disjunct matrices
- Application in CT-FLA case study

Selected References

- C. J. Colbourn, D. W. McClary, *Locating and detecting arrays for interaction faults*, J Comb Optim (2008) 15: pp. 17–48.
- C. Martìnez, L. Moura, D. Panario, B. Stevens, *Locating Errors Using ELAs, Covering Arrays, and Adaptive Testing Algorithms* SIAM J. Discrete Math. 2009, Vol. 23, No. 4, pp. 1776–1799.
- J. Chodoriwsky, L. Moura, *An adaptive algorithm for group testing for complexes*, Theoretical Computer Science 592 (2015) pp. 1–8.
- D.R. Stinson, R. Wei, *Generalized cover-free families*, Discrete Mathematics 279 (2004) 463–477.
- H.-B. Chen, F. K. Hwang, *Exploring the missing link among d-separable, $\bar{d}$-separable and d-disjunct matrices*, DiscreteApplied Mathematics 155 (2007) 662–664.
- D. Du, F. K. Hwang, *Combinatorial group testing and its applications*, World Scientific, 2000.
- C. Shi, Y. Tang, J. Yin, *The equivalence between optimal detecting arrays and super-simple OAs*, Des. Codes Cryptogr. (2012) 62:131–142
- C. Shi, C. M. Wang, *Optimum Detecting Arrays for Independent Interaction Faults*, Acta Mathematica Sinica, English Series Feb., 2016, Vol. 32, No. 2, pp. 199–212.
- Seidel, Sarkar, Colbourn, Syrotiuk, *Separating Interaction Effects Using Locating and Detecting Arrays*, Springer Nature 2018 C. Iliopoulos et al. (Eds.): IWOCA 2018, LNCS 10979, pp. 349–360, 2018.
- C. J. Colbourn, V. Syrotiuk *Detecting arrays for effects of multiple interacting factors*, Information and Computation 301 (2024) 105202.

Thank you for your Attention!

Thank you for your Attention!

Questions? - Comments!