Locating arrays with error correcting ability

Masakazu Jimbo
joint work with Xiao-Nan Lu

†Chubu University

‡Tokyo University of Science

Dedicated to Professor Helleseth's 70-th birthday
MMC Workshop, September 7, 2017.

An example of interaction testing: Software testing

- Applications $F = \{f_1, f_2, \ldots, f_k\}$ are installed in a PC.
- Each application has two states S={1, 2}.
- Some state of some application (f, σ), $(\sigma \in S)$ or such a combination $\{(f_i, \sigma_i), (f_j, \sigma_j)\}$ may cause a "fault" in PC.
- A pair (f, σ) is called a 1-way interaction. A combination of pairs $\{(f_i, \sigma_i), (f_j, \sigma_j)\}$ is called a 2-way interaction.
- We want to find such faulty interactions by designing a testing array of testing suits.

	f_1	f_2	$\cdots$	f_k
t_1	1	2	$\cdots$	1
t_2	2	2	$\cdots$	2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
t_N	2	1	$\cdots$	2

An example of interaction testing: Software testing

- Applications $F = \{f_1, f_2, \ldots, f_k\}$ are installed in a PC.
- Each application has two states S=$\{1, 2\}$.
- Some state of some application (f, σ), $(\sigma \in S)$ or such a combination $\{(f_i, \sigma_i), (f_j, \sigma_j)\}$ may cause a "fault" in PC.
- A pair (f, σ) is called a 1-way interaction. A combination of pairs $\{(f_i, \sigma_i), (f_j, \sigma_j)\}$ is called a 2-way interaction.
- We want to find such faulty interactions by designing a testing array of testing suits.

	f_1	f_2	$\cdots$	f_k
t_1	1	2	$\cdots$	1
t_2	2	2	$\cdots$	2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
t_N	2	1	$\cdots$	2

An example of interaction testing: Software testing

- Applications $F = \{f_1, f_2, \ldots, f_k\}$ are installed in a PC.
- Each application has two states S=$\{1, 2\}$.
- Some state of some application (f, σ), $(\sigma \in S)$ or such a combination $\{(f_i, \sigma_i), (f_j, \sigma_j)\}$ may cause a “fault” in PC.
- A pair (f, σ) is called a 1-way interaction. A combination of pairs $\{(f_i, \sigma_i), (f_j, \sigma_j)\}$ is called a 2-way interaction.
- We want to find such faulty interactions by designing a testing array of testing suits.

	f_1	f_2	$\cdots$	f_k
t_1	1	2	$\cdots$	1
t_2	2	2	$\cdots$	2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
t_N	2	1	$\cdots$	2

An example of interaction testing: Software testing

- Applications $F = \{f_1, f_2, \ldots, f_k\}$ are installed in a PC.
- Each application has two states S=$\{1, 2\}$.
- Some state of some application (f, σ), $(\sigma \in S)$ or such a combination $\{(f_i, \sigma_i), (f_j, \sigma_j)\}$ may cause a "fault" in PC.
- A pair (f, σ) is called a 1-way interaction. A combination of pairs $\{(f_i, \sigma_i), (f_j, \sigma_j)\}$ is called a 2-way interaction.
- We want to find such faulty interactions by designing a testing array of testing suits.

	f_1	f_2	$\cdots$	f_k
t_1	1	2	$\cdots$	1
t_2	2	2	$\cdots$	2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
t_N	2	1	$\cdots$	2

Interaction testing: Terminologies

- Let $F = \{f_1, f_2, \ldots, f_k\}$ be the set of k **factors**.
- For each $f \in F$, let $S = \{1, 2, \ldots, s\}$ be the set of possible **levels** or **values**.
- A t-**way interaction** is a choice of a set K of t factors, and a selection of a value $\sigma_f \in S$ for each factor $f \in K$.

$$T = \{(f, \sigma_f) \mid f \in K\} \text{ with } K \in \binom{F}{t}, \sigma_f \in S$$

- A **test** is a k-tuple indexed by the factors, and the coordinate indexed by f has an entry in S.
- A **test suit** is a collection of tests.
- It is natural to use an $N \times k$ **array** $A = (a_{rf})$ to present a test suit consisting of N **tests** and k **factors**.

Interaction testing: Problems

Assumptions:

- Each test gives a result 0 (pass) or 1 (fail).
- Failures are caused by an i-way interaction with $i \le t$.

Problem:

- Is there an i-way interaction causing faults?
- Which are they?
- Given k and t, how many tests (N) are required?

Combinatorial testing arrays:

- Covering arrays
- Locating arrays
- Detecting arrays

Interaction testing arrays and a t-locating array

- Suppose $A = (a_{rf})$ is an $N \times k$ testing array.
- let K be a t-subset of the column indices of A.
- A t-way interaction $T = \{(f, \sigma_f) \mid f \in K\}$ appears in the r-th row $\Leftrightarrow$ $a_{rf} = \sigma_f$ for each $f \in K$.
- $\rho_A(T)$ consists of the rows indices r of A in which the t-way interaction T appears, namely

$$\rho_A(T) = \{r \mid a_{rf} = \sigma_f \text{ for each } f \in K\}.$$

How can we find faults?

- Let $\mathcal{T}$ be the set of i-way interactions for $i \leq t$. And assume that there is only one i-way interaction which causes failure in $\mathcal{T}$.
- An array A can detect any single failure in $\mathcal{T}$ iff $\rho_A(T)$'s are distinct for all T in $\mathcal{T}$.
- Such an array A is called a $\bar{t}$-locating array.

Interaction testing arrays and a t-locating array

- Suppose $A = (a_{rf})$ is an $N \times k$ testing array.
- let K be a t-subset of the column indices of A.
- A t-way interaction $T = \{(f, \sigma_f) \mid f \in K\}$ appears in the r-th row $\Leftrightarrow$ $a_{rf} = \sigma_f$ for each $f \in K$.
- $\rho_A(T)$ consists of the rows indices r of A in which the t-way interaction T appears, namely

$$\rho_A(T) = \{r \mid a_{rf} = \sigma_f \text{ for each } f \in K\}.$$

How can we find faults?

- Let $\mathcal{T}$ be the set of i-way interactions for $i \leq t$. And assume that there is only one i-way interaction which causes failure in $\mathcal{T}$.
- An array A can detect any single failure in $\mathcal{T}$ iff $\rho_A(T)$'s are distinct for all T in $\mathcal{T}$.
- Such an array A is called a $\bar{t}$-locating array.

Example: Non 1-locating array

An s-ary testing array with $N = 6$, $k = 9$ and $s = 3$ levels for each factor.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	outcome
t_1	1	3	3	3	2	2	1	3	2	1
t_2	2	1	3	3	3	2	2	1	1	1
t_3	2	2	1	3	3	3	1	2	3	0
t_4	3	2	2	1	3	3	3	1	3	0
t_5	3	3	2	2	1	3	2	3	1	0
t_6	3	3	3	2	2	1	3	2	2	0

- Assume there is at most **one** 1-way interaction causing faults.
- Outcome says that t_1, t_2 have the same value σ and t_3, t_4, t_5, t_6 are different from σ.
- Such a 1-way interaction is $(f_6, 2)$.
- $\{t_1, t_2\} = \rho((f_6, 2))$.

Example: Non 1-locating array

An s-ary testing array with $N = 6$, $k = 9$ and $s = 3$ levels for each factor.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	outcome
t_1	1	3	3	3	2	2	1	3	2	0
t_2	2	1	3	3	3	2	2	1	1	0
t_3	2	2	1	3	3	3	1	2	3	1
t_4	3	2	2	1	3	3	3	1	3	1
t_5	3	3	2	2	1	3	2	3	1	0
t_6	3	3	3	2	2	1	3	2	2	0

- Assume there is at most **one** 1-way interaction causing faults.
- Outcome says that t_3, t_4 have the same value σ and t_1, t_2, t_5, t_6 are different from σ.
- Such a 1-way interactions are $(f_2, 2)$ or $(f_9, 3)$. Which is faulty?
- $\{t_3, t_4\} = \rho((f_2, 2)) = \rho((f_9, 3))$.

Example: A locating array with strength 1

An s-ary testing array with $N = 6$, $k = 9$ and $s = 3$ levels for each factor.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	outcome
t_1	1	3	3	3	2	2	1	3	2	0
t_2	2	1	3	3	3	2	2	1	3	0
t_3	2	2	1	3	3	3	1	2	1	1
t_4	3	2	2	1	3	3	3	1	2	1
t_5	3	3	2	2	1	3	2	3	1	0
t_6	3	3	3	2	2	1	3	2	3	0

- Assume there is at most **one** 1-way interaction causing faults.
- Outcome says that t_3, t_4 have the same value σ and t_1, t_2, t_5, t_6 are different from σ.
- $(f_2, 2)$ is faulty.
- For any distinct $T = (f, \sigma)$ and $T' = (f', \sigma')$, $\rho(T) \neq \rho(T')$.

How can we check the locating array property?

A 3-ary **locating array** with
$t = 1$, $N = 6$, $k = 9$, and $s = 3$.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
t_1	1	3	3	3	2	2	1	3	2
t_2	2	1	3	3	3	2	2	1	3
t_3	2	2	1	3	3	3	1	2	1
t_4	3	2	2	1	3	3	3	1	2
t_5	3	3	2	2	1	3	2	3	1
t_6	3	3	3	2	2	1	3	2	3

For any distinct $T = (f, \sigma)$ and
$T' = (f', \sigma')$, $\rho(T) \neq \rho(T')$.

All supports for $1, 2, 3$ are distinct.

	1	2	3
f_1	$\{1\}$	$\{2,3\}$	$\{4,5,6\}$
f_2	$\{2\}$	$\{3,4\}$	$\{1,5,6\}$
f_3	$\{3\}$	$\{4,5\}$	$\{1,2,6\}$
f_4	$\{4\}$	$\{5,6\}$	$\{1,2,3\}$
f_5	$\{5\}$	$\{1,6\}$	$\{2,3,4\}$
f_6	$\{6\}$	$\{1,2\}$	$\{3,4,5\}$
f_7	$\{1,3\}$	$\{2,5\}$	$\{4,6\}$
f_8	$\{2,4\}$	$\{3,6\}$	$\{1,5\}$
f_9	$\{3,5\}$	$\{1,4\}$	$\{2,6\}$

The above is regarded as **a spread system** with 9 spreads on 6 points with 3 parts

How can we check the locating array property?

A 3-ary **locating array** with $t = 1$, $N = 6$, $k = 9$, and $s = 3$.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
t_1	1	3	3	3	2	2	1	3	2
t_2	2	1	3	3	3	2	2	1	3
t_3	2	2	1	3	3	3	1	2	1
t_4	3	2	2	1	3	3	3	1	2
t_5	3	3	2	2	1	3	2	3	1
t_6	3	3	3	2	2	1	3	2	3

For any distinct $T = (f, \sigma)$ and $T' = (f', \sigma')$, $\rho(T) \neq \rho(T')$.

All supports for $1, 2, 3$ are distinct.

	1	2	3
f_1	$\{1\}$	$\{2, 3\}$	$\{4, 5, 6\}$
f_2	$\{2\}$	$\{3, 4\}$	$\{1, 5, 6\}$
f_3	$\{3\}$	$\{4, 5\}$	$\{1, 2, 6\}$
f_4	$\{4\}$	$\{5, 6\}$	$\{1, 2, 3\}$
f_5	$\{5\}$	$\{1, 6\}$	$\{2, 3, 4\}$
f_6	$\{6\}$	$\{1, 2\}$	$\{3, 4, 5\}$
f_7	$\{1, 3\}$	$\{2, 5\}$	$\{4, 6\}$
f_8	$\{2, 4\}$	$\{3, 6\}$	$\{1, 5\}$
f_9	$\{3, 5\}$	$\{1, 4\}$	$\{2, 6\}$

The above is regarded as **a spread system** with 9 spreads on 6 points with 3 parts

A binary locating array of strength $\bar{t} = 2 (t \leq 2)$

A binary $\bar{t}$-**locating array** with $N = 11$, $k = 6$.

$$
\begin{bmatrix}
 & f_0 & f_1 & f_2 & f_3 & f_4 & f_5 \\
t_0 & 1 & 1 & 2 & 1 & 1 & 1 \\
t_1 & 2 & 1 & 1 & 2 & 1 & 1 \\
t_2 & 1 & 2 & 1 & 1 & 2 & 1 \\
t_3 & 2 & 1 & 2 & 1 & 1 & 2 \\
t_4 & 2 & 2 & 1 & 2 & 1 & 1 \\
t_5 & 2 & 2 & 2 & 1 & 2 & 1 \\
t_6 & 1 & 2 & 2 & 2 & 1 & 2 \\
t_7 & 1 & 1 & 2 & 2 & 2 & 1 \\
t_8 & 1 & 1 & 1 & 2 & 2 & 2 \\
t_9 & 2 & 1 & 1 & 1 & 2 & 2 \\
t_{10} & 1 & 2 & 1 & 1 & 1 & 2
\end{bmatrix}
$$

For any distinct 2-way interactions $T = \{(f_1, \sigma_1), (f_2, \sigma_2)\}$, $\rho(T)$ are distinct.

	(1, 1)	(1, 2)	(2, 1)	(2, 2)
(f_0, f_1)	$\{0, 7, 8\}$	$\{2, 6, 10\}$	$\{1, 3, 9\}$	$\{4, 5\}$
(f_0, f_2)	$\{2, 8, 10\}$	$\{0, 6, 7\}$	$\{1, 4, 9\}$	$\{3, 5\}$
(f_0, f_3)	$\{0, 2, 10\}$	$\{6, 7, 8\}$	$\{3, 5, 9\}$	$\{1, 4\}$
(f_0, f_4)	$\{0, 6, 10\}$	$\{2, 7, 8\}$	$\{1, 3, 4\}$	$\{5, 9\}$
(f_0, f_5)	$\{0, 2, 7\}$	$\{6, 8, 10\}$	$\{1, 4, 5\}$	$\{3, 9\}$
(f_1, f_2)	$\{1, 8, 9\}$	$\{0, 3, 7\}$	$\{2, 4, 10\}$	$\{5, 6\}$
(f_1, f_3)	$\{0, 3, 9\}$	$\{1, 7, 8\}$	$\{2, 5, 10\}$	$\{4, 6\}$
(f_1, f_4)	$\{0, 1, 3\}$	$\{7, 8, 9\}$	$\{4, 6, 10\}$	$\{2, 5\}$
(f_1, f_5)	$\{0, 1, 7\}$	$\{3, 8, 9\}$	$\{2, 4, 5\}$	$\{6, 10\}$
(f_2, f_3)	$\{2, 9, 10\}$	$\{1, 4, 8\}$	$\{0, 3, 5\}$	$\{6, 7\}$
(f_2, f_4)	$\{1, 4, 10\}$	$\{2, 8, 9\}$	$\{0, 3, 6\}$	$\{5, 7\}$
(f_2, f_5)	$\{1, 2, 4\}$	$\{8, 9, 10\}$	$\{0, 5, 7\}$	$\{3, 6\}$
(f_3, f_4)	$\{0, 3, 10\}$	$\{2, 5, 9\}$	$\{1, 4, 6\}$	$\{7, 8\}$
(f_3, f_5)	$\{0, 2, 5\}$	$\{3, 9, 10\}$	$\{1, 4, 7\}$	$\{6, 8\}$
(f_4, f_5)	$\{0, 1, 4\}$	$\{3, 6, 10\}$	$\{2, 5, 7\}$	$\{8, 9\}$

A binary locating array of strength $\bar{t} = 2 (t \le 2)$

A binary $\bar{t}$-**locating array** with $N = 11$, $k = 6$.

$$\begin{bmatrix} & f_0 & f_1 & f_2 & f_3 & f_4 & f_5 \\ t_0 & 1 & 1 & 2 & 1 & 1 & 1 \\ t_1 & 2 & 1 & 1 & 2 & 1 & 1 \\ t_2 & 1 & 2 & 1 & 1 & 2 & 1 \\ t_3 & 2 & 1 & 2 & 1 & 1 & 2 \\ t_4 & 2 & 2 & 1 & 2 & 1 & 1 \\ t_5 & 2 & 2 & 2 & 1 & 2 & 1 \\ t_6 & 1 & 2 & 2 & 2 & 1 & 2 \\ t_7 & 1 & 1 & 2 & 2 & 2 & 1 \\ t_8 & 1 & 1 & 1 & 2 & 2 & 2 \\ t_9 & 2 & 1 & 1 & 1 & 2 & 2 \\ t_{10} & 1 & 2 & 1 & 1 & 1 & 2 \end{bmatrix}$$

	1	2
f_0	$\{0, 2, 6, 7, 8, 10\}$	$\{1, 3, 4, 5, 9\}$
f_1	$\{0, 1, 3, 7, 8, 9\}$	$\{2, 4, 5, 6, 10\}$
f_2	$\{1, 2, 4, 8, 9, 10\}$	$\{0, 3, 5, 6, 7\}$
f_3	$\{0, 2, 3, 5, 9, 10\}$	$\{1, 4, 6, 7, 8\}$
f_4	$\{0, 1, 3, 46, 10\}$	$\{2, 5, 7, 8, 9\}$
f_5	$\{0, 1, 2, 4, 5, 7\}$	$\{3, 6, 8, 9, 10\}$

For any distinct $1, 2$-way interactions, $\rho(T)$'s are distinct.

If error-correction is taken in mind...

- Assumption: (1) Faults occur only in 1-way interactions and (2) at most one error may happen in the outcome vector.
- The outcome is $\{1, 4\}$, which does not fit any supports of 1-way interactions.
- $\{1, 2, 4\}$ is the nearest among all supports of 1-way interactions.
- Hence, we find $T = (f_1, 1)$ is faulty 1-way interaction.
- Actually, the minimum distance of these supports are 3, Hence, all 1-way interactions can be detected even if there is at most one error in the outcome.

A locating array with strength 1

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	outcome
t_1	1	2	2	2	1	2	1	1
t_2	1	1	2	2	2	1	2	0
t_3	2	1	1	2	2	2	1	0
t_4	1	2	1	1	2	2	2	1
t_5	2	1	2	1	1	2	2	0
t_6	2	2	1	2	1	1	2	0
t_7	2	2	2	1	2	1	1	0

	1	2
f_1	$\{1, 2, 4\}$	$\{3, 5, 6, 7\}$
f_2	$\{2, 3, 5\}$	$\{1, 4, 6, 7\}$
f_3	$\{3, 4, 6\}$	$\{1, 2, 5, 7\}$
f_4	$\{4, 5, 7\}$	$\{1, 2, 3, 6\}$
f_5	$\{1, 5, 6\}$	$\{2, 3, 4, 7\}$
f_6	$\{2, 6, 7\}$	$\{1, 3, 4, 5\}$
f_7	$\{1, 3, 7\}$	$\{2, 4, 5, 6\}$

If error-correction is taken in mind...

- Assumption: (1) Faults occur only in 1-way interactions and (2) at most one error may happen in the outcome vector.
- The outcome is $\{1, 4\}$, which does not fit any supports of 1-way interactions.
- $\{1, 2, 4\}$ is the nearest among all supports of 1-way interactions.
- Hence, we find $T = (f_1, 1)$ is faulty 1-way interaction.
- Actually, the minimum distance of these supports are 3, Hence, all 1-way interactions can be detected even if there is at most one error in the outcome.

A locating array with strength 1

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	outcome
t_1	1	2	2	2	1	2	1	1
t_2	1	1	2	2	2	1	2	0
t_3	2	1	1	2	2	2	1	0
t_4	1	2	1	1	2	2	2	1
t_5	2	1	2	1	1	2	2	0
t_6	2	2	1	2	1	1	2	0
t_7	2	2	2	1	2	1	1	0

	1	2
f_1	$\{1, 2, 4\}$	$\{3, 5, 6, 7\}$
f_2	$\{2, 3, 5\}$	$\{1, 4, 6, 7\}$
f_3	$\{3, 4, 6\}$	$\{1, 2, 5, 7\}$
f_4	$\{4, 5, 7\}$	$\{1, 2, 3, 6\}$
f_5	$\{1, 5, 6\}$	$\{2, 3, 4, 7\}$
f_6	$\{2, 6, 7\}$	$\{1, 3, 4, 5\}$
f_7	$\{1, 3, 7\}$	$\{2, 4, 5, 6\}$

If error-correction is taken in mind...

- Assumption: (1) Faults occur only in 1-way interactions and (2) at most one error may happen in the outcome vector.
- The outcome is $\{1, 4\}$, which does not fit any supports of 1-way interactions.
- $\{1, 2, 4\}$ is the nearest among all supports of 1-way interactions.
- Hence, we find $T = (f_1, 1)$ is faulty 1-way interaction.
- Actually, the minimum distance of these supports are 3, Hence, all 1-way interactions can be detected even if there is at most one error in the outcome.

A locating array with strength 1

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	outcome
t_1	1	2	2	2	1	2	1	1
t_2	1	1	2	2	2	1	2	0
t_3	2	1	1	2	2	2	1	0
t_4	1	2	1	1	2	2	2	1
t_5	2	1	2	1	1	2	2	0
t_6	2	2	1	2	1	1	2	0
t_7	2	2	2	1	2	1	1	0

	1	2
f_1	$\{1, 2, 4\}$	$\{3, 5, 6, 7\}$
f_2	$\{2, 3, 5\}$	$\{1, 4, 6, 7\}$
f_3	$\{3, 4, 6\}$	$\{1, 2, 5, 7\}$
f_4	$\{4, 5, 7\}$	$\{1, 2, 3, 6\}$
f_5	$\{1, 5, 6\}$	$\{2, 3, 4, 7\}$
f_6	$\{2, 6, 7\}$	$\{1, 3, 4, 5\}$
f_7	$\{1, 3, 7\}$	$\{2, 4, 5, 6\}$

If error-correction is taken in mind...

- If a t-locating array tolerates e errors, namely, even if the outcome has **at most** e **errors**, the t-locating property still holds, then the array is called e-**error correcting** t-**locating array** or a (t, e)-**locating array**.

Known bound on N or k when $e = 0$

- For a $(\bar{t}, e)$-locating array with s levels and k factors, let $\mathrm{LAN}_{(\bar{t},e)}(k, s)$ be the minimum number N of tests (rows).
- For a $(\bar{t}, e)$-locating array with s levels and N tests, let $\mathrm{LAk}_{(\bar{t},e)}(N, s)$ be the maximum number k of factors (columns).

Problem 1

Given k, s, t and e, find the value of $\mathrm{LAN}_{(t,e)}(k, s)$. Or, given N, s, t and e, find the value of $\mathrm{LAk}_{(\bar{t},e)}(N, s)$.

- Not many has been known for such bounds even when $e = 0$.

Known bound on N or k when $e = 0$

- For a $(\bar{t}, e)$-locating array with s levels and k factors, let $\mathrm{LAN}_{(\bar{t},e)}(k, s)$ be the minimum number N of tests (rows).
- For a $(\bar{t}, e)$-locating array with s levels and N tests, let $\mathrm{LAk}_{(\bar{t},e)}(N, s)$ be the maximum number k of factors (columns).

Problem 1

Given k, s, t and e, find the value of $\mathrm{LAN}_{(t,e)}(k, s)$. Or, given N, s, t and e, find the value of $\mathrm{LAk}_{(\bar{t},e)}(N, s)$.

- Not many has been known for such bounds even when $e = 0$.

Lower bounds on N for $e = 0$

Theorem 1 (Tang, Colbourn and Yin(2012))

- *For $k \geq t \geq 2$ and $s \geq 2$,*

$$\mathrm{LAN}_{(t,0)}(k,s) \geq \left\lceil \frac{2\binom{k}{t}s^t}{1+\binom{k}{t}} \right\rceil .$$

- *For $s \geq t \geq 2$,*

$$\mathrm{LAN}_{(t,0)}(k,s) \geq \left\lceil -\frac{3}{2} - \binom{k}{t} + \sqrt{\binom{k}{t}^2 + (3+6s^t)\binom{k}{t} + \frac{9}{4}} \right\rceil .$$

Theorem 2 (A simple bound)

$$\mathrm{LAN}_{(t,0)}(k,s) \geq \left\lceil \log_2 s \left(t + \log_s \binom{k}{t} \right) \right\rceil .$$

When k is large, the simple bound is much better than T-C-Y bounds.

An improvement of the lower bounds on N

The bound below is an improvement of Tang-Colbourn-Yin's bounds.

Theorem 3 (An improved bound)

For any given k, s, N and t, we fix an integer $\tau > 1$ arbitirary. Then a lower bound for N satisfies

$$\sum_{\ell=1}^{\tau-1}(\tau-\ell)\binom{N}{\ell} \geq \binom{k}{t}(\tau s^t - N).$$

T-C-Y's bound can be derived by setting $\tau = 2, 3$.

A lower bounds on N with e-correcting ability

Theorem 4 (An improved bound)

For any given k, s, t, and e,

$$2^N \geq \left(\sum_{i=0}^{e} \binom{N}{i} \right) \binom{k}{t} s^t$$

holds. Especially, for $e = 0$, we have

$$N \geq \left\lceil \log_2 s \left(t + \log_s \binom{k}{t} \right) \right\rceil .$$

Even when $e = 0$, no construction attaininng the bound is known.

Upper bounds $\mathsf{LAk}_{(1,e)}(N,s)$ and an optimal binary $(1,1)$-locating array

- Let $A(n,d)$ denote the maximum possible size of a binary code $\mathcal{C}$ of length n and Hamming distance d.

Proposition (Hamming bound and Johnson bound)

- $$\mathsf{LAk}_{(t,e)}(N,s) \leq \frac{A(N,2e+1)}{s} \leq \frac{2^N}{s^t \sum_{\ell=0}^{e} \binom{N}{\ell}}.$$

- For $t=e=1$, recall that $\mathsf{LAk}_{(1,1)}(N,2) \leq \frac{2^N}{2(N+1)}$.
- A $(1,1)$-locating array generated from the $[N=2^m-1, 2^m-m-1, 3]$ Hamming code attains the above bound.

Theorem 5

$\mathsf{LAk}_{(1,1)}(2^m-1,2) = 2^{2^m-m-2}$ *for any* $m \geq 3$.

A construction of (t, e)-locating arrays with $N = O(k)$ for $t \geq 2$

- Some other optimal $(1, e)$-locating arrays can be derived from affine geometry.

- Now, we consider the case of $t \geq 2$. As stated before no constructions with $N \leq O(k)$ are known.
- We will derive such construction by utilizing Payley type matrices.

A construction of (t, e)-locating arrays with $N = O(k)$ for $t \geq 2$

- Some other optimal $(1, e)$-locating arrays can be derived from affine geometry.

- Now, we consider the case of $t \geq 2$. As stated before no constructions with $N \leq O(k)$ are known.
- We will derive such construction by utilizing Payley type matrices.

A Paley type locating array

Example 6 (A Paley matrix of order 11)

−	0	1	2	3	4	5	6	7	8	9	10
0	1	1	2	1	1	1	2	2	2	1	2
1	2	1	1	2	1	1	1	2	2	2	1
2	1	2	1	1	2	1	1	1	2	2	2
3	2	1	2	1	1	2	1	1	1	2	2
4	2	2	1	2	1	1	2	1	1	1	2
5	2	2	2	1	2	1	1	2	1	1	1
6	1	2	2	2	1	2	1	1	2	1	1
7	1	1	2	2	2	1	2	1	1	2	1
8	1	1	1	2	2	2	1	2	1	1	2
9	2	1	1	1	2	2	2	1	2	1	1
10	1	2	1	1	1	2	2	2	1	2	1

2: square,
1: non-square or 0
in $\mathbb{F}_{11}$

We correspond each row to a test ($N = 11$) and each column to a factor ($k = 11$). Then the array is a binary 2-locating array.

Paley type s-ary (t, e)-locating arrays

- Let $\mathbb{F}_q$ be the finite fields of order q. (Consider an array with $N = k = q$.)
- Let χ_s be a primitive multiplicative character of order s on $\mathbb{F}_q$ and ζ_s be a primitive s-th root of 1 in $\mathbb{C}$.
- We define a $q \times q$ array $A = (a_{xy})$ $(x, y \in \mathbb{F}_q)$ by

$$a_{xy} = \begin{cases} 0 & \text{if } x = y, \\ i & \text{if } \chi_s(x - y) = \zeta_s^i \end{cases}$$

Then the following can be shown by utilizing some number theoretic technique including Weil's theorem.

Theorem 7

For any given s, t and e, a $q \times q$ array A is an s-ary (t, e)-LA if q is large enough.

Examples of Paley type binary (t, e)-locating arrays

Theorem 8

Let $q \equiv 3 \pmod 4$ be a prime power. Then, for $s = 2$,

- *A is a binary $(1, e)$-LA with $e = \frac{q-7}{4}$ if $q \geq 7$.*
- *A is a binary $(2, e)$-LA with $e = \frac{3q-10\sqrt{q}-43}{16}$ if $q \geq 11$.*
- *A is a binary $(3, e)$-LA with $e = \frac{7q-114\sqrt{q}-215}{64}$ if $q > 293$.*

Theorem 9

Let $q \equiv 1 \pmod 4$ be a prime power. Then, for $s = 2$,

- *A is a binary $(1, e)$-LA with $e = \frac{q-11}{4}$ if $q > 11$.*
- *A is a binary $(2, e)$-LA with $e = \frac{3q-10\sqrt{q}-91}{16}$ if $q > 51$.*
- *A is a binary $(3, e)$-LA with $e = \frac{7q-114\sqrt{q}-535}{64}$ if $q > 370$.*

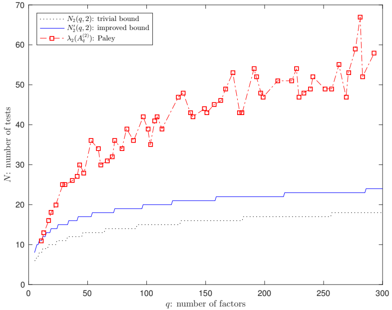
Remark to Paley type locating arrays and truncation of rows

- A Paley type matrix A can generate an s-ary (t, e)-locating array with $N = k = q$ if q is sufficiently large.
- A Paley matrix is utilized to construct a Hadamard matrix and a Hadamard matrix is an orthogonal array of strength 2 (not strength t).
- Constructions of t-locating arrays with $N \le O(k)$ is not known for general t even if $e = 0$ except for our construction, which is $N = k$.
- But known lower bound $\mathrm{LAN}_{(t,0)}(k, s)$ is $N \ge O(\log k)$. Our construction requires $N = O(k)$. It is not known whether $N = O(\log k)$ can be attained, or not.
- We try to truncate rows from our Paley type locating array to reduce the number of tests without loosing the property of a t-locating array. (Here, we do not care the error correcting ability e.)

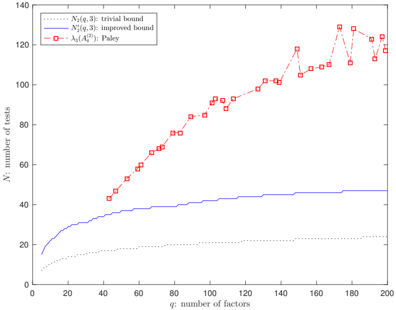
Remark to Paley type locating arrays and truncation of rows

- A Paley type matrix A can generate an s-ary (t, e)-locating array with $N = k = q$ if q is sufficiently large.
- A Paley matrix is utilized to construct a Hadamard matrix and a Hadamard matrix is an orthogonal array of strength 2 (not strength t).
- Constructions of t-locating arrays with $N \leq O(k)$ is not known for general t even if $e = 0$ except for our construction, which is $N = k$.
- But known lower bound $\mathrm{LAN}_{(t,0)}(k, s)$ is $N \geq O(\log k)$. Our construction requires $N = O(k)$. It is not known whether $N = O(\log k)$ can be attained, or not.
- We try to truncate rows from our Paley type locating array to reduce the number of tests without loosing the property of a t-locating array. (Here, we do not care the error correcting ability e.)

Truncation of rows from Paley type locating arrays



$t = 2$



$t = 3$

More research are required for locating arrays!!!

Takk.

Gratulere med dagen Professor Helleseths sin 70 Ärsdag!