Fault Location and Resolvable Set Systems

Charles J. Colbourn, colbourn@asu.edu
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

School of Computing, Informatics, and Decision Systems Engineering
Arizona State University

ACA 2015

(Interaction) Testing

- Let $\mathcal{F} = \{F_1, \ldots, F_k\}$ be a set of k *factors*.
- For each $F_f \in \mathcal{F}$ let V_f be the set of possible *levels* or *values* for factor F_f.
- A *t-way interaction* is a set F of t factors, and a value $\nu_f \in V_f$ for each factor $F_f \in F$. The parameter t is the *strength*; we assume that $t \leq k$.
- A *test* is a k-tuple indexed by the factors, so that the coordinate indexed by F_i contains an entry of V_i.
- A *test suite* is a collection of tests; when there are N tests, it is natural to write this as an $N \times k$ array.

Charles J. Colbourn, colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

(Interaction) Testing

- When tests are run, each may *pass* or *fail*.
- We suppose that failures are caused by *s*-way interactions with $s \leq t$.
- And our goal is to first determine whether there are any *s*-way interactions with $s \leq t$ causing faults, and if so to determine the interactions that cause the faults.

(Interaction) Testing

Example

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	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	1	2	2	2	2	2	2	2	2	2	2	2
0	1	1	1	1	0	1	2	2	0	0	0	0	1	1	1	2	2	2	2
0	2	2	2	2	2	2	0	1	0	0	0	0	1	2	2	0	1	1	1
1	0	1	1	1	2	2	0	1	0	1	1	2	0	0	1	1	0	1	2
1	1	2	2	2	1	0	1	0	2	1	1	0	0	2	1	2	2	1	0
1	2	0	1	2	0	2	1	0	2	0	2	2	1	0	2	1	0	2	1
1	2	1	0	2	2	1	2	0	1	2	1	1	0	1	2	0	2	0	1
1	2	1	2	0	2	1	1	2	2	1	0	1	2	0	0	2	1	0	0
2	0	2	2	2	0	1	2	2	1	2	2	0	2	2	0	1	0	1	2
2	1	0	2	1	2	0	2	2	2	1	2	2	0	1	2	0	1	2	0
2	1	2	0	1	1	2	0	2	1	0	1	1	2	1	0	2	0	2	1
2	1	2	1	0	1	2	2	1	1	2	0	2	1	0	0	1	2	0	0
2	2	1	1	1	1	0	1	0	0	2	2	1	2	2	1	0	1	0	2

Occurrences of Interactions

Charles J. Colbourn, colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- Let A be an $N \times k$ array forming a test suite for factors $\mathcal{F} = \{F_1, \ldots, F_k\}$, indexing the rows by $\{1, \ldots, N\}$.
- The t-way interaction $T = \{(f_i, \nu_i) : 1 \leq i \leq t\}$ *appears in* row j if, for $1 \leq i \leq t$, the entry of A in row j, column f_i is symbol ν_i.
- Let $\rho(T)$ be the set of row indices of A in which interaction T appears.

Covering Arrays

- If interaction T causes a failure, test suite A can detect it only if $\rho(T) \neq \emptyset$ (and under our assumptions, it will detect it if $\rho(T) \neq \emptyset$).
- A *covering array* of *strength t* is a test suite so that $\rho(T) \neq \emptyset$ whenever T is a t-way interaction.
- There is a huge literature on covering array construction!
- If some interaction of strength at most t causes a fault, at least one test in the covering array must fail, so we can certify the presence of a fault (or the absence of any fault caused by such an interaction).
- But can we find the faults?

Locating Arrays

- Let $\mathcal{T}$ be a set of interactions. Define
$$\rho(\mathcal{T}) = \bigcup_{T \in \mathcal{T}} \rho(T)$$
- If the set of interactions each causing failure is precisely $\mathcal{T}$, then the set of tests that fail is precisely $\rho(\mathcal{T})$.
- We would like to use the set of tests in which failure has occurred to determine the set of interactions causing failures.

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

Locating Arrays

- A $(\overline{d}, t)$-*locating array* is a test suite in which whenever $\mathcal{T}_1$ and $\mathcal{T}_2$ are sets, each containing at most d t-way interactions,

$$\rho(\mathcal{T}_1) = \rho(\mathcal{T}_2) \text{ if and only if } \mathcal{T}_1 = \mathcal{T}_2$$

- We limit *a priori* the number of interactions causing faults, because if this number is arbitrary, location cannot be accomplished even using all possible tests.
- In contrast with covering arrays, very little is known about locating arrays.

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

Locating Arrays

- How can we use a locating array to determine the faulty interactions?
- We can start by looking at all tests that pass, and declaring every interaction in each as not faulty. However, we cannot be sure in general that the interactions that remain are in fact the faulty ones.
- So we could ask for a stronger condition that would ensure that faulty interactions are precisely those that do not appear in any test that passes.

Detecting Arrays

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- A $(\overline{d}, t)$-*detecting array* is a test suite in which whenever $\mathcal{T}_1$ is a set containing at most d t-way interactions, and T is a t-way interaction,

$$\rho(T) \subseteq \rho(\mathcal{T}_1) \text{ if and only if } T \in \mathcal{T}_1$$

Locating Arrays

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- Because little is known about locating and detecting arrays, it makes sense to start at the beginning.
- For covering arrays, the story begins at strength $t = 2$, because covering arrays of strength 1 are trivial – just pick tests so that every value of every factor appears at least once.
- But what about locating and detecting arrays?

Locating Arrays

- Even for strength 1, the number of tests needed is unknown.
- We focus on the case when every factor has the same number of values, and in which there is at most one "interaction" causing failure.
- We recast the problem in somewhat different language.
- In an $N \times k$ test suite on v symbols, every *column* is a partition of $\{1, \ldots, N\}$ into v parts; the parts correspond exactly to the $\rho()$s for the v values for this factor.

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

Resolvable Set Systems

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- So looking at the partitions arising from the columns, we get k parallel classes of sets (partitions) on $\{1, \ldots, N\}$, each having v sets (parts).
- What does $(\overline{1}, 1)$-detecting mean? No set contains another. This leads to the *Sperner partition systems* studied by Meagher and Li.
- What does $(\overline{1}, 1)$-locating mean? No two sets are equal. This appears not to have been studied!

Resolvable Set Systems

- So, in our new language, here is the problem: Let $(S_{ij} : 1 \leq i \leq k, 1 \leq j \leq v)$ be subsets of $\{1, \ldots, N\}$ so that for every $1 \leq i \leq k$, the sets $(S_{ij} : 1 \leq j \leq v)$ partition $\{1, \ldots, N\}$, and $S_{ij} = S_{i'j'}$ only if $i = i'$ and $j = j'$.
- Call such a collection of sets an (N, v)-*disjoint set partition*.
- We are to determine the largest value of k in an (N, v)-disjoint set partition.

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

The Easy Case: $v = 2$

- If, for every set X with $\{1\} \subseteq X \subseteq \{1, \ldots, N\}$, we form the partition $(X, \{1, \ldots, N\} \setminus X)$, then no two such partitions can share a set.
- So it is easy to make 2^{N-1} partitions.
- And, because there are 2^N sets, with two in each partition, no more than 2^{N-1} partitions can be made.
- So the *exact* answer when $v = 2$ is 2^{N-1}.

Charles J. Colbourn, colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

Necessary Conditions

Set Partitions Yield Integer Partitions

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

$$S_{i1} \quad \cdots \quad S_{iv} \quad \Rightarrow \qquad |S_{i1}| \quad \cdots \quad |S_{iv}|$$

- Call a multiset of integer partitions of N each into v (nonegative integer) parts (N, v)*-admissible* if the total number of parts equal to ℓ is at most $\binom{N}{\ell}$ for every $0 \leq \ell \leq N$
- What is the largest number of partitions in an (N, v)-admissible set?
- This is a basic upper bound on the number of partitions in the *set* partition as well.

Integer Partitions

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- Formulate a “simple” linear program.
  - Variables correspond to (all possible) integer partitions of N with v parts each, and are constrained to be nonnegative.
  - There is an inequality for each part size ℓ saying that the number of parts of size ℓ does not exceed $\binom{N}{\ell}$.
  - Maximize the sum of variables.

Integer Partitions

Greedy

- Inspection of the structure of solutions leads to a fairly simple characterization of the maximum when $N \not\equiv v - 1 \pmod{v}$:
  - for $0 \leq \ell < \lfloor \frac{N}{v} \rfloor$, take $\binom{N}{\ell}$ partitions with one part of ℓ and $v - 1$ parts equal to $\lfloor \frac{N-\ell}{v-1} \rfloor$ or $\lceil \frac{N-\ell}{v-1} \rceil$.
  - Now take partitions with all parts $\lfloor \frac{N}{v} \rfloor$ or $\lceil \frac{N}{v} \rceil$ to use as many parts of size $\lfloor \frac{N}{v} \rfloor$ as possible.

  Messy calculations with binomial coefficients verify that this is feasible for all $\lfloor \frac{N}{v} \rfloor < \ell \leq N$, and this gives the maximum number of (integer) partitions.

Charles J. Colbourn, colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

Sufficient Conditions

Integer Partitions Yield Set Partitions

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- Given an (N, v)-admissible set of k integer partitions $\{(n_{i1}, \ldots, n_{iv} : 1 \leq i \leq v\}$, can we form an (N, v)-disjoint set partition having k set partitions?
- How can we turn the set sizes into the actual sets, so that no set is used twice?

Baranyai's Theorem

Charles J. Colbourn, colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- The $(\ell v, v)$-admissible set of $\binom{\ell v}{\ell}/v$ integer partitions each equal to $(\ell, \ldots, \ell)$ yields an $(\ell v, v)$-disjoint set partition having $\binom{\ell v}{\ell}/v$ set partitions.
- Key things used in the proof:
  - All sets used have the same size.
  - All sets of size ℓ are used, each exactly once.
- So Baranyai's result is a very very special case of what we want.

Brouwer-Schrijver proof: Baranyai's Theorem

- Idea: We build up the sets one element at a time, writing $N = \ell v$
- We start with $\binom{N}{\ell}/v$ set partitions of $\emptyset$ into parts $(\emptyset, \ldots, \emptyset)$.
- For σ from 1 to N in turn, each set partition $(S_1, \ldots, S_v)$ is a set partition of $\{1, \ldots, \alpha - 1\}$.
- We will determine to which of $S_1, \ldots, S_v$ we add symbol σ.
- Before iteration σ, every j-subset of $\{1, \ldots, \sigma - 1\}$ must appear as a part exactly $\binom{N-\alpha-1}{\ell-j}$ times.
- After iteration σ, every j-subset of $\{1, \ldots, \sigma\}$ must appear as a part exactly $\binom{N-\alpha}{\ell-j}$ times.

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

Brouwer-Schrijver proof: Baranyai's Theorem

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- Form a directed graph, with vertices s, t, and a vertex for each set partition for $\{1, \ldots, \sigma - 1\}$, and a vertex for each set of size at most ℓ that contains σ.
- Add an arc from t to s carrying flow $\binom{N}{\ell}/v$.
- Add an arc from s to each partition carrying flow 1.
- Add an arc from every j-subset of $\{1, \ldots, \sigma\}$ containing σ to t carrying flow $\binom{N-\alpha}{\ell-j}$.
- When partition P contains a j-set S, add an arc from P to $S \cup \{\sigma\}$ with flow $\frac{\ell-j}{N-\sigma-1}$.
- This is a circulation – flow conservation holds everywhere.

Brouwer-Schrijver proof: Baranyai's Theorem

Charles J. Colbourn, colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- But then by the integer flow theorem, there is an *integer* flow obtained by rounding each of the fractions up or down.
- Look at the edges from partitions to sets that get a flow of 1, and add σ to the corresponding sets.

Extending Baranyai's Theorem

Charles J. Colbourn, colbourn@asu.ed with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- Key things used in the proof:
  - All sets used have the same size.
  - All sets of the allowed size are used, each exactly once.
- To relax the first, allow sets of different sizes but *keep track not only of the set built so far, but also its target size*. Specifically, introduce vertices for the sets containing σ for each of the allowed set sizes.
- A j-subset of $\{1, \ldots, \sigma\}$ that is to form a set of size ν has an arc to t with flow $\binom{N-\alpha}{\nu-j}$.
- A partition P that is to contain a set of size ν but that contains a j-subset S of $\{1, \ldots, \sigma - 1\}$ in this position now has an arc to $(S \cup \{\sigma\}, \nu)$ with flow $\frac{\nu-j}{N-\sigma-1}$.
- So we can deal with many set sizes at the same time.

Extending Baranyai's Theorem

- Key things used in the proof:
  - (All sets used have the same size no longer.)
  - *All sets of the allowed sizes are used, each exactly once.*
- This is a problem! For general collections of sizes of sets, we cannot simultaneously use all sets of each size while keeping them as partitions.
- Try, for example, sets of size 8 and smaller on 15 points with three sets per partition. Then we have to exhaust $\binom{15}{8}$ sets of size 8, but each must appear with a set of size at most 3, and $\sum_{i=0}^{3} \binom{15}{i} < \binom{15}{8}$.

Extending Baranyai's Theorem

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- To repair this, we could try to relax the requirement that every set of each allowed size appears, but this seems problematic.
- Instead, extend from integer partitions of N to integer partitions whose sum is *at most* N, and to set partitions of *subsets* of $\{1, \ldots, N\}$.
- So if we have an (N, v)-admissible set of k integer partitions $\{(n_{i1}, \ldots, n_{iv} : 1 \leq i \leq v\}$, whenever a part of size ℓ is present but does not appear $\binom{N}{\ell}$ times, we can add sufficiently many copies of partitions of ℓ into a single part to make up the deficit.
- The extension of the integer flow argument is "easy".

Extending Baranyai's Theorem

- The extension of the integer flow argument is "easy".
- We must account for partitions of integers less than N, because in these cases the new symbol α may not be added to the set in the partition; rather it is just skipped.
- To do this, we add a single vertex γ. Then whenever a partition of ℓ with single part ℓ is being built, if S is the set already built, add the directed edge from this partition to γ with flow $\frac{N-\sigma-\ell+|S|+1}{N-\sigma-1}$. And add a directed edge from γ to t, computing the flow value so as to ensure flow conservation.
- This establishes the result that we need.

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

Extending Baranyai's Theorem

Charles J. Colbourn, colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

Theorem

Let $\mathcal{I} = \{(\pi_{i1}, \ldots, \pi_{i\ell_i}) : 1 \le i \le k\}$ *be a collection of integer partitions of integers each at most N, so that every part of size* ℓ *for* $0 \le \ell \le N$ *appears at most* $\binom{N}{\ell}$ *times. Then there exist sets* $\{(S_{i1}, \ldots, S_{i\ell_i}) : 1 \le i \le k\}$ *of subsets of* $\{1, \ldots, N\}$ *so that* S_{ij} *has* π_{ij} *elements,* $S_{ij} \cap S_{ij'} = \emptyset$ *unless* $j = j'$*, and* $S_{ij} \ne S_{i'j'}$ *unless* $i = i'$ *and* $j = j'$*.*

Back to Locating

Charles J. Colbourn,
colbourn@asu.ed
with Bingli Fan (Beijing) and Daniel Horsley (Melbourne)

- Using this theorem, we can determine the largest number of columns in a $(1, 1)$-locating array on v symbols with N rows *exactly* when every column has the same number of symbols.
- This generalizes easily to arrays in which different columns have possible different numbers of symbols.
- What about locating *two* faults? We would need (1) no two sets equal, (2) no set equal to the union of two others, and (3) the union of any two sets is not the same as the union of any other two.