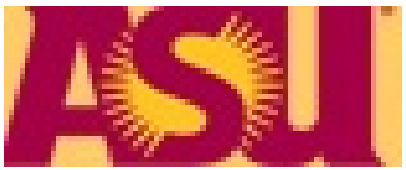
Testing Arrays for Fault Localization

Charles J. Colbourn
Arizona State University

Portorož, June 2021



Combinatorial Testing

Levels of Factors Can Affect Response

13	14	15	16	17	18	19
Sunny	Mostly Sunny	Mostly Sunny	Sunny	Partly Cloudy	Partly Cloudy	Sunny
Forecast:	Forecast:	Forecast:	Forecast:	Forecast:	Forecast:	Forecast:
45° | 28°	47° | 30°	48° | 31°	47° | 32°	48° | 33°	48° | 33°	46° | 31°
0 mm	0 mm	0 mm	0 mm	0 mm	0 mm	0 mm

13	14	15	16	17	18	19
Mostly Sunny	Sunny	Sunny	Mostly Sunny	Mostly Sunny	Mostly Sunny	Sunny
Actual:	Forecast:	Forecast:	Forecast:	Forecast:	Forecast:	Forecast:
113° | 81°	116° | 86°	118° | 89°	117° | 91°	118° | 93°	119° | 90°	116° | 88°
0 in	0 in	0 in	0 in	0 in	0 in	0 in

Charles J. Colbourn
Arizona State University

Combinatorial Testing

Charles J. Colbourn
Arizona State University

- Suppose that there are k factors $F_1, \ldots, F_k$ that control the operation of a complex system.
- Each factor F_i can take on any of a finite set V_i of levels.
- A test is a k-tuple $(\sigma_1, \ldots, \sigma_k)$ with $\sigma_i \in V_i$ for $1 \leq i \leq k$.
- A t-way interaction is a set $\{(\gamma_i, \nu_i) : 1 \leq i \leq t, \nu_i \in V_{\gamma_i}\}$ in which $\gamma_1, \ldots, \gamma_t$ are all distinct.

Combinatorial Testing

Matrix Representation

Charles J. Colbourn
Arizona State University

- A set of N tests is a test suite, usually represented as an $N \times k$ array whose N rows are the N tests.
- Test $(\sigma_1, \ldots, \sigma_k)$ covers interaction $\{(\gamma_i, \nu_i) : 1 \leq i \leq t\}$ when $\nu_i = \sigma_{\gamma_i}$ for $1 \leq i \leq t$.
- For test suite A and t-way interaction T, denote by $\rho_A(T)$ the set of indices of rows of A in which T is covered.

Combinatorial Testing

How does one test?

- Execute every test in a test suite A to get a response for each test.
- Classify each response according to whether it is significant or not.
- Let R be the set of row indices for tests having significant response.
- Goal: Determine a (minimum) set of interactions $\{T_1, \ldots, T_d\}$ so that $\bigcup_{i=1}^{d} \rho_A(T_i) = R$.
- Assume that d and t are both fixed (and 'small'), and both known.

Charles J. Colbourn
Arizona State University

Detecting Arrays

Charles J. Colbourn
Arizona State University

- An $N \times k$ array A is (d, t)-detecting if for every t-way interaction T and every set $\mathcal{T} = \{T_1, \ldots, T_d\}$ of d t-way interactions with $T \notin \mathcal{T}$, there is a row in which T is covered but no interaction in $\mathcal{T}$ is covered.
- In other words, defining $\rho(\mathcal{T}) = \bigcup_{i=1}^{d} \rho_A(T_i)$, we insist that $\rho_A(T) \setminus \rho(\mathcal{T}) \neq \emptyset$.
- Members of $\rho_A(T) \setminus \rho(\mathcal{T})$ are witnesses for T despite the presence of $\mathcal{T}$.

A Test Suite

Rows and Columns Indexed from 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	1	0	0	0	0	1
1	0	1	1	0	1	0	1	0	0
1	0	0	0	1	1	1	0	0	0
0	**1**	1	0	0	1	0	0	1	0
0	0	1	0	1	0	1	1	1	0
1	1	0	1	0	0	1	0	1	0
0	0	0	1	1	1	0	0	1	1
0	0	1	1	0	0	1	0	0	1
0	**1**	0	1	1	0	0	1	0	0
1	0	0	0	0	0	0	1	1	1
0	**1**	0	0	0	1	1	1	0	1

Look at the interaction $T = \{(0,0),(1,1)\}$.

Charles J. Colbourn
Arizona State University

A Test Suite

Rows covering $T = \{(0,0),(1,1)\}$

0	1	1	0	0	1	0	0	1	0
0	1	0	1	1	0	0	1	0	0
0	1	0	0	0	1	1	1	0	1

$\rho(T) \setminus \rho(\{\{(0,0),(2,0)\}\})$ has one witness
$\rho(T) \setminus \rho(\{\{(0,0),(2,1)\}\})$ has two witnesses
$\rho(T) \setminus \rho(\{\{(2,1),(3.1)\}\})$ has three witnesses
If we check all 2-way interactions, we find that this is a a (1,2)-detecting array
$\rho(T) \setminus \rho(\{\{(0,0),(2,0)\},\{(0,0),(2,1)\}\} = \emptyset$ so this is not a (2,2)-detecting array

Charles J. Colbourn
Arizona State University

A Test Suite

Rows covering $T = \{(0,0),(1,1)\}$

0	1	1	0	0	1	0	0	1	0
0	1	0	1	1	0	0	1	0	0
0	1	0	0	0	1	1	1	0	1

$\rho(T) \setminus \rho(\{\{(0,0),(2,0)\}\})$ has one witness
$\rho(T) \setminus \rho(\{\{(0,0),(2,1)\}\})$ has two witnesses
$\rho(T) \setminus \rho(\{\{(2,1),(3.1)\}\})$ has three witnesses
If we check all 2-way interactions, we find that this is a a (1,2)-detecting array
$\rho(T) \setminus \rho(\{\{(0,0),(2,0)\},\{(0,0),(2,1)\}\} = \emptyset$ so this is not a (2,2)-detecting array

Charles J. Colbourn
Arizona State University

Detecting Arrays

How many witnesses? – *separation*

One witness shall not rise up against a man for any iniquity, or for any sin, in any sin that he sinneth: at the mouth of two witnesses, or at the mouth of three witnesses, shall the matter be established.

Deuteronomy 19:15 (KJV)

Charles J. Colbourn
Arizona State University

Detecting Arrays

How many witnesses? – *separation*

Charles J. Colbourn
Arizona State University

- Sometimes test responses are not available for certain tests, resulting in the loss of some or all witnesses.
- To get $\delta \geq 1$ witnesses, we impose the stronger requirement that

$$|\rho_A(T) \setminus \rho(\mathcal{T})| \geq \delta.$$

- The parameter δ is the separation, and the array is (d, t, δ)-detecting.

Detecting Arrays

Separation

- In a (d, t, δ)-detecting array, each t-way interaction T must be covered in at least $d + \delta$ rows.
- But more is true when $d \geq 1$: For every t-way interaction T and every column γ not in T, there must be at least $d + 1$ distinct symbols in column γ in the rows of $\rho(T)$.
- So one might think of *levels* serving as witnesses rather than *rows* or *tests*.

Charles J. Colbourn
Arizona State University

Detecting Arrays

Corroboration

Charles J. Colbourn
Arizona State University

- Fusion is the operation of merging two levels within a factor.
- Suppose that we are allowed to perform s fusions of levels for each factor, so that we always have a $(d, t, 1)$-detecting array.
- Then the initial array has corroboration s.
- This parameter is useful because we can (hope to) make detecting arrays with different numbers of levels per factor from ones with the same number of levels per factor (uniform ones).

Detecting Arrays

and Covering Arrays

Charles J. Colbourn
Arizona State University

- A covering array of strength t and index (or separation) δ – a $CA_\delta(N; t, k, v)$ – is (defined to be) a $(0, t, \delta)$-detecting array.
- This requires simply that $|\rho(T)| \geq \delta$ for every t-way interaction T, i.e., every interaction is covered in at least δ tests.

Detecting Arrays

from covering arrays

Charles J. Colbourn
Arizona State University

- A covering array of strength $d + t$ and index/separation δ is
  1. a $(d, t, \delta(v - d)v^{d-1})$-detecting array with corroboration 1, and
  2. a $(d, t, \delta(d + 1)^{d-1})$-detecting array with corroboration $v - d$.

  where each is uniform with v levels per symbol.
- These have far from the fewest tests in general!

Detecting Arrays

from hash families

Charles J. Colbourn
Arizona State University

- We explore a construction of detecting arrays from hash families that leads to fewer tests.

Hash Family

- A hash family $HF(N; k, v)$ is an $N \times k$ array in which each cell contains a single element from a fixed set of size v.
- The term comes from applications in which each row encodes a 'hashing' function from a k-set (the columns) to a v-set (the symbols).

Charles J. Colbourn
Arizona State University

Separating Hash Family

Charles J. Colbourn
Arizona State University

- An $HF(N; k, v)$ is separating with index λ of type $\{w_1, \ldots, w_s\}$ when for every way to choose a subset T containing $\sum_{i=1}^{s} w_i$ of the k columns, and every way to partition T into classes $C_1, \ldots, C_s$ with $|C_i| = w_i$ for $1 \leq i \leq s$, there are at least λ rows in which two columns from different classes always contain distinct symbols.
- Sometimes denoted by $SHF_\lambda(N; k, v, \{w_1, \ldots, w_s\})$.
- Often exponential notation is used for types: For example, $\{1, 2, 1, 3, 2, 1, 1\}$ can be denoted as $1^4 2^2 3^1$.

Strong Separating Hash Family

Charles J. Colbourn
Arizona State University

- An $SHF_\lambda(N; k, v, \{w_1, \ldots, w_s\})$ is strong separating when $w_1 = \cdots = w_{s-1} = 1$.
- For a strong SHF we can write the type as $1^t d^1$.
- Aside: A strong $SHF_1(N; k, v, 1^t d^1)$ with $d = 1$ is a perfect hash family of strength $t + 1$. These have been studied extensively!

Covering Perfect Hash Family

Charles J. Colbourn
Arizona State University

- Sherwood, Martirosyan, and C explored hash families in which the 'symbols' in the array are column vectors in $\mathbb{F}_q^s$.
- Such a hash family is covering perfect of strength s if, for every s-set of columns, in some row the s column vectors (each from $\mathbb{F}_q^s$) form an $s \times s$ matrix that is nonsingular over $\mathbb{F}_q$.

Covering Perfect Hash Family

Are they useful?

Charles J. Colbourn
Arizona State University

- These CPHFs lead to
  - substantial improvements in bounds for so-called covering arrays, and algorithms to find them (C, Lanus, Sarkar)
  - the best current asymptotic bounds for covering array sizes (C, Lanus, Sarkar; and Das, Meszaros)
  - elegant connections with finite geometry and with LFSRs (Raaphorst, Tzanakis, Moura, Panario, Stevens)

Covering Strong Separating Hash Family

Charles J. Colbourn
Arizona State University

- A covering strong separating hash family – a $CSSHF_\lambda(N; k, q, t, d)$ – is a strong separating hash family of index λ and type $1^t d^1$ with 'symbols' being column vectors from $\mathbb{F}_q^{t+1}$ when, for every way to choose a set T of t columns and a disjoint set D of d columns, there exist at least λ rows in which
  - the $t \times t$ matrix arising from the first t coordinates of the $(t+1) \times t$ array corresponding to the columns of T in this row is nonsingular; and
  - for every column $\gamma \in D$, the $(t+1) \times (t+1)$ array corresponding to the columns of $T \cup \{\gamma\}$ in this row is nonsingular.

Covering Strong Separating Hash Family

Are they useful?

- Let H be a $CSSHF_\lambda(N; k, q, t, d)$ with $d < q$.
- Let $S \subseteq \mathbb{F}_q$ with $|S| = d + \sigma$ for $q - d \geq \sigma \geq 1$
- Let R be the list of row vectors in $\mathbb{F}_q^t \times S$.
- We form a $N(d + \sigma)q^t \times k$ array with entries from $\mathbb{F}_q$ as follows. Each entry $h \in H$ is a column vector $(h_0, \ldots, h_t)^T$. Each row $r \in R$ is a row vector $(b_0, \ldots, b_t)$.
- We compute the column vector $(h \cdot r : r \in R)^T$, and substitute this for h in H.

Charles J. Colbourn
Arizona State University

Covering Strong Separating Hash Family

Are they useful?

Charles J. Colbourn
Arizona State University

- But what do we get?
  - In every t-subset of columns, for every choice of elements for these columns (i.e. choice from $\mathbb{F}_q^t$), there must be at least λ sets of at least $d + \sigma$ rows in which this t-tuple arises in the chosen columns.
  - So this is a *covering array of index* $\lambda(d + \sigma)$.
  - But much more is true.

Covering Strong Separating Hash Family

Are they useful?

- What do we get from the $CSSHF_\lambda(N; k, q, t, d)$ with $d < q$?
  - a (d, t)-detecting array with
    - $Nq^t(d + \sigma)$ rows, k columns, and q symbols,
    - separation $\lambda\sigma$, and
    - corroboration σ.

Charles J. Colbourn
Arizona State University

Detecting Main Effects

- Let's focus on $t = 1$.

Charles J. Colbourn
Arizona State University

Strong Separating Hash Family

Making Detecting Arrays by *h*-inflation

- Let v be a prime power and let $1 \leq h \leq v$ and let $\{e_0, \ldots, e_{v-1}\}$ be the elements of $\mathbb{F}_v$.
- Let A be an $(N; k, v+1)$-hash family on $\{e_0, \ldots, e_{v-1}\} \cup \{\infty\}$.
- Define 2×1 column vectors $\mathcal{C}_h$ containing $\mathbf{c}_\infty = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\mathbf{c}_x = \begin{pmatrix} x \\ 1 \end{pmatrix}$ for $x \in \mathbb{F}_v$. Form a set of r_h row vectors $\mathcal{R}_h = (\mathbf{r}_1, \ldots, \mathbf{r}_{r_h})$ so that for every $\mathbf{c}_a \in \mathcal{C}_h$, each $\mathbf{d}_a = (\mathbf{r}_i \mathbf{c}_a : 1 \leq i \leq r_h)$ contains each entry of $\mathbb{F}_v$ at least h times.
- Form B by replacing each element a in array A by the column vector $\mathbf{d}_a^T$.
- Then B is a $(r_h N; k, v)$-hash family, an *h*-inflation of A.

Charles J. Colbourn
Arizona State University

Detecting Arrays

from hash families

Charles J. Colbourn
Arizona State University

Via h-inflation, we get

Theorem

Let v be a prime power. When an SHF$(N; k, v+1, \{1, d\})$ with separation δ exists, and $1 \leq s \leq v - d$, there exists a $(d, 1, \delta s)$-detecting array with $r_{s+d}N$ tests, k factors, v symbols; and a $(d, 1, \delta)$-detecting array with $r_{s+d}N$ tests, k factors, v symbols, and corroboration $\lfloor (s + d - 1)/d \rfloor$.

Separating Hash Family

from Error Correcting Codes

Charles J. Colbourn
Arizona State University

- Suppose that there is a v-ary code of length N having k codewords and Hamming distance Δ.
- Consider one codeword C and a set D of d other codewords. There are at most $d(N - \Delta)$ coordinates in which C agrees with one or more codeword in D.
- So there must be at least $N - d(N - \Delta)$ coordinates in which C disagrees with every codeword in D.
- Hence when $\Delta > \frac{d-1}{d}N$, we get a $SHF(N; k, v, \{1, d\})$ with separation $N - d(N - \Delta)$.
- Perhaps unfortunately, we want codes with larger alphabets, and this connection is most effective only when $d = 1$.

Separating Hash Family

Computational Methods

Charles J. Colbourn
Arizona State University

- How can we construct the necessary hash families with many factors (columns) for specified d, v, and δ, so that N is "small"?
- Easy idea: Use the basic probabilistic method. Choose a random hash family and check until one has the desired type and separation.
- Better idea: Select in stages by conditional expectation.

Separating Hash Family

Conditional Expectation

- Given a set of rows already chosen, you can efficiently calculate the *expected* number of rows needed to complete the array.
- We always choose a row to add that reduces the expected number of rows to complete by at least one.
- To choose such a row, we choose it one-symbol-at-a-time, so as never to increase the expected number of rows to complete.
- This can all be done in polynomial time provided that d is fixed, and it guarantees a size no larger than the basic probabilistic bound.

Charles J. Colbourn
Arizona State University

Separating Hash Family

Conditional Expectation with Oversampling

- To end up with k columns, proceed as above to make $k + x$ columns, stopping when there are at most x sets of $d + 1$ columns for which a $\{1, d\}$ separation fails to occur at least δ times.
- Then delete at most x columns to remove all of the 'blemishes', leaving at least k columns.

Charles J. Colbourn
Arizona State University

Separating Hash Family

Lovász Local Lemma

- The symmetric LLL yields a bound on N that is smaller than that of the basic probabilistic method (but not better than oversampling).
- But can we guarantee to meet the LLL bound efficiently?

Charles J. Colbourn
Arizona State University

Separating Hash Family

Algorithmic Lovász Local Lemma

Charles J. Colbourn
Arizona State University

- The Moser-Tardos algorithm guarantees to meet this better bound, but runs in *expected* polynomial time.
- The idea is simple: Make a random array. Repeatedly check every set of $d+1$ columns for the $\{1, d\}$ separation occurring at least δ times. If it does not, resample all entries in each of the $d+1$ columns.
- A variation: Just replace one of the $d+1$ columns randomly.

Separating Hash Family

Random Column Extension

Charles J. Colbourn
Arizona State University

- If we are to resample only one column, why not the last?
- And if we just keep sampling until we get a column that is ok, we are in essence doing a random extension of the array by one column.
- This is remarkably fast, and surprisingly accurate compared to the probabilistic methods.
- We report some computational results to give the flavour of the arrays produced.

Number k of columns found for an $SHF_\delta(N; k, 6, \{1, 2\})$

Charles J. Colbourn
Arizona State University

$\delta \downarrow N \rightarrow$	1	2	3	4	5	6	7	8	9	10	11	12
1	6	10	36	51	154	201	373	634	1003	1751	2825	4578
2		6	7	34	39	142	152	262	342	529	805	1257
3			6	6	30	32	72	80	168	195	328	486
4				6	6	27	27	56	58	125	134	231

$\delta \downarrow N \rightarrow$	13	14	15	16	17	18	19	20	21	22	23	24
1	8068	10000										
2	2041	3163	4920	8431	10000							
3	716	1086	1695	2543	3891	6290	9878	10000				
4	311	466	696	1005	1540	2310	3387	5181	8242	10000		

Detecting Arrays

for t-way interactions

Charles J. Colbourn
Arizona State University

- We have discussed the detection of main effects ("1-way interactions") and the need for separation.
- In the process, we outlined the need for a further parameter, corroboration.
- But in practice small sets of factors interact.
- The algebraic construction developed here over finite fields can be adapted to make (d, t)-detecting arrays with different separations, and in principle different corroborations as well.
- Nevertheless, the h-inflation used for main effects differs from the inflation used for $t \geq 2$, so this special case remains individually interesting.

Detecting Arrays

Charles J. Colbourn
Arizona State University

- The connection to hash families leads to
  - a very compact representation when v is a prime power (because we need a field);
  - the best current asymptotics for detecting array sizes;
  - randomized and derandomized algorithms for their construction via the algorithmic LLL, via conditional expectation, and via oversampling.
  - a powerful mechanism for dealing with larger values of t and d.

A Challenge

- Can one effectively use the algebraic/geometric interpretation of the entries in covering strong separating hash families to obtain good direct constructions?

Charles J. Colbourn
Arizona State University