Linear Extensions and Sequence Covering Arrays

Charles J. Colbourn

School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
(part 1: with Yeow Meng Chee, Daniel Horsley, Junling Zhou)
(part 2: with Patrick Murray)

Testing Event Sequences

- In some processes, for example in manufacturing, a set of tasks must be carried out in a certain sequence.
- But people are not good at following instructions, and sometimes do the steps (or events) in the wrong order.
- If certain subsets are done in the wrong order, this may cause the wrong behaviour.
- So we want to test the system to make sure that when users do some events in the wrong order, *either* the process still works *or* the user gets an error message.

Testing Event Sequences

- Suppose that there are v events in the process.
- Suppose that errors can be caused by performing some subset of t or fewer events in a certain order.
- We want to ensure that for every subset of t or fewer events, we perform the events in each of the $t!$ orders at least once – in doing this, we still perform all v events in some order.

Testing Event Sequences

- This problem was introduced by Kuhn, Higdon, Kacker, Lawrence, and Lei in April 2012 at the Workshop on Combinatorial Testing.
- They give a basic lower bound on the number of tests needed, and a heuristic algorithm for constructing tests.
- But this problem is closely related to many that have been studied already.

Permutation t-Coverings

- A *t-subpermutation* of $\{0, \ldots, v-1\}$ is a t-tuple $(x_1, \ldots, x_t)$ with $x_i \in \{0, \ldots, v-1\}$ for $1 \leq i \leq t$, and $x_i \neq x_j$ when $i \neq j$.
- A permutation π of $\{0, \ldots, v-1\}$ *covers* the t-subpermutation $(x_1, \ldots, x_t)$ if $\pi^{-1}(x_i) < \pi^{-1}(x_j)$ whenever $i < j$.
- For example, with $v = 5$ and $t = 3$, $(4, 0, 3)$ is a 3-subpermutation that is covered by the permutation 4 2 0 3 1.

Permutation t-Coverings

- A permutation covering of order v and strength t is a set $\Pi = \{\pi_1, \ldots, \pi_N\}$ where π_i is a permutation of $\{0, \ldots, v-1\}$, and every t-subpermutation of $\{0, \ldots, v-1\}$ is covered by at least one of the permutations $\{\pi_1, \ldots, \pi_N\}$.
- Call one a PermC$(N; t, v)$.
- When written as an array, often called a sequence covering array SeqCA$(N; t, v)$.

Permutation t-Covering

Example

$t = 3$, $v = 5$, $N = 8$

SeqCA	CSSP
4 2 0 3 1	2 4 1 3 0
1 4 3 0 2	3 0 4 2 1
3 1 2 0 4	3 1 2 0 4
0 2 4 1 3	0 3 1 4 2
2 1 3 4 0	4 1 0 2 3
0 3 4 1 2	0 3 4 1 2
3 0 2 1 4	1 3 2 0 4
4 1 2 0 3	3 1 2 4 0

Scrambling Sets

- A completely t-scrambling set of permutations, CSSP$(N; t, v)$ is an $N \times v$ array $A = (a_{ij})$ for which
  - every row forms a permutation of the v symbols, and
  - in every set of t columns $c_1, \ldots, c_t$, and for every permutation ψ of $\{1, \ldots, t\}$, there is a row ρ such that $a_{\rho c_{\psi(i)}} < a_{\rho c_{\psi(i+1)}}$ for $1 \leq i < t$.
  - (in other words, in every set of t columns, every 'pattern' appears on these t columns in at least one row)
- This is *equivalent* to a SeqCA$(N; t, v)$ – just interchange the roles of columns and symbols.

Sequence Covering Arrays

The Existence Question

- Given t and v, what is the smallest N for which a SeqCA$(N; t, v)$ exists?
- Call this number SeqCAN(t, v).
  - SeqCAN$(t, v) \geq t!$
  - SeqCAN$(t, v) \leq \binom{v}{t} t!$
- SeqCAN$(2, v) = 2$ for all $v \geq 2$ – Just take any permutation and its reversal!

Sequence Covering Arrays

The Case of Equality

- For $t \geq 3$, when is $\mathrm{SeqCAN}(t, v) = t!$?
- Levenshtein (1991) showed that it always is when $v \leq t + 1$, and conjectured that it never is when $v > t + 1$.
- Mathon and Tran Van Trung (1999) showed that $\mathrm{SeqCAN}(4, 6) = 4!$, and showed that $\mathrm{SeqCAN}(t, v) > t!$ when v exceeds a certain quadratic function of t.
- We have shown that $\mathrm{SeqCAN}(t, v) > t!$ when $v \geq 2t$.
- We need a way to produce lower bounds.

Covering Array. Definition

- Let N, k, t, and v be positive integers.
- Let C be an $N \times k$ array with entries from an alphabet Σ of size v; we typically take $\Sigma = \{0, \ldots, v-1\}$.
- When $(\nu_1, \ldots, \nu_t)$ is a t-tuple with $\nu_i \in \Sigma$ for $1 \leq i \leq t$, $(c_1, \ldots, c_t)$ is a tuple of t column indices ($c_i \in \{1, \ldots, k\}$), and $c_i \neq c_j$ whenever $\nu_i \neq \nu_j$, the t-tuple $\{(c_i, \nu_i) : 1 \leq i \leq t\}$ is a *t-way interaction*.
- The array covers the t-way interaction $\{(c_i, \nu_i) : 1 \leq i \leq t\}$ if, in at least one row ρ of C, the entry in row ρ and column c_i is ν_i for $1 \leq i \leq t$.
- Array C is a covering array CA$(N; t, k, v)$ of strength t when every t-way interaction is covered.

Covering Array

CA(13;3,10,2)

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

Covering Array

- CAN(t, k, v) is the minimum N for which a CA$(N; t, k, v)$ exists.
- The basic goal is to minimize CAN(t, k, v).
- When t and v are both fixed, CAN(t, k, v) is $\Theta(\log k)$.

Lower Bounds for Sequence Covering Arrays

Theorem

Let k and t be integers satisfying $k \geq t \geq 3$. Whenever $0 \leq a < t$, the size of a sequence covering array for k events with strength t is at least

$$a! CAN(t - a, k - a, a + 1)$$

Lower Bounds for Sequence Covering Arrays

- Choose any a events.
- For each ordering $(e_1, \ldots, e_a)$ of the a events, select the permutations of the permutation covering in which the a selected events appear in the chosen order; suppose that there are n such permutations.
- Form an $n \times (k - a)$ array A whose columns are indexed by the remaining events, and whose rows are indexed by the permutations selected.
- In the row indexed by π and the column indexed by event e,
  - place 0 if $\pi(e) < \pi(e_1)$
  - place a if $\pi(e) > \pi(e_a)$
  - otherwise, place i when $\pi(e_i) < \pi(e) < \pi(e_{i+1})$.

Lower Bound

Example

SeqCA(9;3,7) – $t = 3$, $v = 7$, $N = 9$
take $a = 1$ and use symbol 6

0 5 6 4 3 2 1	0 1 1 1 1 0
2 1 6 5 0 3 4	1 0 0 1 1 1
3 4 5 1 2 0 6	0 0 0 0 0 0
6 1 2 4 3 0 5	1 1 1 1 1 1
0 1 4 3 6 2 5	0 0 1 0 0 1
5 2 3 4 6 0 1	1 1 0 0 0 0
3 6 1 5 0 2 4	1 1 1 0 1 1
4 0 1 2 5 6 3	0 0 0 1 0 0
6 2 5 1 3 4 0	1 1 1 1 1 1

Lower Bounds for Sequence Covering Arrays

- The theorem can give better estimates.
- For $t = 4$, we obtain the lower bounds
  - $\mathrm{SeqCAN}(4, v) \geq \mathrm{CAN}(4, v, 1)$ by taking $a = 0$,
  - $\mathrm{SeqCAN}(4, v) \geq \mathrm{CAN}(3, v - 1, 2)$ by taking $a = 1$,
  - $\mathrm{SeqCAN}(4, v) \geq 2\mathrm{CAN}(2, v - 2, 3)$ by taking $a = 2$,
  - $\mathrm{SeqCAN}(4, v) \geq 6\mathrm{CAN}(1, v - 3, 4) = 24$ by taking $a = 3$.
- Taking $a = 0$ always gives a trivial bound.

Upper Bounds for Sequence Covering Arrays

Random Arrays

- Pick a permutation of $\{0, \ldots, v-1\}$ uniformly at random. Then a specific t-subpermutation fails to be covered with probability $\frac{t!-1}{t!}$.
- Picking N permutations uniformly at random and independently fails to cover a specific t-subpermutation with probability $\left(\frac{t!-1}{t!}\right)^N$.
- There are $\frac{v!}{(v-t)!}$ t-subpermutations.

Upper Bounds for Sequence Covering Arrays

Random Arrays

- So when $\frac{v!}{(v-t)!}\left(\frac{t!-1}{t!}\right)^N < 1$, a SeqCA$(N; t, v)$ must exist!
- When t is fixed, this gives an $O(\log v)$ upper bound on the number of permutations needed.

Upper Bounds for Sequence Covering Arrays

Greedy Random Arrays

- Instead generate the set of permutations *one permutation at a time*.
- Idea: Always pick the next permutation so that it covers the largest possible number of as-yet-uncovered t-subpermutations.
- Suppose that after i permutations have already been chosen, there remain R_i t-subpermutations to be covered.
- Then $R_0 = \frac{v!}{(v-t)!}$.

Upper Bounds for Sequence Covering Arrays

Greedy Random Arrays

- Selecting the $(i+1)$st permutation uniformly at random, every (as-yet-uncovered) t-subpermutation is covered with probability $\frac{1}{t!}$.
- By the linearity of expectations, the expected number of t-subpermutations covered *for the first time* is $\frac{R_i}{t!}$.
- But then

$$R_{i+1} \leq R_i - \frac{R_i}{t!},$$

and because $R_0 = \frac{v!}{(v-t)!}$,

$$R_i \leq \left(\frac{t! - 1}{t!}\right)^i \frac{v!}{(v-t)!}.$$

- When $R_N < 1$, a SeqCA$(N; t, v)$ exists – and the greedy method guarantees to find one!

Stein-Lovász-Johnson

- The greedy strategy has proved remarkably successful for a variety of combinatorial covering problems.
- Most surprising is that in many cases it has been shown that we can select the next row efficiently (in time polynomial in the number of columns when the strength t is fixed).
- Such efficient methods have been found for
  - covering arrays (Bryce and Colbourn (2007, 2009))
  - perfect hash families (Colbourn (2008))
  - separating, distributing, strengthening, scattering, ... hash families (Colbourn (2011); Colbourn, Horsley, and Syrotiuk (2012))

Stein-Lovász-Johnson

- The key observations in these efficient methods are
  - It is enough to choose the next “row” so that it covers *at least the average number of as-yet-uncovered “entities”*
  - We can fill in the row one entry at a time so that the expected number covered by any completion of the row never decreases, and what we need to do is to find an entry – efficiently – that does not decrease this expected number.

Tarui's Method

- $SeqCAN(3, \binom{s}{\lfloor s/2 \rfloor}) \leq 2s$:
  - Form all binary vectors of length s and weight $\lfloor s/2 \rfloor$. These are the events to be ordered.
  - For each coordinate c, let A_c contain all vectors with a 1 in coordinate c, and B_c contain all those with a 0.
  - Form two permutations as follows. In the first list all events of A_c then all events of B_c. In the second list all events of A_c in the reverse order, then all events of B_c in the reverse order.

A Product Construction

- Using a notion of "signing" a sequence covering array, we have developed a direct product construction.
- Whenever (signed) SeqCA($N; 3, v$) and SeqCA($M; 3, w$) both exist, there is a (signed) SeqCA($N + M; 3, vw$).
- This improves on all of the known computational methods when $v \geq 40$, but only applies for strength three.
- But it does not (asymptotically) beat Tarui's method.

Some Computation

Events	$t = 3$								
v	TA	U	U_R	K	ER	D_R	D	BTI	BR
5	8	17	16	8		10	8	7	7
10	10	30	28	14	11	14	12	9	9
15	12	37	34	18		16	14	10	10
20	12	42	38	22	19	18	16	12	12
25	14	46	42	24		20	18	14	14
30	14	49	46	26	23	22	19	15	15
40	16	54	50	32	27	24	21	17	17
50	16	58	52	34	31	26	23	19	18
60	16	61	56	38	34	26	24	21	20
70	16	64	58	40	36	28	25	22	22
80	18	66	60	42	38	30	26	24	23
90	18	68	62	44		30	27		

The Product Construction and Tarui's Method

- Using the product construction, SeqCA$(N; 3, v)$ for $(v, N) \in \{(40, 15), (80, 17), (128, 18), (160, 19), (256, 20), (288, 21)\}$
- Using Tarui's method, SeqCA$(N; 3, v)$ for $(v, N) \in \{(35, 14), (70, 16), (126, 18), (252, 20), (462, 22)\}$

Some Computation

Events	$t = 4$							$t = 5$				
v	U	$\mathrm{U_R}$	K	$\mathrm{D_R}$	D	BTI	BR	U	$\mathrm{U_R}$	$\mathrm{D_R}$	D	BA
5	54	54	29	24	26	24		120	120	120	120	
10	140	138	72	56	57	58	55	731	728	318	322	383
15	184	180	108	78	78	84		1027	1024	470	475	
20	214	210	134	92	92	105	104	1223	1218	582	590	
25	236	232	152	104	104			1370	1366	674	682	
30	255	250	166	114	113		149	1488	1482	748	760	
40	283	278	198	132	128		181	1671	1664			
50	305	298	214	146	141			1811	1804			
60	322	316	238	154	151			1924	1916			
70	337	330	250	166	160			2019	2012			
80	350	342	264	174	168			2101	2092			
90	361	354	-	180	176			2173	2164			

A Post-Optimization Method

- Choose an arbitrary order on the permutations.
- Determine all t-permutations covered by each permutation that is not covered by an earlier one.
- For each permutation, form a poset on the v events in which $x \prec y$ when there is some subpermutation in which x precedes y and that is covered for the first time by this permutation.
- Choose an arbitrary linear extension of each poset, and replace the permutation using this linear extension.
- If there is a permutation that covers no subpermutation for the first time, remove it.
- Repeat the steps above until some stopping criterion is met.

A Post-Optimization Method

- Choose an arbitrary order on the permutations.
- Determine all t-permutations covered by each permutation that is not covered by an earlier one.

Example

SeqCA	First Covered
4 2 0 3 1	031 201 203 231 401 403 420 421 423 431
1 4 3 0 2	102 130 132 140 142 143 302 402 430 432
3 1 2 0 4	104 120 124 204 304 310 312 314 320 324
0 2 4 1 3	013 021 023 024 041 043 213 241 243 413
2 1 3 4 0	134 210 214 230 234 240 340
0 3 4 1 2	012 032 042 034 341 342 412
3 0 1 4 2	014 301
1 4 2 0 3	103 123
3 2 4 1 0	321 410

A Post-Optimization Method

- Choose an arbitrary order on the permutations.
- Determine all t-permutations covered by each permutation that is not covered by an earlier one.
- For each permutation, form a poset on the v events in which $x \prec y$ when there is some subpermutation in which x precedes y and that is covered for the first time by this permutation.
- Choose an arbitrary linear extension of each poset, and replace the permutation using this linear extension.
- Example: From permutation 1 4 2 0 3, $\{103, 123\}$ has the poset $1 \prec 0$, $1 \prec 2$, $0 \prec 3$, $2 \prec 3$; one linear extension is 4 1 2 0 3.

Example

SeqCA	First Covered
4 2 0 3 1	031 201 203 231 401 403 420 421 423 431
1 4 3 0 2	102 130 132 140 142 143 302 402 430 432
3 1 2 0 4	104 120 124 204 304 310 312 314 320 324
0 2 4 1 3	013 021 023 024 041 043 213 241 243 413
2 1 3 4 0	134 210 214 230 234 240 340
0 3 4 1 2	012 032 042 034 341 342 412
3 0 2 1 4	014 301 321
4 1 2 0 3	103 123 410
3 2 4 1 0	—

Using the Post-Optimization Method

$t = 4$		
v	Initial	Final
5	26	24
6	34	24
7	41	36
8	44	41
9	52	46
10	57	51
13	71	62
15	78	67
25	104	91
90	180	162

$t = 5$		
v	Initial	Final
6	148	122
7	198	175
8	242	218
9	284	261
10	318	300
11	354	335
12	386	360
13	419	390
15	475	451
20	590	574
30	748	725

Open Problems # 1

- Find explicit direct constructions (like Tarui's) for strength $t \geq 4$.
  - Can we construct a sequence covering array from a covering array? What conditions are necessary? sufficient?
  - Can one find an explicit construction to determine SeqCAN(3, v)?

Open Problems # 2

- Can we determine when SeqCAN$(t, v) = t!$? This is related to *Directed Steiner t-designs*.
- Find recursive constructions (like the product construction) for strength $t \geq 4$.

Variations on the Theme

- What if certain ℓ-subpermutations must be avoided (constraints)? How does this affect the size of the array? the complexity of finding one?
- What if certain ℓ-subpermutations need not be tested (avoids)? This is related to the problem of "separation dimension".

Conclusion

- An interesting practical application to event sequence testing.
- Strong connections to design theory, permutation patterns, extremal set theory.
- And *lots* of interesting open problems!