International Journal of Group Theory
ISSN (print): 2251-7650, ISSN (on-line): 2251-7669
Vol. 13 No. 3 (2024), pp. 293-305.

http://ijgt.ui.ac.ir/

www.ui.ac.ir

COVERING PERFECT HASH FAMILIES AND COVERING ARRAYS OF HIGHER INDEX

CHARLES J. COLBOURN

Abstract. By exploiting symmetries of finite fields, covering perfect hash families provide a succinct representation for covering arrays of index one. For certain parameters, this connection has led to both the best current asymptotic existence results and the best known efficient construction algorithms for covering arrays. The connection generalizes in a straightforward manner to arrays in which every t-way interaction is covered $\lambda > 1$ times, i.e., to covering arrays of index more than one. Using this framework, we focus on easily computed, explicit upper bounds on numbers of rows for various parameters with higher index.

To Daniela Nikolova on her 70th Birthday

1. Introduction

Combinatorial arrays for testing large-scale systems have been a topic of substantial interest due to their many applications and to the challenges of constructing such arrays efficiently. Because the arrays that are needed are large and must satisfy stringent conditions, constructions based on finite fields have been explored. Perhaps surprisingly, these constructions have led to dramatic improvements in our knowledge about the sizes of such testing arrays both in theory and in practice. In this paper, our objective is to explore generalizations of these techniques to testing scenarios in which multiple coverage is desirable. At the same time, we aim to strengthen bridges among the combinatorial aspects, the algebraic aspects, the probabilistic aspects, and the testing applications.

For the most part, we follow a well-trodden path that has been earlier followed for covering arrays [15] and perfect hash families [22]. However, our interest in this paper is primarily to address a basic

Keywords: covering array, covering perfect hash family, finite field, probabilistic method.

MSC(2010): Primary: 05B15; Secondary: 20B25, 51E20.

Communicated by Patrizia Longobardi.

Article Type: 2022 CCGTA IN SOUTH FLA.

Received: 11 Aprill 2023, Accepted: 26 September 2023.

Cite this article: C. J. Colbourn, Covering perfect hash families and covering arrays of higher index, Int. J. Group Theory, **13** no. 3 (2024) 293–305. http://dx.doi.org/10.22108/ijgt.2023.137230.1836 .

question. Many algebraic, geometric, number-theoretic, or other approaches might be applied to the construction of testing arrays. How can one decide which appear promising, in the sense that they yield testing arrays of competitive size? For "small" parameters with index 1, one might check online tables (such as [10]) to make a comparison. But for larger parameters, or when the index is greater than one, it is desirable to have simple methods to compute reasonable upper bounds for use as targets with which to compare a construction. Our use of probabilistic methods here is to derive such methods; in this paper, their asymptotic consequences are less relevant.

Now we describe the testing arrays formally. Let N, t, k, v, and λ be positive integers with $k \geq t \geq 2$, $v \geq 2$, and $\lambda \geq 1$. A *covering array* $\mathsf{CA}_\lambda(N; t, k, v)$ is an $N \times k$ array A in which each entry is from a v-ary alphabet Σ, and for every $N \times t$ sub-array B of A and every $\mathbf{x} \in \Sigma^t$, there are at least λ rows of B that each equal $\mathbf{x}$.

For k a positive integer, denote by $[k]$ the set $\{1, \ldots, k\}$. A *t-way interaction* is $\{(c_i, a_i) : 1 \leq i \leq t\}$ where $c_i \in [k]$, $c_i \neq c_j$ for $i \neq j$, and $a_i \in \Sigma$. Row r of an $N \times k$ array A *covers* the interaction $\iota = \{(c_i, a_i) : 1 \leq i \leq t, c_i \in [k], c_i \neq c_j \text{ for } i \neq j, \text{ and } a_i \in \Sigma\}$ when $A(r, c_i) = a_i$ for $1 \leq i \leq t$. Array A *λ-covers* the interaction ι when at least λ rows of A cover ι. In this vernacular, a $\mathsf{CA}_\lambda(N; t, k, v)$ λ-covers each t-way interaction on k columns on an alphabet of size v.

Covering arrays are widely used for combinatorial interaction testing [28, 31, 39]. The k columns represent *factors* that might affect test outcomes; the v *levels* of each factor indicate possible settings for the factor; each of the N rows forms a *test* or *run* of a test plan; t is the coverage *strength*; and λ is the *index* or *repetition* of coverage. Until this time, essentially all research effort has focussed on the case of index $\lambda = 1$ to ensure that each t-way interaction is tested, but recently it has been argued that many experimental environments benefit from increasing the index [1, 23]; for related work, see [16, 17, 34, 40].

Applications require the effective construction of actual covering arrays. Costs of testing include the effort to generate a test plan, the effort to execute the plan; and the analysis of the outcomes. In order to reduce execution costs, it is essential to ensure that the number of tests is 'small'; one hopes to produce covering arrays with the fewest rows possible. The smallest value of N for which a $\mathsf{CA}_\lambda(N; t, k, v)$ exists is a *covering array number*, denoted by $\mathsf{CAN}_\lambda(t, k, v)$. At the same time, methods to generate the test plan cannot be too time-intensive. Computational methods are challenged both by the large number of interactions to cover, $v^t \binom{k}{t}$, and by the large number of rows required. A natural strategy is to assume and exploit symmetries on the rows, columns, or symbols. For group actions on the symbols, see [7, 8, 13]; on columns see [12]; and on both see [11, 32, 35], for example. However, these all concern the action of relatively small groups, and the resulting methods have remained focussed on 'small' parameters.

For index one, Sherwood *et al.* [45] extended the Bose-Bush construction of orthogonal arrays [3, 6] to produce a construction of covering arrays over the finite field employing actions on t-sets of columns and on symbols. In the process, Sherwood *et al.* [45] introduced a class of combinatorial arrays called

http://dx.doi.org/10.22108/ijgt.2023.137230.1836

'covering perfect hash families', later generalized in [14, 15, 48]. These have proved to be remarkably effective in establishing the best known asymptotic upper bound on covering array numbers of index one [15, 18] and in the development of efficient algorithms for constructing covering arrays of index one; see [14, 15, 47, 48, 51, 52], for example. Connections with linear feedback shift registers, projective geometries, and linear codes are developed in [37, 40, 41, 49]. Essentially all of the cited work focusses on index one, but Dougherty *et al.* [23] explicitly extend the definitions to higher indices.

In this paper, we first recall the 'covering perfect hash family' framework for arbitrary index. Then, by examining probabilistic methods to obtain bounds, we explore the dependence between the index and number of rows needed for specified strength and number of factors.

2. Covering perfect hash families

Let q be a prime power. Let $\mathbb{F}_q$ be the finite field of order q. Let $\mathcal{R}_{t,q} = \{\mathbf{r}_0, \ldots, \mathbf{r}_{q^t-1}\}$ be the set of all (row) vectors of length t with entries from $\mathbb{F}_q$. Let $\mathcal{V}_{t,q}$ be the set of all column vectors of length t with entries from $\mathbb{F}_q$, not all 0, in which the first nonzero coordinate is the multiplicative identity element. Vectors in $\mathcal{V}_{t,q}$ are called *permutation vectors*. The essence of the Bose-Bush construction [6] is: When $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$ is a set of permutation vectors, the array $A = (a_{ij})$ formed by setting a_{ij} to be the product of $\mathbf{r}_i$ and $\mathbf{x}_j$ is a $\mathsf{CA}(q^t; t, t, q)$ if and only if the $t \times t$ matrix $X = [\mathbf{x}_1 \cdots \mathbf{x}_t]$ is nonsingular (over $\mathbb{F}_q$).

For nonzero $\mu \in \mathbb{F}_q$, substituting $\mathbf{x}_i$ by $\mu\mathbf{x}_i$ simply reorders the rows of the covering array produced; this justifies our restriction to 'normalized' vectors in which the first nonzero coordinate is the multiplicative identity element. Then $|\mathcal{V}_{t,q}| = \frac{q^t-1}{q-1} = \sum_{i=0}^{t-1} q^i$.

Let $C = (\mathbf{c}_{ij})$ be an $n \times k$ array with entries from $\mathcal{V}_{t,q}$. Let $T = \{\gamma_1, \ldots, \gamma_t\}$ be a set of distinct column indices of C. For row ρ of C and column set T, the entries $\{\mathbf{c}_{\rho\gamma_1} \cdots \mathbf{c}_{\rho\gamma_t}\}$ form a *covering t-set for T* when $[\mathbf{c}_{\rho\gamma_1} \cdots \mathbf{c}_{\rho\gamma_t}]$ is nonsingular, *non-covering* otherwise. Then C is a *covering perfect hash family* $\mathsf{CPHF}_\lambda(n; k, q, t)$ when every set $\{\gamma_1, \ldots, \gamma_t\}$ of distinct column indices has a covering t-set in at least λ rows. An equivalent formulation can be useful. A t-set T of columns is *μ-covered* if it has a covering t-set in at least μ rows; it is *$\mu^=$-covered* if it has a covering t-set in exactly μ rows. Then the CPHF condition asks for every t-set to be λ-covered. The following is straightforward (see [15, 45]).

Lemma 2.1. *Suppose that a* $\mathsf{CPHF}_\lambda(n; k, q, t)$ *exists. Then there exists a* $\mathsf{CA}_\lambda(n(q^t - 1) + \lambda; t, k, q)$.

As observed in [14, 15], one can often reduce the number of rows by enforcing restrictions on the entries of the CPHF. For our purposes, what matters is that a CPHF with few rows represents a covering array with far more rows. This succinctness has evident advantages for the construction and storage of a covering array, but it also facilitates their analysis.

3. Probabilistic Methods and Bounds

We employ probabilistic upper bounds to explore how increases in the index impact sizes of the CPHFs and the resulting covering arrays. (For applications of similar methods to covering arrays of higher index

http://dx.doi.org/10.22108/ijgt.2023.137230.1836

directly, see [21, 23].) For covering array numbers with index one, there is an extensive literature on both lower bounds and upper bounds. For lower bounds, see [26, 44]. The basic probabilistic method (see [2], for example) is applied in [9] to obtain an upper bound and a greedy construction algorithm. Later an efficient algorithm that guarantees to construct an array whose number of rows never exceeds that upper bound was developed using the Stein–Lovász–Johnson paradigm [30, 33, 46], by calculating exact conditional expectations [4, 5]. Subsequently, the symmetric version of the Lovász local lemma [2, 24] was used to improve on these bounds [27]; see also [19, 38]. Various techniques using tiling [20, 53], entropy compression [25] and multiphase construction [42, 43] have further improved on these bounds.

Imposing some symmetry on the covering arrays led to similar improvements in the conditional expectation methods [13], but the best known asymptotic bounds arise by imposing the strong symmetry of CPHFs [15, 18]. In [15], three upper bounds on covering array numbers are derived, all using CPHFs. The first uses a conditional expectation method in the vein of Stein–Lovász–Johnson. The second uses the Lovász Local Lemma for CPHFs. The third, and best, is a conditional expectation method that constructs an array that has more factors than necessary and permits a small number of uncovered t-sets, so that deleting one column from each uncovered t-set yields the desired array. This process has been variously called 'random selection with postprocessing' [50], 'expurgation' [19], or 'oversampling' [17].

Here we extend this best known bound to obtain an oversampling conditional expectation bound for CPHFs of higher index. Suppose that the entries of an $n \times k$ array A are chosen uniformly at random from $\mathcal{V}_{t,q}$. Let T be a set of t columns of A. The probability that row ρ of A does not contain a covering t-set for T can easily be computed. The total number of t-sets is $\left(\frac{q^t-1}{q-1}\right)^t$, and the number that are covering t-sets is $\left(\frac{1}{q^t-1}\right)^t \prod_{i=0}^{t-1}(q^t-q^i)$. So within row ρ of A, the probability that the columns of T are *not* covering is

$$\phi_{t,q} := 1 - \frac{\prod_{i=0}^{t-1}(q^t-q^i)}{(q^t-1)^t} = 1 - \prod_{i=1}^{t-1}\frac{q^t-q^i}{q^t-1}.$$

In [15], it is shown that for all $q \geq 3$ and $t \geq 3$, $\frac{1}{q} < \phi_{t,q} \leq \frac{q+1}{q^2}$.

Because coverage in different rows is independent, the probability that T is $\mu^{=}$-covered is

$$\binom{n}{\mu}\phi_{t,q}^{n-\mu}(1-\phi_{t,q})^{\mu} = \phi_{t,q}^{n}\binom{n}{\mu}\left(\frac{1-\phi_{t,q}}{\phi_{t,q}}\right)^{\mu}.$$

Excluding all cases in which T is $\mu^{=}$-covered for $\mu < \lambda$, the probability that T is **not** λ-covered is

$$\phi_{t,q}^{n}\left[\sum_{\mu=0}^{\lambda-1}\binom{n}{\mu}\left(\frac{1-\phi_{t,q}}{\phi_{t,q}}\right)^{\mu}\right].$$

http://dx.doi.org/10.22108/ijgt.2023.137230.1836

There are $\binom{k}{t}$ choices for T. Using the union bound, the probability that at least one is **not** λ-covered is no more than

$$\psi_{k,t,n,\lambda} := \binom{k}{t}\phi_{t,q}^n\left[\sum_{\mu=0}^{\lambda-1}\binom{n}{\mu}\left(\frac{1-\phi_{t,q}}{\phi_{t,q}}\right)^\mu\right].$$

Indeed, by linearity of expectations, the expected number of t-sets in the random array A that are not λ-covered is exactly $\psi_{k,t,n,\lambda}$.

Via the basic probabilistic method [2], existence of a $\mathsf{CPHF}_\lambda(n;k,q,t)$ is guaranteed when $\psi_{k,t,n,\lambda} < 1$. Moreover, a conditional expectation construction yields an efficient algorithm to produce a CPHF that meets the bound [23].

Because coverage for each t-set of columns is only dependent on at most $\binom{k}{t} - \binom{k-t}{t}$ other t-sets, the symmetric version of the Lovász Local Lemma improves on the basic bound, requiring the weaker condition that

$$e\left[\binom{k}{t} - \binom{k-t}{t}\right]\phi_{t,q}^n\left[\sum_{\mu=0}^{\lambda-1}\binom{n}{\mu}\left(\frac{1-\phi_{t,q}}{\phi_{t,q}}\right)^\mu\right] < 1.$$

Then using the Moser-Tardos strategy [36], a $\mathsf{CPHF}_\lambda(n;k,q,t)$ can be generated in expected polynomial time (in the number of events) by resampling t-sets that are not λ-covered.

The best bound, however, arises from the basic probabilistic method by using oversampling (or expurgation). We derive this bound next. The essential idea to construct a $\mathsf{CPHF}_\lambda(n;k,q,t)$ follows:

(1) Choose $k' \geq k$.
(2) Construct a random $n \times k'$ array B.
(3) Let $\mathcal{T} = \{T_1, \ldots, T_\tau\}$ be the set of t-sets of columns that are not λ-covered in B.
(4) Choose $C = \{c_1, \ldots, c_\sigma\} \subset [1, k']$ so that $C \cap T_i \neq \emptyset$ for each $1 \leq i \leq \tau$.
(5) Delete all columns whose column index is in C.
(6) The resulting array is a $\mathsf{CPHF}_\lambda(n; k' - \sigma, q, t)$.

Naturally we require that $k' - \sigma \geq k$ for this to succeed, yet our only choice is that of k'. Now $\sigma \leq \tau$ because we can choose a single column index for each $T_i \in \mathcal{T}$ to form C. Hence it suffices if $k' - \tau \geq k$. Because B has been chosen uniformly at random, the expected value of τ is known to be $\psi_{k',t,n,\lambda}$. In order to choose k', we want $k' - \lfloor\psi_{k',t,n,\lambda}\rfloor \geq k$. Then to obtain the largest k, choose k' so that $\psi_{k',t,n,\lambda} - \psi_{k'-1,t,n,\lambda} \leq 1$ but $\psi_{k'+1,t,n,\lambda} - \psi_{k',t,n,\lambda} > 1$. The intuition is that k' should be increased by 1 when the number of additional uncovered t-sets shows an expected increase of no more than 1. Now

$$\begin{aligned}\psi_{k',t,n,\lambda} - \psi_{k'-1,t,n,\lambda} &= \left[\binom{k'}{t} - \binom{k'-1}{t}\right]\phi_{t,q}^n\left[\sum_{\mu=0}^{\lambda-1}\binom{n}{\mu}\left(\frac{1-\phi_{t,q}}{\phi_{t,q}}\right)^\mu\right] \\ &= \binom{k'-1}{t-1}\phi_{t,q}^n\left[\sum_{\mu=0}^{\lambda-1}\binom{n}{\mu}\left(\frac{1-\phi_{t,q}}{\phi_{t,q}}\right)^\mu\right]\end{aligned}$$

Theorem 3.1. *Let t, k, q, λ be positive integers, with $t \geq 2$ and $q \geq 2$ a prime power. A $\mathsf{CPHF}_\lambda(n;k,t,q)$ exists whenever*

$$\binom{\frac{t}{t-1}k}{t-1}\phi_{t,q}^n\left[\sum_{\mu=0}^{\lambda-1}\binom{n}{\mu}\left(\frac{1-\phi_{t,q}}{\phi_{t,q}}\right)^\mu\right] < 1.$$

http://dx.doi.org/10.22108/ijgt.2023.137230.1836

Proof. Let $\Theta = \left[\sum_{\mu=0}^{\lambda-1} \binom{n}{\mu} \left(\frac{1-\phi_{t,q}}{\phi_{t,q}}\right)^{\mu}\right]$. The number of t-sets of columns not λ-covered in a random $n \times \frac{t}{t-1}k$ array B is $\binom{\frac{t}{t-1}k}{t}\phi_{t,q}^{n}\Theta = \frac{\frac{t}{t-1}k}{t}\binom{\frac{t}{t-1}k-1}{t-1}\phi_{t,q}^{n}\Theta$. But $\binom{\frac{t}{t-1}k-1}{t-1}\phi_{t,q}^{n}\Theta < \binom{\frac{t}{t-1}k}{t-1}\phi_{t,q}^{n}\Theta < 1$, by hypothesis. Hence there exists an $n \times \frac{t}{t-1}k$ array B with at most $\frac{\frac{t}{t-1}k}{t} = \frac{k}{t-1}$ t-sets of columns that are not λ-covered. □

4. **The dependence on the index**

One could pursue the strategy in [23] to extract an asymptotic upper bound on the number of rows as a function of k and λ (for fixed t and q) from Theorem 3.1. However, our goal is to examine explicit sizes of CPHF_λs to understand the effect of oversampling and the relationship between the index λ and the number n of rows.

Denote by $\Theta_{n,t,q,\lambda}$ the quantity $\left[\sum_{\mu=0}^{\lambda-1} \binom{n}{\mu} \left(\frac{1-\phi_{t,q}}{\phi_{t,q}}\right)^{\mu}\right]$. Fix $k > t \geq 3$ and $q \geq 3$ a prime or prime power. Theorem 3.1 ensures that a $\mathsf{CPHF}(n; k, t, q)$ exists when

$$\binom{\frac{t}{t-1}k}{t-1}\phi_{t,q}^{n} < 1, \tag{4.1}$$

and a $\mathsf{CPHF}_\lambda(n'; k, t, q)$ exists when

$$\binom{\frac{t}{t-1}k}{t-1}\phi_{t,q}^{n'}\Theta_{n',t,q,\lambda} < 1. \tag{4.2}$$

When the inequality (4.1) holds, the inequality (4.2) also holds when

$$\phi_{t,q}^{n'-n}\Theta_{n',t,q,\lambda} = \phi_{t,q}^{n'-n}\left[\sum_{\mu=0}^{\lambda-1} \binom{n'}{\mu} \left(\frac{1-\phi_{t,q}}{\phi_{t,q}}\right)^{\mu}\right] < 1.$$

This yields an upper bound on the number of *additional* rows that suffice to ensure index λ rather than index one. Although this number appears to have no dependence on the number k of columns, we remind the reader that n grows logarithmically in k and $n' \geq n + \lambda - 1$. (Dougherty [21] establishes that as $\lambda \to \infty$, increasing λ by 1 increases the number of rows by a constant.)

It can (and does) happen that a bound obtained by CPHFs is not as tight as one obtained by a similar method applied to covering arrays (see §5). To justify our focus on covering perfect hash families, let us note that Table 1 reports the existence of a $\mathsf{CPHF}_{10}(39; 1000, 4, 4)$. Then by Lemma 2.1, there is a $\mathsf{CA}_{10}(9955; 4, 1000, 4)$. Contrast this with bounds obtained by choosing a covering array at random rather than a CPHF. The basic probabilistic method ensures the existence of a $\mathsf{CA}_{10}(13579; 4, 1000, 4)$, and employing oversampling yields a $\mathsf{CA}_{10}(12138; 4, 1000, 4)$. As anticipated, the symmetry imposed by CPHFs produces not just a more compact representation (and a more efficient construction algorithm), it also results in a worthwhile reduction in the number of rows needed. In our experience, this single example is typical when λ is 'small' and q is 'large'; see §5 for further discussion. It is also noteworthy that the best known bound on $\mathsf{CAN}_1(4, 1000, 4)$ is 3404 [29], and hence 10-fold coverage is achieved with less than three times the number of rows as simple coverage.

http://dx.doi.org/10.22108/ijgt.2023.137230.1836

TABLE 1. Upper bounds on the number n of rows in a $\mathsf{CPHF}_\lambda(n;k,4,4)$ via the basic probabilistic method and via oversampling, for various values of k and λ.

	Number k of columns									
λ	10^2		10^3		10^4		10^5		10^6	
1	13	11	21	17	28	23	36	28	44	34
2	16	14	24	20	32	26	40	32	48	38
3	19	17	27	23	35	29	43	35	51	41
5	24	21	32	28	41	34	49	41	57	47
10	34	32	44	39	53	46	62	53	71	60
15	44	41	55	50	65	57	74	64	83	72
25	63	60	75	69	86	77	96	85	106	93
50	107	103	121	114	134	124	146	134	157	143
100	190	185	207	199	223	211	237	223	251	234
200	349	342	371	360	391	376	409	391	425	404
500	809	800	842	826	870	849	894	869	917	888

TABLE 2. Upper bounds on the number n of rows in a $\mathsf{CPHF}_\lambda(n;k,4,7)$ via the basic probabilistic method and via oversampling, for various values of k and λ.

	Number k of columns									
λ	10^2		10^3		10^4		10^5		10^6	
1	21	19	35	31	48	43	62	55	76	67
2	24	23	38	35	52	47	67	59	80	71
3	27	26	42	38	56	50	70	63	84	75
5	32	31	48	44	63	57	77	69	91	82
10	44	43	61	57	77	71	92	84	107	97
15	55	53	73	69	89	83	105	97	121	110
25	75	73	95	90	113	106	130	121	146	135
50	122	120	145	140	166	158	185	175	203	191
100	210	207	238	231	262	253	285	273	306	292
200	377	373	412	404	442	430	469	454	494	477
500	855	850	904	893	945	930	982	963	1016	993

For applications, we are most concerned with the sizes of the covering perfect hash families whose existence is guaranteed. For certain values of t and q, we tabulate the upper bounds on n for various choices of k and λ. In the tables provided, for each choice of t, q, k, and λ we provide the upper bound from the basic probabilistic method first and then the upper bound obtained using oversampling from Theorem 3.1.

http://dx.doi.org/10.22108/ijgt.2023.137230.1836

TABLE 3. Upper bounds on the number n of rows in a $\mathsf{CPHF}_\lambda(n;k,7,4)$ via the basic probabilistic method and via oversampling, for various values of k and λ.

	Number k of columns									
λ	10^2		10^3		10^4		10^5		10^6	
1	9	8	14	11	19	15	24	19	29	23
2	11	10	16	14	22	18	27	22	32	25
3	13	12	19	16	24	20	29	24	34	28
5	17	15	22	20	28	24	33	28	39	32
10	25	23	31	28	37	33	43	37	49	42
15	33	31	39	36	46	41	52	46	58	51
25	47	45	55	51	62	57	69	62	75	67
50	82	80	91	87	100	93	107	100	115	106
100	149	146	160	155	170	163	180	170	188	177
200	278	274	293	286	305	296	317	305	327	314
500	656	650	677	667	694	681	710	694	724	706

TABLE 4. Upper bounds on the number n of rows in a $\mathsf{CPHF}_\lambda(n;k,25,4)$ via the basic probabilistic method and via oversampling, for various values of k and λ.

	Number k of columns									
λ	10^2		10^3		10^4		10^5		10^6	
1	5	5	8	7	11	9	14	11	17	13
2	7	6	10	8	13	11	16	13	19	15
3	8	7	11	10	14	12	18	14	21	17
5	11	10	14	13	17	15	21	17	24	20
10	17	16	21	19	24	22	28	24	31	27
15	23	22	27	25	31	28	34	31	38	34
25	35	34	39	37	43	40	47	43	51	46
50	63	62	69	66	73	70	77	73	82	76
100	119	117	125	122	130	126	136	130	140	134
200	228	226	235	232	242	237	248	242	254	247
500	550	547	560	555	569	562	577	569	585	575

Tables 1, 2, 3, and 4 give both upper bounds on n, for each

$$\lambda \in \{1,2,3,5,10,15,25,50,100,200,500\}$$

and $k \in \{100,1000,10000,100000,1000000\}$ in a $\mathsf{CPHF}_\lambda(n;k,q,t)$ with $(q,t) = (4,4)$, $(4,7)$, $(7,4)$, and (25.4), respectively. For each parameter selection, oversampling makes a clear improvement. One might conclude that these improvements are relatively small, but remember that each row of the CPHF underlies $q^t - 1$ rows of the covering array that it generates. In Table 2, for example, each row of

http://dx.doi.org/10.22108/ijgt.2023.137230.1836

a $\mathsf{CPHF}_\lambda(n; k, 4, 7)$ generates 16,383 rows in the covering array. Although the relative improvement appears to be small, the reduction in the size of the covering array generated is nonetheless substantial.

What is more striking is the relatively small number of additional rows needed to obtain λ-fold coverage. Examining $k = 10^4$ in Table 2, for example, twice the number of rows that suffice for $\lambda = 1$ is enough to ensure at least 15-fold coverage, while four times the number suffices to provide at least 50-fold coverage.

5. Limitations of CPHFs

Dougherty [21] applies similar probabilistic methods, both to covering arrays directly and to covering arrays from CPHFs, with the goal of obtaining the best asymptotic bounds for higher index. He remarks that "bounds achieved for the covering arrays resulting from [covering perfect] hash families are not as asymptotically strong as the results we have obtained [for covering arrays directly]."

TABLE 5. Upper bounds on the number N of rows in a $\mathsf{CA}_\lambda(N; 4, k, 4)$ via oversampling for CAs directly (first entry) and via CPHFs (second entry), for various k and λ.

	Number k of columns									
λ	10^2		10^3		10^4		10^5		10^6	
1	4708	2806	6475	4336	8240	5866	10005	7141	11770	8671
2	5504	3572	7342	5102	9162	6632	10972	8162	12775	9692
3	6181	4338	8079	5868	9948	7398	11798	8928	13635	10458
5	7370	5360	9375	7145	11328	8675	13250	10460	15149	11990
10	9919	8170	12138	9955	14266	11740	16337	13525	18367	15310
25	16328	15325	19007	17620	21522	19660	23927	21700	26254	23740
50	25658	26315	28884	29120	31862	31670	34674	34220	37365	36515
100	42626	47275	46646	50845	50301	53905	53712	56965	56944	59770

Look at Table 5. When $\lambda \leq 25$, the bound via CPHFs always improves on that from CAs directly (sometimes dramatically). Yet when $\lambda = 100$, the route via CPHFs discussed here yields weaker bounds than treating covering arrays directly.

What is the explanation for this behaviour? When a t-set of columns is not covering in a row of a CPHF, the corresponding rows of the generated array nevertheless cover some (but not all) t-way interactions. As the number of rows with non-covering sets increases, the partial coverage of interactions from each of these rows may combine to cover all t-way interactions, despite no single row of the CPHF having a covering set! This partial coverage is ignored in the all-or-nothing analysis carried out for CPHFs here. When the number of symbols is small, the probability that a row contains a non-covering set in a particular t-set of columns increases. Also as the index increases (and hence the number of rows in the CPHF increases), the number of occurrences of non-covering sets increases. Indeed when q is 'small' and/or when λ is 'large', ignoring the partial coverage from non-covering sets results in bounds

http://dx.doi.org/10.22108/ijgt.2023.137230.1836

that are weak. Even when the index is 1, it was earlier observed that the CPHF techniques fare poorly when $q = 2$ [15].

It is feasible to improve the probabilistic analysis to account for the partial coverage that has been thus far ignored. We plan to treat this in a later paper. Despite this limitation of the current analysis, we believe that the current bounds from CPHFs serve their intended purpose of providing a sensible target for comparison to evaluate the promise of proposed algebraic constructions, particularly when the index is small and the number of symbols is not too small.

6. Concluding Remarks

The probabilistic approach leads to efficient and effective construction algorithms for covering perfect hash families, and hence to covering arrays, in a well-understood manner. However, in closing we focus on a different direction. We have seen that adopting the algebraic framework of finite fields in certain cases improves both the sizes of the covering arrays generated and the efficiency of their construction. Nevertheless, we have employed probabilistic analysis, choosing arrays at random. Can we choose the covering perfect hash families deterministically in general to achieve arrays with fewer (or at least, no more) rows? In one special case, that of a $\mathsf{CPHF}_1(n; 25^2 + 25 + 1, 25, 3)$, Raaphorst *et al.* [41] give a construction with $n = 2$ from the desarguesian projective plane. By way of contrast, the basic probabilistic method yields a $\mathsf{CPHF}_1(6; 25^2 + 25 + 1, 25, 3)$ which oversampling improves to a $\mathsf{CPHF}_1(5; 25^2 + 25 + 1, 25, 3)$. Hence, even when probabilistic techniques yield the best general results, there can be much room for improvement in specific situations. It would be of substantial practical interest to discover other group-theoretic, algebraic, or geometric connections and thereby devise even more efficient methods for the construction of covering arrays with fewer rows.

Acknowledgments

Research supported by the National Science Foundation under Grant 1813729. The author thanks Ryan Dougherty, Dimitris Simos, and Michael Wagner for helpful discussions about covering arrays with higher index. Thanks also to a referee whose comments clarified the presentation

References

[1] Y. Akhtar, C. J. Colbourn and V. R. Syrotiuk, Mixed covering, locating, and detecting arrays via cyclotomy, *Proceedings of the 52nd Southeastern Conference on Combinatorics, Graph Theory and Computing, to appear.*

[2] N. Alon and J. H. Spencer, *The probabilistic method*, Third edition, With an appendix on the life and work of Paul Erdős. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., Hoboken, NJ, 2008.

[3] R. C. Bose and K. A. Bush, Orthogonal arrays of strength two and three, *Ann. Math. Statistics*, **23** (1952) 508–524.

[4] R. C. Bryce and C. J. Colbourn, The density algorithm for pairwise interaction testing, *Softw. Test. Verification Reliab.*, **17** (2007) 159–182.

[5] R. C. Bryce and C. J. Colbourn, A density-based greedy algorithm for higher strength covering arrays, *Softw. Test. Verification Reliab.*, **19** (2009) 37–53.

[6] K. A. Bush, Orthogonal arrays of index unity, *Ann. Math. Statistics*, **23** (1952) 426–434.

[7] M. A. Chateauneuf, C. J. Colbourn and D. L. Kreher, Covering arrays of strength three, *Des. Codes Cryptogr.*, **16** (1999) 235–242.

[8] M. A. Chateauneuf and D. L. Kreher, On the state of strength-three covering arrays, *J. Combin. Des.*, **10** (2002) 217–238.

[9] D. M. Cohen, S. R. Dalal, M. L. Fredman and G. C. Patton, The AETG system: An approach to testing based on combinatorial design, *IEEE Trans. Software Eng.*, **23** (1997) 437–444.

[10] C. J. Colbourn, Covering array tables: $2 \leq v \leq 25$, $2 \leq t \leq 6$, $t \leq k \leq 10000$, 2005-23. https://www.public.asu.edu/ccolbou/src/tabby.

[11] C. J. Colbourn, Strength two covering arrays: existence tables and projection, *Discrete Math.*, **308** (2008) 772–786.

[12] C. J. Colbourn, Covering arrays from cyclotomy, *Des. Codes Cryptogr.*, **55** (2010) 201–219.

[13] C. J. Colbourn, Conditional expectation algorithms for covering arrays, *J. Combin. Math. Combin. Comput.*, **90** (2014) 97–115.

[14] C. J. Colbourn and E. Lanus, Subspace restrictions and affine composition for covering perfect hash families, *Art Discrete Appl. Math.*, **1** (2018) 19 pp.

[15] C. J. Colbourn, E. Lanus and K. Sarkar, Asymptotic and constructive methods for covering perfect hash families and covering arrays, *Des. Codes Cryptogr.*, **86** (2018) 907–937.

[16] C. J. Colbourn and D. W. McClary, Locating and detecting arrays for interaction faults, *J. Comb. Optim.*, **15** (2008) 17–48.

[17] C. J. Colbourn and V. R. Syrotiuk, On a combinatorial framework for fault characterization, *Math. Comput. Sci.*, **12** (2018) 429–451.

[18] S. Das and T. Mészáros, Small arrays of maximum coverage, *J. Combin. Des.*, **26** (2018) 487–504.

[19] D. Deng, D. R. Stinson and R. Wei, The Lovász local lemma and its applications to some combinatorial arrays, *Des. Codes Cryptogr.*, **32** (2004) 121–134.

[20] M. S. Donders and A. P. Godbole, t-covering arrays generated by a tiling probability model, *Congr. Numer.*, **218** (2013) 111–116.

[21] R. E. Dougherty, An asymptotically optimal bound for covering arrays of higher index, 2022. arXiv: 2211.01209.

[22] R. E. Dougherty and C. J. Colbourn, *Perfect hash families: the generalization to higher indices*, Discrete mathematics and applications, Springer Optim. Appl., **165**, Springer, Cham, 2020 177–197.

[23] R. E. Dougherty, K. Kleine, M. Wagner, C. J. Colbourn, and D. E. Simos, Algorithmic methods for covering arrays of higher index, *J. Comb. Optim.*, **45** (2023) 21 pp.

[24] P. Erdős and L. Lovász, *Problems and results on* 3*-chromatic hypergraphs and some related questions*, Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), **I, II, III**, Colloq. Math. Soc. János Bolyai, **10**, North-Holland, Amsterdam-London, 1975 609–627.

http://dx.doi.org/10.22108/ijgt.2023.137230.1836

[25] N. Francetić and B. Stevens, Asymptotic size of covering arrays: an application of entropy compression, *J. Combin. Des.*, **25** (2017) 243–257.

[26] L. Gargano, J. Körner and U. Vaccaro, Sperner capacities, *Graphs Combin.*, **9** (1993) 31–46.

[27] A. P. Godbole, D. E. Skipper and R. A. Sunley, t-covering arrays: upper bounds and Poisson approximations, *Combin. Probab. Comput.*, **5** (1996) 105–118.

[28] A. Hartman, *Software and hardware testing using combinatorial covering suites*, Graph theory, combinatorics and algorithms, Springer, New York, 2005 237–266.

[29] I. Izquierdo-Marquez, J. Torres-Jimenez, B. Acevedo-Juárez and H. Avila-George, A greedy-metaheuristic 3-stage approach to construct covering arrays, *Inf. Sci.*, **460-461** (2018) 172–189.

[30] D. S. Johnson, Approximation algorithms for combinatorial problems, *J. Comput. System Sci.*, **9** (1974) 256–278.

[31] D. R. Kuhn, R. Kacker and Y. Lei, *Introduction to Combinatorial Testing*, CRC Press, Boca Raton, FL, 2013.

[32] J. R. Lobb, C. J. Colbourn, P. Danziger, B. Stevens and J. Torres-Jimenez, Cover starters for covering arrays of strength two, *Discrete Math.*, **312** (2012) 943–956.

[33] L. Lovász, On the ratio of optimal integral and fractional covers, *Discrete Math.*, **13** (1975) 383–390.

[34] C. Martínez, L. Moura, D. Panario and B. Stevens, Locating errors using ELAs, covering arrays, and adaptive testing algorithms, *SIAM J. Discrete Math.*, **23** (2009/10) 1776–1799.

[35] K. Meagher and B. Stevens, Group construction of covering arrays, *J. Combin. Des.*, **13** (2005) 70–77.

[36] R. A. Moser and G. Tardos, A constructive proof of the general Lovász local lemma, *J. ACM*, **57** (2010) 15 pp.

[37] L. Moura, G. L. Mullen and D. Panario, Finite field constructions of combinatorial arrays, *Des. Codes Cryptogr.*, **78** (2016) 197–219.

[38] L. Moura, S. Raaphorst and B. Stevens, Upper bounds on the sizes of variable strength covering arrays using the Lovász local lemma, *Theoret. Comput. Sci.*, 800 (2019) 146–154.

[39] C. Nie and H. Leung, A survey of combinatorial testing, *ACM Computing Surveys*, **43** (2011) 1–29.

[40] D. Panario, M. Saaltink, B. Stevens and D. Wevrick, An extension of a construction of covering arrays, *J. Combin. Des.*, **28** (2020) 842–861.

[41] S. Raaphorst, L. Moura and B. Stevens, A construction for strength-3 covering arrays from linear feedback shift register sequences, *Des. Codes Cryptogr.*, **73** (2014) 949–968.

[42] K. Sarkar and C. J. Colbourn, Upper bounds on the size of covering arrays, *SIAM J. Discrete Math.*, **31** (2017) 1277–1293.

[43] K. Sarkar and C. J. Colbourn, Two-stage algorithms for covering array construction, *J. Combin. Des.*, **27** (2019) 475–505.

[44] K. Sarkar, C. J. Colbourn, A. De Bonis and U. Vaccaro, Partial covering arrays: algorithms and asymptotics, *Theory Comput. Syst.*, **62** (2018) 1470–1489.

[45] G. B. Sherwood, S. S. Martirosyan and C. J. Colbourn, Covering arrays of higher strength from permutation vectors, *J. Combin. Des.*, **14** (2006) 202–213.

[46] S. K. Stein, Two combinatorial covering theorems, *J. Combinatorial Theory Ser. A*, *16* (1974) 391–397.

http://dx.doi.org/10.22108/ijgt.2023.137230.1836

[47] J. Torres-Jimenez and I. Izquierdo-Marquez, A simulated annealing algorithm to construct covering perfect hash families, *Math. Probl. Eng.*, **2018** (2018) 14 pp.

[48] J. Torres-Jimenez and I. Izquierdo-Marquez, Improved covering arrays using covering perfect hash families with groups of restricted entries, *Appl. Math. Comput.*, **369** (2020) 17 pp.

[49] G. Tzanakis, L. Moura, D. Panario and B. Stevens, Constructing new covering arrays from LFSR sequences over finite fields, *Discrete Math.*, **339** (2016) 1158–1171.

[50] E. van den Berg, E. Candès, G. Chinn, C. Levin, P. D. Olcott and C. Sing-Long, Single-photon sampling architecture for solid-state imaging sensors, *Proc. Natl. Acad. Sci. USA*, **110** (2013) 2752–2761.

[51] M. Wagner, C. J. Colbourn and D. E. Simos, In-parameter-order strategies for covering perfect hash families, *Appl. Math. Comput.*, **421** (2022) 21 pp.

[52] R. A. Walker II and C. J. Colbourn, Tabu search for covering arrays using permutation vectors, *J. Statist. Plann. Inference.* **139** (2009) 69–80.

[53] R. Yuan, Z. Koch and A. P. Godbole, Covering array bounds using analytical techniques, *Congr. Numer.*, **222** (2014) 65–73.

Charles J. Colbourn

Computing and Augmented Intelligence, Arizona State University, PO Box 878809, Tempe, AZ, 85287-8809, U.S.A.

Email: colbourn@asu.edu

http://dx.doi.org/10.22108/ijgt.2023.137230.1836