Information and Computation 301 (2024) 105202

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/yinco

Detecting arrays for effects of multiple interacting factors

Charles J. Colbourn *, Violet R. Syrotiuk

Computing and Augmented Intelligence, Arizona State University, P.O. Box 878809, Tempe, AZ 85287, USA

ARTICLE INFO

Article history:
Received 10 September 2019
Received in revised form 31 May 2023
Accepted 29 January 2024
Available online 31 July 2024

Keywords:
Combinatorial testing
Detecting array
Covering array
Finite field

ABSTRACT

Detecting arrays provide test suites for complex engineered systems in which many factors interact. The determination of which interactions have a significant impact on system behaviour requires not only that each interaction appear in a test, but also that its effect can be distinguished from those of other significant interactions. In this paper, compact representations of detecting arrays using vectors over the finite field are developed. Covering strong separating hash families exploit linear independence over the field, while the weaker elongated covering perfect hash families permit some linear dependence. For both, probabilistic analyses are employed to establish effective upper bounds on the number of tests needed in a detecting array for a wide variety of parameters. The analyses underlie efficient algorithms for the explicit construction of detecting arrays.

1. Introduction

Complex engineered systems are pervasive. Evidently a major design goal is to ensure that the system, once deployed, not only operates correctly but also meets specified performance objectives. The operation of such a system is dictated by many *factors*; some are controllable by the system operator, while others are environmental. We consider systems in which the number of factors, k, is known. The factors are $F_1, \dots, F_k$. An individual factor may be set to a level, such as 'on' or 'off'; 'green', 'blue', or 'red'; or a numerical value such as the temperature, for example. When a set of possible levels is infinite, category partitioning identifies a discrete subset of allowed levels that are meant to be representative [40]. The selection of categories and representative levels can be challenging (e.g., [26]), but it is not possible to check each of infinitely many levels. Here we assume that, for a specific system, each factor F_i has a set $S_i = \{v_{i1}, \dots, v_{is_i}\}$ of s_i allowed *levels*. The *type* of the system is $(s_1, \dots, s_k)$.

Setting each factor F_i to a level $v_i \in S_i$ yields a particular configuration of the system; we call $(v_1, \dots, v_k)$ a *test* or a *run*. Executing a test (that is, executing the system as configured by the test) results in a *test response*. In some settings the response is the binary 'failed' or 'operating'; when it is more complicated, typically one classifies the responses as binary. For example, when the response is a metric such as throughput, a threshold value is chosen so that performance that is at least the threshold value is acceptable and performance below is unacceptable. A primary objective is to determine how the factors and their levels impact the correctness or performance of the system by examining the test responses from a number of tests. Here we consider nonadaptive testing, in which all tests are selected before any are executed. The tests selected form a *test suite* or *experimental design*; the *size* of the test suite is the number of tests that it contains. To implement testing in this way, two main problems must be addressed:

* Corresponding author.
E-mail addresses: charles.colbourn@asu.edu (C.J. Colbourn), syrotiuk@asu.edu (V.R. Syrotiuk).

https://doi.org/10.1016/j.ic.2024.105202

Design: Given k and $(S_1, \ldots, S_k)$, choose a number N of tests, and a test suite of size N.
Analysis: Execute all N tests to obtain the N test responses. From the test responses (and without further testing), determine the causes of failure (if any) in the system.

Design and analysis are tied together by the possible causes of failure. To give a simple example, if setting a particular factor to a specific level triggers a system failure, the design step must emit at least one test in which the particular factor is set to the specific level, for otherwise the analysis receives no evidence of the failure. It is therefore necessary that the potential causes of failure be known prior to design.

Here we suppose that failures (when present) can be attributed to the level selections for a 'small' number of factors. This is the province of combinatorial interaction testing; see [31,32,37,38] and references therein, particularly for the empirical justification for considering level-wise interactions among few factors. To make matters more precise, when $\{i_1, \ldots, i_t\} \subseteq \{1, \ldots, k\}$ and $\sigma_{i_j} \in S_{i_j}$, the set $T = \{(i_j, \sigma_{i_j}) : 1 \le j \le t\}$ is an *interaction* of *strength t*, or *t-way interaction*. A test $(\nu_1, \ldots, \nu_k)$ *covers* the t-way interaction $T = ((i_1, \sigma_{i_1}), \ldots (i_t, \sigma_{i_t}))$ when $\nu_{i_j} = \sigma_{i_j}$ for $1 \le j \le t$.

Often a test suite is written as an $N \times k$ array $A = (a_{i,j})$ in which $a_{i,j} \in S_j$ when $1 \le i \le N$ and $1 \le j \le k$. For a test suite $A = (a_{i,j})$ of size N and type $(s_1, \ldots, s_k)$ and a t-way interaction $T = \{(i_j, \sigma_{i_j}) : 1 \le j \le t\}$, denote by $\rho_A(T)$ the set of indices of rows of A in which T is covered. That is,

$$\rho_A(T) := \{r : a_{r,i_j} = \sigma_{i_j}, 1 \le j \le t\}$$

If an interaction T has $\rho_A(T) = \emptyset$, no test in A incorporates the possible effect of T; if T would trigger a fault, no evidence of this is found. Hence the basis of combinatorial interaction testing is that every interaction that may cause a fault (typically, every interaction of strength at most t) is covered in at least one test of A. Formally, a *covering array* $\mathsf{CA}_\delta(N; t, k, (s_1, \ldots, s_k))$ of *size N*, *type* $(s_1, \ldots, s_k)$, *index* δ, and *strength t* is an $N \times k$ array A for which every t-way interaction T has $|\rho_A(T)| \ge \delta$. The notation $\mathsf{CA}_\delta(N; t, k, v)$ is used when the type is uniform. Covering arrays have been studied extensively when $\delta = 1$ (see [7,9,27,31,38,53] and references therein), but more recently for $\delta > 1$ as well [22].

When a covering array of strength t is adopted as the test design, the test responses certify the presence or absence of a faulty interaction of strength at most t. Indeed when $\delta > 1$, this can be done even when a small number of test responses are missing or inaccurate. However, isolating particular interactions causing faults may not be possible. A simple example arises when two interactions T and T' have $\rho_A(T) = \rho_A(T')$. If the test responses exhibit faulty behaviour exactly in rows $\rho_A(T)$, we cannot tell whether T causes the fault or T' does (or both).

When $\mathcal{T}$ is a set of interactions, $\rho_A(\mathcal{T})$ denotes $\cup_{T\in\mathcal{T}} \rho_A(T)$. Suppose that the set of interactions triggering a fault is $\mathcal{T}$. Then the set of tests whose responses show a fault is precisely $\rho_A(\mathcal{T})$. The analysis of the test results asks that we determine $\mathcal{T}$ given $\rho_A(\mathcal{T})$. Without limitations on $\mathcal{T}$, this cannot be done. Consider the situation when every test response indicates a fault, for example. More generally, see [35]. For this reason, we only ask to determine $\mathcal{T}$ given $\rho_A(\mathcal{T})$ when there are at most d faulty interactions, i.e., when $|\mathcal{T}| \le d$. (This restriction is analogous to the one made in the related problem of combinatorial group testing [23,28].)

In the remainder of the paper we treat testing arrays to isolate faults when few faults of small strength are present. In §2, we provide the necessary combinatorial foundation for detecting arrays. In §3, we describe the construction of uniform detecting arrays when the number of symbols is a prime power. First we introduce elongated covering perfect hash families to produce covering arrays of index greater than 1. Then we enforce further conditions to define covering strong separating hash families to produce detecting arrays. In §4, we explore probabilistic methods for both types of covering hash family and illustrate the upper bounds obtained on the corresponding detecting arrays. We further comment on efficient construction algorithms meeting the bounds obtained. Conclusions are given in §5.

2. Detecting arrays

Following [11,12,44], we formally define certain test suites. Let $\mathscr{I}_t$ denote the set of t-way interactions for an array of type $(s_1, \ldots, s_k)$. An $N \times k$ array A of type $(s_1, \ldots, s_k)$ is $(\overline{d}, t, \delta)$*-locating* if

$$|\rho_A(\mathcal{R}) \cap \rho_A(\mathscr{T})| < \delta \Leftrightarrow \mathcal{R} = \mathcal{T} \text{ whenever } \mathscr{R}, \mathscr{T} \subseteq \mathscr{I}_t, |\mathscr{R}| \le d, \text{ and } |\mathscr{T}| \le d.$$

Numerous variants on this basic theme are developed in [11]; except when d is 'small', no such array exists [11,35]. For this reason, in [29], the definition is relaxed to require that the condition holds only when $\mathcal{R}$ and $\mathcal{T}$ are *distinguishable*, i.e., there exists an array B with $\rho_B(\mathcal{R}) \ne \rho_B(\mathscr{T})$. This agrees with the original definition when d is small enough. In principle, when $\delta \ge 1$, test responses from such a locating array enable one to uniquely determine the set of interactions that trigger faults provided that there are at most d, and each has strength t. However, in practice no efficient analysis algorithm is known. Naturally, any interaction that is covered in a test whose response indicates no fault cannot itself trigger a fault. But using a locating array, this may not suffice to isolate the faults.

In order to obtain an efficient analysis technique, we strengthen the conditions on the design, again following [11]. An $N \times k$ array A of type $(s_1, \ldots, s_k)$ is (d, t, δ)*-detecting* if

$$|\rho_A(T) \setminus \rho_A(\mathscr{T})| < \delta \Leftrightarrow T \in \mathscr{T} \text{ whenever } \mathscr{T} \subseteq \mathscr{I}_t \text{ and } |\mathscr{T}| = d.$$

The notation $\mathsf{DA}_\delta(N; d, t, k, (s_1, \ldots, s_k))$ is used when different factors may have different numbers of levels (i.e., a *mixed* detecting array). Because a covering array $\mathsf{CA}_\delta(N; t, k, (s_1, \ldots, s_k))$ is equivalent to a $\mathsf{DA}_\delta(N; 0, t, k, (s_1, \ldots, s_k))$, detecting arrays enforce coverage. The notation is simplified to $\mathsf{DA}_\delta(N; d, t, k, v)$ when all factors have the same number, v, of levels (i.e., the array is *uniform*). The original definition from [11] sets $\delta = 1$, and it is relaxed in [30] to enforce the condition only when T is distinguishable from $\mathcal{T}$. Rows in $\rho_A(T) \setminus \rho_A(\mathscr{T})$ are *witnesses* for T that are not masked by interactions in $\mathscr{T}$; the number of witnesses is the *separation* of the detecting array [44].

To support an operation known as fusion (see, e.g., [17]) for detecting arrays, in [13,14,16] a further parameter (corroboration) is defined; however we do not pursue that in this paper.

Locating and detecting arrays have been employed in practical testing applications; examples include [3,19,43,44,49]. Moreover, it has been argued that they exhibit desirable statistical properties as experimental designs for level-wise screening [1].

Our interest here is in the effective construction of detecting arrays with few rows. A connection with covering arrays (from [11]) gives us a launching pad:

Lemma 2.1. *A* $\mathsf{CA}_\lambda(N; t, k, v)$ *is*

1. *a* $\mathsf{DA}_\delta(N; d, t-d, k, v, 1)$ *with* $\delta = \lambda(v-d)v^{d-1}$, *and*
2. *a* $\mathsf{DA}_\delta(N; d, t-d, k, v, v-d)$ *with* $\delta = \lambda(d+1)^{d-1}$

whenever $1 \le d < \min(t, v)$.

Beyond the construction from covering arrays of higher strength, relatively few constructions are available. When the number of factors is very small, see [46–48,50]. For larger values of k, algorithmic methods are developed in [44] that are inspired by probabilistic methods using conditional expectations (similar to [5] for covering arrays) and random resampling (similar to [18] for covering arrays). Although these methods produce $(1, t)$-mixed detecting arrays for a variety of separation values, they have not been applied for $d > 1$. For an extension to larger d for locating (but not detecting) arrays, see [33].

When $d = t = 1$, detecting arrays can be constructed via a correspondence with a so-called Sperner partition system [6,25,34,36]. When $t = 1$ and $d \ge 1$, probabilistic methods are explored in [13,16]. An algebraic approach to treat higher strengths is introduced in [15] and further generalized in §3. Related algebraic approaches are discussed in [14] and in [2]. See [12,30] for pointers to other work on detecting arrays.

3. Constructions over finite fields

3.1. Covering perfect hash families

Sherwood *et al.* [45] developed a construction of covering arrays using an analogue of perfect hash families whose entries are vectors over a finite field. We employ a generalization of their definitions from [18]. Let q be a prime power, and 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 $\mathbb{F}_q$, and let $\mathcal{T}_{t,q}$ be the set of all column vectors of length t with entries from $\mathbb{F}_q$, not all 0. As noted in [45], when $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$ satisfies $\mathbf{x}_i \in \mathcal{T}_{t,q}$ for $1 \le i \le t$, the array $A = (a_{ij})$ in which $a_{ij} = \mathbf{r}_i \mathbf{x}_j$ (the dot 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.

For any nonzero $\mu \in \mathbb{F}_q$, substituting $\mu \mathbf{x}_i$ for $\mathbf{x}_i$ does not alter the non-singularity. Define $\langle \mathbf{x} \rangle = \{\mu \mathbf{x} : \mu \in \mathbb{F}_q, \mu \neq 0\}$. Let $\mathcal{V}_{t,q}$ be a set of representatives of $\{\langle \mathbf{x} \rangle : \mathbf{x} \in \mathcal{T}_{t,q}\}$. Then $|\mathcal{V}_{t,q}| = \frac{q^t-1}{q-1} = \sum_{i=0}^{t-1} q^i$.

A *covering perfect hash family* $\mathsf{CPHF}(n; k, q, t)$ is an $n \times k$ array $C = (\mathbf{c}_{ij})$ with entries from $\mathcal{V}_{t,q}$ so that, for every set $\{\gamma_1, \ldots, \gamma_t\}$ of distinct column indices, there is at least one row index ρ of C for which $[\mathbf{c}_{\rho\gamma_1} \cdots \mathbf{c}_{\rho\gamma_t}]$ is nonsingular; this is a *covering t-set* and the t-set of columns is *covered*.

Suppose that C is a $\mathsf{CPHF}(n; k, q, t)$. Then there exists a $\mathsf{CA}(n(q^t-1)+1; t, k, q)$. The proof is straightforward [18]: Replace each entry $\mathbf{c}_{ij}$ of C by the column vector obtained by multiplying $\mathbf{c}_{ij}$ by each $\mathbf{r}_\ell \in \mathcal{R}_{t,q}$. This produces a $\mathsf{CA}(nq^t; t, k, q)$; because the all-zero row appears (at least) n times, $n-1$ copies can be removed.

Because this CPHF construction of covering arrays employs a compact representation of certain covering arrays as covering perfect hash families, in practice it makes feasible the explicit construction of covering arrays for 'large' k and t [18,51,52]. Surprisingly, the imposition of such structure yields covering array numbers among the best known at present. Indeed CPHFs lead to the best current asymptotic existence upper bounds for covering array numbers with fixed strength t [18,20] and to efficient (and practical) algorithms to construct covering arrays realizing the given bounds [10,18]. Specific constructions of CPHFs also arise from constructions in projective geometries [41,54].

Combining this approach to construct covering arrays with Lemma 2.1 produces detecting arrays with large separation. Thus it appears that we have found a method to make detecting arrays with desired separation. But a closer look reveals that when d and v are large, the route via covering arrays results in far too many rows. We treat this next.

3.2. Covering strong separating hash families

An approach to construct detecting arrays over the finite field is outlined in [15]. We refine that approach here. Let $d \geq 0$ and $t \geq 2$. As before, let $\mathcal{R}_{t+1,q} = \{\mathbf{r}_0, \dots, \mathbf{r}_{q^{t+1}-1}\}$ be the set of all (row) vectors of length $t+1$ with entries from $\mathbb{F}_q$. For a subset S of $\mathbb{F}_q$ of cardinality g, let $\mathcal{R}_{t+1,q,g}$ be the set of all vectors in $\mathcal{R}_{t+1,q}$ whose entry in coordinate $t+1$ belongs to S. Let $\mathcal{W}_{t+1,q}$ be a set of representatives of all column vectors of length $t+1$ with entries from $\mathbb{F}_q$, so that at least one of the first t coordinates is nonzero. (The latter condition eliminates precisely one vector from $\mathcal{V}_{t+1,q}$.) When $\mathbf{v} \in \mathcal{W}_{t+1,q}$, denote by $\partial(\mathbf{v})$ the column vector of length t obtained from $\mathbf{v}$ by deleting the entry in coordinate $t+1$.

An *elongated covering perfect hash family* $\mathsf{ECPHF}_\delta(n; k, q, t)$ is an $n \times k$ array $C = (\mathbf{c}_{ij})$ with entries from $\mathcal{W}_{t+1,q}$ so that, for every set $\{\gamma_1, \dots, \gamma_t\}$ of distinct column indices, there are at least δ row indices ρ of C for which $[\partial(\mathbf{c}_{\rho\gamma_1}) \cdots \partial(\mathbf{c}_{\rho\gamma_t})]$ is nonsingular.

Lemma 3.1. *Let q be a prime power, and let t, k, and g be integers with $g \leq q$. Whenever an $\mathsf{ECPHF}_\delta(n; k, q, t)$ exists, a $\mathsf{CA}_{\delta g}(ngq^t; t, k, q)$ exists.*

Proof. Let $\mathcal{R} = \mathcal{R}_{t+1,q,g} = \{\mathbf{r}_1, \dots, \mathbf{r}_{gq^t}\}$ be a set of gq^t row vectors. Let $C = (\mathbf{c}_{ij})$ be an $\mathsf{ECPHF}_\delta(n; k, q, t)$. Form a $ngq^t \times k$ array A with rows indexed by $\{1, \dots, gq^t\} \times \{1, \dots, n\}$ and columns indexed by $\{1, \dots, k\}$. In the cell in row (σ, ρ) and column κ of A, place the entry $\mathbf{r}_\sigma \mathbf{c}_{\rho,\kappa}$. Then A has ngq^t rows, k columns, and q symbols. For a row ρ of C, denote by A_ρ the $gq^t \times k$ array generated by row ρ. We show that

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_n \end{pmatrix}$$

is a $\mathsf{CA}_{\delta g}(ngq^t; t, k, q)$.

Let $T = \{(\gamma_1, \nu_1), \dots, (\gamma_t, \nu_t)\}$ be a t-way interaction. There are δ rows $D = \{\ell_1, \dots, \ell_\delta\}$ of C in which $[\partial(\mathbf{c}_{\ell_j\gamma_1}) \cdots \partial(\mathbf{c}_{\ell_j\gamma_t})]$ is nonsingular. For each row $\ell_j \in D$, A_{ℓ_j} contains exactly g rows that cover T, because the unique solution $\mathbf{r}$ to

$$\mathbf{r}[\partial(\mathbf{c}_{\ell\gamma_1}) \cdots \partial(\mathbf{c}_{\ell\gamma_t})] = (\nu_1, \cdots, \nu_t)$$

yields g solutions for $\mathcal{R}$. It follows that $|\rho_{A_\ell}(T)| = g$ whenever $\ell \in D$, and hence that $|\rho_A(T)| = g\delta$. □

Lemma 3.1 can increase coverage of interactions, but does not guarantee that different interactions appear in different sets of rows. To obtain detecting arrays, we can impose a further condition. A *covering strong separating hash family* $\mathsf{CSSHF}_\delta(n; k, q, d, t)$ is an $\mathsf{ECPHF}_\delta(n; k, q, t)$ so that, for every set $\{\gamma_1, \dots, \gamma_t\} \cup \{a_1, \dots, a_d\}$ of distinct column indices, there are at least δ row indices ρ of C for which $[\mathbf{c}_{\rho\gamma_1} \cdots \mathbf{c}_{\rho\gamma_t} \mathbf{c}_{\rho a_i}]$ is nonsingular whenever $1 \leq i \leq d$. The nomenclature combines the notions of 'strong separating' hash families [39,42] and CPHFs. Increased separation can arise both by selecting larger δ or by increasing g (when possible).

The following is straightforward:

Proposition 3.2. *Let $C = (\mathbf{c}_{ij})$ be an $n \times k$ array with entries from $\mathcal{W}_{t+1,q}$. Then when $d \geq 1$ and C is a $\mathsf{CSSHF}_\delta(n; k, q, d, t)$, C is both a $\mathsf{CSSHF}_\delta(n; k, q, d-1, t)$ and a $\mathsf{CPHF}_\delta(n; k, q, t+1)$.*

Theorem 3.3. *Let q be a prime power, and let t, d, and k be integers with $t + d \leq k$ and $1 \leq d < q$. Whenever a $\mathsf{CSSHF}_\delta(n; k, q, d, t)$ exists and $d < g \leq q$, a $\mathsf{DA}_{\delta(g-d)}(ngq^t; d, t, k, q)$ exists.*

Proof. Let $\mathcal{R} = \mathcal{R}_{t+1,q,g} = \{\mathbf{r}_1, \dots, \mathbf{r}_{gq^t}\}$ be a set of gq^t row vectors. Let $C = (\mathbf{c}_{ij})$ be a $\mathsf{CSSHF}_\delta(n; k, q, d, t)$. Because C is an $\mathsf{ECPHF}_\delta(n; k, q, t)$, we can construct $A_1, \dots, A_n$ and A as in the proof of Lemma 3.1. We shall show that A is a $\mathsf{DA}_{\delta(g-d)}(ngq^t; d, t, k, q)$.

Consider a t-way interaction $T = \{(\gamma_1, \nu_1), \dots, (\gamma_t, \nu_t)\}$ and a set of d other t-way interactions, $\mathcal{T} = \{T_1, \dots, T_d\}$. Without loss of generality, no T_i has the same column support as T, for if it did, $\rho_A(T) \cap \rho_A(T_i) = \emptyset$. Let a_i be a column in T_i that does not appear in T for $1 \leq i \leq d$. (It is not required that $\{a_1, \dots, a_d\}$ be all distinct.) By the statement of the theorem, there are δ rows $\rho \in \{\ell_1, \dots, \ell_\delta\}$ of C in which $[\partial(\mathbf{c}_{\rho\gamma_1}) \cdots \partial(\mathbf{c}_{\rho\gamma_t})]$ is nonsingular, and $[\mathbf{c}_{\rho\gamma_1} \cdots \mathbf{c}_{\rho\gamma_t} \mathbf{c}_{\rho a_i}]$ is nonsingular whenever $1 \leq i \leq d$. Following the proof of Lemma 3.1, when $1 \leq j \leq \delta$, there are g rows of A_{ℓ_j} that cover T. For each column a_i, because $[\mathbf{c}_{\ell_j\gamma_1} \cdots \mathbf{c}_{\ell_j\gamma_t} \mathbf{c}_{\ell_j a_i}]$ is nonsingular, the symbols in column a_i in the g rows that cover T in A_ℓ are all distinct. Hence when $\ell \in \{\ell_1, \dots, \ell_\delta\}$, we have $|\rho_{A_\ell}(T) \setminus \rho_{A_\ell}(\mathcal{T})| \geq g - d$. □

3.3. Elongated covering perfect hash families

Covering strong separating hash families impose conditions that suffice to generate detecting arrays, but in a sense the conditions are more stringent than needed. To explore this, suppose that $C = (\mathbf{c}_{ij})$ is an $\mathsf{ECPHF}_\delta(n; k, q, t)$. Consider a t-tuple $G = (\gamma_1, \ldots, \gamma_t)$ of columns, and a row ρ for which $[\partial(\mathbf{c}_{\rho\gamma_1}) \cdots \partial(\mathbf{c}_{\rho\gamma_t})]$ is nonsingular. According to Lemma 3.1, each interaction T on column tuple G is covered the same number (g) of times in A_ρ. For a different column a, $[\mathbf{c}_{\rho\gamma_1} \cdots \mathbf{c}_{\rho\gamma_t}\mathbf{c}_{\rho a}]$ may be singular (because C need not be a CSSHF). When this occurs, the g symbols in the rows covering T in A_ρ are not all distinct. Indeed they are all the same! More precisely, it must happen that $\mathbf{c}_{\rho a} = \sum_{i=1}^{t} \mu_i \mathbf{c}_{\rho\gamma_i}$ for $\mu_1, \ldots, \mu_t \in \mathbb{F}_q$. Then when column γ_i contains ν_i in a row of A_ρ, column a must contain the symbol $\sum_{i=1}^{t} \mu_i \nu_i$. Hence by focussing on A_ρ alone, no detection of T is possible despite the repeated coverage.

This overlooks an important point, however. The particular symbol that arises repeatedly in the additional column a depends entirely on the coefficients $\mu_1, \ldots, \mu_t$ specifying the linear combination. A second such row ρ' may have column a being a different linear combination of the t columns in G; then it can happen that T appears with one symbol in column a in A_ρ but with a different symbol in $A_{\rho'}$. When this happens, separation is possible. Unfortunately, this does not overcome the issue for all interactions. Suppose, for example, that the interaction has $\nu_i = 0$ for each $1 \le i \le t$. Then the symbol in column a must also be 0, no matter the linear combination.

To address this, we pursue a different approach. Let $\mathcal{R} = \mathcal{R}_{t+1,q,g} = \{\mathbf{r}_1, \ldots, \mathbf{r}_{gq^t}\}$ be a set of gq^t row vectors. Let $C = (\mathbf{c}_{ij})$ be an $\mathsf{ECPHF}_\delta(n; k, q, t)$. Let $F = (\alpha_{ij})$ be an $n \times k$ array with entries from $\mathbb{F}_q$; entries in F are *adders*. Form a $ngq^t \times k$ array A with rows indexed by $\{1, \ldots, gq^t\} \times \{1, \ldots, n\}$ and columns indexed by $\{1, \ldots, k\}$. In the cell in row (σ, ρ) and column κ of A, place the entry $(\mathbf{r}_\sigma \mathbf{c}_{\rho,\kappa}) + \alpha_{\rho,\kappa}$. In effect, we have modified the construction of Lemma 3.1 by permuting symbols within each column of each of the A_ρ arrays. Lemma 3.1 and Theorem 3.3 continue to hold with this modification. Moreover, for a column tuple G and another column a, when G carries a nonsingular array in row ρ, we find g different entries in column a when $G \cup \{a\}$ is also nonsingular, or a single entry when it is singular. But the adders ensure that in the latter case, for any t-way interaction T, two such rows can lead to different entries in column a.

One might therefore choose the $\mathsf{ECPHF}_\delta(n; k, q, t)$ and the adders so as to ensure detection even when the array is not a $\mathsf{CSSHF}_\delta(n; k, q, d, t)$. This appears to be a challenging problem; in the next section, we explore a probabilistic analysis.

4. Asymptotics and existence

To apply Theorem 3.3 directly, CSSHFs are needed. The same strategy might be useful even for certain ECPHFs that need not be CSSHFs. In either case, our aim is to construct detecting arrays with few rows and many columns, to support testing for many factors with few tests. A probabilistic analysis of the more restrictive situation for CPHFs underlies the best known general asymptotic existence results for covering arrays [18,20]. These approaches also lead to many covering arrays having the fewest rows known for practical sizes [8,18].

Here we extend this to construct CSSHFs and ECPHFs. We start with CSSHFs (see also [15]) and later generalize.

4.1. The basic probabilistic method for CSSHFs

Elements of an $n \times k$ array are chosen uniformly at random from $\mathcal{W}_{t+1,q}$. Let G be a set of t distinct columns, and D be a set of d distinct columns disjoint from G. Focus on a particular row, and consider the selections in the columns of G sequentially. Check that after σ columns are examined, the $t \times \sigma$ matrix thus far has rank σ. The σ columns generate a space of q^σ column vectors. None of these can be adjoined if the selections on G are to be nonsingular. Hence the next column has probability $\frac{q^t - q^\sigma}{q^t - 1}$ of maintaining full rank. After the $t \times t$ matrix on G is guaranteed to be nonsingular, for each column of D, only the last coordinate of the vector chosen matters. Columns of D can now be treated independently. For each one, the entry chosen in the last coordinate leads to full rank of the corresponding $(t+1) \times (t+1)$ matrix with probability $\frac{q-1}{q}$, because only one choice from $\mathbb{F}_q$ can make the matrix singular. The probability that both conditions are met for G and D in the row under consideration is therefore

$$\psi_{d,t,q} = \left(\frac{\prod_{i=0}^{t-1} (q^t - q^i)}{(q^t - 1)^t} \right) \left(\frac{q-1}{q} \right)^d.$$

Rows are selected independently. Therefore the probability that the conditions are met *fewer than* δ times within n rows is

$$\sum_{i=0}^{\delta-1} \binom{n}{i} (1 - \psi_{d,t,q})^i (\psi_{d,t,q})^{n-i}.$$

Using linearity of expectations, when

$$\binom{k}{t} \binom{k-t}{d} (\psi_{d,t,q})^n \left[\sum_{i=0}^{\delta-1} \binom{n}{i} \left(\frac{1 - \psi_{d,t,q}}{\psi_{d,t,q}} \right)^i \right] < 1, \tag{1}$$

Table 1
Upper bounds on N in a $\mathsf{DA}_1(N; d, t, 100000, 7)$ by the basic probabilistic method.

$t \downarrow$ $d \rightarrow$	0	1	2	3
2	833	2450	6468	10584
3	8232	24010	58653	93296
4	72030	206486	497007	768320
5	605052	1714314	3983259	6050520
6	4823609	13882582	31412283	47059600

Table 2
Upper bounds on N in a $\mathsf{DA}_1(N; d, t, 100000, 7)$ by oversampling.

$t \downarrow$ $d \rightarrow$	0	1	2	3
2	588	1764	5145	8820
3	6517	19208	49392	80948
4	60025	172872	424977	681884
5	521017	1479016	3529470	5445468
6	4353013	12235496	28588707	43294832

a $\mathsf{CSSHF}_\delta(n; k, q, d, t)$ exists. This is the basic probabilistic method (see [4], for example). For CPHFs (with $d = 0$ and $\delta = 1$) [18], this simplifies to

$$\binom{k}{t}(\psi_{0,t,q})^n < 1.$$

For fixed d, t, q, k, and δ, one can easily compute the smallest n for which (1) is satisfied; then Theorem 3.3 can be applied to make a detecting array. We illustrate the consequences for various values of t and d when $q = 7$, $k = 100000$, and $\delta = 1$ (but note that the approach is fully general).

4.2. Oversampling for CSSHFs

Improvements using the Lovász local lemma [4,24] can be obtained in the same manner as for CPHFs [18]. Here we pursue a different approach that appears to yield better bounds. The strategy that we explore is variously known as oversampling [18], postprocessing [55], and expurgation [21].

The left side of the inequality (1) is precisely the expected number of choices of a tuple G of t columns and d other columns, for which not all interactions on columns in G can be detected because of choices on the d other columns. Call such a bad event a *blemish*, so that the left side is the expected number of blemishes. By requiring that the expected number of blemishes be less than 1, and observing that every array has an integral number of blemishes, the basic probabilistic method ensures that at least one array has no blemishes. Then it is a CSSHF.

Given an $n \times k$ array that has at most b blemishes, one can choose one column from each of the blemishes. Then by deleting the chosen columns (of which there are at most b), no blemishes remain and the resulting array is a CSSHF. To make this effective, one wants to consider arrays on $k + k'$ columns having an expected number of blemishes less than $k' + 1$. Then choosing an array whose number of blemishes does not exceed the expectation, we can delete k' columns to produce an $n \times k$ CSSHF. The analysis parallels the basic probabilistic argument closely. When

$$\binom{k+k'}{t}\binom{k+k'-t}{d}(\psi_{d,t,q})^n\left\lceil\sum_{i=0}^{\delta-1}\binom{n}{i}\left(\frac{1-\psi_{d,t,q}}{\psi_{d,t,q}}\right)^i\right\rceil < k'+1, \tag{2}$$

a $\mathsf{CSSHF}_\delta(n; k, q, d, t)$ exists. Naturally, (2) leads to an upper bound on n for each $k' \geq 1$. Guided by the discussion in [18], we treat the bounds obtained when $k' = \lceil k/t \rceil$.

To contrast with the basic probabilistic method, in Table 2 we report results for the same parameters as in Table 1. Evidently, oversampling can yield substantial improvements on the basic probabilistic argument.

4.3. Using ECPHFs

Until this point, our analyses have treated each row of a CSSHF independently. Within the corresponding array A_ρ, suppose that some t-way interaction T is covered by d others. Then *every* interaction on the same columns as T is also covered by the union of d others within A_ρ. This symmetry allowed us to treat the rows of the CSSHF independently in the probabilistic calculations, while focussing on the columns supporting the interaction rather than the symbols that it contains. As discussed in §3.3, however, different rows may combine to support detection despite no single one sufficing.

In order to understand this, we treat the situation when $d = 1$ and $\delta = 1$, using the adder method from §3.3. Suppose that $T = \{(\gamma_1, \nu_1), \ldots, (\gamma_t, \nu_t)\}$ is a t-way interaction and a is another column. Let $G = \{\gamma_1, \ldots, \gamma_t\}$ be the column support of T. For a row ρ, one of three situations can arise.

1. $[\partial(\mathbf{c}_{\rho\gamma_1}) \cdots \partial(\mathbf{c}_{\rho\gamma_t})]$ is singular.
2. $[\partial(\mathbf{c}_{\rho\gamma_1}) \cdots \partial(\mathbf{c}_{\rho\gamma_t})]$ is nonsingular but $[\mathbf{c}_{\rho\gamma_1} \cdots \mathbf{c}_{\rho\gamma_t} \mathbf{c}_{\rho a}]$ is singular.
3. $[\partial(\mathbf{c}_{\rho\gamma_1}) \cdots \partial(\mathbf{c}_{\rho\gamma_t}))$ and $[\mathbf{c}_{\rho\gamma_1} \cdots \mathbf{c}_{\rho\gamma_t} \mathbf{c}_{\rho a}]$ are both nonsingular.

If any row is in the third group, there is no blemish on columns G and a. We treat all rows in the first group as if they make no contribution to coverage for G and a and instead focus on rows in the second group. We can determine, for $0 \le \mu \le n$, the probability with which no row is in the third group, μ rows are in the second group, and the remaining $n - \mu$ are in the first.

In this situation, a *blemish* is a set G of t columns and another column a so that at least one interaction with column support G appears with at most one symbol in column a. A *flaw* for G and a is a *specific interaction* T with column support G that appears with at most one symbol in column a. When none of the q^t such interactions forms a flaw, there is no blemish for G and a.

Our concern is to develop an upper bound on the expected number of blemishes. To do so, we consider flaws. Consider a tuple G of t columns and another column a for which μ rows are in the second group and the remaining $n - \mu$ are in the first. Let $x_1, \ldots, x_\mu$ be the symbols in column a with which a specific interaction T with column support G appears in the rows of the second group. Now there is a flaw for T and a only when all of $\{x_1, \ldots, x_\mu\}$ are the same. Because adders are chosen independently, there is a flaw for T and a with probability 1 if $\mu = 0$ and $\frac{q}{q^\mu} = \frac{1}{q^{\mu-1}}$ if $\mu \ge 1$. Then a naive upper bound on the probability that there is a blemish for G and a is obtained by the union bound as $\min(1, \frac{q^t}{q^{\mu-1}})$.

An improvement is possible, as follows. When $m \in \mathbb{F}_q$ is nonzero, interactions

$$\begin{aligned} T &= \{(\gamma_1, \nu_1), \ldots, (\gamma_t, \nu_t)\} \text{ and} \\ T' &= \{(\gamma_1, m\nu_1), \ldots, (\gamma_t, m\nu_t)\} \end{aligned}$$

are either both flaws for additional column a or neither is. Hence it suffices to check representatives of the q^t symbol vectors under multiplication, of which there are $1 + \frac{q^t - 1}{q-1}$. Again by the union bound, an upper bound on the probability that there is a blemish for G and a is $\min(1, \frac{1}{q^{\mu-1}}(1 + \frac{q^t-1}{q-1}))$.

It is routine to adjust (1) and (2) to account for the reduced probability of a blemish. We call the resulting bounds the ECPHF bounds. The analysis for $d > 1$ or $\delta > 1$ is more involved, but can be carried out in a similar manner.

At first glance, it appears that small reductions in the probability of a blemish may not warrant the consideration of ECPHFs that are not CSSHFs. In order to explore this, in Table 3 we present a large number of bounds with and without oversampling, both for the initial analysis of CSSHFs and incorporating the additional detection of ECPHFs.

The ECPHF bounds appear to provide worthwhile improvements, at least for the parameters shown. Naturally our interest lies in explicitly constructing a detecting array whose number of rows does not exceed the computed bound. For this purpose, conditional expectation one-row-at-a-time algorithms can be used as for CPHFs [18,22]. One chooses elements in a row being built so as to reduce the expected number of blemishes as much as possible. The correct computation of the expected number of blemishes depends not only on the rows that have been built, but also on the number yet to build. We do not give the details here, as they parallel the algorithm in [22] closely.

Because we have focussed on the detection provided by rows in the second group when none are in the third, it may be possible to improve further by examining coverage of specific interactions within rows of the first group. We do not pursue this extension here, but we believe that it can be useful when q is 'small', δ is 'large', or both.

5. Concluding remarks

Using arithmetic over the finite field, CSSHFs provide a compact representation for uniform detecting arrays for various values of t, d, and k when the number of symbols is a prime power. Moreover, they support larger separation. While weakening the conditions on CSSHFs, ECPHFs can also yield detecting arrays with fewer rows. We have established basic probabilistic analyses both for CSSHFs and for ECPHFs, and treated extensions using oversampling. These provide effective upper bounds on the sizes of detecting arrays. More importantly, they underlie efficient construction algorithms.

We do not expect that the detecting arrays produced have the fewest rows possible. Although random selections of a covering hash family provide useful bounds and arrays, we expect that a disciplined method for selecting the array may yield fewer rows. The structure of the finite field may be valuable in specifying a deterministic structure for the covering hash family. For CPHFs, this has been fruitful for some specific parameters [41,54]. For detecting arrays, such deterministic constructions would be of interest.

Table 3
Upper bounds on N in a $\mathsf{DA}_1(N; 1, t, k, 5)$.

		Basic		Oversampling	
t	k	CSSHF	ECPHF	CSSHF	ECPHF
2	10^2	600	500	500	450
2	10^3	950	750	700	600
2	10^4	1250	950	900	750
2	10^5	1550	1200	1150	900
2	10^6	1900	1400	1350	1000
2	10^7	2200	1600	1550	1150
3	10^2	4250	3750	3750	3250
3	10^3	6750	5500	5500	4750
3	10^4	9000	7250	7250	6000
3	10^5	11500	9000	9000	7250
3	10^6	13750	10750	10750	8500
3	10^7	16250	12250	12500	9750
4	10^2	26250	23750	23750	21250
4	10^3	42500	35000	36250	31250
4	10^4	57500	46250	48750	40000
4	10^5	72500	57500	60000	48750
4	10^6	87500	68750	72500	57500
4	10^7	102500	80000	85000	66250
5	10^2	156250	137500	143750	131250
5	10^3	243750	212500	218750	187500
5	10^4	337500	281250	293750	250000
5	10^5	431250	350000	375000	306250
5	10^6	525000	412500	450000	362500
5	10^7	612500	481250	525000	412500
6	10^2	875000	781250	812500	750000
6	10^3	1406250	1218750	1281250	1125000
6	10^4	1937500	1625000	1718750	1468750
6	10^5	2468750	2000000	2187500	1812500
6	10^6	3031250	2406250	2656250	2125000
6	10^7	3562500	2781250	3125000	2468750

CRediT authorship contribution statement

Charles J. Colbourn: Conceptualization and design of the study, writing the first draft of the manuscript, commenting on previous versions of the manuscript and contributing improvements, reading and approving the final manuscript. **Violet R. Syrotiuk:** Conceptualization and design of the study, commenting on previous versions of the manuscript and contributing improvements, reading and approving the final manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Declaration of generative AI and AI-assisted technologies in the writing process

No AI-assisted tools were used in the writing of this paper.

Acknowledgments

Research supported by NSF grants #1421058 and #1813729.
Thanks to Yasmeen Akhtar, Randy Compton, Ryan Dougherty, Erin Lanus, and Stephen Seidel for helpful discussions.

References

[1] Y. Akhtar, F. Zhang, C.J. Colbourn, J. Stufken, V.R. Syrotiuk, Scalable level-wise screening experiments using locating arrays, J. Qual. Technol. 55 (5) (2023) 584–597.

[2] Y. Akhtar, C.J. Colbourn, V.R. Syrotiuk, Mixed-level covering, locating, and detecting arrays via cyclotomy, in: F. Hoffman, S. Holliday, Z. Rosen, F. Shahrokhi, J. Wierman (Eds.), Combinatorics, Graph Theory and Computing, Springer International Publishing, Cham, 2024, pp. 37–50.
[3] A.N. Aldaco, C.J. Colbourn, V.R. Syrotiuk, Locating arrays: a new experimental design for screening complex engineered systems, Oper. Syst. Rev. 49 (1) (2015) 31–40.
[4] N. Alon, J.H. Spencer, The Probabilistic Method, John Wiley & Sons, Inc., Hoboken, NJ, 2008.
[5] R.C. Bryce, C.J. Colbourn, A density-based greedy algorithm for higher strength covering arrays, Softw. Test. Verif. Reliab. 19 (1) (2009) 37–53.
[6] Y. Chang, C.J. Colbourn, A. Gowty, D. Horsley, J. Zhou, New bounds on the maximum size of Sperner partition systems, Eur. J. Comb. 90 (2020) 103165.
[7] C.J. Colbourn, Combinatorial aspects of covering arrays, Matematiche (Catania) 58 (2004) 121–167.
[8] C.J. Colbourn, Covering array tables: $2 \le v \le 25$, $2 \le t \le 6$, $t \le k \le 10000$, 2005–23, https://www.public.asu.edu/~ccolbou/src/tabby.
[9] C.J. Colbourn, Covering arrays and hash families, in: D. Crnković, V. Tonchev (Eds.), Information Security and Related Combinatorics, in: NATO Science for Peace and Security Series - D: Information and Communication Security, vol. 29, 2011, pp. 99–136.
[10] C.J. Colbourn, E. Lanus, Subspace restrictions and affine composition for covering perfect hash families, Art Discrete Appl. Math. 1 (2) (2018) #P2.03.
[11] C.J. Colbourn, D.W. McClary, Locating and detecting arrays for interaction faults, J. Comb. Optim. 15 (1) (2008) 17–48.
[12] C.J. Colbourn, V.R. Syrotiuk, On a combinatorial framework for fault characterization, Math. Comput. Sci. 12 (4) (2018) 429–451.
[13] C.J. Colbourn, V.R. Syrotiuk, Detecting arrays for main effects, in: M. Ćirić, M. Droste, J.-É. Pin (Eds.), Algebraic Informatics. CAI 2019, in: Lecture Notes in Computer Science, vol. 11545, Springer International Publishing, Cham, 2019, pp. 112–123.
[14] C.J. Colbourn, V.R. Syrotiuk, There must be fifty ways to miss a cover, in: F. Chung, R.L. Graham, F. Hoffman, R.C. Mullin, L. Hogben, D.B. West (Eds.), 50 Years of Combinatorics, Graph Theory, and Computing, CRC Press, 2019, pp. 319–333.
[15] C.J. Colbourn, V.R. Syrotiuk, Covering strong separating hash families, in: J.A. Davis (Ed.), Finite Fields and Their Applications, De Gruyter, Berlin, Boston, 2020, pp. 189–198.
[16] C.J. Colbourn, V.R. Syrotiuk, Detecting arrays for effects of single factors, in: European Congress of Mathematics, EMS Press, Berlin, 2023, pp. 693–718.
[17] C.J. Colbourn, G. Kéri, P.P. Rivas Soriano, J.-C. Schlage-Puchta, Covering and radius-covering arrays: constructions and classification, Discrete Appl. Math. 158 (11) (2010) 1158–1180.
[18] C.J. Colbourn, E. Lanus, K. Sarkar, Asymptotic and constructive methods for covering perfect hash families and covering arrays, Des. Codes Cryptogr. 86 (4) (2018) 907–937.
[19] R. Compton, M.T. Mehari, C.J. Colbourn, E. De Poorter, V.R. Syrotiuk, Screening interacting factors in a wireless network testbed using locating arrays, in: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016, pp. 650–655.
[20] S. Das, T. Mészáros, Small arrays of maximum coverage, J. Comb. Des. 26 (10) (2018) 487–504.
[21] D. Deng, D.R. Stinson, R. Wei, The Lovász local lemma and its applications to some combinatorial arrays, Des. Codes Cryptogr. 32 (1–3) (2004) 121–134.
[22] R.E. Dougherty, K. Kleine, M. Wagner, C.J. Colbourn, D.E. Simos, Algorithmic methods for covering arrays of higher index, J. Comb. Optim. 45 (1) (2023) 28.
[23] D.-Z. Du, F.K.-M. Hwang, D. Ding-Zhu, Combinatorial Group Testing and Its Applications, 2 edition, Series on Applied Mathematics, World Scientific Publishing, Singapore, 1999.
[24] P. Erdős, L. Lovász, Problems and results on 3-chromatic hypergraphs and some related questions, in: Infinite and Finite Sets, North-Holland, Amsterdam, 1975, pp. 609–627.
[25] A. Gowty, D. Horsley, More constructions for Sperner partition systems, J. Comb. Des. 29 (9) (2021) 579–606.
[26] D. Hamlet, R. Taylor, Partition testing does not inspire confidence (program testing), IEEE Trans. Softw. Eng. 16 (12) (1990) 1402–1411.
[27] A. Hartman, Software and hardware testing using combinatorial covering suites, in: M.C. Golumbic, I.B.A. Hartman (Eds.), Graph Theory, Combinatorics and Algorithms, Springer, Boston, MA, 2005, pp. 237–266.
[28] T.B. Idalino, L. Moura, A survey of cover-free families: constructions, applications, and generalizations, in: New Advances in Designs, Codes and Cryptography, Springer Nature, Switzerland, Cham, 2024, pp. 195–239.
[29] H. Jin, T. Tsuchiya, Constrained locating arrays for combinatorial interaction testing, J. Syst. Softw. 170 (2020) 110771.
[30] H. Jin, C. Shi, T. Tsuchiya, Constrained detecting arrays: mathematical structures for fault identification in combinatorial interaction testing, Inf. Softw. Technol. 153 (2023) 107045.
[31] D.R. Kuhn, R. Kacker, Y. Lei, Introduction to Combinatorial Testing, CRC Press, Boca Raton, FL, 2013.
[32] V.V. Kuliamin, A.A. Petukhov, A survey of methods for constructing covering arrays, Program. Comput. Softw. 37 (3) (2011) 121–146.
[33] E. Lanus, C.J. Colbourn, D.C. Montgomery, Partitioned search with column resampling for locating array construction, in: 2019 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), IEEE Press, 2019.
[34] P.C. Li, K. Meagher, Sperner partition systems, J. Comb. Des. 21 (7) (2013) 267–279.
[35] C. Martínez, L. Moura, D. Panario, B. Stevens, Locating errors using ELAs, covering arrays, and adaptive testing algorithms, SIAM J. Discrete Math. 23 (4) (2010) 1776–1799.
[36] K. Meagher, L. Moura, B. Stevens, A Sperner-type theorem for set-partition systems, Electron. J. Comb. 12 (1) (2005) N20.
[37] J. Morgan, Combinatorial testing: an approach to systems and software testing based on covering arrays, in: R.S. Kenett, F. Ruggeri, F.W. Faltin (Eds.), Analytic Methods in Systems and Software Testing, John Wiley & Sons, Nashville, TN, 2018, pp. 131–178.
[38] C. Nie, H. Leung, A survey of combinatorial testing, ACM Comput. Surv. 43 (2) (2011) 1–29.
[39] X. Niu, H. Cao, Constructions and bounds for separating hash families, Discrete Math. 341 (9) (2018) 2627–2638.
[40] T.J. Ostrand, M.J. Balcer, The category-partition method for specifying and generating functional tests, Commun. ACM 31 (6) (1988) 676–686.
[41] S. Raaphorst, L. Moura, B. Stevens, A construction for strength-3 covering arrays from linear feedback shift register sequences, Des. Codes Cryptogr. 73 (3) (2014) 949–968.
[42] P. Sarkar, D.R. Stinson, Frameproof and IPP codes, in: Progress in Cryptology — INDOCRYPT 2001. INDOCRYPT 2001, in: Lecture Notes in Computer Science, vol. 2247, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 117–126.
[43] S.A. Seidel, M.T. Mehari, C.J. Colbourn, E. De Poorter, I. Moerman, V.R. Syrotiuk, Analysis of large-scale experimental data from wireless networks, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE Press, 2018, pp. 535–540.
[44] S.A. Seidel, K. Sarkar, C.J. Colbourn, V.R. Syrotiuk, Separating interaction effects using locating and detecting arrays, in: C. Iliopoulos, H. Leong, W.K. Sung (Eds.), Combinatorial Algorithms. IWOCA 2018, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2018, pp. 349–360.
[45] G.B. Sherwood, S.S. Martirosyan, C.J. Colbourn, Covering arrays of higher strength from permutation vectors, J. Comb. Des. 14 (3) (2006) 202–213.
[46] C. Shi, C.M. Wang, Optimum detecting arrays for independent interaction faults, Acta Math. Sin. Engl. Ser. 32 (2) (2016) 199–212.
[47] C. Shi, Y. Tang, J. Yin, The equivalence between optimal detecting arrays and super-simple OAs, Des. Codes Cryptogr. 62 (2) (2012) 131–142.
[48] C. Shi, Y. Tang, J. Yin, Optimum mixed level detecting arrays, Ann. Stat. 42 (4) (2014) 1546–1563.
[49] V.R. Syrotiuk, C.J. Colbourn, E. De Poorter, M.T. Mehari, I. Moerman, An efficient screening method for identifying parameters and interactions that impact wireless network performance, in: New Advances in Designs, Codes and Cryptography, Springer Nature, Switzerland, 2023, pp. 409–425.
[50] Y. Tang, J.X. Yin, Detecting arrays and their optimality, Acta Math. Sin. Engl. Ser. 27 (12) (2011) 2309–2318.
[51] J. Torres-Jimenez, I. Izquierdo-Marquez, A simulated annealing algorithm to construct covering perfect hash families, Math. Probl. Eng. 2018 (2018) 1860673.

[52] J. Torres-Jimenez, I. Izquierdo-Marquez, Covering arrays of strength three from extended permutation vectors, Des. Codes Cryptogr. 86 (11) (2018) 2629–2643.
[53] J. Torres-Jimenez, I. Izquierdo-Marquez, H. Avila-George, Methods to construct uniform covering arrays, IEEE Access 7 (2019) 42774–42797.
[54] G. Tzanakis, L. Moura, D. Panario, B. Stevens, Constructing new covering arrays from LFSR sequences over finite fields, Discrete Math. 339 (3) (2016) 1158–1171.
[55] E. van den Berg, E. Candès, G. Chinn, C. Levin, P.D. Olcott, C. Sing-Long, Single-photon sampling architecture for solid-state imaging sensors, Proc. Natl. Acad. Sci. 110 (30) (2013).