**Journal of Algorithms and Computation**

journal homepage: http://jac.ut.ac.ir

# Profiles of covering arrays of strength two

Charles J. Colbourn*[1,2] and Jose Torres-Jimenez†[3]

[1]Arizona State University, P.O. Box 878809, , Tempe, AZ 85287-8809, U.S.A.
[2]State Key Laboratory of Software Development Environment,, Beihang University, Beijing 100191, China.
[3]CINVESTAV-Tamaulipas, Information Technology Laboratory,, Km. 6 Carretera Victoria-Monterrey,, 87276 Victoria Tamps., Mexico

## ABSTRACT

Covering arrays of strength two have been widely studied as combinatorial models of software interaction test suites for pairwise testing. While numerous algorithmic techniques have been developed for the generation of covering arrays with few columns (factors), the construction of covering arrays with many factors and few tests by these techniques is problematic. Random generation techniques can overcome these computational difficulties, but for strength two do not appear to yield a number of tests that is competitive with the fewest known. Consequently, effective construction of covering arrays with many factors and few tests relies on recursive construction techniques.

*Keyword:* covering array, interaction testing, direct product, simulated annealing.

## ARTICLE INFO

*Article history:*
Received June 10, 2013
Received in revised form October 20, 2013
Accepted November, 15, 2013
Available online December, 01, 2013

AMS subject Classification: 05C38.

*Corresponding author:Charles J. Colbourn.
E-mail: charles.colbourn@asu.edu
†E-mail: jtj@cinvestav.mx

# 1 Abstract continued

Among these, a standard direct product has been particularly effective. Necessarily, any recursive method results in substantial duplication of coverage of pairs; by reducing this duplication when possible, the number of tests can sometimes be reduced. In order to reduce duplication, two key features of a covering array are exploited: the number of disjoint rows, and its profile (the distribution of flexible positions). First, the direct product construction is extended to employ different numbers of disjoint rows and different profiles. Then combinatorial and computational constructions for covering arrays with different profiles are developed. Finally some applications of the generalized direct product, with the various profiles so produced, are examined. Of key importance is that, quite frequently, the covering array with fewest tests does not arise as a product of ingredients with the fewest tests; rather, the utility of the ingredient depends in a crucial way on its profile.

# 2 Covering Arrays

Let $N$, $k$, $t$, and $v$ be positive integers. Let $\mathsf{C}$ be an $N \times k$ array with entries from an alphabet $\Sigma$ of size $v$; we typically take $\Sigma = \{0, \dots, v-1\}$. When $(\nu_1, \dots, \nu_t)$ is a $t$-tuple with $\nu_i \in \Sigma$ for $1 \le i \le t$, $(c_1, \dots, c_t)$ is a tuple of $t$ column indices ($c_i \in \{1, \dots, k\}$), and $c_i \neq c_j$ whenever $\nu_i \neq \nu_j$, the $t$-tuple $\{(c_i, \nu_i) : 1 \le i \le t\}$ is a *$t$-way interaction*. The array *covers* the $t$-way interaction $\{(c_i, \nu_i) : 1 \le i \le t\}$ if, in at least one row $\rho$ of $\mathsf{C}$, the entry in row $\rho$ and column $c_i$ is $\nu_i$ for $1 \le i \le t$. Array $\mathsf{C}$ is a *covering array* $\mathsf{CA}(N; t, k, v)$ of *strength* $t$ when every $t$-way interaction is covered.
Suppose that the $i$th factor takes values from a set $\Sigma_i$ of size $v_i$, not containing the special value $\star$. A *mixed covering array*, $\mathsf{MCA}(N; t, k, v_1 v_2 \cdots v_k)$, is a collection of $N$ rows such that for any $t$ distinct column indices, $i_1, i_2, \cdots, i_t$, every $t$-tuple from $\Sigma_{i_1} \times \Sigma_{i_2} \times \cdots \times \Sigma_{i_t}$ occurs in columns $i_1, i_2, \cdots, i_t$ in at least one of the $N$ rows.
Covering arrays are employed in numerous testing applications in which experimental factors interact to detect the presence of faults (see [16, 28] and references therein), and in many related applications (see [19] for a recent list). Applications to interaction testing, in particular to testing component-based software, have driven much recent research; see [9, 10, 13, 15, 38, 53]. In applications in testing, columns of the array correspond to experimental *factors*, and the symbols in the column form *values* or *levels* for the factor. Each row specifies the values to which to set the factors for an experimental *run*. The array is 'covering' in the sense that every $t$-way interaction appears in at least one run. Figure 1 gives an example of a covering array with $N = 26$ rows, 15 factors having four levels each, and strength two. Consider, for example, the 2-way interaction $\{(1, 2), (2, 0)\}$; it is covered in the seventh and eighth rows. The reader can check that all of the $4^2\binom{15}{2} = 1680$ 2-way interactions are covered. The $\star$ entry can be replaced by any symbol, and the result is a $\mathsf{CA}$(26;2,15,4).
Testing cost is incurred for every test to be run, so a primary objective is to produce a test suite (covering array) with as few tests as possible. At the same time, if interactions are the sources of faults in the system, complete coverage of the interactions is important. When the number of factors is small, between 5 and 50, one can rely on powerful computational methods. But as the number

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 0 | 0 | 3 | 3 | 1 | 0 | 1 | 3 | 2 | 3 | 3 | 0 | 0 |
| 0 | 3 | 0 | 2 | 1 | 0 | 1 | 1 | 2 | 3 | 3 | 0 | 1 | 2 | 2 |
| 1 | 1 | 3 | 3 | 0 | 0 | 2 | 3 | 2 | 3 | 1 | 3 | 3 | 0 | 2 |
| 0 | 0 | 3 | 1 | 2 | 3 | 2 | 2 | 1 | 1 | 0 | 3 | 1 | 2 | 2 |
| 1 | 3 | 2 | 1 | 0 | 2 | 3 | 0 | 3 | 1 | 0 | 3 | 2 | 1 | 1 |
| 1 | 0 | 1 | 2 | 3 | 1 | 0 | 2 | 0 | 0 | 3 | 3 | 3 | 3 | 3 |
| 2 | 0 | 2 | 2 | 1 | 1 | 3 | 2 | 3 | 3 | 0 | 1 | 2 | 0 | 2 |
| 2 | 0 | 3 | 2 | 1 | 1 | 0 | 3 | 2 | 1 | 1 | 2 | 3 | 1 | 1 |
| 3 | 1 | 2 | 2 | 0 | 3 | 0 | 0 | 1 | 2 | 3 | 0 | 2 | 2 | 0 |
| 3 | 2 | 1 | 1 | 3 | 0 | 3 | 3 | 2 | 2 | 0 | 2 | 1 | 3 | 0 |
| 2 | 1 | 2 | 3 | 0 | 0 | 1 | 2 | 3 | 0 | 0 | 2 | 1 | 2 | 3 |
| 3 | 3 | 2 | 1 | 2 | 2 | 3 | 1 | 3 | 0 | 0 | 0 | 3 | 3 | 2 |
| 3 | 3 | 0 | 0 | 2 | 1 | 2 | 2 | 3 | 2 | 1 | 1 | 1 | 1 | 0 |
| 3 | 0 | 3 | 3 | 1 | 2 | 1 | 1 | 0 | 2 | 2 | 3 | 0 | 1 | 0 |
| 2 | 3 | 1 | 1 | 2 | 2 | 0 | 1 | 0 | 3 | 3 | 1 | 1 | 3 | 1 |
| 3 | 2 | 1 | 0 | 3 | 3 | 2 | 1 | 0 | 1 | 3 | 2 | 2 | 0 | 1 |
| 0 | 2 | 1 | 3 | 0 | 1 | 0 | 0 | 3 | 3 | 2 | 2 | 0 | 1 | 2 |
| 0 | 1 | 0 | 3 | 2 | 0 | 1 | 3 | 1 | 0 | 2 | $\star$ | 2 | 1 | 1 |
| 1 | 3 | 0 | 1 | 2 | 2 | 3 | 0 | 1 | 2 | 2 | 2 | 1 | 0 | 3 |
| 0 | 1 | 2 | 0 | 3 | 2 | 3 | 3 | 0 | 2 | 1 | 1 | 3 | 2 | 2 |
| 0 | 1 | 0 | 3 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 0 | 3 | 3 |
| 1 | 2 | 3 | 0 | 1 | 3 | 2 | 0 | 2 | 0 | 1 | 1 | 2 | 3 | 3 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 2 | 0 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 2 | 1 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 1 | 3 |

Figure 1: CA(26;2,15,4)

of factors increases to hundreds or thousands, these computational methods either consume too much execution time, or fail to find covering arrays of competitive size. We advocate a different strategy, that develops powerful recursive constructions to make covering arrays with many factors from ingredient arrays with fewer. While stated in this paper as theorems, each has a constructive proof that yields an easy algorithm to produce a large covering array and to verify that it is one. These constructive techniques enable us to adapt computational methods to find *better* small ingredient arrays, letting the recursive method exploit these to make large arrays as needed. In order to pursue this, we adopt a combinatorial viewpoint, and rely heavily on that literature. Therefore we discuss background in the combinatorics of covering arrays next.

We denote by $\mathsf{CAN}(t, k, v)$ the minimum $N$ for which a $\mathsf{CA}(N; t, k, v)$ exists, because fewer rows means fewer tests to be run. Because $\mathsf{CAN}(1, k, v) = v$, $\mathsf{CAN}(t, k, v) = v^t$ when $k < t$, and $\mathsf{CAN}(t, k, 1) = 1$, we generally assume that $k \geq t \geq 2$ and $v \geq 2$. Nevertheless, the definition employed herein allows $t$, $k$, and $v$ to be arbitrary positive integers.

The determination of $\mathsf{CAN}(t, k, v)$ has been the subject of much research; see [8, 16, 28, 29] for survey material. For fixed $t$ and $v$, only $\mathsf{CAN}(2, k, 2)$ has been determined exactly [32, 33, 41]. In fact, an explicit construction of covering arrays with the fewest rows when $t = v = 2$ is given there. Beyond this, when $t$ and $v$ are fixed, exact numbers are known only for a few small values of $k$ (see [20], for example). Therefore most effort has focussed on constructions of covering arrays that have ‘few’ rows, that is, on upper bounds for $\mathsf{CAN}(t, k, v)$. Asymptotic results can be used to determine the growth rate of $\mathsf{CAN}(t, k, v)$ for fixed $t$ and $v$ as a function of $k$ (see [23, 24] for $t = 2$ and [26] in general, for example). Nevertheless the explicit construction of covering arrays is required for many of the applications mentioned. In this paper, we concentrate on existence for strength $t = 2$.

Random techniques, while easily implemented, do not appear to be competitive with combinatorial and computational methods [19]. Orthogonal arrays provide a number of specific examples [30]; a covering array $\mathsf{CA}(v^2; 2, k, v)$ is an *orthogonal array* of strength two and index one. In [18, 46], the structure of the finite field leads to a projection technique that reduces the number of symbols while increasing the number of columns. Computational methods produce many more arrays. For example, simulated annealing [14, 20, 25, 44, 45], tabu search [39], backtracking [52], integer programming [6, 43], and constraint satisfaction [31] have proved successful. Local optimization can often reduce the number of rows required [37]. However, the most prevalent are greedy methods. One basic strategy adds one test at a time [5, 9, 51]; when a suitable test is chosen, it provides a strong theoretical guarantee on the size of the test suite produced [3, 4], and it provides a natural method to prioritize tests [2]. A second greedy strategy adds one factor at a time [22, 34, 48]. Assuming the presence of certain automorphisms also can reduce the difficulty of computational search [7, 18, 35, 36].

An extensive amount of research has concentrated on recursive methods. We focus on strength two here (for higher strengths, see [19] and references therein). Cut-and-paste (or Roux-type, after Roux [42]) constructions operate by juxtaposing copies of smaller covering arrays. For strength two, the prototypical method of this type is given in [21], and a small extension is described in [19]. These methods rely on the presence of “nearly disjoint” rows in the array, and hence exploit the structure of the ingredient arrays in an essential manner. The direct product, in its simplest form, does not exploit the structure of the ingredient arrays. Consequently it rarely outperforms the Roux-type methods [21].

However, as we shall see, controlling the structure of the ingredients in the direct product can not only make it competitive, it provides consistent and useful improvements on all other available techniques. The remainder of the paper is organized as follows. In Section 3, required definitions are introduced. In Section 4, the generalized direct product is developed. Section 5 examines the profiles of arrays produced by generalized direct product, and by the cut–and–paste method. In Section 6, direct construction of covering arrays with different profiles is explored, adapting a variety of known constructions. In Section 7, computational constructions of covering arrays with different profiles are developed, using simulated annealing and a post-optimization method. Finally, in Section 8, consequences for the existence of covering arrays are briefly considered.

# 3 Properties of covering arrays

In order to extend the direct product construction, we develop some further definitions and notation. First we extend the definition of covering arrays to permit a symbol that is not used for coverage, as follows: An $N \times k$ array, each cell of which contains one of $v$ distinct symbols or a different symbol $\star$, is a *covering array* $\mathsf{CA}(N; t, k, v)$ of *strength* $t$ when, for every way to select $t$ columns, each of the $v^t$ possible tuples containing no $\star$ arises in at least one row.

## 3.1 Compatibility

A $\mathsf{CA}(M; 2, \ell, v)$ $\mathsf{B}$ and a $\mathsf{CA}(M'; 2, \ell', v)$ $\mathsf{B}'$ are $(L, r)$-*compatible* if for every $0 \leq \sigma < r$, $1 \leq j \leq \ell$, and $1 \leq j' \leq \ell'$, there exists a $\rho$ with $1 \leq \rho \leq L$ so that the entry in cell $(\rho, j)$ of $\mathsf{B}$ is $\sigma$ and the entry in cell $(\rho, j')$ of $\mathsf{B}'$ is $\sigma$.

## 3.2 Constant Rows

Two rows of a $\mathsf{CA}(N; t, k, v)$ are *disjoint* if, in each column, either they do not agree or both contain $\star$. A set of $r$ rows in which every two are disjoint is a *partial parallel class* on $r$ rows; two partial parallel classes are *disjoint* if they have no rows in common. A *parallel class* is a partial parallel class on $v$ rows. A row is *constant* if, for some symbol $\nu$, every entry in the row is either $\nu$ or $\star$. A row is *pure* if it contains no $\star$. A row is *pure constant* if it is constant and pure. Because symbols within each column can be permuted independently, one has:

**Observation 3.1** *If a* $\mathsf{CA}(N; t, k, v)$ *exists having* $\rho$ *rows that are pairwise disjoint, there is a* $\mathsf{CA}(N; t, k, v)$ *having* $\rho$ *constant rows. These can without loss of generality be assumed to be on any* $\rho$ *of the* $v$ *symbols.*

In a *standardized* $\mathsf{CA}(N; t, k, v)$ one row is constant. Any $\mathsf{CA}(N; t, k, v)$ can be rewritten by choosing a column, and applying an arbitrary permutation to the symbols in the column.

**Observation 3.2** *If a* $\mathsf{CA}(N; t, k, v)$ *exists, then a standardized* $\mathsf{CA}(N; t, k, v)$ *exists.*

### 3.3 Profiles

The *profile* $(d_1, \ldots, d_k)$ of an $N \times k$ array is a $k$-tuple in which the entry $d_i$ is the number of $\star$ entries in the $i$th column. A single covering array can often admit many different profiles, by filling the $\star$ cells and changing a (possibly different) set of flexible cells to $\star$. A profile $(d_1, \ldots, d_k)$ *dominates* profile $(e_1, \ldots, e_k)$ when $d_i \geq e_i$ for $1 \leq i \leq k$, and we write $(d_1, \ldots, d_k) \geq (e_1, \ldots, e_k)$ in this case.

### 3.4 Flexible positions

In general, some $t$-way interactions may be covered more than once. Now consider each row $r$ of a CA$(N; t, k, v)$; for every subset $C$ of $t$ columns, let $T$ be the $t$-way interaction that is covered in row $r$ and the columns of $C$. If $T$ is not covered in any other row of the array, each of the cells $\{(r, c) : c \in C\}$ is *necessary*. All cells that are not necessary in this way are *flexible*. If we ignore a flexible cell $(r, c)$ in the computation of coverage, all $t$-way interactions remain covered. By convention, when flexible cell $(r, c)$ is to be ignored, we place the entry $\star$ (*"don't care"*) in cell $(r, c)$. In general, one cannot simply convert all flexible cells to $\star$, because two flexible cells can each rely on the value in the other for its flexibility. Nevertheless, one can repeatedly choose any one flexible cell to convert to $\star$, and then recalculate the flexible cells for this modified CA, until none remain.

### 3.5 Examples

Figure 2 gives examples to illustrate the definitions given. Each of the five CAs shown is a CA(14;2,7,3) derived from the first one shown. It has one constant row, the thirteenth; the others are not constant. Because it has a constant row, it is standardized. Rows 11 and 13 are disjoint; indeed interchanging symbols 0 and 1 in column 7 would make row 11 constant, while keeping row 13 constant. Hence there is an equivalent CA(14;2,7,3) with two constant rows.

In the second array shown, the flexible entries are shown in boxes, while the necessary ones are shown without. For example, the six pairs involving the entry in the first row and first column are all covered in rows 5, 13, or 14; hence this entry is flexible. The entry in the second row and first column is necessary because the pair with 0 in column 1 and 2 in column 4 appears only in this row. As remarked earlier, we cannot change all flexible positions to $\star$. If we were to do so, the pair (2,2) would no longer be covered in the first two columns, for example.

The third, fourth, and fifth arrays are all CA(14;2,7,3)s in which certain flexible positions have been changed to don't-care positions, so that there remain no flexible positions. Their profiles are (2,1,2,1,1,1,1), (2,1,2,1,2,1,1), and (2,2,2,2,2,2,2), respectively. Profile (2,1,2,1,2,1,1) dominates (2,1,2,1,1,1,1), and is dominated by (2,2,2,2,2,2,2). The fifth array has two rows that contain only $\star$. These can both be deleted to form a CA(12;2,7,3). Similarly, a row can be removed immediately from the fourth array shown, while the third array has no row containing only don't-care positions.

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 0 | 0 | 1 |
| 0 | 0 | 2 | 2 | 2 | 2 | 2 |
| 1 | 1 | 2 | 1 | 2 | 0 | 1 |
| 0 | 1 | 2 | 0 | 1 | 1 | 0 |
| 2 | 2 | 1 | 0 | 2 | 0 | 2 |
| 2 | 1 | 0 | 2 | 1 | 2 | 2 |
| 1 | 2 | 0 | 0 | 0 | 2 | 0 |
| 0 | 2 | 1 | 1 | 1 | 2 | 1 |
| 1 | 0 | 1 | 2 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 0 | 1 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 0 | 0 | 1 |
| 0 | 0 | 2 | 2 | 2 | 2 | 2 |
| 1 | 1 | 2 | 1 | 2 | 0 | 1 |
| 0 | 1 | 2 | 0 | 1 | 1 | 0 |
| 2 | 2 | 1 | 0 | 2 | 0 | 2 |
| 2 | 1 | 0 | 2 | 1 | 2 | 2 |
| 1 | 2 | 0 | 0 | 0 | 2 | 0 |
| 0 | 2 | 1 | 1 | 1 | 2 | 1 |
| 1 | 0 | 1 | 2 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 0 | 1 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 0 | 0 | 1 |
| 0 | 0 | 2 | 2 | 2 | 2 | 2 |
| 1 | 1 | 2 | 1 | 2 | 0 | 1 |
| 0 | 1 | 2 | 0 | 1 | 1 | 0 |
| 2 | $\star$ | 1 | 0 | 2 | 0 | 2 |
| 2 | 1 | 0 | 2 | 1 | 2 | 2 |
| 1 | 2 | 0 | 0 | 0 | 2 | 0 |
| 0 | 2 | 1 | 1 | 1 | 2 | 1 |
| 1 | 0 | 1 | 2 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 2 |
| $\star$ | 2 | $\star$ | $\star$ | 2 | $\star$ | $\star$ |
| $\star$ | 2 | $\star$ | 2 | $\star$ | 1 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 0 | 0 | 1 |
| 0 | 0 | 2 | 2 | 2 | 2 | 2 |
| 1 | 1 | 2 | 1 | 2 | 0 | 1 |
| 0 | 1 | 2 | 0 | 1 | 1 | 0 |
| 2 | 2 | 1 | 0 | 2 | 0 | 2 |
| 2 | 1 | 0 | 2 | 1 | 2 | 2 |
| 1 | 2 | 0 | 0 | 0 | 2 | 0 |
| 0 | 2 | 1 | 1 | 1 | 2 | 1 |
| 1 | 0 | 1 | 2 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 2 |
| $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ |
| $\star$ | 2 | $\star$ | 2 | $\star$ | 1 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ |
| 0 | 0 | 2 | 2 | 2 | 2 | 2 |
| 1 | 1 | 2 | 1 | 2 | 0 | 1 |
| 0 | 1 | 2 | 0 | 1 | 1 | 0 |
| 2 | 2 | 1 | 0 | 2 | 0 | 2 |
| 2 | 1 | 0 | 2 | 1 | 2 | 2 |
| 1 | 2 | 0 | 0 | 0 | 2 | 0 |
| 0 | 2 | 1 | 1 | 1 | 2 | 1 |
| 1 | 0 | 1 | 2 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 2 |
| $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ |
| 2 | 2 | 2 | 2 | 0 | 1 | 1 |

Figure 2: Examples of a CA(14;2,7,3)

# 4 Generalized direct products

In this section, a general recursive construction is developed, along with many specializations. Section 5 then develops constructions for ingredients to be used in applying the results. We start with the simplest direct product, essentially introduced in [11, 40, 47]; we employ an array operation $\otimes$ depicted in Figure 3.

**Theorem 4.1** [40] *When a* $\mathsf{CA}(N;2,k,v)$ *and a* $\mathsf{CA}(M;2,\ell,v)$ *both exist, a* $\mathsf{CA}(N+M;2,k\ell,v)$ *also exists.*

*Proof.* Let $\mathsf{A} = (a_{ij})$ be a $\mathsf{CA}(N;2,k,v)$ and let $\mathsf{B} = (b_{ij})$ be a $\mathsf{CA}(M;2,\ell,v)$. Form an $(N+M)\times k\ell$ array $\mathsf{C} = (c_{i,j}) = \mathsf{A} \otimes \mathsf{B}$ by setting $c_{i,(f-1)k+g} = a_{i,g}$ for $1 \leq i \leq N$, $1 \leq f \leq \ell$, and $1 \leq g \leq k$. Then set $c_{N+i,(f-1)k+g} = b_{i,f}$ for $1 \leq i \leq M$, $1 \leq f \leq \ell$, and $1 \leq g \leq k$. In essence, $k$ copies of $\mathsf{B} = (b_{ij})$ are being appended to $\ell$ copies of $\mathsf{A} = (a_{ij})$ as shown in Figure 3. Because two different columns of $\mathsf{C}$ arise either from different columns of $\mathsf{A}$ or from two different columns of $\mathsf{B}$, the result is a $\mathsf{CA}(N+M;2,k\ell,v)$. ■

Stevens [45] improved on this by exploiting constant rows. Exploiting both "don't care" cells and constant rows is considered in [19, 21]; we develop general mechanisms for doing so here. We suppose that a factor with $v$ values always takes on values from $\{0,\ldots,v-1\}$, and hence the corresponding column of the array contains only these symbols, and possibly $\star$.

**Theorem 4.2** *Let* $v, r_1, r_2, r_3, s_1, s_2, s_3, N, M$ *be positive integers satisfying* $r_1+r_2 \leq v$, $s_1 \leq v-r_1$, $s_2 \leq r_1$, *and* $M \geq s_1+s_2+s_3$. *Suppose that there exist*

$$
\begin{array}{r|cccc|c|cccc|}
\cline{2-10}
 & a_{11} & a_{12} & \cdots & a_{1k} & \cdots & a_{11} & a_{12} & \cdots & a_{1k} \\
 & a_{21} & a_{22} & \cdots & a_{2k} & \cdots & a_{21} & a_{22} & \cdots & a_{2k} \\
N \text{ rows} & \vdots & & & & \cdots & \vdots & & & \\
 & a_{N1} & a_{N2} & \cdots & a_{Nk} & \cdots & a_{N1} & a_{N2} & \cdots & a_{Nk} \\
\cline{2-10}
 & b_{11} & b_{11} & \cdots & b_{11} & \cdots & b_{1\ell} & b_{1\ell} & \cdots & b_{1\ell} \\
 & b_{21} & b_{21} & \cdots & b_{21} & \cdots & b_{2\ell} & b_{2\ell} & \cdots & b_{2\ell} \\
M \text{ rows} & \vdots & & & & \cdots & \vdots & & & \\
 & b_{M1} & b_{M1} & \cdots & b_{M1} & \cdots & b_{M\ell} & b_{M\ell} & \cdots & b_{M\ell} \\
\cline{2-10}
\end{array}
$$

Figure 3: The structure of $\mathsf{A} \otimes B$

- *a* $\mathsf{CA}(N; 2, k, v)$, $\mathsf{A}$, *with profile* $(d_1 + e_1, \ldots, d_k + e_k)$, *having two disjoint partial parallel classes, one on* $r_1 + r_2$ *pure rows and the other on* $r_3$ *pure rows;*
- *for each* $1 \le i \le k$ *and some* $0 \le \delta_i \le v - r_1 - s_1$, *a* $\mathsf{CA}(M + d_i + \delta_i; 2, \ell_i, v)$, $\mathsf{B}_i$, *with profile* $(\gamma_{i1}, \ldots, \gamma_{i\ell_i})$, *in which*
  - *the first* $s_2$ *rows are pure constant on symbols* $\{0, \ldots, s_2 - 1\}$ *and the last* $s_1 + \delta_i$ *rows are pure constant on symbols* $v - s_1 - \delta_i, \ldots, v - 1$*;*
  - *rows* $s_2 + 1, \ldots, s_2 + s_3$ *form a partial parallel class of* $s_3$ *pure rows; and*
  - *for every* $1 \le i_1 < i_2 \le k$, $\mathsf{B}_{i_1}$ *and* $\mathsf{B}_{i_2}$ *are* $(M - s_1, r_1)$*-compatible.*

*Let* $\mu = \max(1, r_2 + s_2, r_3, s_3)$. *Let* $L$ *be the list with entries* $e_i + \gamma_{ij}$ *for* $1 \le i \le k$ *and* $1 \le j \le \ell_i$. *Then there exists a* $\mathsf{CA}(N + M - r_1 - s_1; 2, \sum_{i=1}^{k} \ell_i, v)$ *having* $\mu$ *constant rows and profile* $L$.

*Proof.* Permute symbols and rows of $\mathsf{A}$ so that the $r_1 + r_2$ disjoint pure rows form $r_1$ pure constant rows on $\{0, \ldots, r_1 - 1\}$ as the last $r_1$ rows and $r_2$ pure constant rows on $\{v - r_2, \ldots, v - 1\}$ as the first $r_2$ rows; remove the last $r_1$ rows to form $\mathsf{A}'$ with $N - r_1$ rows.
Form an array $\mathsf{C}$ with $N + M - r_1 - s_1$ rows and $\sum_{i=1}^{k} \ell_i$ columns, indexing columns as $(c, j)$ for $1 \le c \le k$ and $1 \le j \le \ell_i$. Fill the first $N - r_1$ rows in column $(c, j)$ as a copy of column $c$ of $\mathsf{A}'$. For $c = 1, \ldots, k$, let $R_c$ be the indices of the $d_c$ rows in which $\mathsf{A}'$ contains a $\star$ in column $c$ of $\mathsf{A}$. For $1 \le c \le k$, $1 \le j \le \ell_c$, and $1 \le \rho \le M - s_1$ place in row $N - r_1 + \rho$ and column $(c, j)$ of $\mathsf{C}$ the entry in cell $(\rho, j)$ of $\mathsf{B}_i$. Then let $\rho_1, \ldots, \rho_{d_c}$ be the entries of $R_c$. For $1 \le c \le k$, $1 \le j \le \ell_c$, and $1 \le x \le d_c$, in row $\rho_x$ and column $(c, j)$ place the entry in cell $(M - s_1 + x, j)$ of $\mathsf{B}_i$.
Consider columns $(i_1, j_1)$ and $(i_2, j_2)$ of the result. When $i_1 = i_2$, all pairs of the form $(\sigma, \sigma)$ are covered in the first $N - r_1$ rows excluding those in $R_{i_1}$. Then in rows $R_{i_1}$ and the last $M - s_1$ rows, all remaining pairs are covered because two different columns of $\mathsf{B}_{i_1}$ are selected (and $\delta_i \le v - r_1 - s_1$). So suppose that $i_1 \ne i_2$. The first $N - r_1$ rows cover all pairs except possibly for $(\sigma, \sigma)$ when $0 \le \sigma < r_1$, which are covered by the remaining rows as a consequence of compatibility.

Consider the largest number $\mu$ of constant rows. By Observation 3.1, $\mu \geq 1$. In the result, rows $1, \dots, r_2$ are constant on symbols $v - r_2, \dots, v - 1$. Rows $N - r_1 + 1, \dots, N - r_1 + s_2$ are constant on the symbols $0, \dots, s_2 - 1$. So $\mu \geq r_2 + s_2$. Because $\mathsf{A}$ contains a partial parallel class on $s_3$ rows, and these rows are all pure, they yield a partial parallel class in $\mathsf{C}$ of the same size; hence $\mu \geq r_3$. Because $M \geq s_1 + s_2 + s_3$, and each $\mathsf{B}_i$ contains a partial parallel class of size $s_3$, $\mathsf{C}$ also contains a partial parallel class of the same size; hence $\mu \geq s_3$. ■

Theorem 4.2 is quite general, but consequently it can be challenging to verify all of the conditions required on the ingredients. Therefore we state some corollaries obtained by specializing the parameters. Usually we take $r_3 = s_3 = 0$, so the disjoint partial parallel classes are not needed. Compatibility poses the most severe constraint, so the easiest application arises when all of the $\{\mathsf{B}_i\}$ are identical:

**Corollary 4.3** *If a* $\mathsf{CA}(N; 2, k, v)$ *with* $r$ *disjoint rows and a* $\mathsf{CA}(M; 2, \ell, v)$ *with* $s$ *disjoint rows both exist, then a* $\mathsf{CA}(N + M - \min(v, r + s); 2, k\ell, v)$ *exists having* $\min(1, r + s - v)$ *constant rows.*

*Proof.* A $\mathsf{CA}(N; 2, k, v)$ with $r$ disjoint rows yields a $\mathsf{CA}(N; 2, k, v)$ with $r$ pure constant rows; take $r_1 = \min(r, v-s)$ and $r_2 = r - r_1$. Take $s_1 = s$ and $s_2 = 0$, and write the $\mathsf{CA}(M; 2, \ell, v)$ with $s$ disjoint rows as a $\mathsf{CA}(M; 2, \ell, v)$, $\mathsf{B}$, in which the last $s$ rows are constant on the symbols $v - s, \dots, v - 1$, in that order. Take $d_i = e_i = \delta_i = 0$ and $\mathsf{B}_i = \mathsf{B}$ for $1 \leq i \leq k$. Apply Theorem 4.2. If $r_2 = 0$, one row can nonetheless be made constant. ■

**Corollary 4.4** *Suppose that* $\mathsf{A}$ *is a* $\mathsf{CA}(N; 2, k, v)$ *having* $r$ *pure constant rows and profile* $(d_1, \dots, d_k)$*. Let* $0 \leq s \leq v$ *be an integer. Further suppose that for each* $1 \leq i \leq k$ *and some* $0 \leq \delta_i \leq v - s$*, there exists a* $\mathsf{CA}(M + d_i + \delta_i; 2, \ell_i, v)$*,* $\mathsf{B}_i$*, having* $s + \delta_i$ *constant rows. Then there exists a* $\mathsf{CA}(N + M - s; 2, \sum_{i=1}^{k} \ell_i, v)$ *having* $r$ *constant rows.*

*Proof.* Apply Theorem 4.2 using $r_1 = 0$, $r_2 = r$, $s_1 = s$, and $s_2 = 0$. Because $r_1 = 0$, compatibility holds trivially. ■

**Corollary 4.5** *Suppose that there exists a* $\mathsf{CA}(N; 2, k, v)$*,* $\mathsf{A}$*, with profile* $(d_1, \dots, d_k)$*, having* $r_1 + r_2$ *pure constant rows. Let* $s_2 = r_1$ *and* $0 \leq s_1 \leq v - r_1$*. Further suppose that for each* $1 \leq i \leq k$ *and some* $0 \leq \delta_i \leq v - r_1 - s_1$*, there exists a* $\mathsf{CA}(M + d_i + \delta_i; 2, \ell_i, v)$*,* $\mathsf{B}_i$*, having* $s_1 + s_2 + \delta_i$ *constant rows. Then there exists a* $\mathsf{CA}(N + M - r_1 - s_1; 2, \sum_{i=1}^{k} \ell_i, v)$ *having* $r_2 + s_2$ *constant rows.*

*Proof.* Apply Theorem 4.2. Because $s_2 \geq r_1$, compatibility holds. ■

# 5 Recursive Constructions

In order to develop applications of Theorem 4.2, we examine profiles of covering arrays. As we have seen, Theorem 4.2 itself can be used to make a variety of profiles. The other main recursive construction, the cut–and–paste method, can also be employed.

Consider a $\mathsf{CA}(N;2,k_1+k_2,v)$, shown in Figure 4. The arrays $\mathsf{A}_1$, $\mathsf{A}_2$, and $\mathsf{X}$ are $(N-v)\times k_1$, $(N-v)\times k_2$, and $v\times k_2$, respectively. Array $\mathsf{D}$ is a $v\times k_1$ has a specific structure, namely that every column is a permutation of $\{1,\ldots,v\}$. A $\mathsf{CA}(N;2,k_1+k_2,v)$ admitting such a partition is a *partitioned covering array* $\mathsf{PCA}(N;2,(k_1,k_2),v)$. (Figure 1 gives a PCA(26;2,(11,4),4).) Without loss of generality $\mathsf{D}$ can be assumed to be the matrix $\mathsf{P}$ in which each column is the identity permutation. When $q$ is a prime power, an orthogonal array with the maximum number of factors yields an $\mathsf{PCA}(q^2;2,(q,1),q)$.

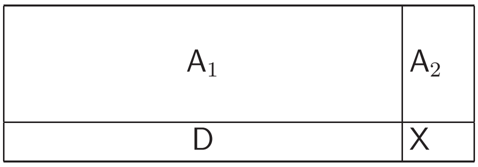


Figure 4: A partitioned covering array (PCA)

The main cut–and–paste construction for covering arrays of strength two is established in [21]; we give a small extension of it proved in [19]:

**Theorem 5.1** *If a* $\mathsf{PCA}(N;2,(k_1,k_2),v)$ *and a* $\mathsf{PCA}(M;2,(\ell_1,\ell_2);v)$ *both exist, then a* $\mathsf{PCA}(N+M-v;2,(k_1\ell_1,k_1\ell_2+k_2\ell_1),v)$ *also exists.*

The construction follows. Take a $\mathsf{PCA}(N;2,(k_1,k_2),v)$ with a partition as in Figure 4 into $\mathsf{A}_1$, $\mathsf{A}_2$, D and X; and an $\mathsf{PCA}(M;2,(\ell_1,\ell_2),v)$ with partition $\mathsf{B}_1$, $\mathsf{B}_2$, E, and Y. Without loss of generality, suppose that D and E consist of column identity permutations, and write each as P. Further suppose that each of the columns of X and Y has the property that the $i+1$st entry does not exceed $i$.
Form an array as in Figure 5. In the products of the form $\mathsf{A}_i\otimes\mathsf{B}_j$, the first $N-v$ rows arise from $\mathsf{A}_i$ while the next $M-v$ arise from $\mathsf{B}_j$, as shown in Figure 3. Here $\ell_1\mathsf{X}$ is obtained by repeating the array X $\ell_1$ times and $k_1(\mathsf{Y})$ is obtained by repeating each column of Y $k_1$ times. P is a $v\times k_1\ell_1$ matrix of (column) identity permutations.

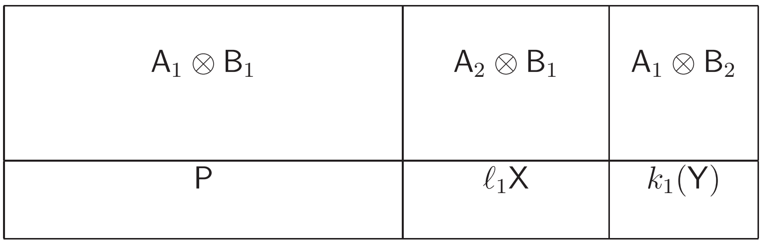


Figure 5: The product of two PCAs

Theorem 5.1 does not exploit don't care positions in either array. In a sense, this is a primary reason why the generalized direct product is able to make the improvements that we have found. However, cut–and–paste preserves (and indeed inflates) don't care positions. Indeed the following employs an easy analysis of the don't care positions that must arise in the resulting array:

**Theorem 5.2** *Suppose that a* $\mathsf{PCA}(N;2,(k_1,k_2),v)$ $\mathsf{A}$ *partitioned into into* $\mathsf{A}_1$, $\mathsf{A}_2$, $\mathsf{D}$ *and* $\mathsf{X}$ *exists, and that there are* $d_i$ *don't care positions in column* $i$ *of* $\mathsf{A}_1$ *for* $1 \leq i \leq k_1$*;* $d_i$ *don't care positions in column* $i$ *of* $\mathsf{A}_2$ *for* $k_1 < i \leq k_1+k_2$*; and* $e_i$ *don't care positions in column* $i$ *of* $\mathsf{X}$ *for* $k_1 < i \leq k_1+k_2$*. Suppose that a* $\mathsf{PCA}(M;2,(\ell_1,\ell_2),v)$ $\mathsf{B}$ *partitioned into into* $\mathsf{B}_1$, $\mathsf{B}_2$, $\mathsf{E}$, *and* $\mathsf{Y}$ *exists, and that there are* $\delta_j$ *don't care positions in column* $j$ *of* $\mathsf{B}_1$ *for* $1 \leq j \leq \ell_1$*;* $\delta_i$ *don't care positions in column* $j$ *of* $\mathsf{B}_2$ *for* $\ell_1 < j \leq \ell_1+\ell_2$*; and* $\varepsilon_i$ *don't care positions in column* $j$ *of* $\mathsf{X}$ *for* $\ell_1 < j \leq \ell_1+\ell_2$*. Form a list* $L$ *of* $(k_1+k_2)(\ell_1+\ell_2)-k_2\ell_2$ *entries, containing* $d_i+\delta_j$ *for* $1 \leq i \leq k_1$ *and* $1 \leq j \leq \ell_1$*;* $d_i+e_i+\delta_j$ *for* $k_1 < i \leq k_1+k_2$ *and* $1 \leq j \leq \ell_1$*; and* $d_i+\delta_j+\varepsilon_j$ *for* $1 \leq i \leq k_1$ *and* $\ell_1 < j \leq \ell_1+\ell_2$*. Then a* $\mathsf{PCA}(N+M-v;2,(k_1\ell_1,k_1\ell_2+k_2\ell_1),v)$ *with profile* $L$ *exists.*

# 6 Direct Constructions of Profiles

In order to apply generalized direct products effectively, ingredients are needed that have suitable combinations of constant rows and profiles. Moreover, to make suitable ingredients using a recursive construction, smaller ingredients are then needed. Here we examine some direct constructions to form basic ingredient arrays.

## 6.1 Fusion

An easy way to form don't care positions is to form a covering array on a larger number of symbols and then omit the 'extra' symbols.

**Theorem 6.1** *Suppose that an* $\mathsf{MCA}(N;2,k,(v_1,\ldots,v_k))$ *exists that*

- *has profile* $(d_1,\ldots,d_k)$*,*
- *has* $\rho \leq v_c-1$ *pure constant rows on symbols* $\{0,\ldots,\rho-1\}$*, and*
- *in which the* $c$*th column contains* $r_{\sigma c}$ *occurrences of* $\sigma$ *for* $\sigma \in \{0,\ldots,v_c-1\}$*.*

*Let* $\varepsilon(\sigma)=1$ *if* $0 \leq \sigma < \rho$*, 0 otherwise. Let* $\delta_c = \max\{r_{\sigma c}-\varepsilon(\sigma) : \sigma \in \{0,\ldots,v_c-1\}\}$*. Then an* $\mathsf{MCA}(N;2,k,(v_1,\ldots,v_{c-1},v_c-1,v_{c+1},\ldots,v_k))$ *exists with profile* $(d_1,\ldots,d_{c-1},d_c+\delta_c,d_{c+1},\ldots,d_k)$ *having* $\rho$ *pure constant rows on symbols* $\{0,\ldots,\rho-1\}$*.*

*Proof.* Choose a symbol $\sigma$ for which $r_{\sigma c}-\varepsilon(\sigma)=\delta_c$. Interchange symbols $\sigma$ in $v_c-1$ in column $c$. Replace all occurrences of $v_c-1$ in column $c$ by $\star$. If $\sigma < \rho$, replace the single star in the constant row containing $\sigma$ by $\sigma$ to restore the pure constant row. ■

After each application of Theorem 6.1, additional flexible positions may arise both in column $c$ and in other columns, and converting these to $\star$ may enable additional reduction in a further application. The particular selections made can affect the benefit from this, so we state a conclusion for covering arrays that holds even when additional $\star$ entries are not deduced.

**Corollary 6.2** *Let* $\mathsf{A}$ *be a* $\mathsf{CA}(N; 2, k, v)$ *on symbol set* $\{0, \ldots, v-1\}$ *and columns* $\{1, \ldots, k\}$*, with profile* $(d_1, \ldots, d_k)$*, in which symbol* $i$ *appears in column* $c$ *exactly* $r_{ic}$ *times. For* $1 \leq c \leq k$*, let* $\psi_c$ *be a permutation on* $\{0, \ldots, v-1\}$ *for which* $r_{\psi_c(j-1),c} \leq r_{\psi_c(j),c}$ *for* $0 < j < v$*. Then for* $2 \leq x < c$*, there is a* $\mathsf{CA}(N; 2, k, x)$ *with profile* $(d_1 + \sum_{j=x}^{v-1} r_{\psi_1(j),1}, \ldots, d_k + \sum_{j=x}^{v-1} r_{\psi_k(j),k})$*.*

*Proof.* Apply Theorem 6.1, taking the CA to have no pure constant rows, to remove $v - x$ symbols from each column. ■

**Corollary 6.3** *If a* $\mathsf{CA}(N; 2, k, v)$ *with profile* $(d_1, \ldots, d_k)$ *exists, then for* $2 \leq x \leq v$*, a* $\mathsf{CA}(N; 2, k, x)$ *with profile* $(d_1 + \lceil \frac{(v-x)(N-d_1)}{v} \rceil, \ldots, d_k + \lceil \frac{(v-x)(N-d_k)}{v} \rceil)$ *also exists.*

*Proof.* For each $c$, $r_{\psi_c(j-1),c} \leq r_{\psi_c(j),c}$ for $0 < j < v$, and $\sum_{j=0}^{v-1} r_{\psi_c(j),c} = N - d_c$. Hence $\sum_{j=x}^{v-1} r_{\psi_c(j),c} \geq (v-x)\frac{N-d_c}{v}$, and the result follows by Corollary 6.2. ■

This variant of fusion does not remove any rows, unlike the use in [20]. We can remove a row if all entries in the row are changed to $\star$, and this can be ensured if symbols are renamed so that all but one entry in the row becomes $\star$. This is the basic operation in the method of post-optimization from [37]; see Section 7.2.

In applying Corollary 6.2 or 6.3, the result cannot be guaranteed to have a pure row, and hence Observation 3.1 need not lead to a pure constant row unless the profile is changed. This can be effectively addressed.

**Lemma 6.4** *If a* $\mathsf{CA}(N; 2, k, v)$ *with profile* $(d_1, \ldots, d_k)$ *having* $\rho$ *pure rows exists, then for* $2 \leq x \leq v$*, a* $\mathsf{CA}(N; 2, k, x)$ *with profile* $(d_1 + \lceil \frac{(v-x)(N-d_1-\rho)}{v} \rceil, \ldots, d_k + \lceil \frac{(v-x)(N-d_k-\rho)}{v} \rceil)$ *having* $\min(\rho, x)$ *pure constant rows also exists.*

*Proof.* Permute so that the pure rows are pure constant on symbols $0, \ldots, \rho - 1$. Then, in each column $c$, select the $v - x$ elements that appear the most frequently in the $N - d_c - \rho$ other rows. Permute symbols in column $c$ so that these symbols are named $v - x, \ldots, v - 1$, ensuring that none of $0, \ldots, \rho-1$ is renamed unless it appears among the $v-x$ chosen symbols. Now change all occurrences of $v - x, \ldots, v - 1$ in column $c$ to $\star$, except in any row that was pure constant; if one appears in such a row, make it again pure constant. ■

**Lemma 6.5** *Suppose that there exists a* $\mathsf{CA}(N'; 2, k, v)$ *in which at least one symbol occurs exactly* $v$ *times in one of the columns. (This always holds when* $v^2 \leq N' < v(v + 1)$*.) Then there exists a* $\mathsf{CA}(N' - 1; 2, k, v - 1)$ *having profile* $(v - 1)^k$*, and a* $\mathsf{CA}(N' - 1; 2, k, v - 1)$ *with* $v - 1$ *pure constant rows having profile* $(v - 1)^{k-1}0^1$*.*

*Proof.* Without loss of generality there is a $\mathsf{CA}(N'; 2, k, v)$ with a pure constant row of symbol $v - 1$, so that symbol $v - 1$ occurs exactly $v$ times in the last column. Delete this constant row and change all occurrences of symbol $v - 1$ to $\star$ to form a $\mathsf{CA}(N' - 1; 2, k, v - 1)$ having profile $(v - 1)^k$. (It may have more $\star$ entries in columns other than the last, but its profile dominates $(v - 1)^k$.) Replace the $v-1$ $\star$ entries in the final column by entries $0,\dots,v-2$, using each symbol exactly once. Then the $v-1$ corresponding rows are pairwise disjoint, and the result is a $\mathsf{CA}(N'-1;2,k,v-1)$ having $v-1$ pure constant rows, and profile $(v-1)^{k-1}0^1$. ■

**Lemma 6.6** *Suppose that there exists a* $\mathsf{CA}(q^2;2,q+1,q)$ *with* $q$ *odd. Then there exists a* $\mathsf{CA}(q^2-2;2,q+1,q-1)$ *having profile* $((q-1)/2)^q(q-2)^1$*, and a* $\mathsf{CA}(q^2-2;2,q+1,q-1)$ *with profile* $((q-1)/2)^q0^1$ *having* $q-2$ *pure constant rows.*

*Proof.* Permute symbols so that the $q$ rows in which $q-1$ appears in the last column are constant on columns $1,\dots,q$. Delete the pure constant row containing $q-1$, and replace all other $q-1$ entries by $\star$. Next delete the constant row containing $q-2$. Select an arbitrary tournament $T$ on vertices $1,\dots,q$ that is in- and out-regular of degree $(q-1)/2$. Whenever $(x,y)$ is an arc of $T$, locate the row that contains $\star$ in column $x$ and $q-2$ in column $y$, and replace the $\star$ by $q-2$. The result is a $\mathsf{CA}(q^2-2;2,q+1,q-1)$ with $((q-1)/2)$ $\star$ entries in each of the first $q$ columns and $q-2$ in the last. Extending each of the $q-2$ constant rows to be pure constant yields the second CA. ■

Using these constructions, we obtain the following:

**Lemma 6.7** *Suppose that there exists a* $\mathsf{CA}(N';2,k,v)$ *in which for some symbol* $\sigma$ *and some column* $c$*,* $\sigma$ *appears exactly* $v$ *times in column* $c$*. Suppose further that there exists a* $\mathsf{CA}(M';2,\ell_1,v-1)$ *having* $s$ *constant rows.*

1. *If a* $\mathsf{CA}(M'-(v-1);2,\ell_2,v)$ *having* $s$ *constant rows exists, there exists a* $\mathsf{CA}((M'-s)+(N'-v);2,(k-1)\ell_1+\ell_2,v-1)$ *having* $v-1$ *constant rows.*
2. *If* $M'\geq v(v-1)$*, there exists a* $\mathsf{CA}((M'-s)+(N'-v);2,(k-1)\ell+2,v-1)$ *having* $v-1$ *constant rows.*
3. *There always exists a* $\mathsf{CA}((M'-s)+(N'-v);2,(k-1)\ell+1,v-1)$ *having* $v-1$ *constant rows.*

*Proof.* Taking $\ell_2=2$ and $\ell_1=\ell$ in the first statement implies the second, because a $\mathsf{CA}(M'-(v-1);2,2,v)$ exists with $v-1$ disjoint rows when $M'-(v-1)\geq(v-1)^2$. Taking $\ell_2=1$ and $\ell_1=\ell$ in the first statement implies the third. So we establish the first statement. By Lemma 6.5, there is a $\mathsf{CA}(N'-1;2,k,v-1)$ with $v-1$ pure constant rows having profile $(v-1)^{k-1}0^1$ Apply Corollary 4.5 with $r_1=s_2=s$, $r_2=v-1-s$, $s_1=0$, $N=N'-1$, and $M=M'-(v-1)$. ■

**Lemma 6.8** *Suppose that there exists a* $\mathsf{CA}(N';2,k,v)$ *that contains a pure constant row on symbol* $\sigma$*, and symbol* $\sigma$ *appears exactly* $v$ *times in columns* $1,\dots,v-1$ *and* $k$*. If there exist a* $\mathsf{CA}(M'-(v-2);2,\ell_2,v-1)$*, a* $\mathsf{CA}(M';2,\ell_1,v-1)$*, and a* $\mathsf{CA}(M'-(v-1);2,\ell_3,v-1)$ *each having* $s$ *constant rows, then a* $\mathsf{CA}((M'-s)+(N'-1);2,v\ell_1+(k-v)\ell_2+\ell_3,v-1)$ *having* $v-1$ *constant rows.*

*Proof.* By Lemma 6.13, a $\mathsf{CA}(N-1;2,k+1,v-1)$ exists having profile $1^{v-1}(v-1)^{k-v+1}0^1(N-v(v-1))^1$ and having $v-1$ pure constant rows. Apply Corollary 4.5 with $r_1 = s_2 = s$, $r_2 = v-1-s$, $s_1 = 0$, $N = N'-1$, and $M = M'-(v-1)$. ∎

**Lemma 6.9** *Suppose that there exists a* $\mathsf{CA}((z-1)(v-1)+v;2,z,v)$ *in which symbol* $v-1$ *appears exactly* $v$ *times in each column. Let* $1 \leq \alpha < v$. *Suppose that there is a* $\mathsf{CA}(M;2,\ell,v-\alpha)$ *containing* $v-\alpha$ *constant rows and a disjoint set of* $v-\alpha$ *rows containing* $\gamma$ *permutation columns. Then there is a* $\mathsf{PCA}((z-1)(v-1)+M-(\alpha-1)z-(v-\alpha);2,(z\ell,(z-1)\ell+\gamma),v-\alpha)$.

*Proof.* Suppose that the rows of the $\mathsf{CA}((z-1)(v-1)+v;2,z,v)$ that contain $v-1$ in column $z$ are otherwise constant. Replace all entries of $v-1,\dots,v-\alpha$ by $\star$ and delete the $\alpha$ rows that contain only $\star$, leaving an $((z-1)(v-1)+v-\alpha)\times z$ array $\mathsf{A}$. Then $\mathsf{A}$ has $(\alpha-1)(z-1)+(v-1)$ $\star$ entries in the first $z-1$ columns, and $(\alpha-1)z+(v-\alpha)$ in column $z$, so has profile $(v-1+(\alpha-1)(z-1))^z$. Let $\mathsf{B}$ be the $\mathsf{CA}(M;2,\ell,v-\alpha)$, with $\mathsf{B}_1$ being the $v-\alpha$ constant rows, $\mathsf{B}_2$ being the $v-\alpha$ rows containing $\gamma$ permutation columns, $\mathsf{B}_3$ being a further set of $(\alpha-1)z$ rows, and $\mathsf{B}_4$ being the remaining $M-(\alpha-1)z-2(v-\alpha)$ rows. Let $\mathsf{C}$ be $\mathsf{A}\otimes\mathsf{B}_4$. Column $z$ of $\mathsf{A}$ leads to a $(v-\alpha)\times\ell$ block of $\star$ entries in the first $v-\alpha$ rows. Replace this block with $\mathsf{B}_2$, so that the first $v-\alpha$ rows now have $(z-1)\ell+\gamma$ permutation columns. The remaining $\star$ entries in column $z$ yield $(\alpha-1)z$ rows, in which we place the rows of $\mathsf{B}_3$. For each of the remaining columns, the $\star$ entries form a $((\alpha-1)(z-1)+(v-1))\times\ell$ block of $\star$ entries, which we replace with the rows of $\mathsf{B}_2$ and $\mathsf{B}_3$.
The verification parallels the proof of Theorem 4.2, but we must verify that all constant pairs are covered, because the first $v-\alpha$ rows may not be pure constant. This is routine, however, because every constant pair is covered in $\mathsf{A}$. ∎

In the absence of other information, Lemma 6.9 can always be applied with $\gamma = 2$. When the $\mathsf{CA}(M;2,\ell,v-\alpha)$ is known to have two disjoint parallel classes, the resulting PCA in fact has a parallel class.
In certain situations, the manner in which the $\star$ positions are used can be varied to ensure the presence of many disjoint parallel classes. We pursue this next.

**Theorem 6.10** *Suppose that there exists a* $\mathsf{CA}(N+1;2,k,v+1)$ *with* $k \geq v$ *in which some row covers pairs, none of which are covered in another row. Suppose further that there is a* $\mathsf{CA}(M;2,\ell_1,v)$ *containing a parallel class and a* $\mathsf{CA}(M;2,\ell_2,v)$ *containing two disjoint parallel classes. Then there exists a* $\mathsf{CA}(N+M-2v;2,v\ell_2+(k-v)\ell_1+1,v)$. *When* $N+1=(v+1)^2$, *for* $1 \leq \alpha \leq k$, *there also exists a* $\mathsf{CA}(N+M-2v;2,(k-\alpha)\ell_1+\alpha\ell_2,v)$ *having* $\alpha$ *disjoint parallel classes, and a* $\mathsf{CA}(N+M-2v;2,(k-\alpha)\ell_1+\alpha\ell_2+1,v)$ *having* $\alpha-v$ *disjoint parallel classes when* $\alpha \geq v$.

*Proof.* Let $\mathsf{A}$ be the $\mathsf{CA}(N;2,k,v)$ obtained from the $\mathsf{CA}(N+1;2,k,v+1)$ by making the row in which every pair is uniquely covered constant on symbol $v+1$, deleting this row, and changing all remaining occurrences of $v+1$ to $\star$. Every row of $\mathsf{A}$ contains at most one $\star$. In the $\mathsf{CA}(M;2,\ell_2,v)$, make one parallel class contain constant rows, and then delete these rows. Let $\mathsf{D}_2$ be the $v$ rows of the second parallel class, and the remaining $M-2v$ rows be $\mathsf{B}_2$. In the $\mathsf{CA}(M;2,\ell_1,v)$, make one

parallel class contain constant rows, and then delete these rows. Let $\mathsf{D}_1$ be any $v$ remaining rows, and the remaining $M-2v$ rows be $\mathsf{B}_1$.
Choose $\alpha$ so that $0 \leq \alpha \leq k$. Form an $(N+M-2v)\times((k-\alpha)\ell_1+\alpha\ell_2)$ matrix $\mathsf{C}$ as follows. In the first $N$ rows, replicate $\alpha$ columns of $\mathsf{A}$ $\ell_2$ times each, then the remaining $k-\alpha$ columns of $\mathsf{A}$ $\ell_1$ times each. Each of the first $\alpha$ columns of $\mathsf{A}$ contains at least $v$ $\star$ entries and hence forms a $v\times\ell_2$ subarray containing only $\star$ entries in the first $N$ rows of $\mathsf{C}$; replace this subarray with $\mathsf{D}_2$. Each of the last $k-\alpha$ columns of $\mathsf{A}$ contains at least $v$ $\star$ entries and hence forms a $v\times\ell_1$ subarray containing only $\star$ entries in the first $N$ rows of $\mathsf{C}$; replace this subarray with $\mathsf{D}_1$. In the remaining $M-2v$ rows of $\mathsf{C}$, concatenate $\alpha$ copies of $\mathsf{B}_2$ followed by $k-\alpha$ copies of $\mathsf{B}_1$.
Then $\mathsf{C}$ is a $\mathsf{CA}(N+M-2v;2,v\ell_2+(k-v)\ell_1,v)$, but we can say more. Suppose that $1\leq c\leq \min(\alpha,v)$. Consider the rows of $\mathsf{A}$ that contain $\star$ in column $c$. Adjoin a new column to $\mathsf{C}$, and place symbol $c$ in the added column in each of these rows. This ensures that symbol $c$ in the added column appears with each symbol in each other column. Hence taking $\alpha=v$, we obtain a $\mathsf{CA}(N+M-2v;2,v\ell_2+(k-v)\ell_1+1,v)$.
Now consider cases when $N+1=(v+1)^2$. Then every column $c$ of $\mathsf{A}$ contains exactly $v$ $\star$ entries, and the rows containing these $\star$ entries yield a parallel class in $\mathsf{C}$ whenever $1\leq c\leq\alpha$. When $\alpha\geq v$, a new column can again be added, placing $\star$ in the rows arising from parallel classes with $c>v$. Then $\alpha-v$ parallel classes remain in the extended covering array. ■

## 6.2 Projection of orthogonal arrays

Projection was introduced in [46] and generalized in [18]. We apply it here to a specific family of orthogonal arrays.

**Theorem 6.11** *Let $q$ be a prime power. Let $x$ be an integer with $0<x<q$. Then there is a* $\mathsf{CA}(q^2-x;2,q+1+x,q-x)$

1. *with profile* $0^{q-1}(q-1)^1(q-1)^1(2(q-1))^1$ *when* $x=1$*;*
2. *with profile* $2^{q-a_1-a_2}(q)^{a_1}(2q-2)^{a_2}(2q-3)^1(3q-4)^2$ *when* $x=2$*, whenever* $(a_1,a_2)\in\{(4,0),(2,1),(0,2)\}$ *and* $a_1+a_2\leq q$*;*
3. *with profile* $6^{q-a_1-a_2-a_3}(q+3)^{a_1}(2q)^{a_2}(3q-3)^{a_3}(3q-3)^1(4q-6)^3$ *when* $x=3$*, whenever* $(a_1,a_2,a_3)\in\{(9,0,0),(7,1,0),(6,0,1),(5,2,0),(4,1,1),(3,3,0),(3,0,2),(2,2,1),(1,4,0),(1,1,2),(0,3,1),(0,0,3)\}$ *and* $a_1+a_2+a_3\leq q$*.*

*There is also a* $\mathsf{CA}(q^2-x;2,q+1+x,q-x)$ *having* $q-x$ *pure constant rows*

1. *with profile* $1^{q-1}(q-1)^1 0^1(q-1)^1$ *when* $x=1$*;*
2. *with profile* $4^{q-a_1-a_2}(q+1)^{a_1}(2q-2)^{a_2}q^1(2q-2)^2$ *when* $x=2$*, whenever* $(a_1,a_2)\in\{(4,0),(2,1),(0,2)\}$ *and* $a_1+a_2\leq q$*;*

3. *with profile* $9^{q-a_1-a_2-a_3}(q+5)^{a_1}(2q+1)^{a_2}(3q-3)^{a_3}(2q)^1(3q-3)^3$ *when* $x = 3$*, whenever* $(a_1, a_2, a_3) \in \{(9,0,0), (7,1,0), (6,0,1), (5,2,0), (4,1,1), (3,3,0), (3,0,2), (2,2,1), (1,4,0), (1,1,2), (0,3,1), (0,0,3)\}$ *and* $a_1 + a_2 + a_3 \leq q$.

*Proof.* Let $q$ be a power of a prime, and let $\mathbb{F}_q$ be the finite field on $q$ elements, with multiplication $\otimes$ and addition $\oplus$. We form a $q^2 \times q+1$ array $\mathsf{A}$ that is a $\mathsf{CA}(q^2; 2, q+1, q)$ as follows. Rows are indexed by polynomials of degree less than two over $\mathbb{F}_q$. We refer to the elements of $\mathbb{F}_q$ as $\{0, \ldots, q-1\}$, where $0 \otimes z = z \otimes 0 = 0$ for $z \in \mathbb{F}_q$. Index columns by $0, \ldots, q-1$ and $\infty$ in that order. For $a, b \in \mathbb{F}_q$, in the row indexed by $ax+b$ and the column indexed by $z_i$, place the entry $(a \otimes z_i) \oplus b$. In the row indexed by $ax+b$ and the column indexed by $\infty$, place the entry $a - 1 \bmod q$. The $q$ rows arising from polynomials of the form $0x+b$ are *near-constant rows,* because they are constant on the first $q$ columns (and have entry $q-1$ in the column headed by $\infty$). The remaining rows are *transverse rows*.
For $x \geq 1$ we 'project' to form an array $\mathsf{A}_x$ as follows. First delete the near-constant rows from $\mathsf{A}$ that contain symbols $q-y$ for $y \leq x$, in the process removing $x$ rows. Then adjoin $x$ additional columns indexed by $\infty_x, \ldots, \infty_1$. Place $\star$ in each of these new columns in each near-constant row.
For each $1 \leq y \leq x$, choose a permutation $\pi_y$ of $\{0, \ldots, q-1\}$. In every transverse row, each symbol occurs exactly once in columns $0, \ldots, q-1$. Then for $1 \leq y \leq x$, if in transverse row $\rho$ we find $q-y$ in column $\pi_y(a)$ and $a < q-x$, place $a$ in column $\infty_y$ in that row. For each $0 \leq a < q-x$, among the $q-1$ transverse rows having $q-y$ in column $\pi_y(a)$ place the symbols $\{0, \ldots, q-x-1\}$ in column $\pi_y(a)$ once each and set the remainder to $\star$. Replace all remaining entries from $\{q-x, \ldots, q-1\}$ by $\star$. The resulting array $\mathsf{A}_x$ is a $\mathsf{CA}(q^2 - x; 2, q+1+x, q-x)$ [18], and so the issue is to determine its profile.
Column $\infty$ has $q-x$ $\star$ entries on the near-constant rows, and $(x-1)q$ on the transverse rows. Each of columns $\infty_y$ for $1 \leq y \leq x$ has $q-x$ $\star$ entries on the near-constant rows, and $x(q-1)$ on the transverse rows. Column $i$ for $0 \leq i < q$ has no $\star$ entries on the near-constant rows, and has $\alpha_i(q-1) + (x-\alpha_i)(x-1)$ $\star$ entries on the transverse rows, where $\alpha_i = |\{y : \pi_y(i) \geq q-x, 1 \leq y \leq x\}|$. By choosing the permutations $\{\pi_y : 1 \leq y \leq x\}$ appropriately, we can select any integers $\alpha_0, \ldots, \alpha_{q-1}$ that satisfy $0 \leq \alpha_i \leq x$ and $\sum_{i=0}^{q-1} \alpha_i = x^2$.
For $x = 1$, column $\infty$ has $q-1$ $\star$ entries and column $\infty_1$ has $2(q-1)$. One of columns $\{0, \ldots, q-1\}$ has $q-1$, and the rest have 0. For $x = 2$, column $\infty$ has $2q-3$ and columns $\infty_1$ and $\infty_2$ have $3q-4$. For columns $\{0, \ldots, q-1\}$, we can choose the $\{\alpha_i\}$ values to get different results. Taking two of them to be 2, we get two columns with $2q-2$ and $q-2$ with 2. Taking one to be 2 and two to be 1, we get one column with $2q-2$, two with $q$, and $q-3$ with 2. Taking four to be 1, we get four columns with $q$ and $q-4$ with 2.
For $x = 3$, column $\infty$ has $3q-3$ and columns $\infty_1$, $\infty_2$, and $\infty_3$ have $4q-6$. When $\alpha_i = 3$, column $i$ has $3q-3$ $\star$ entries; when $\alpha_i = 2$, it has $2q$; when $\alpha_i = 1$, it has $q+3$; and when $\alpha_i = 0$ it has 6.
Pure constant rows in $\mathsf{A}_x$ could be formed by making each of the near-constant rows constant. However, one can then form further $\star$ positions, as follows. Suppose that $q-y$ appears in column $\pi_y(a)$ and $a < q-x$. Once the near-constant rows are made constant, the placement of $a$ in column $\pi_y(a)$ to replace $q-y$ is no longer needed, and can be changed to $\star$. Now we adjust the counts of $\star$ entries appropriately. Column $\infty$ has $(x-1)q$ $\star$ entries on the transverse rows. Each of columns

$\infty_y$ for $1 \leq y \leq x$ has $x(q-1)$ $\star$ entries on the transverse rows. Column $i$ for $0 \leq i < q$ has $\alpha_i(q-1)+(x-\alpha_i)x$ $\star$ entries on the transverse rows, where $\alpha_i = |\{y : \pi_y(i) \geq q-x, 1 \leq y \leq x\}|$. Then a similar analysis establishes the results stated. ■

We can also use projection to construct arrays with multiple disjoint parallel classes:

**Theorem 6.12** *Let $q$ be a prime power. Let $x$ be an integer with $0 < x < q$. Then there is a* $\mathsf{CA}(q^2-x;2,q+1+x,q-x)$ *having $x+1$ disjoint parallel classes.*

*Proof.* Let $\mathsf{A}$ be the $\mathsf{CA}(q^2;2,q+1,q)$ formed in Theorem 6.11. Form $\mathsf{B}$ by first adjoining $x$ columns to $\mathsf{A}$, indexed by $\infty_x,\dots,\infty_1$, changing all entries in transverse rows in column $\infty$ that belong to $\{q-x,\dots,q-1\}$ to $\star$. Then delete the $x$ near-constant rows containing symbols from $\{q-x,\dots,q-1\}$, and make the remaining near-constant rows pure constant (by extending the value in columns $0,\dots,q-1$ through columns $\{\infty,\infty_x,\dots,\infty_1\}$. For $1 \leq y \leq x$, whenever symbol $q-y$ appears in column $c$ of a row, place $c$ in column $\infty_y$ of that row if $c < q-x$. All entries of columns $\infty_1,\dots,\infty_x$ not so determined are set to $\star$.
The $q^2-q$ transverse rows are partitioned into classes as follows: class $W_i$ contains the $q-1$ row indices of rows in which $q-1$ appears in column $i$. We partition $W_i$ arbitrarily into two sets, $R_i$ containing $q-x$ row indices and $S_i$ containing $x-1$. For $0 \leq c < q-x$, place a permutation of symbols $\{0,\dots,q-x-1\}$ in the cells of column $c$ in the rows of $R_c$ and place $\star$ in those of the rows of $S_c$. For $q-x \leq c < q$, place $\star$ in the cells of column $c$ in each row of $W_c$.
Now for $2 \leq y \leq x$, we proceed differently. For $q-x \leq c < q$, when $q-y$ appears in column $c$, simply replace it by $\star$. For $0 \leq c < q-x$, let $T$ be the set of row indices in which $q-y$ appears in column $c$. Now $T \cap W_j$ contains one row index when $c \neq j$, and none otherwise. Therefore $T \cap (\bigcup_{j=0}^{q-x-1} W_j)$ contains $q-x-1$ row indices, so in column $c$ place a permutation of $\{0,\dots,q-x-1\} \setminus \{c\}$ in the corresponding rows. (Note that the pair containing $c$ in column $c$ and $c$ in column $\infty_y$ appears in a pure constant row.) Place a $\star$ in the columns of the remaining $x$ rows of $T$.
Now we recover the parallel classes. By construction, one is the set of pure constant rows. The remaining $x$ parallel classes are those indexed by $R_c$ for $q-x \leq c < q$. Indeed, no replacement of an element of any row of $W_{q-x},\dots,W_{q-1}$ is made except for replacement of entries by $\star$. The rows of $W_c$ initially agreed only in having $q-1$ in column $c$, and hence no two rows of $W_c$ (for $c \geq q-x$) agree in any position except for the entry $\star$. ■

## 6.3 Projection and cover starters

In [18, 35, 36], covering arrays are produced that admit a sharply transitive group action on $k-1$ columns, and a second sharply transitive group action on $v-1$ symbols. In [35], this is generalized to allow a sharply transitive group action on $v-f$ symbols, fixing the remaining $f$. The basic device is to produce a single row, a $(v,k,f)$-*cover starter*, to be developed under the action of the two chosen groups. As discussed in [35], when the cover starter itself contains a don't-care position, covering arrays with many (predictable) profiles result. We pursue a different avenue here, focussing on the case when $f=1$.

Cover starters for $f = 1$ produce a $\mathsf{CA}(k(v-1)+1; 2, k, v)$ with the property that there is a pure constant row (containing only the fixed symbol), and every other row contains the fixed symbol in exactly one position. This property is precisely what is needed to apply projection [18].

**Lemma 6.13** *If a* $\mathsf{CA}(N; 2, k, v)$ *exists that contains a pure constant row on symbol* $\sigma$*, and symbol* $\sigma$ *appears exactly* $v$ *times in columns* $1, \ldots, v-1$ *and* $k$*, then a* $\mathsf{CA}(N-1; 2, k+1, v-1)$ *exists having profile* $1^{v-1}(v-1)^{k-v}0^1(N-1-v(v-1))^1$ *and having* $v-1$ *pure constant rows. In particular, if a* $(v, k, 1)$*-cover starter exists, then a* $\mathsf{CA}(k(v-1); 2, k+1, v-1)$ *exists having profile* $1^{v-1}(v-1)^{k-v}0^1((k-v+1)(v-1))^1$ *and having* $v-1$ *pure constant rows.*

*Proof.* Form the $\mathsf{CA}(N; 2, k, v)$ so that the pure constant row is on symbol $v-1$, and every other row contains (at most) one $v-1$ among columns $\{1, \ldots, v-2\} \cup \{k\}$. Rename symbols in each column, always fixing $v-1$, so that $v$ rows are constant on the first $k-1$ columns and contain $v-1$ in the last. Delete the pure constant row on symbol $v-1$. Add a new column. For $1 \leq c \leq k$, let $R_c$ be the set of row indices of rows that contain $v-1$ in column $c$. For $1 \leq c \leq v-1$, in each row in $R_c$ place $c-1$ in the new column, and place a permutation of $\{0, \ldots, v-2, \star\} \setminus \{c-1\}$ in column $c$ in the rows of $R_c$. For $v \leq c < k$, place $\star$ in each row of $R_c$ both in column $c$ and in the new column. For $c = k$, extend each row in $R_c$ to be pure constant. Then columns $1, \ldots, v-1$ have 1 $\star$ entry each; columns $v, \ldots, k-1$ have at least $v-1$; column $k$ has none; and the added column has $N-1-v(v-1)$. ■

As a construction for covering arrays, Lemma 6.13 is not terribly useful because a $(k+1, v-1, 1)$-cover starter typically yields a smaller array. However, the array constructed by projection has both a full set of pure constant rows, and a 'large' profile.

## 6.4 Holey Transversal Designs

Let $V$ be a set of $hn$ symbols partitioned into $n$ sets $V_1, \ldots, V_n$, each of size $h$. An $((nh)^2 - nh^2) \times k$ array $\mathsf{A}$ is a *holey transversal design* $\mathsf{HTD}(k; nh, h)$ if every symbol in $\mathsf{A}$ is in $V$, and for every two different columns $\gamma_1$ and $\gamma_2$ of $\mathsf{A}$ and every two symbols $x \in V_i$ and $y \in V_j$, exactly one row of $\mathsf{A}$ contains $x$ in column $\gamma_1$ and $y$ in column $\gamma_2$ if and only if $i \neq j$. When $x, y \in V_i$ for some $i$, there is no row of $\mathsf{A}$ with $x$ in column $\gamma_1$ and $y$ in column $\gamma_2$.
By placing a $\mathsf{CA}(N; 2, k, h)$ on the symbols of $V_i$ for each $1 \leq i \leq n$, we obtain a $\mathsf{CA}((nh)^2 - nh^2) + nN; 2, k, hn)$. Let us consider an application of this. In [1] it is shown that an $\mathsf{HTD}(7; 2q, 2)$ exists whenever $q$ is an odd prime power and $7 \leq q \leq 61$. Because $\mathsf{CAN}(2, 7, 2) = 6$, we obtain that $\mathsf{CAN}(2, 7, 2q) \leq 4q^2 - 4q + 6q = 4q^2 + 2q$ whenever $q$ is an odd prime power and $7 \leq q \leq 61$. In fact, we can do somewhat better. The HTDs constructed in [1] have $V = \mathbb{F}_q \times \{0, 1\}$ and $V_i = \{i\} \times \{0, 1\}$ for $i \in \mathbb{F}_q$. The additive group of $\mathbb{F}_q$ is an automorphism group acting on the symbols. So choose any row $((\nu_1, \mu_1), \ldots, (\nu_7, \mu_7))$. Under the action of $\mathbb{F}_q$, this row generates $q$ disjoint rows of the HTD. These $q$ rows use exactly half of the symbols in each column; in fact, whenever $(\nu, i)$ is in one of these rows, $(\nu, 1-i)$ is not in any of them. Then in placing the $\mathsf{CA}(6; 2, 7, 2)$ on symbols in $V_i$, ensure that

one of the rows uses none of the symbols in the $q$ disjoint rows already produced. In this way, we produce a further disjoint row for each $V_i$, to establish that there is a $\mathsf{CA}(4q^2+2q; 2, 7, 2q)$ having $2q$ disjoint rows.
Using the construction in [1], in addition to the row $((\nu_1, \mu_1), \ldots, (\nu_7, \mu_7))$ one can select another row $((\nu_1', \mu_1'), \ldots, (\nu_7', \mu_7'))$ with $(\mu_1, \ldots, \mu_7) \neq (\mu_1', \ldots, \mu_7')$. By selecting the CA(6;2,7,2) so that $(1-\mu_1, \ldots, 1-\mu_7)$ and $(1-\mu_1', \ldots, 1-\mu_7')$ are two rows, which can always be done, a second set of $2q$ disjoint rows can be found, that is disjoint from the first, to establish:

**Lemma 6.14** *There is a* $\mathsf{CA}(4q^2+2q; 2, 7, 2q)$ *having two disjoint parallel classes whenever* $q$ *is an odd prime power and* $7 \le q \le 61$.

Now let $\mathsf{A}$ be the $\mathsf{HTD}(7; 2q, 2)$ and let $\mathsf{A}'$ be the result of interchanging the names of symbols $(\nu, i)$ and $(\nu, 1-i)$ for every $\nu \in \mathbb{F}_q$. Then $\mathsf{A} \otimes \mathsf{A}'$ is an $(8q^2 - 8q) \times 49$ array with columns indexed by $\{1, \ldots, 7\} \times \{1, \ldots, 7\}$. For every two different columns $(\gamma_1, \delta_1)$ and $(\gamma_2, \delta_2)$ and every two symbols $x \in V_i$ and $y \in V_j$, at least one row of $\mathsf{A} \otimes \mathsf{A}'$ contains $x$ in column $(\gamma_1, \delta_1)$ and $y$ in column $(\gamma_2, \delta_2)$ when $i \neq j$. When $x, y \in V_i$ for some $i$, there is no row of $\mathsf{A} \otimes \mathsf{A}'$ with $x$ in column $(\gamma_1, \delta_1)$ and $y$ in column $(\gamma_2, \delta_2)$, *unless* $x = y$ and either $\gamma_1 = \gamma_2$ or $\delta_1 = \delta_2$. There is a $8 \times 70$ array $F$ on two symbols $\{\sigma_1, \sigma_2\}$, formed by including as columns each of the $\binom{8}{4}$ vectors containing each symbol exactly four times. For every two columns of $F$, some row includes $(\sigma_1, \sigma_2)$ and some row includes $(\sigma_2, \sigma_1)$. However, the pairs $(\sigma_1, \sigma_1)$ and $(\sigma_2, \sigma_2)$ are covered if and only if the chosen columns do not differ in each position. For each column of $F$, exactly one other column differs in each position; call these an *antipodal pair of columns*. Now form an $8 \times 49$ array $F'$ as follows: Choose 49 columns of $F$, and index them by $\{1, \ldots, 7\} \times \{1, \ldots, 7\}$, so that if both columns of an antipodal pair are selected, their column indices $(\gamma_1, \delta_1)$ and $(\gamma_2, \delta_2)$ satisfy $\gamma_1 = \gamma_2$ or $\delta_1 = \delta_2$. For each $V_i$, place a copy of $F'$ on the symbols of $V'$. Together with $\mathsf{A} \otimes \mathsf{A}'$ these form a $\mathsf{CA}(8q^2; 2, 49, 2q)$.
There are $2q(2q-2)$ rows in $\mathsf{A}$ and therefore $4q-4$ orbit representatives under $\mathbb{F}_q$. Thus the rows of $\mathsf{A} \otimes \mathsf{A}'$ can be partitioned into $4q-4$ parallel classes. Add a new column. Suppose that the symbols in $\mathbb{F}_q \times \{0, 1\}$ are $\{\nu_1, \ldots, \nu_{2q}\}$. In the added column, in the rows for the $i$th parallel class, place symbol $\nu_i$. Then in the new column, in the rows for the $(2q+1)$st parallel class, place each of the symbols $\nu_1, \ldots, \nu_{2q}$ in one row. For the remaining $q(2q-5)$ rows, place $\star$ in the new column. This establishes:

**Lemma 6.15** *There is a* $\mathsf{CA}(8q^2; 2, 50, 2q)$ *having* $2q$ *pure constant rows and profile* $0^{49}(q(2q-5))^1$ *whenever* $q$ *is an odd prime power and* $7 \le q \le 61$.

Using these, we obtain:

**Lemma 6.16** *If* $7 \le q \le 61$, $q$ *is an odd prime power, and* $2q+1$ *is a prime power, then there exists a* $\mathsf{CA}(8q^2+2q; 2, 14q+14, 2q)$ *having* $2q$ *disjoint rows, and a* $\mathsf{CA}(8q^2+2q+1; 2, 14q+14, 2q)$.

*Proof.* Apply Theorem 6.10 using a $\mathsf{CA}((2q+1)^2; 2, 2q+2, 2q+1)$ and a $\mathsf{CA}(4q^2+2q; 2, 7, 2q)$ having two disjoint parallel classes from Lemma 6.14. ■

# 7 Computational Construction of Profiles

As we have discussed, numerous computational methods have been developed for the construction of covering arrays. In each case, the methods have concentrated on minimizing the number of tests (rows), and have typically not been concerned with other metrics. Stevens [45] and Stardom [44] develop simulated annealing methods that seek disjoint (constant) rows, and Cohen [12] employs simulated annealing to construct covering arrays with a specified pattern of "near-constant" rows. Their methods could in principle be extended to consider profiles as well. We instead adapt a simulated annealing method from [50] using a branch-and-bound procedure to find flexible positions from [27].

## 7.1 Simulated Annealing

The general procedure to obtain different profiles for an specific CA consists of four basic steps. First, make a small number of random changes to a CA, to produce an array of the same dimensions in which a small number of pairs may not be covered. Then, using simulated annealing [50], convert this quasi-CA to a CA. Then, using the exact algorithm reported in [27], detect the profile of the (possibly) new CA. Finally, if not recorded previously, register the new profile. This can be iterated any number of times.
In the implementation, a geometric cooling schedule is used. It starts with an initial temperature of $4.0$. This is repeatedly decremented by a factor of $\alpha$ determined by a Markov chain of length $L = (Nkv)^2$. The algorithm terminates when at least of one of the following conditions is met:

- the number of uncovered interactions is zero;
- the temperature reaches $1^{-10}$;
- eleven consecutive Markov chains do not improve the 'best' solution found; or
- a timeout of 4 minutes of computation is reached.

Using this procedure many profiles were detected. Using the CA(42; 2, 8, 6), a grand total of 36 different profiles were detected. Of these, 10 are not dominated by one of the others: $0^1 1^7$, $0^2 1^4 2^2$, $0^3 1^2 2^3$, $0^3 1^3 3^2$, $0^4 1^1 2^1 3^2$, $0^4 1^2 4^2$, $0^5 1^1 5^2$, $0^5 2^1 4^2$, $0^5 3^3$, $0^6 6^2$.

## 7.2 Post-optimization

Nayeri *et al.* [37] adopt a different strategy, using the observations in Section 3.4. They always start with a covering array, and hence their method is designed to optimize a covering array after its construction by another means. Their key idea is to repeatedly fill all don't-care positions with randomly selected values, then to identify the flexible positions, and finally to examine the flexible positions one by one, changing each to a don't-care position if it remains flexible at this point of the computation. In their case, the goal is to construct an entire row of don't care positions, which can then be eliminated. Whether or not the method succeeds in eliminating rows in this way, after each iteration the specific pattern of don't-care positions may change. By keeping track of the profiles of each intermediate covering array, we have found that a single covering array can lead to many different profiles.

A simple modification of the method in [37] enables one to treat many variants as well. Ensuring that no element in a specific set of rows contains a $\star$ entry or is permitted to have an entry changed to $\star$, the rows in this set are never changed. In this way we can make a set of rows *forced*, so that every solution produced by the algorithm always contains the forced rows. Hence the method can find a variety of profiles for covering arrays with a specified (minimum) number of constant rows, or a specified (minimum) number of parallel classes, for example. We have employed this primarily in cases when the number of constant rows is the maximum possible.

We provide one example. We again consider the CA(42; 2, 8, 6), but require a specified number $c$ of pure constant rows. Again dominated profiles are not listed. Because a solution for $c$ pure constant rows is also a solution for $c' \leq c$ pure constant rows, solutions that are so implied are also omitted.

| $c$ | Profiles |
|---|---|
| 0 | $0^2 1^6$, $0^3 1^4 2^1$, $0^4 1^2 2^2$, $0^5 1^1 2^1 3^1$, $0^5 2^3$, $0^6 3^2$ |
| 1 | |
| 2 | $0^4 1^3 2^1$ |
| 3 | $0^3 1^5$ |
| 4 | $0^5 1^1 2^2$, $0^7 3^1$ |
| 5 | $0^6 1^1 2^1$ |
| 6 | $0^4 1^4$ |

The simulated annealing method explores a larger search space than does post-optimization, and this may account for its success in finding a richer set of solutions.

## 7.3 Some Results on Parallel Classes

Of particular importance for applying generalized direct products is the construction of covering arrays with a parallel class, or two disjoint parallel classes. Theorems 6.12 and 6.14 yield some. For example, there is a CA(100;2,4,10) with two parallel classes and there is a CA(120;2,12,10) with profile $10^{12}$; hence there is a CA(200;2,48,10) with a parallel class. But effective application of the constructions requires many small ingredients.

Tables 1, 2, and 3 give the best current upper bound $N_c$ on $\mathsf{CAN}(2, k, v)$ for covering arrays with $c$ parallel classes for $0 \leq c \leq 2$, $7 \leq v \leq 25$, and $3 \leq k \leq 50$. We do not attempt to give detailed authorities for each entry. When data for a specific value of $k$ is omitted, employ the results for the next larger value of $k$. When an entry for $N_2$ is left blank, it can be determined by adding $v-1$ to the value given for $N_1$. The majority of the values given are computed using simulated annealing [50] or post-optimization [37] on arrays from orthogonal arrays [30], projection [18], cover starters [35], and direct products. Arrays are available on request from the authors.

| $v$ | $k$ | $N_0$ | $N_1$ | $N_2$ | $k$ | $N_0$ | $N_1$ | $N_2$ | $k$ | $N_0$ | $N_1$ | $N_2$ | $k$ | $N_0$ | $N_1$ | $N_2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 7 | 49 | 49 | 49 | 8 | 49 | 55 | 61 | 9 | 59 | 61 | 63 | 10 | 61 | 63 | 63 |
| | 11 | 65 | 67 | 72 | 12 | 71 | 72 | 73 | 13 | 71 | 76 | 81 | 14 | 76 | 79 | 84 |
| | 15 | 79 | 84 | 85 | 16 | 82 | 86 | 88 | 17 | 82 | 87 | 90 | 18 | 85 | 88 | 90 |
| | 19 | 86 | 88 | 90 | 20 | 87 | 88 | 90 | 22 | 88 | 88 | 90 | 23 | 90 | 90 | 90 |
| | 26 | 90 | 90 | 91 | 56 | 91 | 91 | 91 | | | | | | | | |
| 8 | 8 | 64 | 64 | 64 | 9 | 64 | 71 | 78 | 10 | 76 | 77 | 80 | 11 | 78 | 79 | 80 |
| | 13 | 84 | 88 | 95 | 14 | 96 | 101 | 103 | 15 | 96 | 102 | 105 | 16 | 102 | 108 | 112 |
| | 18 | 104 | 110 | 112 | 19 | 107 | 112 | 112 | 20 | 108 | 112 | 112 | 21 | 112 | 112 | 113 |
| | 22 | 112 | 113 | 113 | 23 | 113 | 113 | 113 | 24 | 113 | 113 | 114 | 25 | 114 | 114 | 115 |
| | 26 | 114 | 115 | 115 | 27 | 115 | 115 | 119 | 28 | 115 | 118 | 120 | 30 | 119 | 119 | 120 |
| | 72 | 120 | 120 | 120 | | | | | | | | | | | | |
| 9 | 9 | 81 | 81 | 81 | 10 | 81 | 89 | 97 | 11 | 102 | 102 | 106 | 12 | 105 | 105 | 111 |
| | 13 | 105 | 110 | 111 | 14 | 112 | 112 | 114 | 15 | 118 | 123 | 127 | 16 | 118 | 125 | |
| | 17 | 125 | 133 | | 19 | 129 | 136 | 143 | 20 | 132 | 139 | 143 | 21 | 139 | 143 | 143 |
| | 22 | 141 | 144 | 150 | 23 | 142 | 144 | 150 | 25 | 144 | 144 | 150 | 26 | 145 | 145 | 152 |
| | 28 | 146 | 151 | 153 | 29 | 147 | 151 | 153 | 30 | 148 | 151 | 153 | 31 | 152 | 153 | 153 |
| | 90 | 153 | 153 | 153 | | | | | | | | | | | | |
| 10 | 4 | 100 | 100 | 100 | 5 | 102 | 102 | 102 | 6 | 102 | 105 | 108 | 7 | 113 | 114 | 116 |
| | 8 | 115 | 115 | 116 | 9 | 115 | 116 | 117 | 10 | 116 | 116 | 117 | 11 | 116 | 117 | 118 |
| | 12 | 117 | 118 | 120 | 13 | 120 | 120 | 120 | 14 | 127 | 136 | 145 | 15 | 136 | 145 | 149 |
| | 16 | 145 | 149 | 149 | 17 | 149 | 150 | 152 | 18 | 150 | 158 | 167 | 20 | 155 | 163 | 172 |
| | 21 | 162 | 170 | 178 | 22 | 166 | 174 | 182 | 23 | 171 | 179 | | 24 | 178 | 186 | 191 |
| | 25 | 178 | 187 | 191 | 26 | 185 | 190 | 191 | 27 | 190 | 190 | 191 | 29 | 190 | 191 | 191 |
| | 30 | 190 | 191 | 192 | 32 | 191 | 191 | 194 | 36 | 191 | 192 | 198 | 37 | 198 | 199 | 200 |
| | 39 | 199 | 199 | 200 | 48 | 200 | 200 | 200 | 49 | 200 | 200 | 202 | 65 | 202 | 202 | 202 |
| 11 | 11 | 121 | 121 | 121 | 12 | 121 | 131 | 141 | 13 | 153 | 156 | 159 | 14 | 155 | 157 | 159 |
| | 15 | 158 | 160 | 160 | 16 | 161 | 162 | 163 | 17 | 171 | 181 | 184 | 18 | 177 | 183 | 191 |
| | 19 | 178 | 187 | 196 | 20 | 186 | 196 | 206 | 21 | 192 | 200 | 208 | 22 | 192 | 201 | 210 |
| | 23 | 200 | 209 | 218 | 24 | 203 | 212 | 220 | 25 | 204 | 213 | 222 | 26 | 211 | 220 | 226 |
| | 27 | 218 | 222 | 226 | 28 | 221 | 223 | 226 | 29 | 224 | 225 | 227 | 30 | 225 | 226 | 231 |
| | 32 | 226 | 229 | 231 | 33 | 227 | 229 | 231 | 35 | 229 | 231 | 231 | 132 | 231 | 231 | 231 |
| 12 | 6 | 144 | 144 | 144 | 7 | 144 | 147 | 150 | 8 | 162 | 163 | 163 | 9 | 163 | 163 | 164 |
| | 10 | 163 | 164 | 165 | 11 | 164 | 164 | 165 | 12 | 164 | 165 | 165 | 13 | 164 | 165 | 166 |
| | 14 | 165 | 166 | 168 | 15 | 168 | 168 | 168 | 16 | 188 | 188 | 199 | 17 | 188 | 199 | 210 |
| | 18 | 199 | 210 | 221 | 19 | 210 | 221 | 226 | 20 | 221 | 226 | 227 | 21 | 227 | 228 | 229 |
| | 22 | 227 | 237 | 245 | 23 | 232 | 242 | 248 | 24 | 232 | 242 | 252 | 25 | 242 | 252 | |
| | 27 | 245 | 255 | | 28 | 254 | 264 | | 29 | 262 | 272 | 276 | 32 | 269 | 276 | 276 |
| | 42 | 276 | 276 | 276 | 47 | 276 | 276 | 282 | 49 | 276 | 279 | 282 | 84 | 288 | 288 | 288 |
| 13 | 13 | 169 | 169 | 169 | 14 | 169 | 181 | 193 | 15 | 215 | 217 | 218 | 16 | 217 | 217 | 218 |
| | 17 | 217 | 217 | 229 | 18 | 217 | 229 | 239 | 19 | 229 | 240 | 240 | 20 | 241 | 242 | 243 |
| | 21 | 253 | 261 | 261 | 22 | 262 | 262 | 263 | 24 | 271 | 282 | 293 | 25 | 280 | 282 | 293 |
| | 27 | 281 | 282 | 293 | 28 | 282 | 293 | | 29 | 291 | 302 | | 30 | 299 | 311 | |
| | 32 | 301 | 312 | | 33 | 309 | 320 | 325 | 39 | 317 | 325 | 325 | 182 | 325 | 325 | 325 |

Table 1: Existence of CA$(N; 2, k, v)$s with 0, 1, or 2 parallel classes: $7 \leq v \leq 13$, $k \leq 50$

| $v$ | $k$ | $N_0$ | $N_1$ | $N_2$ | $k$ | $N_0$ | $N_1$ | $N_2$ | $k$ | $N_0$ | $N_1$ | $N_2$ | $k$ | $N_0$ | $N_1$ | $N_2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 14 | 6 | 196 | 196 | 196 | 7 | 210 | 210 | 210 | 8 | 229 | 233 | 237 | 9 | 233 | 236 | 239 |
| | 10 | 236 | 239 | 241 | 11 | 238 | 240 | 242 | 12 | 239 | 241 | 243 | 13 | 240 | 242 | 243 |
| | 14 | 241 | 243 | 244 | 15 | 242 | 244 | 244 | 16 | 243 | 244 | 245 | 17 | 244 | 245 | 246 |
| | 18 | 247 | 248 | 248 | 19 | 248 | 251 | 252 | 20 | 272 | 273 | 274 | 21 | 274 | 276 | 277 |
| | 22 | 287 | 300 | 313 | 23 | 300 | 300 | 313 | 24 | 306 | 314 | 322 | 25 | 322 | 322 | 323 |
| | 27 | 329 | 339 | | 30 | 330 | 342 | | 31 | 341 | 353 | | 33 | 342 | 354 | |
| | 34 | 351 | 363 | | 35 | 360 | 372 | 378 | 36 | 370 | 378 | 378 | 37 | 381 | 391 | 392 |
| | 38 | 388 | 391 | 392 | 40 | 391 | 391 | 392 | 42 | 392 | 392 | 392 | 50 | 392 | 392 | |
| 15 | 6 | 225 | 225 | 226 | 7 | 246 | 247 | 249 | 8 | 248 | 248 | 250 | 9 | 249 | 249 | 251 |
| | 10 | 249 | 250 | 251 | 11 | 250 | 250 | 252 | 13 | 251 | 251 | 252 | 14 | 251 | 252 | 252 |
| | 15 | 252 | 252 | 253 | 16 | 252 | 252 | 253 | 17 | 252 | 254 | 255 | 18 | 255 | 255 | 255 |
| | 19 | 281 | 282 | 282 | 20 | 285 | 285 | 286 | 21 | 331 | 332 | 332 | 22 | 332 | 332 | 332 |
| | 23 | 334 | 335 | 335 | 24 | 335 | 337 | 338 | 25 | 351 | 360 | | 26 | 365 | 379 | |
| | 28 | 376 | 387 | | 32 | 383 | 396 | | 33 | 394 | 407 | | 34 | 405 | 418 | |
| | 36 | 406 | 419 | | 37 | 418 | 431 | | 38 | 428 | 441 | 451 | 39 | 439 | 450 | 451 |
| | 40 | 449 | 450 | 451 | 98 | 450 | 450 | 451 | | | | | | | | |
| 16 | 16 | 256 | 256 | 256 | 17 | 256 | 271 | 286 | 18 | 286 | 287 | 288 | 19 | 288 | 288 | 288 |
| | 20 | 342 | 342 | 344 | 21 | 344 | 344 | 346 | 22 | 346 | 347 | 348 | 23 | 350 | 350 | 351 |
| | 24 | 376 | 376 | | 25 | 376 | 391 | | 26 | 406 | 406 | | 27 | 406 | 421 | |
| | 28 | 421 | 436 | | 29 | 436 | 451 | 455 | 30 | 442 | 456 | 456 | 36 | 442 | 456 | |
| | 38 | 466 | 480 | 493 | 39 | 478 | 483 | 493 | 40 | 483 | 483 | 493 | 45 | 490 | 490 | 496 |
| | 272 | 496 | 496 | 496 | | | | | | | | | | | | |
| 17 | 17 | 289 | 289 | 289 | 18 | 289 | 305 | 321 | 19 | 351 | 351 | 351 | 20 | 352 | 352 | 352 |
| | 21 | 354 | 354 | 354 | 22 | 357 | 357 | 358 | 23 | 465 | 465 | 467 | 24 | 467 | 467 | 468 |
| | 25 | 469 | 469 | 470 | 26 | 471 | 471 | 472 | 27 | 473 | 473 | 474 | 28 | 474 | 474 | 475 |
| | 29 | 477 | 477 | 477 | 30 | 479 | 479 | 480 | 31 | 497 | 513 | | 32 | 513 | 529 | 538 |
| | 34 | 517 | 532 | 538 | 39 | 517 | 532 | | 40 | 529 | 544 | | 41 | 541 | 556 | 561 |
| | 42 | 551 | 561 | 561 | 51 | 560 | 561 | 561 | | | | | | | | |
| 18 | 5 | 324 | 324 | 328 | 7 | 342 | 342 | 342 | 8 | 352 | 352 | 354 | 9 | 355 | 355 | 355 |
| | 11 | 355 | 355 | 356 | 12 | 355 | 355 | 357 | 16 | 356 | 357 | 357 | 17 | 357 | 357 | 358 |
| | 19 | 358 | 358 | 358 | 20 | 358 | 359 | 360 | 21 | 360 | 360 | 360 | 22 | 442 | 443 | 454 |
| | 23 | 484 | 487 | 488 | 24 | 487 | 487 | 488 | 25 | 489 | 489 | 489 | 26 | 490 | 490 | 491 |
| | 27 | 491 | 491 | 491 | 28 | 494 | 494 | 494 | 29 | 496 | 496 | 498 | 30 | 545 | 545 | 554 |
| | 31 | 545 | 545 | 556 | 32 | 545 | 558 | 562 | 33 | 558 | 558 | 571 | 34 | 579 | 596 | 613 |
| | 35 | 593 | 609 | 617 | 40 | 593 | 609 | 626 | 42 | 595 | 611 | | 44 | 603 | 615 | |
| | 45 | 615 | 627 | | 46 | 627 | 639 | 648 | 47 | 639 | 648 | 648 | 50 | 647 | 648 | 648 |
| 19 | 19 | 361 | 361 | 361 | 20 | 361 | 379 | 397 | 21 | 498 | 498 | 498 | 24 | 500 | 500 | 500 |
| | 25 | 502 | 502 | 502 | 26 | 504 | 504 | 506 | 27 | 507 | 507 | 508 | 28 | 509 | 509 | 511 |
| | 29 | 572 | 572 | 573 | 30 | 574 | 574 | 575 | 31 | 576 | 576 | 576 | 32 | 577 | 577 | 579 |
| | 33 | 637 | 637 | 640 | 34 | 638 | 638 | 641 | 35 | 640 | 640 | 643 | 36 | 645 | 645 | 647 |
| | 45 | 660 | 673 | 691 | 46 | 673 | 686 | 703 | 47 | 686 | 699 | 703 | 50 | 698 | 703 | 703 |

Table 2: Existence of CA$(N; 2, k, v)$s with 0, 1, or 2 parallel classes: $14 \leq v \leq 19$, $k \leq 50$

| $v$ | $k$ | $N_0$ | $N_1$ | $N_2$ | $k$ | $N_0$ | $N_1$ | $N_2$ | $k$ | $N_0$ | $N_1$ | $N_2$ | $k$ | $N_0$ | $N_1$ | $N_2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 6 | 400 | 400 | 401 | 7 | 429 | 433 | 437 | 8 | 463 | 468 | 473 | 9 | 482 | 486 | 488 |
| | 10 | 488 | 489 | 491 | 11 | 491 | 492 | 495 | 12 | 493 | 495 | 497 | 13 | 495 | 496 | 499 |
| | 14 | 497 | 498 | 500 | 15 | 499 | 500 | 502 | 16 | 500 | 501 | 503 | 17 | 501 | 502 | 504 |
| | 18 | 503 | 504 | 505 | 19 | 504 | 504 | 506 | 20 | 505 | 506 | 507 | 21 | 507 | 507 | 508 |
| | 22 | 508 | 508 | 509 | 23 | 508 | 508 | 510 | 24 | 511 | 511 | 512 | 25 | 513 | 513 | 515 |
| | 26 | 517 | 517 | 519 | 27 | 520 | 520 | 522 | 28 | 588 | 588 | 590 | 29 | 589 | 591 | 592 |
| | 30 | 591 | 593 | 594 | 31 | 594 | 596 | 596 | 32 | 659 | 660 | 661 | 33 | 661 | 662 | 664 |
| | 34 | 663 | 664 | 665 | 35 | 666 | 668 | 668 | 36 | 679 | 679 | 694 | 37 | 685 | 685 | 700 |
| | 38 | 707 | 707 | 722 | 39 | 740 | 740 | 755 | 43 | 754 | 770 | | 45 | 755 | 774 | |
| | 47 | 756 | 774 | | 49 | 761 | 779 | | 50 | 775 | 793 | | | | | |
| 21 | 6 | 441 | 441 | 441 | 7 | 441 | 446 | 451 | 8 | 499 | 502 | 503 | 9 | 501 | 506 | 507 |
| | 10 | 509 | 509 | 509 | 11 | 511 | 511 | 511 | 12 | 512 | 512 | 512 | 13 | 513 | 513 | 513 |
| | 15 | 514 | 514 | 514 | 17 | 516 | 516 | 516 | 18 | 517 | 517 | 517 | 21 | 518 | 518 | 518 |
| | 22 | 519 | 519 | 519 | 23 | 519 | 519 | 519 | 24 | 521 | 521 | 521 | 25 | 523 | 523 | 523 |
| | 26 | 526 | 526 | 527 | 27 | 600 | 600 | 600 | 28 | 603 | 603 | 603 | 29 | 605 | 605 | 606 |
| | 30 | 609 | 609 | 610 | 31 | 677 | 677 | 680 | 32 | 680 | 680 | 682 | 33 | 680 | 681 | 684 |
| | 34 | 683 | 683 | 685 | 35 | 752 | 753 | 755 | 36 | 754 | 754 | 758 | 37 | 757 | 758 | 761 |
| | 38 | 763 | 765 | 765 | 40 | 810 | 810 | 826 | 41 | 838 | 838 | 838 | 42 | 841 | 843 | 843 |
| | 54 | 847 | 861 | | | | | | | | | | | | | |
| 22 | 5 | 484 | 484 | 485 | 6 | 487 | 488 | 489 | 7 | 506 | 506 | 506 | 8 | 520 | 521 | 521 |
| | 9 | 521 | 522 | 522 | 10 | 521 | 523 | 523 | 11 | 522 | 523 | 524 | 12 | 523 | 524 | 524 |
| | 13 | 524 | 524 | 524 | 16 | 524 | 525 | 525 | 21 | 525 | 526 | 527 | 23 | 526 | 526 | 527 |
| | 24 | 526 | 527 | 528 | 25 | 528 | 528 | 528 | 26 | 609 | 609 | 609 | 27 | 611 | 611 | 611 |
| | 28 | 613 | 614 | 614 | 29 | 617 | 617 | 619 | 30 | 691 | 692 | 692 | 31 | 696 | 697 | 697 |
| | 32 | 698 | 698 | 700 | 33 | 699 | 700 | 702 | 34 | 777 | 777 | 779 | 35 | 779 | 779 | 781 |
| | 36 | 783 | 783 | 784 | 37 | 785 | 785 | 785 | 38 | 857 | 857 | 858 | 39 | 860 | 860 | 861 |
| | 40 | 862 | 862 | 862 | 41 | 868 | 868 | 869 | 42 | 905 | 905 | 905 | 43 | 912 | 912 | 912 |
| | 44 | 925 | 946 | | 54 | 940 | 956 | 969 | | | | | | | | |
| 23 | 23 | 529 | 529 | 529 | 24 | 529 | 551 | 573 | 25 | 616 | 616 | 616 | 26 | 616 | 616 | 617 |
| | 27 | 619 | 619 | 619 | 28 | 622 | 622 | 623 | 29 | 706 | 706 | 706 | 30 | 708 | 708 | 708 |
| | 31 | 712 | 712 | 712 | 32 | 714 | 714 | 716 | 33 | 793 | 793 | 795 | 34 | 793 | 795 | 798 |
| | 35 | 796 | 799 | 800 | 36 | 798 | 802 | 803 | 37 | 877 | 877 | 881 | 38 | 878 | 879 | 883 |
| | 39 | 879 | 883 | 886 | 40 | 882 | 888 | 888 | 41 | 922 | 925 | 925 | 42 | 926 | 931 | 933 |
| | 44 | 1019 | 1019 | 1035 | 58 | 1019 | 1035 | 1035 | | | | | | | | |
| 24 | 8 | 576 | 576 | 576 | 9 | 576 | 585 | 594 | 10 | 619 | 619 | 619 | 12 | 619 | 619 | 620 |
| | 13 | 620 | 620 | 620 | 14 | 620 | 620 | 621 | 17 | 620 | 620 | 622 | 18 | 621 | 621 | 622 |
| | 22 | 621 | 621 | 623 | 25 | 622 | 622 | 623 | 26 | 622 | 622 | 624 | 27 | 624 | 624 | 624 |
| | 28 | 713 | 713 | 714 | 29 | 716 | 716 | 717 | 30 | 718 | 718 | 719 | 31 | 722 | 722 | 724 |
| | 32 | 808 | 808 | 808 | 33 | 810 | 810 | 811 | 34 | 812 | 812 | 815 | 35 | 814 | 814 | 819 |
| | 36 | 896 | 896 | 900 | 37 | 896 | 896 | 904 | 38 | 904 | 904 | 905 | 39 | 907 | 907 | 909 |
| | 40 | 951 | 951 | 954 | 41 | 955 | 955 | 956 | 72 | 1128 | 1128 | 1128 | | | | |
| 25 | 25 | 625 | 625 | 625 | 26 | 625 | 649 | 673 | 28 | 719 | 722 | 722 | 29 | 725 | 725 | 725 |
| | 30 | 726 | 726 | 727 | 31 | 820 | 820 | 820 | 32 | 823 | 823 | 824 | 33 | 826 | 826 | 827 |
| | 34 | 828 | 828 | 830 | 35 | 914 | 914 | 917 | 36 | 917 | 917 | 920 | 37 | 919 | 919 | 922 |
| | 38 | 923 | 925 | 925 | 39 | 970 | 970 | 970 | 40 | 970 | 970 | 974 | 50 | 1181 | 1200 | 1215 |

Table 3: Existence of CA$(N; 2, k, v)$s with 0, 1, or 2 parallel classes: $20 \leq v \leq 25$, $k \leq 50$

## 8 Consequences and Conclusions

In this section, we outline some applications of the constructions of profiles. At [17], covering array numbers are tabulated for strength two when $3 \leq v \leq 25$, and $3 \leq k \leq 20000$. (Explicit solutions for many of these can be found at [49].) A majority of the recorded entries arise from the generalized direct products given in this paper. Given the dramatic number of improvements by generalized direct product, it is infeasible to enumerate even a small fraction of them. In Table 4, we provide summary statistics prior to, and after, the application of the constructions developed here.

| Type | Authority | Before | After |
|---|---|---|---|
| **Computational:** | Simulated annealing [50] | 245 | 208 |
| | Other | 22 | 22 |
| **Direct:** | OAs and Projection [18] | 231 | 231 |
| | Cover starter [35] | 129 | 129 |
| | Other | 7 | 7 |
| **Recursive:** | Cut-and-paste [21] | 777 | 549 |
| | Direct Product | 728 | 228 |
| | Generalized Direct Product | - | 1783 |
| | Total | 2139 | 3257 |

Table 4: Extent of Changes

In principle, in the ranges specified the tables could contain $23 \cdot 19998$ entries, but in practice a bound for $\mathsf{CAN}(2, k, v)$ is reported only when it is better than the bound for $\mathsf{CAN}(2, k + 1, v)$. Authorities are partitioned into three categories: computational, direct, and recursive. Unfortunately, this division is somewhat artificial. Results of direct constructions have often been improved by a computational method, but are attributed here to the underlying direct construction. In the 'after' column, this occurs in 158 of the cases using post-optimization [37] and 87 times using simulated annealing [50]. Similarly, results of direct and recursive constructions often provide the initial array used in simulated annealing, but the result here is attributed to the computational method. Among the 'other' computational results, one finds earlier simulated annealing methods [12, 20] and tabu search [39, 54], but the very popular greedy methods do not account for a single best result.

As expected, the generalized direct product improves upon the simpler direct product. It also improves quite often on the cut-and-paste method. As $v$ and $k$ increase, the generalized direct products provide more improvements. For $3 \leq v \leq 7$, cut-and-paste constructions typically remain more effective; and for $k \leq 50$ and larger $v$, combinations of direct and computational constructions are more effective.

To show the magnitude of the improvements obtained, Figure $6$ displays results with and without generalized direct product for $v = 14$ and $100 \leq k \leq 20000$. Throughout this range, the generalized direct product makes improvements, sometimes reducing the number of rows by 5%.

The success of the generalized direct product at improving on previously known bounds results both from the flexibility of the construction, and the manner in which other constructions can be adapted to

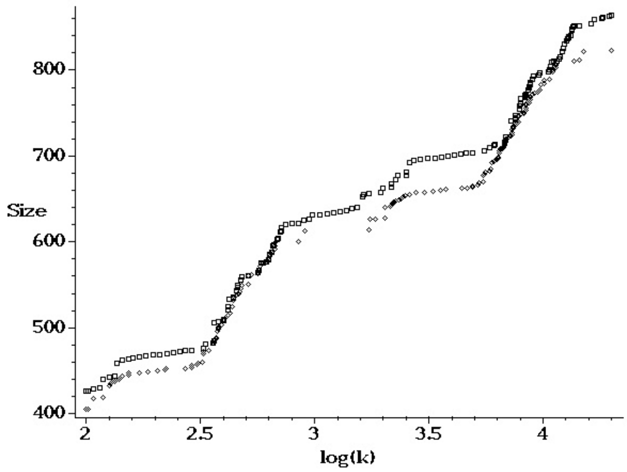


Figure 6: CAN(2,$k$,14) for $100 \leq k \leq 20000$

furnish ingredients. However, its most important aspect is that by investing more effort in ensuring that small ingredients have appropriate profiles and many disjoint rows, the construction can save many rows in the larger arrays constructed. This compounds as these arrays are again used as ingredients. Hence making covering arrays with many factors and few rows is simplified by finding 'better' small ingredient arrays. What we have shown is that, in many situation, 'better' means having the right profile.

The practical importance of the generalized direct products developed here is that by investing computational effort in finding 'good' small ingredient arrays, straightforward constructions of covering arrays with the fewest rows of those known can be employed to produce large arrays. This extends the range of testing problems to ones with many factors, without sacrificing complete coverage, and reducing both the time to construct and the time to run the corresponding test suite.

# References


[1] R. J. R. Abel, F. E. Bennett, and G. Ge. The existence of four HMOLS with equal sized holes. *Des. Codes Cryptogr.*, 26(1-3):7–31, 2002.

[2] R. C. Bryce and C. J. Colbourn. Prioritized interaction testing for pairwise coverage with seeding and avoids. *Information and Software Technology Journal*, 48:960–970, 2006.

[3] R. C. Bryce and C. J. Colbourn. The density algorithm for pairwise interaction testing. *Software Testing, Verification, and Reliability*, 17:159–182, 2007.

[4] R. C. Bryce and C. J. Colbourn. A density-based greedy algorithm for higher strength covering arrays. *Software Testing, Verification, and Reliability*, 19:37–53, 2009.

[5] R. C. Bryce, C. J. Colbourn, and M. B. Cohen. A framework of greedy methods for constructing interaction tests. In *Proceedings of the 27th International Conference on Software Engineering (ICSE)*, pages 146–155, Los Alamitos, CA, 2005. IEEE.

[6] D. A. Bulutoglu and F. Margot. Classification of orthogonal arrays by integer programming. *Journal of Statistical Planning and Inference*, 138:654–666, 2008.

[7] M. A. Chateauneuf, C. J. Colbourn, and D. L. Kreher. Covering arrays of strength 3. *Des. Codes Crypt.*, 16:235–242, 1999.

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

[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 Transactions on Software Engineering*, 23:437–44, 1997.

[10] D. M. Cohen, S. R. Dalal, J. Parelius, and G. C. Patton. The combinatorial design approach to automatic test generation. *IEEE Software*, 13:82–88, 1996.

[11] D. M. Cohen and M. L. Fredman. New techniques for designing qualitatively independent systems. *J. Combin. Des.*, 6:411–416, 1998.

[12] M. B. Cohen. *Designing test suites for software interaction testing*. PhD thesis, The University of Auckland, Department of Computer Science, 2004.

[13] M. B. Cohen, C. J. Colbourn, P. B. Gibbons, and W. B. Mugridge. Constructing test suites for interaction testing. In *Proc. Intl. Conf. on Software Engineering (ICSE 2003)*, pages 38–48, Los Alamitos, CA, 2003. IEEE.

[14] M. B. Cohen, C. J. Colbourn, and A. C. H. Ling. Constructing strength three covering arrays with augmented annealing. *Discrete Math.*, 308:2709–2722, 2008.

[15] M. B. Cohen, M. B. Dwyer, and J. Shi. Constructing interaction test suites for highly-configurable systems in the presence of constraints: a greedy approach. *IEEE Transactions on Software Engineering*, 34:633–650, 2008.

[16] C. J. Colbourn. Combinatorial aspects of covering arrays. *Le Matematiche (Catania)*, 58:121–167, 2004.

[17] C. J. Colbourn. Covering array tables, 2005-2013. http://www.public.asu.edu/∼ccolbou/src/tabby.

[18] C. J. Colbourn. Strength two covering arrays: Existence tables and projection. *Discrete Math.*, 308:772–786, 2008.

[19] C. J. Colbourn. Covering arrays and hash families. In *Information Security and Related Combinatorics*, NATO Peace and Information Security, pages 99–136. IOS Press, 2011.

[20] C. J. Colbourn, G. Kéri, P. P. Rivas Soriano, and J.-C. Schlage-Puchta. Covering and radius-covering arrays: Constructions and classification. *Discrete Applied Mathematics*, 158:1158–1190, 2010.

[21] C. J. Colbourn, S. S. Martirosyan, G. L. Mullen, D. E. Shasha, G. B. Sherwood, and J. L. Yucas. Products of mixed covering arrays of strength two. *J. Combin. Des.*, 14:124–138, 2006.

[22] M. Forbes, J. Lawrence, Y. Lei, R. N. Kacker, and D. R. Kuhn. Refining the in-parameter-order strategy for constructing covering arrays. *J. Res. Nat. Inst. Stand. Tech.*, 113:287–297, 2008.

[23] L. Gargano, J. Körner, and U. Vaccaro. Sperner theorems on directed graphs and qualitative independence. *J. Combinat. Theory (A)*, 61:173–192, 1992.

[24] L. Gargano, J. Körner, and U. Vaccaro. Sperner capacities. *Graphs and Combinatorics*, 9:31–46, 1993.

[25] B. J. Garvin, M. B. Cohen, and M. B. Dwyer. Evaluating improvements to a meta-heuristic search for constrained interaction testing. *Empirical Software Engineering*, 16:61–102, 2011.

[26] A. P. Godbole, D. E. Skipper, and R. A. Sunley. $t$-covering arrays: upper bounds and Poisson approximations. *Combinatorics, Probability and Computing*, 5:105–118, 1996.

[27] L. Gonzalez-Hernandez, J. Torres-Jimenez, and N. Rangel-Valdez. An exact approach to maximize the number of wild cards in a covering array. In *Proceedings of 10th Mexican International Conference on Artificial Intelligence, MICAI 2011*, 2011.

[28] A. Hartman. Software and hardware testing using combinatorial covering suites. In M. C. Golumbic and I. B.-A. Hartman, editors, *Interdisciplinary Applications of Graph Theory, Combinatorics, and Algorithms*, pages 237–266. Springer, Norwell, MA, 2005.

[29] A. Hartman and L. Raskin. Problems and algorithms for covering arrays. *Discrete Math.*, 284:149–156, 2004.

[30] A. S. Hedayat, N. J. A. Sloane, and J. Stufken. *Orthogonal Arrays*. Springer-Verlag, New York, 1999.

[31] B. Hnich, S. Prestwich, E. Selensky, and B. M. Smith. Constraint models for the covering test problem. *Constraints*, 11:199–219, 2006.

[32] G. O. H. Katona. Two applications (for search theory and truth functions) of Sperner type theorems. *Periodica Math.*, 3:19–26, 1973.

[33] D. Kleitman and J. Spencer. Families of k-independent sets. *Discrete Math.*, 6:255–262, 1973.

[34] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence. IPOG/IPOD: Efficient test generation for multi-way software testing. *Software Testing, Verification, and Reliability*, 18:125–148, 2008.

[35] J. R. Lobb, C. J. Colbourn, P. Danziger, B. Stevens, and J. Torres-Jimenez. Cover starters for strength two covering arrays. *Discrete Mathematics*, 312:943–956, 2012.

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

[37] P. Nayeri, C. J. Colbourn, and G. Konjevod. Randomized postoptimization of covering arrays. *European Journal of Combinatorics*, 34:91–103, 2013.

[38] C. Nie and H. Leung. A survey of combinatorial testing. *ACM Computing Surveys*, 43(2):#11, 2011.

[39] K. Nurmela. Upper bounds for covering arrays by tabu search. *Discrete Applied Mathematics*, 138:143–152, 2004.

[40] S. Poljak and Z. Tuza. On the maximum number of qualitatively independent partitions. *J. Combinat. Theory (A)*, 51:111–116, 1989.

[41] A. Réyni. *Foundations of Probability*. Wiley, New York, 1971.

[42] G. Roux. *$k$-Propriétés dans les tableaux de $n$ colonnes: cas particulier de la $k$-surjectivité et de la $k$-permutivité*. PhD thesis, Université de Paris, 1987.

[43] N. J. A. Sloane. Covering arrays and intersecting codes. *J. Combin. Des.*, 1:51–63, 1993.

[44] J. Stardom. Metaheuristics and the search for covering and packing arrays. Master's thesis, Simon Fraser University, 2001.

[45] B. Stevens. *Transversal Covers and Packings*. PhD thesis, Mathematics, University of Toronto, 1998.

[46] B. Stevens, A. C. H. Ling, and E. Mendelsohn. A direct construction of transversal covers using group divisible designs. *Ars Combin.*, 63:145–159, 2002.

[47] B. Stevens and E. Mendelsohn. New recursive methods for transversal covers. *J. Combin. Des.*, 7:185–203, 1999.

[48] K. C. Tai and L. Yu. A test generation strategy for pairwise testing. *IEEE Transactions on Software Engineering*, 28:109–111, 2002.

[49] J. Torres-Jimenez. Covering array tables, 2010-2013. http://www.tamps.cinvestav.mx/∼jtj/.

[50] J. Torres-Jimenez and E. Rodriguez-Tello. New upper bounds for binary covering arrays using simulated annealing. *Information Sciences*, 185(1):137–152, 2012.

[51] Y. W. Tung and W. S. Aldiwan. Automating test case generation for the new generation mission software system. In *Proc. 30th IEEE Aerospace Conference*, pages 431–437, Los Alamitos, CA, 2000. IEEE.

[52] J. Yan and J. Zhang. A backtracking search tool for constructing combinatorial test suites. *J. Systems Software*, 81:1681–1693, 2008.

[53] C. Yilmaz, M. B. Cohen, and A. Porter. Covering arrays for efficient fault characterization in complex configuration spaces. *IEEE Transactions on Software Engineering*, 31:20–34, 2006.

[54] L. Zekaoui. Mixed covering arrays on graphs and tabu search algorithms. Master's thesis, University of Ottawa, 2006.