Covering Arrays of Higher Strength From Permutation Vectors

George B. Sherwood
Testcover.com LLC
41 Clover Hill Road
Colts Neck, NJ 07722
gsherwood@att.net

Sosina S. Martirosyan and Charles J. Colbourn
Dept. of Computer Science and Engineering
Arizona State University
P.O. Box 878809, Tempe, AZ 85287
{sosina.martirosyan,charles.colbourn}@asu.edu

Abstract

A *covering array* $CA(N; t, k, v)$ is an $N \times k$ array such that every $N \times t$ sub-array contains all t-tuples from v symbols *at least* once, where t is the *strength* of the array. Covering arrays are used to generate software test suites to cover all t-sets of component interactions. We introduce a combinatorial technique for their construction, focussing on covering arrays of strength 3 and 4. With a computer search, covering arrays with improved parameters have been found.

1 Introduction

Covering arrays have attracted much attention during recent years. Their study is motivated by their application to software and hardware testing. These and other applications of covering arrays are discussed in [5, 6, 9]. A wide range of mathematics and computation has been used to construct covering arrays. The main problem is to minimize the number of tests N for given number of factors k, number of levels v per factor, and strength t of coverage; or equivalently to maximize k for given values of t, v, and N. Most known results are for strength two or three covering arrays. Constructions for strength two and three covering arrays can be found in [2, 4, 3, 7, 9, 15], for example. However very little is known about covering arrays with larger strength; for an arbitrary value t see [8, 13], for example. Recent surveys appear in [6, 9].

An *orthogonal array* $OA_\lambda(t, k, v)$ is an $\lambda v^t \times k$ array on v symbols such that every $\lambda v^t \times t$ sub-array contains each ordered subset of size t from v symbols *exactly* λ times. When $\lambda = 1$ we omit the subscript. Covering arrays are generalizations of orthogonal arrays.

A *covering array* $CA(N; t, k, v)$ is an $N \times k$ array such that every $N \times t$ sub-array contains all tuples from v symbols of size t *at least* one time each. When N is unknown or unspecified, the notation $CA(t, k, v)$ is also used. The *covering array number* $CAN(t, k, v)$ is the minimum number N of rows required to produce a $CA(N; t, k, v)$. In a covering array $CA(t, k, v)$, t is the *strength*, k the *degree*, and v the *order*. We consider the case where the symbols are taken from the field $\mathbf{GF(v)}$ of order v.

The covering arrays to be constructed consist of columns formed by concatenating permutations of the elements of $\mathbf{GF(v)}$. A permutation vector of length v^t is defined as follows. Let $(\beta_0^{(i)}, \beta_1^{(i)}, \cdots, \beta_{t-1}^{(i)})$ be the base v representation of the symbol $i \in \{0, 1, \cdots, v^t - 1\}$; that is $i = \beta_0^{(i)} + v^1\beta_1^{(i)} + \cdots + v^{t-1}\beta_{t-1}^{(i)}$ where $\beta_j^{(i)} \in \{0, 1, \cdots, v-1\}$ for $0 \le j \le t-1$. For each $(t-1)$-tuple $(h_1, h_2, \cdots, h_{t-1})$ with $h_j \in \{0, 1, \cdots, v-1\}$ for $1 \le j \le t-1$, a *permutation vector* $(\overrightarrow{h_1, h_2, \cdots, h_{t-1}})$

of length v^t is the vector that has the symbol $\beta_0^{(i)} + (h_1 \times \beta_1^{(i)}) + (h_2 \times \beta_2^{(i)}) + \cdots + (h_{t-1} \times \beta_{t-1}^{(i)})$ in position i for $0 \leq i \leq v^t - 1$. Here addition and multiplication are in $\mathbf{GF(v)}$.

For each $(t-1)$-tuple $(h_1, h_2, \cdots, h_{t-1})$ with $h_j \in \{0, 1, \cdots, v-1\}$ for $1 \leq j \leq t-1$, a *reduced permutation vector* $\overrightarrow{(h_1, h_2, \cdots, h_{t-1})}^-$ of length v^{t-1} is the vector that has the symbol $(h_1 \times \beta_0^{(i)}) + (h_2 \times \beta_1^{(i)}) + \cdots + (h_{t-1} \times \beta_{t-2}^{(i)})$ in position i for $0 \leq i \leq v^{t-1} - 1$ and $i = \beta_0^{(i)} + v^1 \beta_1^{(i)} + \cdots + v^{t-2} \beta_{t-2}^{(i)}$.

A set of t different permutation vectors of length v^t is *covering* if the $v^t \times t$ array that has these vectors as columns is a $CA(v^t; t, t, v)$, so that all possible t tuples occur exactly once as a row. A set is *noncovering* otherwise; such a set necessarily covers some t-tuple more than once, and some other t-tuple not at all.

Consider t permutation vectors $\overrightarrow{(h_1^{(1)}, h_2^{(1)}, \cdots, h_{t-1}^{(1)})}$, $\overrightarrow{(h_1^{(2)}, h_2^{(2)}, \cdots h_{t-1}^{(2)})}$, $\cdots$, $\overrightarrow{(h_1^{(t)}, h_2^{(t)}, \cdots, h_{t-1}^{(t)})}$. To check that this set of permutation vectors is noncovering, we check that the array that has these vectors as columns contains some t-tuple twice as a row. In other words, the set of permutation vectors is noncovering if and only if there exist distinct positions $i, j \in \{0, 1, \cdots, v^t - 1\}$ such that:

$$\begin{array}{c}
\beta_0^{(i)} + (h_1^{(1)} \times \beta_1^{(i)}) + (h_2^{(1)} \times \beta_2^{(i)}) + \cdots + (h_{t-1}^{(1)} \times \beta_{t-1}^{(i)}) = \beta_0^{(j)} + (h_1^{(1)} \times \beta_1^{(j)}) + (h_2^{(1)} \times \beta_2^{(j)}) + \cdots + (h_{t-1}^{(1)} \times \beta_{t-1}^{(j)}) \\
\beta_0^{(i)} + (h_1^{(2)} \times \beta_1^{(i)}) + (h_2^{(2)} \times \beta_2^{(i)}) + \cdots + (h_{t-1}^{(2)} \times \beta_{t-1}^{(i)}) = \beta_0^{(j)} + (h_1^{(2)} \times \beta_1^{(j)}) + (h_2^{(2)} \times \beta_2^{(j)}) + \cdots + (h_{t-1}^{(2)} \times \beta_{t-1}^{(j)}) \\
\vdots \\
\beta_0^{(i)} + (h_1^{(t)} \times \beta_1^{(i)}) + (h_2^{(t)} \times \beta_2^{(i)}) + \cdots + (h_{t-1}^{(t)} \times \beta_{t-1}^{(i)}) = \beta_0^{(j)} + (h_1^{(t)} \times \beta_1^{(j)}) + (h_2^{(t)} \times \beta_2^{(j)}) + \cdots + (h_{t-1}^{(t)} \times \beta_{t-1}^{(j)})
\end{array} \tag{1}$$

Let $\alpha_r = \beta_r^{(i)} - \beta_r^{(j)}$ for $0 \leq r \leq t-1$ with the fixed choices of i and j. Rewrite (1) as:

$$\begin{array}{c}
\alpha_0 + (h_1^{(1)} \times \alpha_1) + (h_2^{(1)} \times \alpha_2) + \cdots + (h_{t-1}^{(1)} \times \alpha_{t-1}) = 0 \\
\alpha_0 + (h_1^{(2)} \times \alpha_1) + (h_2^{(2)} \times \alpha_2) + \cdots + (h_{t-1}^{(2)} \times \alpha_{t-1}) = 0 \\
\vdots \\
\alpha_0 + (h_1^{(t)} \times \alpha_1) + (h_2^{(t)} \times \alpha_2) + \cdots + (h_{t-1}^{(t)} \times \alpha_{t-1}) = 0
\end{array} \tag{2}$$

The set of permutation vectors is noncovering if and only if the system of linear equations (2) with unknowns $\{\alpha_r : 0 \leq r \leq t-1\}$ has a nonzero solution over $\mathbf{GF(v)}$.

Bush's well known orthogonal arrays (see [11]) can be described using permutation vectors. Consider the v permutation vectors $\{\overrightarrow{(h, h^2, \cdots, h^{t-1})} : h \in GF(v)\}$ and the reduced permutation vector of length v^t $\overrightarrow{(0, 0, \cdots, 0, 1)}^-$. Let these be the columns of a $v^t \times (v+1)$ array C. That C is an $OA(t, v+1, v)$ follows from the proof of Theorem 3.2 in [9]. Similarly, if $\ell \geq 2$, then the 2^ℓ permutation vectors $\overrightarrow{(h, h^2)}$ of length $(2^\ell)^3$ for every $h \in GF(2^\ell)$ and the reduced permutation vectors $\overrightarrow{(0, 1, 0)}^-$ and $\overrightarrow{(0, 0, 1)}^-$ form an $OA(3, 2^\ell + 2, 2^\ell)$.

2 Covering Perfect Hash Families

Every permutation vector has the same v tuple $(0, 1, ...v-1)$ in the first v positions. Hence we employ a *shortened permutation vector* consisting of the last $v^t - v$ rows. The prototype construction proceeds as follows.

Construction 2.1 *Choose two numbers n and k, and form an $n \times k$ array F in which the entries are integers chosen from $\{0, 1, \ldots, v^{t-1} - 1\}$. Replace each entry e in F by the column that is the* *shortened permutation vector indexed by the base v representation of e. Then append v constant rows, one for each symbol of* $\mathbf{GF(v)}$. *The result is an $(n \cdot (v^t - v) + v) \times k$ array.*

When does Construction 2.1 produce a covering array? To ensure that it does, we require that F meet certain requirements that we determine next.

Perfect hash families are well studied combinatorial objects. A *t-perfect hash family $\mathcal{H}$*, denoted $\mathsf{PHF}(n; k, q, t)$, is a family of n functions $h : A \mapsto B$, where $|A| \geq |B| = q$, such that for any subset $X \subseteq A$ with $|X| = t$, there is at least one function $h \in \mathcal{H}$ that is injective on X. Thus a $\mathsf{PHF}(n; k, q, t)$ can be viewed as an $n \times k$-array $\mathcal{H}$ with entries from a set of q symbols such that for any set of t columns there is at least one row having distinct entries in this set of columns. We require initially that F be a perfect hash family so that, when entries of F are replaced by permutation vectors, these vectors are distinct. We require more to ensure coverage of t-tuples.

For $i \in \{0, 1, \cdots, v^{t-1} - 1\}$ the corresponding permutation vector of length v^t is $(\overrightarrow{h_1, h_2, \cdots, h_{t-1}})$ when $i = v^0 h_1 + v^1 h_2 + \cdots + v^{t-2} h_{t-1}$. A t-tuple with symbols from the set $\{0, 1, \cdots, v^{t-1} - 1\}$ is a *covering tuple* if the set of permutation vectors corresponding to its symbols is covering. Alternatively a t-tuple is a *noncovering tuple* if the set of permutation vectors corresponding to its symbols is noncovering.

To produce a covering array, we require that F be a $\mathsf{PHF}(n; k, q, t)$ in which, for any set of t columns, there exists at least one row having a covering tuple in this set of columns. Such a *covering perfect hash family* is denoted by $\mathsf{CPHF}(n; k, v^{t-1}, t)$. Replacing the symbols of a $\mathsf{CPHF}(n; k, v^{t-1}, t)$ by the corresponding permutation vectors yields a $\mathsf{CA}(n \cdot v^t; t, k, v)$. Using Construction 2.1, we obtain a $\mathsf{CA}(n \cdot (v^t - v) + v; t, k, v)$.

Hence we have:

Theorem 2.2 *If v is a prime or prime power, and a $\mathsf{CPHF}(n; k, v^{t-1}, t)$ exists, then a $\mathsf{CA}(n \cdot (v^t - v) + v; t, k, v)$ exists.*

We give a simple product type construction of CPHF:

Theorem 2.3 *Suppose that a $\mathsf{CPHF}(n; q, v^{t-1}, t)$ and a $\mathsf{PHF}(n'; k, q, t)$ both exist. Then there is a $\mathsf{CPHF}(nn'; k, v^{t-1}, t)$.*

Proof. Suppose the array A is a $\mathsf{CPHF}(n; q, v^{t-1}, t)$ and the array B is a $\mathsf{PHF}(n'; k, q, t)$ with symbols from the set $\{1, 2, \cdots, q\}$. Replace the symbol i in B by the i-th column of A for $1 \leq i \leq q$. The resulting array is a $\mathsf{CPHF}(nn'; k, v^{t-1}, t)$. ■

3 Distinct Noncovering Tuples

The search technique that we used to find a covering perfect hash family computes and stores symbols for noncovering tuples. To determine if they cover, tuples from the candidate CPHF are compared with the stored noncovering tuples. Because any tuple with repeated symbols is noncovering, we can restrict our attention to *distinct noncovering tuples* having distinct symbols from the set $\{0, 1, \cdots, v^{t-1} - 1\}$. The distinct noncovering tuples can be computed from the system (2) as follows. Here we write the base v representation of a t-tuple as

$$((h_1^{(1)}, h_2^{(1)}, \cdots, h_{t-1}^{(1)}), (h_1^{(2)}, h_2^{(2)}, \cdots, h_{t-1}^{(2)}), \cdots, (h_1^{(t)}, h_2^{(t)}, \cdots, h_{t-1}^{(t)})).$$

When $t = 2$ the system of equations (2) cannot have a nonzero solution for any choice of two distinct coefficients. Thus any distinct 2-tuple (and its corresponding permutation vectors) are covering for strength two. Now consider the system of linear equations (2) for $t = 3$.

$$\begin{array}{l}\alpha_0 + (h_1^{(1)} \times \alpha_1) + (h_2^{(1)} \times \alpha_2) = 0\\ \alpha_0 + (h_1^{(2)} \times \alpha_1) + (h_2^{(2)} \times \alpha_2) = 0\\ \alpha_0 + (h_1^{(3)} \times \alpha_1) + (h_2^{(3)} \times \alpha_2) = 0\end{array} \tag{3}$$

We compute all distinct noncovering tuples, partitioned into two cases:

1. ($\alpha_2 = 0$) The system of equations (3) has a nonzero solution such that $\alpha_2 = 0$ if and only if $h_1^{(1)} = h_1^{(2)} = h_1^{(3)}$. Thus for a fixed $\gamma_0 \in \mathbf{GF(v)}$ any tuple having the symbol γ_0 in all first positions is noncovering. A 3-tuple of the form $((\gamma_0, h_2^{(1)}), (\gamma_0, h_2^{(2)}), (\gamma_0, h_2^{(3)}))$ is a distinct noncovering tuple when $h_2^{(1)} \neq h_2^{(2)} \neq h_2^{(3)}$.

2. ($\alpha_2 \neq 0$) The system of equations (3) has a nonzero solution such that $\alpha_2 \neq 0$ if

$$\begin{array}{l}h_2^{(1)} = \gamma_1 \times h_1^{(1)} + \gamma_0\\ h_2^{(2)} = \gamma_1 \times h_1^{(2)} + \gamma_0\\ h_2^{(3)} = \gamma_1 \times h_1^{(3)} + \gamma_0\end{array} \tag{4}$$

   for some γ_0, γ_1 in $\mathbf{GF(v)}$.

   In this case, distinct noncovering tuples can be described by considering the v^2 ordered pairs (γ_0, γ_1) from $\mathbf{GF(v)}$. For each ordered pair, select distinct $h_1^{(1)} \neq h_1^{(2)} \neq h_1^{(3)}$ and compute $h_2^{(1)}, h_2^{(2)}, h_2^{(3)}$ from equations (4). The same tuple cannot occur for two distinct ordered pairs (γ_0, γ_1).

These two cases include all distinct noncovering tuples exactly once. For any fixed element γ_0 in $\mathbf{GF(v)}$ the first case gives $\binom{v}{3}$ distinct noncovering tuples. For each ordered pair (γ_0, γ_1) the second case gives $\binom{v}{3}$ distinct noncovering tuples. Hence for strength three, we get $\binom{v}{3} \times (v^2 + v)$ distinct noncovering tuples. In $\mathbf{GF(2)}$ there are not enough symbols for three choices of $h_j^{(1)}, h_j^{(2)}$and $h_j^{(3)}$, and hence any tuple with distinct symbols is covering.

In the same way we can tabulate all distinct noncovering tuples for strength four.

$$\begin{array}{l}\alpha_0 + (h_1^{(1)} \times \alpha_1) + (h_2^{(1)} \times \alpha_2) + (h_3^{(1)} \times \alpha_3) = 0\\ \alpha_0 + (h_1^{(2)} \times \alpha_1) + (h_2^{(2)} \times \alpha_2) + (h_3^{(2)} \times \alpha_3) = 0\\ \alpha_0 + (h_1^{(3)} \times \alpha_1) + (h_2^{(3)} \times \alpha_2) + (h_3^{(3)} \times \alpha_3) = 0\\ \alpha_0 + (h_1^{(4)} \times \alpha_1) + (h_2^{(4)} \times \alpha_2) + (h_3^{(4)} \times \alpha_3) = 0\end{array} \tag{5}$$

Consider two main cases:

1. ($\alpha_3 = 0$) We are looking for distinct noncovering tuples for which (5) has nonzero solutions with $\alpha_3 = 0$.

   (a) ($\alpha_2 = 0$) In this case, $h_1^{(1)} = h_1^{(2)} = h_1^{(3)} = h_1^{(4)}$. Thus for a fixed $\gamma_0 \in \mathbf{GF(v)}$ any tuple having the symbol γ_0 in all first positions is noncovering. 4-tuples of the form

$$((\gamma_0, h_2^{(1)}, h_3^{(1)}), (\gamma_0, h_2^{(2)}, h_3^{(2)}), (\gamma_0, h_2^{(3)}, h_3^{(3)}), (\gamma_0, h_2^{(4)}, h_3^{(4)}))$$

are distinct noncovering tuples if the ordered pairs $(h_2^{(1)}, h_3^{(1)}), (h_2^{(2)}, h_3^{(2)}), (h_2^{(3)}, h_3^{(3)})$ and $(h_2^{(4)}, h_3^{(4)}))$ are distinct. In this case we obtain $\binom{v^2}{4} \times v$ distinct noncovering tuples.

(b) $(\alpha_2 \neq 0)$ For any fixed ordered pair (γ_0, γ_1) from $\mathbf{GF(v)}$ consider a solution such that

$$\begin{aligned} h_2^{(1)} &= \gamma_1 \times h_1^{(1)} + \gamma_0 \\ h_2^{(2)} &= \gamma_1 \times h_1^{(2)} + \gamma_0 \\ h_2^{(3)} &= \gamma_1 \times h_1^{(3)} + \gamma_0 \\ h_2^{(4)} &= \gamma_1 \times h_1^{(4)} + \gamma_0 \end{aligned} \tag{6}$$

The resulting tuple is distinct and noncovering when the four choices for the first and third positions are distinct. For each fixed pair (γ_0, γ_1), select any four distinct ordered pairs $(h_1^{(1)}, h_3^{(1)}), (h_1^{(2)}, h_3^{(2)}), (h_1^{(3)}, h_3^{(3)})$ and $(h_2^{(4)}, h_3^{(4)}))$ and compute $h_2^{(1)}, h_2^{(2)}, h_2^{(3)}$ and $h_2^{(4)}$ from equations (6). We obtain $\binom{v^2}{4}$ distinct noncovering tuples for each ordered pair. Hence there are a total of $\binom{v^2}{4} \times v^2$ distinct noncovering tuples in this case.

2. $(\alpha_3 \neq 0)$ We are looking for the distinct noncovering tuples for which (5) has nonzero solutions with $\alpha_3 \neq 0$. Nonzero solutions arise with coefficients for which

$$\begin{aligned} h_3^{(1)} &= \gamma_2 \times h_2^{(1)} + \gamma_1 \times h_1^{(1)} + \gamma_0 \\ h_3^{(2)} &= \gamma_2 \times h_2^{(2)} + \gamma_1 \times h_1^{(2)} + \gamma_0 \\ h_3^{(3)} &= \gamma_2 \times h_2^{(3)} + \gamma_1 \times h_1^{(3)} + \gamma_0 \\ h_3^{(4)} &= \gamma_2 \times h_2^{(4)} + \gamma_1 \times h_1^{(4)} + \gamma_0 \end{aligned} \tag{7}$$

for any γ_0, γ_1 and γ_2 in $\mathbf{GF(v)}$. In this case, all distinct noncovering tuples can be described by considering all v^3 ordered triples $(\gamma_0, \gamma_1, \gamma_2)$ from $\mathbf{GF(v)}$. For each fixed triple, select any four distinct ordered pairs $(h_1^{(1)}, h_2^{(1)})$, $(h_1^{(2)}, h_2^{(2)})$,$(h_1^{(3)}, h_2^{(3)})$ and $(h_1^{(4)}, h_2^{(4)})$ and compute $h_3^{(1)}, h_3^{(2)}, h_3^{(3)}$ and $h_3^{(4)}$ from equations (7). For any fixed triple $(\gamma_0, \gamma_1, \gamma_2)$ we obtain $\binom{v^2}{4}$ distinct noncovering tuples. Thus, in this case, there are $\binom{v^2}{4} \times v^3$ distinct noncovering tuples in total.

These two cases specify all distinct noncovering tuples for strength four. Some noncovering tuples are included more than once when $v \geq 4$. Nevertheless the distinct noncovering tuples, of which there are at most $\binom{v^2}{4} \times (v^3 + v^2 + v)$, can be calculated and stored. When $v = 3$ there are 4914 distinct noncovering tuples and 12636 covering tuples.

For strength t in general, consider the system of linear equations (2). We describe a general method that gives an upper bound on number of distinct noncovering tuples of permutation vectors. There are $v(v^{t-1} - 1)$ choices for the t-tuple $(\alpha_0, \alpha_1, \cdots, \alpha_{t-1})$ so that $\alpha_i \in \mathbf{GF(v)}$ for $0 \leq i \leq t-1$, and at least one α_i is nonzero for some $1 \leq i \leq t-1$. For each such tuple consider the equation

$$\alpha_0 + (h_1 \times \alpha_1) + (h_2 \times \alpha_2) + \cdots + (h_{t-1} \times \alpha_{t-1}) = 0$$

with unknowns $h_1, h_2, \cdots, h_{t-1} \in \mathbf{GF(v)}$. The equation has v^{t-2} solutions over $\mathbf{GF(v)}$ since at least one α_i is nonzero for some $1 \leq i \leq t-1$; hence the corresponding h_i is uniquely determined once values have been assigned to the other indeterminates in an arbitrary manner (see [12], for example). Consequently, for each such fixed t-tuple there are $\binom{v^{t-2}}{t}$ noncovering sets of permutation vectors. Now the t-tuple $(\alpha_0, \alpha_1, \cdots, \alpha_{t-1})$ and the t-tuple $(\beta \times \alpha_0, \beta \times \alpha_1, \cdots, \beta \times \alpha_{t-1})$ where β is any nonzero element of $\mathbf{GF(v)}$, have the same set of solutions. We conclude that there are at most $\frac{v(v^{t-1}-1)}{v-1}\binom{v^{t-2}}{t}$ distinct noncovering tuples of permutation vectors. The noncovering tuples can be determined by considering each of the $(v^{t-1} + v^{t-2} + \cdots + v)$ t-tuples.

4 Strength Three

To find a covering perfect hash family, we need to test whether any particular tuple is covering. Using the symbols from $\{0, 1, \cdots, v^2 - 1\}$ we tabulate the $\binom{v}{3} \times (v^2 + v)$ noncovering 3-tuples as $(v^2 + v)$ sets of v symbols. In each set, any subset of three elements comprises a noncovering tuple. Then a tuple is covering if it contains distinct symbols and is not contained in any of the noncovering symbol sets.

Let us apply the general machinery developed to determine the noncovering sets of permutation vectors for strength three. Consider three permutation vectors (a_1, b_1), (a_2, b_2) and (a_3, b_3). If $a_1 = a_2 = a_3$, choose $\alpha_2 = 0$; choose α_1 to be any nonzero element and $\alpha_0 = -(\alpha_1 a_1)$. Then these three permutation vectors are noncovering. Similarly three vectors with $b_1 = b_2 = b_3$ are noncovering. Finally if $(a_1, b_1) = (c, d)$, $(a_2, b_2) = ((c+a), (d+b))$ and $(a_3, b_3) = ((c+\gamma a), (d+\gamma b))$ for nonzero γ, the three vectors are noncovering. All other sets of three permutation vectors are covering.

For strength $t = 3$ the following CPHF arrays have been found by computer search. Here the elements are given as base v strings of symbols $h_{t-1} \cdots h_2 h_1$ over alphabet $\{0, 1, \cdots, v-1\}$.

```
CPHF(3;20,9,3)
00 01 02 10 11 12 20 21 22 00 01 02 10 11 12 20 21 22 00 01
00 01 02 10 11 22 20 21 12 22 20 21 12 10 01 02 00 11 20 00
00 01 10 02 11 21 22 12 20 10 11 01 00 12 20 02 22 21 20 21

CPHF(2;16,16,3)
00 01 02 03 10 11 12 13 20 21 22 23 30 31 32 33
00 01 10 11 02 03 12 13 21 20 31 30 23 22 33 32

CPHF(3;28,16,3)
00 01 02 03 10 11 12 13 20 21 22 23 30 31 32 33 00 01 02 03 10 11 12 13 20 21 22 23
00 01 02 03 10 11 12 13 20 21 22 23 30 31 33 32 11 10 22 23 01 00 20 21 32 33 02 03
00 01 10 11 02 03 12 13 21 20 31 30 23 22 32 33 11 10 01 00 31 30 13 12 22 23 02 03

CPHF(2;24,25,3)
00 01 02 03 04 10 11 12 13 14 20 21 22 23 24 30 31 32 33 34 40 41 42 43
00 01 10 11 23 04 42 03 13 24 14 02 31 41 43 22 30 21 40 32 12 44 33 34

CPHF(2;31,49,3)
00 01 02 03 04 05 06 10 11 12 13 14 15 16 20 21 22 23 24 25 26
00 01 10 11 23 25 63 02 03 12 13 20 60 65 22 43 46 62 55 16 53
```

```
30 31 32 33 34 35 36 40 41 42
50 52 61 42 15 31 40 34 35 41

CPHF(2;40,64,3)
00 01 02 03 04 05 06 07 10 11 12 13 14 15 16 17 20 21 22 23 24
00 01 10 11 24 25 34 35 02 03 12 13 26 27 36 37 21 20 31 30 05

25 26 27 30 31 32 33 34 35 36 37 40 41 42 43 44 45 46 47
04 15 14 23 22 33 32 07 06 17 16 51 50 41 40 75 74 65 64

CPHF(2;39,81,3)
00 01 02 03 04 05 06 07 08 10 11 12 13 14 15 16 17 18 20 21 22
00 01 10 11 24 26 53 57 83 02 03 12 13 20 25 46 48 66 14 15 71

23 24 25 26 27 28 30 31 32 33 34 35 36 37 38 40 41 42
64 30 80 76 85 68 74 67 08 51 41 47 78 62 05 21 52 84

CPHF(2;44,121,3)
00 01 02 03 04 05 06 07 08 09 0A 10 11 12 13 14 15 16 17 18 19 1A
00 01 10 11 24 28 43 49 83 89 A4 02 03 12 13 26 2A 40 45 80 85 A6

20 21 22 23 24 25 26 27 28 29 2A 30 31 32 33 34 35 36 37 38 39 3A
14 21 20 04 0A A7 70 77 19 51 58 50 64 35 86 96 A0 A5 34 99 68 88

CPHF(2;39,169,3)
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 10 11 12 13 14 15 16 17 18
00 01 10 11 24 2A 45 49 57 97 A5 A9 C4 02 03 12 13 26 2C 47 4B 59

19 1A 1B 1C 20 21 22 23 24 25 26 27 28 29 2A 2B 2C
99 A7 AB C6 14 23 20 06 0A 1C 73 84 70 8C B2 B1 9A
```

These produce the following covering arrays:

CA$(75; 3, 20, 3)$ CA$(124; 3, 16, 4)$ CA$(184; 3, 28, 4)$ CA$(245; 3, 24, 5)$
CA$(679; 3, 31, 7)$ CA$(1016; 3, 40, 8)$ CA$(1449; 3, 39, 9)$ CA$(2651; 3, 44, 11)$
CA$(4381; 3, 39, 13)$

These new arrays compare favourably with known results. Below is the list of covering arrays given in [3, Table V].

CA$(83; 3, 20, 3)$ CA$(159; 3, 16, 4)$ CA$(232; 3, 28, 4)$ CA$(385; 3, 24, 5)$
CA$(1029; 3, 30, 7)$ CA$(1536; 3, 30, 8)$ CA$(2187; 3, 30, 9)$ CA$(3993; 3, 30, 11)$
CA$(6591; 3, 30, 13)$

Compare also with arrays given in [14] as follows:

CA(700;3,18,6) CA$(725; 3, 20, 6)$ CA$(878; 3, 20, 7)$ CA$(1506; 3, 22, 8)$
$CA(2171; 3, 24, 9)$ CA$(3635; 3, 30, 10)$ CA$(3907; 3, 28, 11)$

The covering arrays from permutation vectors have fewer tests than the covering arrays from [3] and [14] listed above. In fact, in many cases fewer tests are required for more factors. Finally we compare with results in [4].

$\mathsf{CA}(50;3,9,3)$ $\mathsf{CA}(180;3,17,4)$ $\mathsf{CA}(188;3,25,4)$ $\mathsf{CA}(203;3,30,4)$
$CA(371;3,25,5)$ $\mathsf{CA}(692;3,24,6)$

These improve upon the results in [3]. However, they are larger than covering arrays from permutation vectors, or do not compare with them. For example, from $\mathsf{CA}(679;3,31,7)$ by symbol-fusing we obtain a $\mathsf{CA}(678;3,31,6)$ which improves upon $\mathsf{CA}(692;3,24,6)$ constructed in [4]. Another example is $\mathsf{CA}(184;3,28,4)$ which requires fewer tests for more factors than the $\mathsf{CA}(188;3,25,4)$.

The new arrays compare favourably also with the arrays found by computational methods in [4, 10]. These comparisons demonstrate that constructions based on permutation vectors afford some substantial reductions in numbers of tests in many cases. The new arrays also provide useful ingredients for "Roux-type" recursive constructions as in [3, 4, 13].

5 Strength Four

For strength $t = 4$, the same machinery can be used to produce covering perfect hash families as follows:

```
CPHF(2;8,8,4)
000 001 010 011 100 101 110 111
000 001 010 100 011 110 111 101

CPHF(4;11,8,4)
000 001 010 011 100 101 110 111 000 001 010
000 001 010 011 100 101 111 110 101 111 001
000 001 010 011 100 110 111 101 111 010 101
000 001 010 100 011 111 101 110 011 100 000

CPHF(2;10,27,4)
000 001 002 010 011 100 101 110 120 121
000 001 010 100 111 101 211 122 212 112

CPHF(3;16,27,4)
000 001 002 010 011 012 020 021 100 101 102 110 111 112 211 222
000 001 002 010 011 100 101 110 012 020 221 112 220 022 111 122
000 001 010 100 111 101 211 120 200 121 221 201 220 210 110 102

CPHF(2;9,64,4)
000 001 002 003 010 100 110 121 131
000 001 010 100 112 113 121 211 222

CPHF(2;11,125,4)
000 001 002 003 004 010 100 110 121 212 231
000 001 010 100 111 124 142 214 241 412 421
```

These produce the following covering arrays:

$$\mathsf{CA}(30;4,8,2) \quad \mathsf{CA}(58;4,11,2) \quad \mathsf{CA}(159;4,10,3) \quad \mathsf{CA}(237;4,16,3)$$
$$\mathsf{CA}(508;4,9,4) \quad \mathsf{CA}(1245;4,11,5)$$

We now recall the known "Roux type" constructions for strength 4 covering arrays.

Theorem 5.1 *[9, Theorem 7.8] For v a positive integer,*

$$CAN(4,2k,v) \leq CAN(4,k,v) + (v-1)CAN(3,k,v) + CAN(2,k,v^2)$$

Theorem 5.2 *[13, Corollary 4.4] For $v \geq 2$ a prime power, we have*

$$CAN(4,2k,v) \leq CAN(4,k,v) + (v-1)CAN(3,k,v) + 2v^2 CAN(2,k,v)$$

Theorem 5.3 *[13, Theorem 4.13] For all positive integer v*

$$CAN(4,2k,v) \leq CAN(4,k,v) + (v-1)CAN(3,k,v) + (CAN(2,k,v))^2$$

Other than asymptotic constructions, these are to the best of our knowledge the only known constructions for strength 4. The new arrays have fewer rows (tests) than those derived from Theorems 5.1 - 5.3. For example, a $\mathsf{CA}(40;4,8,2)$ can be obtained from Theorem 5.1 when the component arrays are $\mathsf{CA}(16;2,4,4)$, $\mathsf{CA}(16;4,4,2)$ and $\mathsf{CA}(8;3,4,2)$. A $\mathsf{CA}(37;4,8,2)$ is constructed in [13]. These are larger than the $\mathsf{CA}(30;4,8,2)$ obtained from permutation vectors. One of the disadvantages of "Roux-type" constructions is that ingredient arrays with "good" parameters are often not known. This makes a fair comparison difficult. However, even if we employ *lower* bounds on covering arrays numbers for ingredient arrays in the Roux-type theorems, we still produce more tests than by using permutation vectors.

For example, suppose that the unknown $\mathsf{CA}(81;4,5,3)$ and $\mathsf{CA}(28;3,5,3)$ both exist. Using Theorem 5.1 with optimal component arrays $\mathsf{CA}(81;4,5,3)$, $\mathsf{CA}(28;3,5,3)$ and $\mathsf{CA}(81;2,5,9)$ we would obtain a $\mathsf{CA}(218;4,10,3)$ rather than the $\mathsf{CA}(159;4,10,3)$ obtained from permutation vectors. In a similar way we can compare the other 4-covering arrays from permutation vectors with those that might arise from the Roux-type constructions. Again we find that arrays from permutation vectors have fewer tests.

Indeed they also compare favourably with the arrays found by computational methods [10] listed below.

$$\mathsf{CA}(42;4,8,2) \quad \mathsf{CA}(309;4,10,3) \quad \mathsf{CA}(508;4,9,4) \quad \mathsf{CA}(1725;4,10,5)$$

6 The Computational Search

The search technique that we used to find covering perfect hash families employs a particular tabulation of the noncovering tuples and restricts the search space to improve efficiency. We outline the method here. First, as described earlier, all distinct noncovering tuples are computed and stored. They are tabulated as sets of v^{t-2} symbols representing noncovering permutation vectors rather than sets of $\binom{v^{t-2}}{t}$ noncovering tuples. Each of the $(v^{t-1} + v^{t-2} + \cdots + v)$ sets is ordered to facilitate testing whether a candidate t-tuple is a subset. For example, for $t = 4$ and

$v = 5$, the computation yields 1960750 noncovering 4-tuples. We tabulate these as $5^3+5^2+5^1 = 155$ ordered sets of $5^2 = 25$ to make the coverage check more efficient. (Only 1941375 of the noncovering 4-tuples are unique, so there are 7750000 covering 4-tuples in total.)

Next a candidate array is initialized. To reduce the scope of the search and thus reduce execution time, certain constraints are imposed:

1. When $k < v^{t-1}$, the first row is required to be in ascending order without repetition. The other rows may contain any permutation of $\{0, 1, \cdots, v^{t-1} - 1\}$.
2. When $k \geq v^{t-1}$, two constraints are imposed:
   (a) The array is partitioned into subarrays of at most v^{t-1} contiguous columns. Each row in each subarray consists of distinct entries from $\{0, 1, \cdots, v^{t-1} - 1\}$.
   (b) In the first row within each subarray, the entry in the i-th column is i.

Using backtracking, the candidate array is replaced by a new candidate array, which is checked for coverage. All t distinct columns are examined to verify that they index as a row at least one t-tuple that has distinct entries and that it is not a subset of any of the noncovering symbol sets. Candidate arrays for which all selections of t columns pass the coverage check are covering perfect hash families.

7 Concluding Remarks

The construction of covering arrays of higher strength by direct and computational methods relies to a large extent on restricting attention to those arrays that exhibit much structure. Earlier efforts to construct arrays of strength three have exploited structure arising from assuming the action of a "large" automorphism group (see [2, 3], for example). In this paper, a different strategy is adopted, by using permutation vectors to represent succinctly many entries in a covering array. While the algebraic structure of the finite field used to define these permutation vectors surely provides substantial structure, this structure is only tangentially related to regularity arising from the automorphism group. This opens a new avenue, using the algebraic structure to restrict the search space without requiring the action of any specific automorphism group on the array produced. Indeed, the computational results here demonstrate that this restriction can, in many cases, yield new covering arrays.

Acknowledgments

The first author thanks Testcover.com for support. The second and third authors thank Arizona State University, and the Consortium for Embedded and InterNetworking Technologies, for support.

References

[1] J. Bierbrauer and H. Schellwatt, Almost independent and weakly biased arrays: efficient constructions and cryptologic applications, *Advances in Cryptology (Crypto 2000)*, *Lecture Notes in Computer Science* 1880 (2000), 533–543.

[2] M. A. Chateauneuf, C. J. Colbourn, and D. L. Kreher, Covering arrays of strength 3, *Designs, Codes and Cryptography* **16** (1999) 235–242.

[3] M. A. Chateauneuf and D. L. Kreher. On the state of strength-three covering arrays. *Journal of Combinatorial Designs*, 10(4):217–238, 2002

[4] M. B. Cohen, C. J. Colbourn, and A. C. H. Ling, Augmented simulated annealing to build interaction test suites, *Discrete Mathematics*, to appear.

[5] 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(7):437–44, 1997.

[6] C.J. Colbourn. Combinatorial Aspects of Covering Arrays. *Le Matematiche (Catania)*, to appear.

[7] C. J. Colbourn, S. S. Martirosyan, G. L. Mullen, D. Shasha, G. B. Sherwood, and J. L. Yucas, Products of Mixed Covering Arrays of Strength Two, *preprint*, 2004.

[8] A. P. Godbole, D. E. Skipper, and R. A. Sunley, *t*-covering arrays: upper bounds and Poisson approximations, *Combinatorics, Probab. Comput.* 5 (1966), 105–117.

[9] A. Hartman, Software and Hardware Testing Using Combinatorial Covering Suites, in: *Graph Theory, Combinatorics and Algorithms: Interdisciplinary Applications*, Kluwer Academic Publishers, to appear.

[10] A. Hartman and L. Raskin, Problems and Algorithms for Covering Arrays, *Discrete Math* 284/1-3 (2004) 149-156.

[11] A. S. Hedayat, N. J. A. Sloane, and J. Stufken, *Orthogonal Arrays, Theory and Applications*, Springer, 1999.

[12] R. Lidl, H. Niederreiter(Editors), *Finite Fields*, 2nd ed. Cambridge, England: Cambridge University Press, 1997.

[13] S. Martirosyan and Tran Van Trung. On t-covering arrays. *Designs, Codes and Cryptography* 32 (2004), 323–339.

[14] K. Meagher and B. Stevens. Group construction of covering arrays. *Journal of Combinatorial Designs*, to appear.

[15] N. J. A. Sloane, Covering arrays and intersecting codes, *J. Combin Designs* 1 (1993), 51–63.