Journal of Statistical Planning and Inference 139 (2009) 69 – 80

Contents lists available at ScienceDirect

Journal of Statistical Planning and Inference

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

Tabu search for covering arrays using permutation vectors

Robert A. Walker II, Charles J. Colbourn*

Computer Science and Engineering, Arizona State University, P.O. Box 878809, Tempe, AZ 85287, USA

ARTICLE INFO

Available online 27 May 2008

Keywords:
Covering array
Orthogonal array
Permutation vector
Tabu search
Heuristic search

ABSTRACT

A *covering array* $\mathsf{CA}(N; t, k, v)$ is an $N \times k$ array, in which in every $N \times t$ subarray, each of the v^t possible t-tuples over v symbols occurs at least once. The parameter t is the *strength* of the array. Covering arrays have a wide range of applications for experimental screening designs, particularly for software interaction testing. A compact representation of certain covering arrays employs "permutation vectors" to encode $v^t \times 1$ subarrays of the covering array so that a covering perfect hash family whose entries correspond to permutation vectors yields a covering array. We introduce a method for effective search for covering arrays of this type using tabu search. Using this technique, improved covering arrays of strength 3, 4 and 5 have been found, as well as the first arrays of strength 6 and 7 found by computational search.

1. Introduction

A *covering array* $\mathsf{CA}(N; t, k, v)$ is an $N \times k$ array in which every subarray induced by a selection of t columns contains all possible t-tuples over v symbols. Fig. 1 shows a $\mathsf{CA}(13; 3, 10, 2)$. A $\mathsf{CA}(v^t; t, k, v)$ is an *orthogonal array*, denoted $\mathsf{OA}(t, k, v)$; in this case every t-tuple occurs *exactly* once. The smallest N for which a $\mathsf{CA}(N; t, k, v)$ exists is the *covering array number*, denoted $\mathsf{CAN}(t, k, v)$.

Screening experiments are often used to indicate factors and levels that impact response; once such factors are identified, more detailed models can then be constructed to measure main effects and interactions. A particular case arises in testing a complex system for unexpected interactions; in experimental design, covering arrays arise primarily in this setting. Covering arrays have been the focus of much research, primarily due to their applications in software and hardware interaction testing. These applications are discussed in Cohen et al. (1997) and Colbourn (2004). Applications in biological sciences also arise (Shasha et al., 2001).

Our focus is on construction techniques, rather than on the specific application to experimental design. Techniques used to construct covering arrays include recursive methods (for examples see Hartman and Raskin, 2004; Martirosyan and Van Trung, 2004; Sloane, 1993), algebraic methods (Chateauneuf and Kreher, 2002; Hedayat et al., 1999), and computational search such as in Cohen (2004, 2005) and Nurmela (2004). Recently, Sherwood et al. (2006) exploited a compact representation of covering arrays based on permutation vectors. When v is prime or a prime power, a *covering perfect hash family* $\mathsf{CPHF}(n; k, v^{t-1}, t)$ is an $n \times k$ array on v^{t-1} symbols such that every $n \times t$ subarray contains at least one row which is "covering" in the following sense.

The v^{t-1} symbols in a CPHF can be viewed as a $(t-1)$-tuple on v symbols. This $(t-1)$-tuple represents a permutation vector of length v^t over the elements of the finite field $\mathbb{F}_v$. Given a $(t-1)$-tuple $(h_1, h_2, \ldots, h_{t-1})$ with $h_j \in \{0, 1, \ldots, v-1\}$ for $1 \leqslant j \leqslant t-1$, a *permutation vector* $(\overrightarrow{h_1, h_2, \ldots, h_{t-1}})$ of length v^t has the symbol $(h_{t-1} \cdot \beta_{t-1}^{(i)}) + \cdots + (h_2 \cdot \beta_2^{(i)}) + (h_1 \cdot \beta_1^{(i)}) + \beta_0^{(i)}$ in position i

* Corresponding author. Tel.: +1 480 727 6631; fax: +1 480 965 2751.
E-mail addresses: robby.walker@gmail.com (R.A. Walker), charles.colbourn@asu.edu (C.J. Colbourn).

0378-3758/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.jspi.2008.05.020

$$\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&0&1&0&0&0&0&1&\\
1&0&1&1&0&1&0&1&0&0&\\
1&0&0&0&1&1&1&0&0&0&\\
0&1&1&0&0&1&0&0&1&0&\\
0&0&1&0&1&0&1&1&1&0&\\
1&1&0&1&0&0&1&0&1&0&\\
0&0&0&1&1&1&0&0&1&1&\\
0&0&1&1&0&0&1&0&0&1&\\
0&1&0&1&1&0&0&1&0&0&\\
1&0&0&0&0&0&0&1&1&1&\\
0&1&0&0&0&1&1&1&0&1&
\end{bmatrix}$$

Fig. 1. A CA(13; 3, 10, 2).

where i is represented in base v as $i=\sum_{k=0}^{t-1} v^k\cdot\beta_k^{(i)}$. A row is *covering* if the expansion of the permutation vectors into columns results in an OA(t,t,v). When every symbol in a CPHF is expanded in this manner, the result is a covering array.

When $i<v$, $\beta_k^{(i)}=0$ for $k>0$. Hence, every permutation vector starts with the sequence $0,1,\ldots,v-1$. Eliminating these duplicate rows leads to the key theorem of Sherwood et al. (2006):

Theorem 1.1. *If v is a prime or a prime power, and a* CPHF$(n;k,v^{t-1},t)$ *exists, then a* CA$(n\cdot(v^t-v)+v;t,k,v)$ *exists.*

We typically omit the exponent, and refer to a CPHF$(n;k,v,t)$ instead of a CPHF$(n;k,v^{t-1},t)$.

Using backtracking, Sherwood et al. (2006) found covering arrays for strengths 3 and 4 that improve upon other known constructions. In this paper, we employ the permutation vector representation as the basis of a tabu search method. In this way, we find a number of improved covering arrays for strengths 3–5; more surprisingly, we find the first covering arrays of strength 6 and 7 from computer search. We conclude by presenting the first existence tables for covering arrays of strength 5, partly to demonstrate the utility of the arrays found by the heuristic search method.

2. Forming CAs from CPHFs

In order to understand the construction underlying Theorem 1.1, we show the expansion of the following CPHF$(2;10,3,3)$ into a CA:

```
11        00        22        21        01        02        10        11        02        12
10        01        11        11        00        22        01        02        20        12
```

Write each of the v^{t-1} symbols as a $t-1$ tuple on v symbols (in this case, the 3^2 symbols as 2-tuples on 3 symbols). To convert the symbol `11` ($h_1=1,h_2=1$) into a vector of length 3^3 each row number i is written as a v^t tuple. Hence for example $i=0$ is written as $i=000$ and $i=17$ is written as $i=122$. For row $i=000$, the vector is assigned the value $0\cdot1+0\cdot1+0=0$. Continuing in this manner,

$$\begin{aligned}
&i=001:0\cdot1+0\cdot1+1=1\\
&i=002:0\cdot1+0\cdot1+2=2\\
&i=010:0\cdot1+1\cdot1+0=1\\
&i=011:0\cdot1+1\cdot1+1=2\\
&i=012:0\cdot1+1\cdot1+2=0\\
&i=020:0\cdot1+2\cdot1+0=2\\
&i=021:0\cdot1+2\cdot1+1=0\\
&i=022:0\cdot1+2\cdot1+2=1\\
&\vdots\\
&i=212:2\cdot2+1\cdot1+2=1\\
&i=220:2\cdot2+2\cdot1+0=0\\
&i=221:2\cdot2+2\cdot1+1=1\\
&i=222:2\cdot2+2\cdot1+2=2
\end{aligned}$$