Design of Sequence Family Subsets Using a Branch and Bound Technique

Gregory Cresp, Hai Huyen Dam, and Hans-Jürgen Zepernick, *Member, IEEE*

***Abstract*—The number of spreading sequences required for direct-sequence code-division multiple-access (DS-CDMA) systems depends on the number of simultaneous users in the system. Often a sequence family provides more sequences than are required; in many cases the selection of the employed sequences is a computationally intensive task. This selection is a key consideration, as the properties of the sequences assigned affect the error performance in the system. In this paper, a branch and bound algorithm is presented to perform this selection based on two different cost functions. Numerical results are presented to demonstrate the improved performance of this algorithm over previous work.**

***Index Terms*—Branch and bound technique, sequences, subset design.**

I. Introduction

FOR most classes of spreading sequences, the size of the family is determined by the sequence length, for example [1]–[3]. Many applications provide constraints on both the sequence length and the number of sequences required. Thus, it is often necessary to employ a family containing more sequences than necessary and to assign only as many sequences as required. The problem is to select the sequences employed such that they are optimal with respect to some measure. Of particular interest are correlation measures which, when the sequences are employed in communication systems, are related to the resulting error rates [4].

Given a family of spreading sequences and the number of sequences required, the selection of such a subset of sequences is nontrivial. The number of possible subsets quickly becomes astronomical as the family size increases. For example, the search space for selecting 32 sequences from a family of 64 is 1.83×10^{18}. Previous methods to solve this problem include a brute force solution proposed for multicarrier code-division multiple-access (CDMA) systems [5], a genetic algorithm approach [6] and different approaches which select incrementally optimal sequences [3], [7], [8]. Of these, only the brute force solution is guaranteed to produce the global optimum but, due to its computational complexity, such an approach is only usable for very small families.

Branch and bound algorithms [9] are well suited to large combinatorial problems. For example, in [10], such an approach is employed to examine the merit factor problem.

In this paper a new algorithm is developed, based on a branch and bound approach, to solve the problem of subset selection. Two different cost measures are considered, both related to correlation measures on the sequences. New methods are proposed to obtain bounds on the cost functions used. In order to determine this lower bound for the mean squared cross-correlation measure, a new procedure is developed to approximate the optimal subset. By applying these bounds to the branch and bound algorithm, a reduction of the computational expense of the optimization procedure is obtained. In contrast to previous methods [5]–[8], the branch and bound approach achieves this computational saving without sacrificing the optimality of the result. Computational complexity can be further reduced if some degree of suboptimality is allowed.

The computational saving of the branch and bound approach means that it is suitable for use with much larger families than can be handled by a brute force solution. There will be a limit to the family size for which this approach is suitable. The algorithm is not constrained by the sequence length of the family, only the number of sequences to select from.

Examples of the operation of the branch and bound algorithm are given for a number of different sequence families and a number of different subset sizes. These design examples demonstrate the effectiveness of the branch and bound algorithm, as the sequence families selected have lower correlation values than those produced by the approaches listed above.

The outline of this paper is as follows. Section II introduces the notation used and the correlation measures of interest. The optimization problem is outlined in Section III. A branch and bound algorithm is presented in Section IV. Strict lower bounds on the cost functions are developed in Section V. A new procedure, used to efficiently generate an approximate bound on mean squared cross-correlation values, is presented in Section VI. An example of the operation of the branch and bound procedure is detailed in Section VII. Section VIII provides numerical results and conclusions are presented in Section IX.

Manuscript received January 22, 2008; revised February 01, 2009. Current version published July 15, 2009. This work was supported in part by the Commonwealth of Australia through the Cooperative Research Centre Program and by The University of Western Australia through the Hackett Studentship. Parts of this work were published in the Australian Communications Theory Workshop, Perth, Australia, January 2006.

G. Cresp was with the School of Electrical, Electronic and Computer Engineering, The University of Western Australia, Crawley, Australia. He is now with MRX Technologies, Perth, Australia (e-mail: crespg@watri.org.au).

H. H. Dam is with the Department of Mathematics and Statistics, Curtin University of Technology, Perth, Australia (e-mail: H.Dam@exchange.curtin.edu.au).

H.-J. Zepernick is with the Blekinge Institute of Technology, SE-372 25 Ronneby, Sweden (e-mail: hans-jurgen.zepernick@bth.se).

Communicated by G. Gong, Associate Editor for Sequences.

Digital Object Identifier 10.1109/TIT.2009.2023719

II. Definitions

Consider a sequence family $\mathcal{U}$ with indexes from the finite set $\mathcal{S} \subset \mathbb{Z}$

$$\mathcal{U} = \{\mathbf{u}_k | k \in \mathcal{S}\} \tag{1}$$

where each sequence $\mathbf{u}_k$ has length N:

$$\mathbf{u}_k = [u_k(0), \ldots, u_k(N-1)]. \tag{2}$$

It is assumed that $\mathbf{u}_k \neq \mathbf{u}_r$ for $k \neq r$. The family size is thus

$$\|\mathcal{U}\| = \|\mathcal{S}\| = S \tag{3}$$

where $\|\cdot\|$ denotes the cardinality of a set. In light of this, $\mathcal{S}$ is simplified as $\{1, 2, \ldots, S\}$.

For any $\mathcal{I} \subseteq \mathcal{S}$, the subset of $\mathcal{U}$ consisting of sequences with indexes from $\mathcal{I}$ is denoted by

$$\mathcal{U}(\mathcal{I}) = \{\mathbf{u}_k \in \mathcal{U} | k \in \mathcal{I}\} \tag{4}$$

and $\mathcal{I}$ is referred to as an index set. The collection of subsets of $\mathcal{U}(\mathcal{I})$ of size M is denoted by

$$\mathcal{U}^{(M)}(\mathcal{I}) = \{\mathcal{U}(\mathcal{J}) | \mathcal{J} \subseteq \mathcal{I}, \|\mathcal{J}\| = M\}. \tag{5}$$

The set of all subsets of $\mathcal{U}$ of size M is $\mathcal{U}^{(M)}(\mathcal{S})$ and for simplicity is denoted by

$$\mathcal{U}^{(M)} = \mathcal{U}^{(M)}(\mathcal{S}). \tag{6}$$

The branch and bound procedure of Section IV aims to select the member $\mathcal{U}(\mathcal{K})$ of $\mathcal{U}^{(M)}$ that minimizes some predefined cost function. Such a method requires a means of ordering index sets. A suitable means is lexicographic ordering [11], based on the order of elements of $\mathcal{S}$. Given two sets

$$\mathcal{A} = \{a_0, a_1, \ldots, a_n\} \quad \text{and} \quad \mathcal{B} = \{b_0, b_1, \ldots, b_m\},$$

with $a_0 < a_1 < \cdots < a_n$ and $b_0 < b_1 < \cdots < b_m$, we define $\mathcal{A} < \mathcal{B}$ if either:

1) $n < m$ and $a_i = b_i$ for $0 \leq i \leq n$; or
2) there exists some $0 \leq k \leq \min(m, n)$ such that $a_i = b_i$ for all $0 \leq i < k$ and $a_i < b_i$ for $i = k$.

Using this ordering, a hierarchy of subsets may be defined.

Take some $\mathcal{I} \subseteq \mathcal{S}$ and some integer $1 \leq M \leq S$. The set $\mathcal{U}_{\mathcal{I}}^{(M)}$ equals $\{\mathcal{U}(\mathcal{I})\}$ if $\|\mathcal{I}\| = M$, otherwise if $\|\mathcal{I}\| < M$ it contains all subsets of $\mathcal{U}$ of size M whose index set $\mathcal{K}$ is either formed by appending to $\mathcal{I}$ indices which are larger than any element of $\mathcal{I}$. That is, $\mathcal{K}$ contains $\mathcal{I}$ and is greater than or equal to $\mathcal{I}$ under lexicographic ordering:

$$\mathcal{U}_{\mathcal{I}}^{(M)} = \Big\{\mathcal{U}(\mathcal{K}) \in \mathcal{U}^{(M)} | \mathcal{K} \subseteq \mathcal{S}, \|\mathcal{K}\| = M, \mathcal{I} \subseteq \mathcal{K}, \mathcal{I} \leq \mathcal{K}\Big\}. \tag{7}$$

If $\|\mathcal{I}\| > M$ then this set is empty. If $\|\mathcal{I}\| \leq M$, the size of $\mathcal{U}_{\mathcal{I}}^{(M)}$ is

$$\left\|\mathcal{U}_{\mathcal{I}}^{(M)}\right\| = \binom{S - \max(\mathcal{I})}{M - \|\mathcal{I}\|} \tag{8}$$

where $\max(\mathcal{I})$ denotes the largest element of $\mathcal{I}$ and $\binom{S-\max(\mathcal{I})}{M-\|\mathcal{I}\|}$ is the number of possible ways to choose $M - \|\mathcal{I}\|$ elements from a set of size $S - \max(\mathcal{I})$. Noting that $\binom{a}{b} = 0$ when $b < 0$ or $b > a$, and $\binom{a}{b} = 1$ when $b = 0$ or $b = a$, simple tests may be produced for the special cases where $\mathcal{U}_{\mathcal{I}}^{(M)}$ is empty or contains a single family.

Property 1: The set of subsets $\mathcal{U}_{\mathcal{I}}^{(M)}$ is either empty or contains a single element in the following cases:

$$\left\|\mathcal{U}_{\mathcal{I}}^{(M)}\right\| = \begin{cases} 0, & S - \max(\mathcal{I}) < M - \|\mathcal{I}\| \text{ or } \|\mathcal{I}\| > M \\ 1, & S - \max(\mathcal{I}) = M - \|\mathcal{I}\| \text{ or } \|\mathcal{I}\| = M. \end{cases} \tag{9}$$

In view of application areas such as asynchronous CDMA systems, we consider aperiodic correlation measures. Aperiodic correlation have been demonstrated to be a useful and popular measure for the suitability of a sequence family to such systems [4]. The aperiodic cross-correlation (CC) between two sequences $\mathbf{u}_k$ and $\mathbf{u}_r$ is defined as

$$C_{k,r}(l) = \begin{cases} \frac{1}{N} \displaystyle\sum_{i=0}^{N-l-1} u_k(i) u_r^*(i+l), & 0 \leq l < N \\ \frac{1}{N} \displaystyle\sum_{i=0}^{N+l-1} u_k(i-l) u_r^*(i), & -N < l < 0 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where the argument l represents the discrete shift between the two sequences and $(\cdot)^*$ denotes the complex conjugate of $(\cdot)$.

A quantification of the CC properties with respect to all possible shifts and all possible sequences in a given family is provided by the mean squared aperiodic cross-correlation (MSCC). For any $\mathcal{U}(\mathcal{I})$ defined as in (4), the MSCC value is defined as [3]

$$R_{\text{cc}}(\mathcal{U}(\mathcal{I})) = \frac{1}{\|\mathcal{I}\|(\|\mathcal{I}\| - 1)} \sum_{\substack{k,r \in \mathcal{I} \\ k \neq r}} \sum_{l=1-N}^{N-1} |C_{k,r}(l)|^2. \tag{11}$$

The maximum cross-correlation, c_{cm}, is a worst case measure of multi-user interference and is defined for $\mathcal{U}(\mathcal{I})$ via

$$c_{\text{cm}}(\mathcal{U}(\mathcal{I})) = \max_{\substack{k,r \in \mathcal{I} \\ k \neq r \\ -N < l < N}} \{|C_{k,r}(l)|\}. \tag{12}$$

The use of c_{cm} may overestimate interference, particularly in systems where each user's delay changes often [12]. In such cases, MSCC can give a more accurate indication of interference [13]. Where delay is more static, c_{cm} can be a more accurate measure [14].

III. Problem Formulation

For a given sequence length N, traditional spreading sequence families, such as Gold [1], Frank–Zadoff–Chu [2], and Oppermann sequences [3] produce a fixed number S of spreading sequences. For many practical applications the number M of sequences required, related to the number of users, is less than the total number of sequences in the given family. Thus, the design problem can be formulated as selecting

the subset $\mathcal{V}$ of M sequences from the entire set $\mathcal{U}$ of S sequences that minimizes a cost function f relating to correlation properties. This may be formulated as the optimization problem

$$P: \begin{cases} \text{minimize } f(\mathcal{V}) \\ \text{subject to } \mathcal{V} \in \mathcal{U}^{(M)}. \end{cases} \tag{13}$$

The minimal value for P is denoted by

$$z = \min\left\{ f(\mathcal{V}) | \mathcal{V} \in \mathcal{U}^{(M)} \right\}. \tag{14}$$

In this paper we concentrate on the following two design problems: i) minimize the mean squared cross-correlation; and ii) minimize the maximum cross-correlation. These problems can be formulated as in (13) with

$$f(\mathcal{V}) = R_{\text{cc}}(\mathcal{V}) \tag{15}$$

and

$$f(\mathcal{V}) = c_{\text{cm}}(\mathcal{V}) \tag{16}$$

respectively. A similar approach to that described here can be used for other cost functions.

Using the same approach as (8), the size of the search space of P is $\binom{S}{M}$, which becomes increasingly large as M approaches $S/2$. In light of this, an efficient optimization technique is required to solve problem P.

IV. A Branch and Bound Procedure

Problem P, given in (13), can be solved by a branch and bound method, which operates by partitioning the problem into several subproblems, each easier to solve [11]. Although it is not considered in this implementation, parallelization of branch and bound algorithms is a well developed area of research [15].

A set of subproblems is said to partition P if the subproblems' search spaces are disjoint and their union is the search space of P. In order to perform this partitioning, a notation for subproblems of P is required. For any $\mathcal{I} \subset \mathcal{S}$ define the subproblem $P_{\mathcal{I}}$ as

$$P_{\mathcal{I}}: \begin{cases} \text{minimize } f(\mathcal{V}) \\ \text{subject to } \mathcal{V} \in \mathcal{U}_{\mathcal{I}}^{(M)} \end{cases} \tag{17}$$

where $\mathcal{U}_{\mathcal{I}}^{(M)}$ is as defined in (7). Denote by $z_{\mathcal{I}}$ the minimum cost for the subproblem $P_{\mathcal{I}}$, that is

$$z_{\mathcal{I}} = \begin{cases} \min\left\{ f(\mathcal{V}) \,\middle|\, \mathcal{V} \in \mathcal{U}_{\mathcal{I}}^{(M)} \right\}, & \mathcal{U}_{\mathcal{I}}^{(M)} \neq \emptyset \\ \infty, & \mathcal{U}_{\mathcal{I}}^{(M)} = \emptyset. \end{cases} \tag{18}$$

Take some $\mathcal{I}$ with $\|\mathcal{I}\| < M$. We wish to show that the set of subproblems

$$\{P_{\mathcal{I}\cup\{i\}} | \max(\mathcal{I}) < i \leq S\} \tag{19}$$

is a partition of subproblem $P_{\mathcal{I}}$. The subproblem $P_{\mathcal{I}\cup\{i\}}$ is defined similarly to $P_{\mathcal{I}}$. The search space is $\mathcal{U}_{\mathcal{I}\cup\{i\}}^{(M)}$, that is the smallest $\|\mathcal{I}\|$ elements of the index set are $\mathcal{I}$, and the next smallest is i.

For any family $\mathcal{U}(\mathcal{K}) \in \mathcal{U}_{\mathcal{I}}^{(M)}$, noting that $\|\mathcal{I}\| < M$, the value $j = \min(\mathcal{K} \setminus \mathcal{I})$ is well defined. From this definition of j, it follows that $\mathcal{U}(\mathcal{K}) \in \mathcal{U}_{\mathcal{I}\cup\{j\}}^{(M)}$, and $j > \max(\mathcal{I})$. Conversely, for any $i > \max(\mathcal{I})$ and any $\mathcal{U}(\mathcal{K}) \in \mathcal{U}_{\mathcal{I}\cup\{i\}}^{(M)}$ we have $\mathcal{U}(\mathcal{K}) \in \mathcal{U}_{\mathcal{I}}^{(M)}$. Hence

$$\mathcal{U}_{\mathcal{I}}^{(M)} = \bigcup_{i=\max(\mathcal{I})+1}^{S} \mathcal{U}_{\mathcal{I}\cup\{i\}}^{(M)}. \tag{20}$$

Next, consider $i, j > \max(\mathcal{I})$ with $i \neq j$. For any $\mathcal{U}(\mathcal{K}) \in \mathcal{U}_{\mathcal{I}\cup\{i\}}^{(M)}$, the smallest $\|\mathcal{I}\| + 1$ elements of $\mathcal{K}$ are the elements of $\mathcal{I} \cup \{i\}$, hence $\mathcal{U}(\mathcal{K}) \notin \mathcal{U}_{\mathcal{I}\cup\{j\}}^{(M)}$. The same reasoning applies to show that for any $\mathcal{U}(\mathcal{K}) \in \mathcal{U}_{\mathcal{I}\cup\{j\}}^{(M)}$ we have $\mathcal{U}(\mathcal{K}) \notin \mathcal{U}_{\mathcal{I}\cup\{i\}}^{(M)}$. Hence

$$\mathcal{U}_{\mathcal{I}\cup\{i\}}^{(M)} \cap \mathcal{U}_{\mathcal{I}\cup\{j\}}^{(M)} = \emptyset, \quad \forall i \neq j. \tag{21}$$

Thus the set described in (19) is a partition for $P_{\mathcal{I}}$. The optimum cost for $P_{\mathcal{I}}$ can be found from this partition via [11]

$$z_{\mathcal{I}} = \min\{z_{\mathcal{I}\cup\{i\}} | \max(\mathcal{I}) < i \leq S\}. \tag{22}$$

The branch and bound algorithm is aimed at producing the optimal value z of problem P. As such, if for some subproblem $P_{\mathcal{I}}$ the optimal value satisfies $z_{\mathcal{I}} > z$, then the exact value $z_{\mathcal{I}}$ is not of interest. To this end, we introduce the term $y_{\mathcal{I}}$, which is the output produced for $P_{\mathcal{I}}$. This output satisfies

$$y_{\mathcal{I}} = z_{\mathcal{I}}, \text{ when } z_{\mathcal{I}} = z$$

and

$$y_{\mathcal{I}} \geq z_{\mathcal{I}}, \text{ when } z_{\mathcal{I}} > z. \tag{23}$$

When it can be determined that $P_{\mathcal{I}}$ does not contain the optimal value for P, that is $z_{\mathcal{I}} > z$, the output $y_{\mathcal{I}}$ can be set to ∞, eliminating the need to solve $P_{\mathcal{I}}$.

The branch and bound procedure is started by partitioning the original problem and recursively solving each subproblem in the resulting partition. The solution to the original problem is then found from the solutions to the first partition. This is detailed in Procedure 1 and summarized in the flow chart of Fig. 1.

Procedure 1 (Branch and Bound):

Inputs	Sequence family $\mathcal{U}$, subset size M, cost function f.
Step 1	Construct a partition for problem P: $\left\{P_{\{i\}} | i = 1, 2, \ldots, S\right\}$. (24)
Step 2	For each $i = 1, \ldots, S$, use Step 3 with input $\mathcal{I} = \{i\}$ to produce $y_{\{i\}}$.
Step 3	Recursive step. The input is an index set $\mathcal{I} \subset \mathcal{S}$.
Step 3A	If $\mathcal{U}_{\mathcal{I}}^{(M)} = \emptyset$, then the subproblem is *infeasible*. Set $y_{\mathcal{I}} = \infty$ and terminate. As per Property 1, this is the case if $S - \max(\mathcal{I}) < M - \|\mathcal{I}\|$ or $\|\mathcal{I}\| > M$.
Step 3B	If $z_{\mathcal{I}}$ is trivially determinable, then $y_{\mathcal{I}} = z_{\mathcal{I}}$ can be returned immediately. This case is referred to as *optimality*. We define trivially determinable as $\|\mathcal{U}_{\mathcal{I}}^{(M)}\| = 1$. Following Property 1, optimality applies if $S - \max(\mathcal{I}) = M - \|\mathcal{I}\|$ or $\|\mathcal{I}\| = M$.

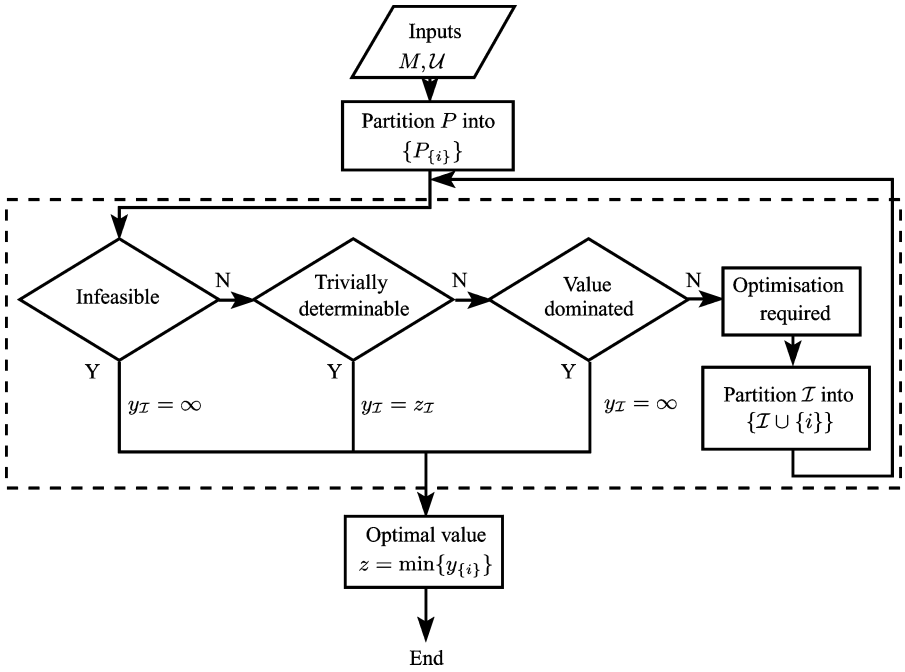

Fig. 1. Flowchart of the branch and bound procedure (Procedure 1).

Step 3C If $z_{\mathcal{I}}$ is not trivially determinable, test for *value dominance*. This is done by finding a lower bound $z_{\mathcal{I},\text{low}}$ for $z_{\mathcal{I}}$ and an upper bound $z_{\max}$ for z. If $z_{\max} < z_{\mathcal{I},\text{low}}$ then $z < z_{\mathcal{I}}$ and hence this subproblem cannot produce a solution which is optimal for P. Terminate and return $y_{\mathcal{I}} = \infty$. The derivation of $z_{\mathcal{I},\text{low}}$ and $z_{\max}$ is considered in Section V.

Step 3D If none of the above possibilities apply, $P_{\mathcal{I}}$ *requires optimization*. The problem $P_{\mathcal{I}}$ is partitioned as in (19). For each $i = \max(\mathcal{I}) + 1, \ldots, S$ recursively call Procedure 1 Step 3 with input $\mathcal{I} \cup \{i\}$ to produce $y_{\mathcal{I}\cup\{i\}}$.

Step 3E The output $y_{\mathcal{I}}$ is found, in a similar manner to (22) via

$$y_{\mathcal{I}} = \min\left\{y_{\mathcal{I}\cup\{i\}} \middle| \max(\mathcal{I}) < i \leq S\right\}. \quad (25)$$

Step 4 The optimal value z is found as per (22)

$$z = \min_{i=1,\ldots,S}\left\{y_{\{i\}}\right\}. \quad (26)$$

End

The subproblem $\mathcal{P}_{\{i\}}$ considered above has search space $\mathcal{U}_{\{i\}}^{(M)}$, containing all sequence families whose minimum sequence index is i.

It follows from (23) that the output produced by Procedure 1 is z, the optimal value for P. Noting that $\left\{P_{\{i\}} | i = 1, 2, \ldots, S\right\}$ is a partition for P, at least one subproblem $P_{\{i\}}$ must contain the optimal point in the search space for P, and hence for that i we have $z = z_{\{i\}} = y_{\{i\}}$.

Given that Procedure 1 contains a recursive step, it must be verified that the procedure will terminate. The partitioning performed in Step 3D strictly reduces the size of the search space considered in each recursive call. Since Procedure 1 Step 3 terminates when the search space reaches size 1 or 0, and the initial search space is finite, it follows that there is a finite limit on the number of recursive calls.

V. Strict Bounds for Value Dominance

The key to an efficient branch and bound procedure is the production of tight and computationally simple bounds $z_{\max}$ and $z_{\mathcal{I},\text{low}}$, used in Step 3C.

The upper bound $z_{\max}$ can be calculated the same way regardless of the cost function used. A suitable value is the smallest value of $y_{\mathcal{I}}$ produced by any previous execution of Procedure 1 Step 3. This holds since for all $\mathcal{I}$ we have $z \leq z_{\mathcal{I}} \leq y_{\mathcal{I}}$, hence $z \leq \min\{y_{\mathcal{I}} | \mathcal{I} \subset \mathcal{S}, ||\mathcal{I}|| = M\}$. If no $y_{\mathcal{I}}$ values have been produced, the upper bound $z_{\max} = \infty$ is used.

The lower bound on the optimal value of the current subproblem must be formulated separately for each different cost function. In the following we propose lower bounds corresponding to the two cost functions presented in Section III.

A. Minimizing Mean Squared Cross-Correlation

Let $P_{\mathcal{I}}$ be the current subproblem and define $||\mathcal{I}|| = I$ and $\mathcal{L} = \{\max(\mathcal{I}) + 1, \ldots, S\}$. Let the optimal solution to $P_{\mathcal{I}}$ be

$$\mathcal{U}(\mathcal{K}) = \mathcal{U}(\mathcal{I}) \cup \mathcal{U}(\mathcal{J}) \quad (27)$$

where $\mathcal{K} = \mathcal{I} \cup \mathcal{J}$ for some $\mathcal{J} \subset \mathcal{L}$ with $\|\mathcal{J}\| = M - I$. The MSCC of this optimal solution $\mathcal{U}(\mathcal{K})$ is then

$$\begin{aligned} &R_{\mathrm{cc}}(\mathcal{U}(\mathcal{K})) \\ &\quad = \frac{1}{M(M-1)} \sum_{\substack{k,r\in\mathcal{I}\\ k\neq r}} \sum_{l=1-N}^{N-1} |C_{k,r}(l)|^2 \\ &\quad + \frac{2}{M(M-1)} \sum_{k\in\mathcal{J}} \sum_{r\in\mathcal{I}} \sum_{l=1-N}^{N-1} |C_{k,r}(l)|^2 \\ &\quad + \frac{1}{M(M-1)} \sum_{\substack{k,r\in\mathcal{J}\\ k\neq r}} \sum_{l=1-N}^{N-1} |C_{k,r}(l)|^2. \end{aligned} \tag{28}$$

A lower bound on $R_{\mathrm{cc}}(\mathcal{U}(\mathcal{K}))$ is produced by considering each of the three terms in (28) separately. The bound, denoted $z_{\mathcal{I},\mathrm{low}}$, is calculated via

$$z_{\mathcal{I},\mathrm{low}} = R_1 + R_{2,\mathrm{low}} + R_{3,\mathrm{low}} \tag{29}$$

where R_1 is the exact value of the first term of (28) and $R_{2,\mathrm{low}}$, $R_{3,\mathrm{low}}$ are lower bounds for the second and third terms, respectively. The terms $R_1, R_{2,\mathrm{low}}$ and $R_{3,\mathrm{low}}$ are derived in the following.

- Since $\mathcal{I}$ is known, R_1 may be calculated via

$$R_1 = \begin{cases} \frac{I(I-1)}{M(M-1)} R_{\mathrm{cc}}(\mathcal{U}(\mathcal{I})), & I > 1 \\ 0, & \text{otherwise.} \end{cases} \tag{30}$$

- The second term of (28) can not be directly calculated since it sums over $\mathcal{J}$, which is unknown. Noting that $\mathcal{J} \subset \mathcal{L}$, a lower bound on this term may be found by considering the elements of $\mathcal{L}$ which give the smallest contribution to such a sum. For each $k \in \mathcal{L}$ let

$$K(k) = \frac{2}{M(M-1)} \sum_{r\in\mathcal{I}} \sum_{l=1-N}^{N-1} |C_{k,r}(l)|^2. \tag{31}$$

Let $\mathcal{M}$ denote the indices k of the $\|\mathcal{J}\| = M - I$ smallest elements $K(k)$ of the set $\{K(k)|k \in \mathcal{L}\}$. The set $\mathcal{M}$ is thus the subset of $\mathcal{L}$ which gives the smallest possible sum in the style of the second term of (28). If we define

$$R_{2,\mathrm{low}} = \begin{cases} \sum_{k\in\mathcal{M}} K(k), & I < M \\ 0, & \text{otherwise} \end{cases} \tag{32}$$

then, as required

$$R_{2,\mathrm{low}} \leq \frac{2}{M(M-1)} \sum_{k\in\mathcal{J}} \sum_{r\in\mathcal{I}} \sum_{l=1-N}^{N-1} |C_{k,r}(l)|^2. \tag{33}$$

- To produce $R_{3,\mathrm{low}}$, note that the third term sums over the $2\binom{M-I}{2}$ possible pairs (k,r) such that $k,r \in \mathcal{J}$ with $k \neq r$. Whilst $\mathcal{J}$ is unknown, it is known that $\mathcal{J} \subset \mathcal{S}$. A lower bound on the contribution of each pair (k,r) with $k,r \in \mathcal{J}$ is thus the minimum contribution from any pair (k,r) with $k,r \in \mathcal{S}$ and $k \neq r$. Let

$$c = \min_{\substack{k,r\in\mathcal{S}\\ k\neq r}} \sum_{l=1-N}^{N-1} |C_{k,r}(l)|^2 \tag{34}$$

be this minimum. The lower bound on the third term of (28) is thus

$$R_{3,\mathrm{low}} = \begin{cases} \frac{1}{M(M-1)} 2\binom{M-I}{2} c, & I < M-1 \\ 0, & \text{otherwise.} \end{cases} \tag{35}$$

Clearly for any $k,r \in \mathcal{J}$ with $k \neq r$ it follows that $c \leq \sum_{l=1-N}^{N-1} |C_{k,r}(l)|^2$ and hence

$$R_{3,\mathrm{low}} \leq \frac{1}{M(M-1)} \sum_{\substack{k,r\in\mathcal{J}\\ k\neq r}} \sum_{l=1-N}^{N-1} |C_{k,r}(l)|^2. \tag{36}$$

B. Minimizing Maximum Cross-Correlation

Given the index set $\mathcal{I}$ for the current subproblem $P_{\mathcal{I}}$, minimizing maximum cross-correlation, a lower bound, $z_{\mathcal{I},\mathrm{low}}$, for $z_{\mathcal{I}}$ is

$$z_{\mathcal{I},\mathrm{low}} = c_{\mathrm{cm}}(\mathcal{U}(\mathcal{I})). \tag{37}$$

This follows from (12), since if $\mathcal{U}(\mathcal{K})$ is the optimal solution to $P_{\mathcal{I}}$, we have $\mathcal{I} \subset \mathcal{K}$ and hence $c_{\mathrm{cm}}(\mathcal{U}(\mathcal{I})) \leq c_{\mathrm{cm}}(\mathcal{U}(\mathcal{K}))$.

VI. An Approximate Lower Bound for Mean Squared Cross-Correlation

A. The Use of an Approximate Lower Bound

If value dominance is never applied, the branch and bound algorithm simply performs global optimization. As value dominance is applied more often, the efficiency of the algorithm increases. The frequency with which value dominance can be applied, and hence the efficiency of the branch and bound algorithm, is dependent on the tightness of $z_{\mathcal{I},\mathrm{low}}$, the lower bound on $z_{\mathcal{I}}$. A tighter bound will result in a more efficient algorithm.

Rather than use an exact lower bound $z_{\mathcal{I},\mathrm{low}}$, an approximate lower bound $z_{\mathcal{I},\mathrm{app}}$ can be employed. As the bound is no longer required to be a strict lower bound, a larger value can be selected. However, since the actual value $z_{\mathcal{I}}$ will typically be larger than the strict lower bound, the approximate bound $z_{\mathcal{I},\mathrm{app}}$ will generally be closer to the actual value than the strict bound $z_{\mathcal{I},\mathrm{low}}$. The price for this gain is that $z_{\mathcal{I},\mathrm{app}}$ may sometimes be larger than $z_{\mathcal{I}}$ and hence lead to value dominance being incorrectly applied. An incorrect application of value dominance may lead to $z_{\mathcal{I}} < y_{\mathcal{I}}$ when $z_{\mathcal{I}} = z$, which violates (23). Since the optimality of the output z is dependent on (23), the use of an approximate bound may lead to suboptimal output.

The approximate bound should be selected such that this risk of suboptimal output is offset by a large decrease in computational cost. In the study performed by the authors, the results of which are given in Section VIII, a significant reduction in running time was achieved when the bound $z_{\mathcal{I},\mathrm{app}}$ described below was used and the resulting output z was in all cases still optimal.

Using the same framework as in Section V-A we consider an approximate bound for use in Procedure 1 when the cost function is the MSCC. The first two terms of (28) are calculated as before. The third term remains to be considered. An ideal lower bound $R_{3,\text{ideal}}$ for this term would satisfy

$$R_{3,\text{ideal}} = \frac{(M-I)(M-I-1)}{M(M-1)} \min\left\{R_{\text{cc}}(\mathcal{V}) \middle| \mathcal{V} \in \mathcal{U}^{(M-I)}(\mathcal{L})\right\}. \quad (38)$$

Identifying this minimal value involves solving an optimization problem very similar to the original problem P. As such an approximate lower bound can be found via methods which find an approximately optimal solution to the selection of a subset with respect to MSCC.

B. The Fillin Method

We first present a new method to find an approximate solution to such a problem, which we will refer to as the Fillin method. The advantage of this new method is its very small computational cost compared to other techniques, such as Mourad *et al.*'s algorithm [8] or a Monte Carlo algorithm [14].

The Fillin method approximates the optimal cost of a subset of size M of a given family by recursively considering subsets of size $M-1$. The results of these considerations are stored in two tables, $\mathbf{C}$ and $\mathbf{S}$. The jth column of $\mathbf{C} = [c_{i,j}]$ contains the costs of subsets of size j, the jth column of $\mathbf{S} = [\mathcal{J}_{i,j}]$ contains the corresponding index sets. Within the tables, the row indicates how much of the original family is being considered for inclusion. The entry in row i, column j corresponds to selecting a subset of size j whose index set contains values greater than or equal to i, that is an element of $\mathcal{U}^{(j)}(\{i,\ldots,S\})$.

In the case where $\mathcal{U}^{(j)}(\{i,\ldots,S\})$ is empty, the corresponding entries in $\mathbf{C}$ and $\mathbf{S}$ can be immediately filled as

$$c_{i,j} = \infty, \quad \forall S-j+1 < i \leq S \quad (39)$$
$$\mathcal{J}_{i,j} = \emptyset, \quad \forall S-j+1 < i \leq S. \quad (40)$$

Similarly, when the subsets in $\mathcal{U}^{(j)}(\{i,\ldots,S\})$ contain a single sequence the MSCC is undefined and hence

$$c_{i,1} = 0, \quad \forall 1 \leq i \leq S \quad (41)$$
$$\mathcal{J}_{i,1} = \{i\}, \quad \forall 1 \leq i \leq S. \quad (42)$$

In the remaining cases, $c_{i,j}$ is an approximation of the optimal MSCC for an element of $\mathcal{U}^{(j)}(\{i,\ldots,S\})$

$$c_{i,j} \approx \min\left\{\frac{j(j-1)}{M(M-1)} R_{\text{cc}}(\mathcal{V}) \middle| \mathcal{V} \in \mathcal{U}^{(j)}(\{i,\ldots,S\})\right\}, \quad \forall 2 \leq j \leq M, 1 \leq i \leq S-j+1 \quad (43)$$

and $\mathcal{J}_{i,j}$ is the index set of the corresponding subset $\mathcal{V}$.

For an (i,j) with $2 \leq j \leq M$ and $1 \leq i \leq S-j+1$ the values $c_{i,j}$ and $\mathcal{J}_{i,j}$ can be found as described below, with reference to the values $c_{k,j-1}$ for $i < k \leq S$ and $c_{i+1,j}$. In this manner the two tables can be recursively filled in in order of increasing j and decreasing i.

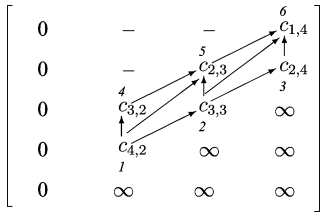

Fig. 2. The steps taken to calculate cost $c_{1,4}$ using the Fillin method, including the order of calculation. "$-$" indicates a value not calculated.

Take such an (i,j) and consider the elements of $\mathcal{U}^{(j)}(\{i,\ldots,S\})$. First there are those subsets which do not contain $\mathbf{u}_i$. These subsets are all members of $\mathcal{U}^{(j)}(\{i+1,\ldots,S\})$, the best known element of which is $\mathcal{J}_{i+1,j}$. Alternatively, there are those subsets which contain $\mathbf{u}_i$. Each of these subsets can be constructed by adding $\mathbf{u}_i$ to a subset in $\mathcal{U}^{(j-1)}(\{k,\ldots,S\})$ for some $i < k \leq S$. As such, suitable candidates for the optimal such subset are $\mathcal{J}_{k,j-1} \cup \{i\}$ for $i < k \leq S$.

If we define

$$k_1 = \operatorname*{argmin}_{i<k\leq S-j+2} \left(c_{k,j-1} + \frac{2}{M(M-1)} \sum_{r\in\mathcal{J}_{k,j-1}} \sum_{l=1-N}^{N-1} |C_{i,r}(l)|^2\right) \quad (44)$$

and

$$f_{\min} = c_{k_1,j-1} + \frac{2}{M(M-1)} \sum_{r\in\mathcal{J}_{k_1,j-1}} \sum_{l=1-N}^{N-1} |C_{i,r}(l)|^2 \quad (45)$$

then

$$c_{i,j} = \min(c_{i+1,j}, f_{\min})$$
$$\mathcal{J}_{i,j} = \begin{cases} \mathcal{J}_{i+1,j}, & c_{i+1,j} \leq f_{\min} \\ \mathcal{J}_{k_1,j-1} \cup \{i\}, & f_{\min} < c_{i+1,j}. \end{cases} \quad (46)$$

The value $c_{1,M}$ can be used as an approximate solution to P. Of more interest is the value $c_{(\max\mathcal{I}+1),(M-||\mathcal{I}||)}$, which is an approximation of the ideal bound given in (38).

The operation of the Fillin method is illustrated in Fig. 2. This figure shows the matrix $\mathbf{C}$ for the case $S = 5, M = 4$. The values which are required to calculate the entry $c_{1,4}$ are indicated, as well as the order in which these values will themselves be calculated.

C. The Approximate Lower Bound

An approximate lower bound to the third term of (28) is obtained by employing both the Fillin method and the Removal method of [7]. The final result is the minimum of the two results

$$R_{3,\text{app}} = \min\left(c_{(\min\mathcal{L}),(M-I)}, d_{(\min\mathcal{L}),(M-I)}\right). \quad (47)$$

Here $c_{(\min\mathcal{L}),(M-I)}$, as defined in (43), is the Fillin method approximation of the optimal MSCC of an element of $\mathcal{U}^{(M-I)}(\mathcal{L})$ and $d_{(\min\mathcal{L}),(M-I)}$ is the corresponding approximation produced by the Removal method. Noting that there are only a

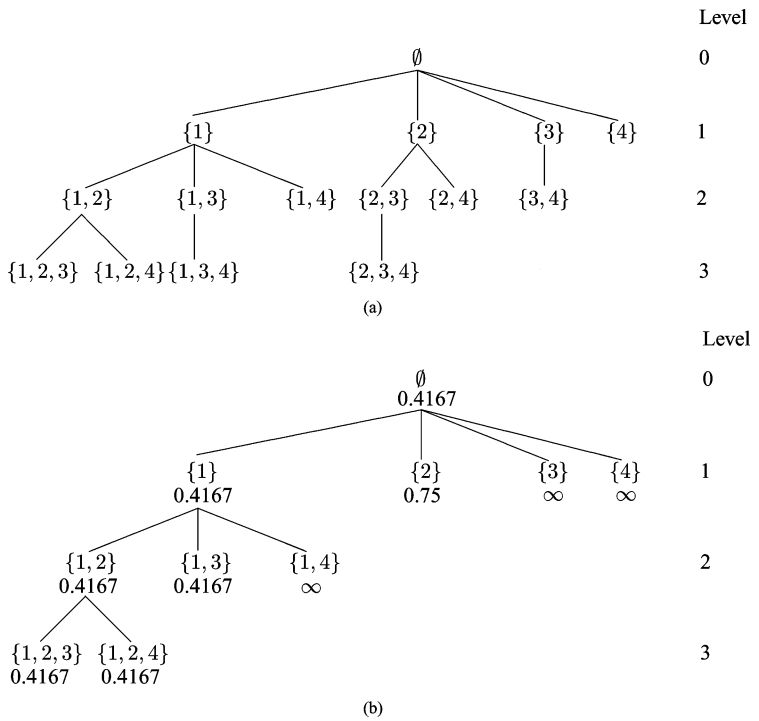

Fig. 3. Search trees for subsets of size 3 of the Walsh Hadamard family of length 4. (a) The unpruned tree and (b) the pruned and labeled tree.

small number of possible values for $\mathcal{L}$ and $M-I$, these values can be calculated before the start of the branch and bound procedure and stored for later reference.

The resulting approximate lower bound on $z_{\mathcal{I}}$ is

$$z_{\mathcal{I},\text{app}} = R_1 + R_{2,\text{low}} + R_{3,\text{app}}. \tag{48}$$

As will be seen in Section VIII, the branch and bound procedure using $z_{\mathcal{I},\text{app}}$ gives the same, optimal, results as one using $z_{\mathcal{I},\text{low}}$ in all the tests performed here. It is important in (47) to use both the Fillin method and the Removal method together. If either of these methods is used on its own, there are cases where the output of the branch and bound algorithm is suboptimal.

VII. Branch and Bound Example

To demonstrate the use of the branch and bound procedure, we now consider an example where $\mathcal{U}$ is the Walsh Hadamard family [16] of size $S=4$, that is $\mathcal{S}=\{1,2,3,4\}$, and a subset of size $M=3$ is optimized for MSCC.

The subproblems produced by the branch and bound process form a tree structure. For simplicity, we identify each subproblem $P_{\mathcal{I}}$ by the set $\mathcal{I}$. The empty set, $\emptyset$, corresponding to problem P, is the root of the tree, denoted level 0. For any $\mathcal{I} \subset \mathcal{S}$ the children of node $\mathcal{I}$, that is nodes one level below $\mathcal{I}$ and directly connected to $\mathcal{I}$, correspond to the subproblems forming the partition of $\mathcal{P}_{\mathcal{I}}$ given in (19). If $\|\mathcal{I}\| \geq M$ then there are no such children, otherwise the child nodes are the sets $\mathcal{I} \cup \{i\}$ for each i with $\max(\mathcal{I}) < i \leq S$. Nodes with no children are referred to as leaves, by the definition above every node at level M will be a leaf. If $\mathcal{I}$ is a leaf at level M, then $\mathcal{U}(\mathcal{I})$ is a member of the search space of P. It follows that a node which has no level M descendants will correspond to an infeasible subproblem. At any node corresponding to an infeasible or value dominated subproblem the tree can be pruned, since no children of such a node need to be considered.

TABLE I
Branch and Bound Procedure for the Walsh Hadamard Family With Size $S=4$ and $M=3$

Node $\mathcal{I}$	Case	Result
$\emptyset$	Start	$z=\min(0.4167, 0.75, \infty, \infty)$
$\{1\}$	Optimisation	$y_{\mathcal{I}}=\min(0.4167, 0.4167, \infty)$
$\{1,2\}$	Optimisation	$y_{\mathcal{I}}=\min(0.4167, 0.4167)$
$\{1,2,3\}$	Optimality	$y_{\mathcal{I}}=0.4167$
$\{1,2,4\}$	Optimality	$y_{\mathcal{I}}=0.4167$
$\{1,3\}$	Optimality	$y_{\mathcal{I}}=0.4167$
$\{1,4\}$	Infeasible	$y_{\mathcal{I}}=\infty$
$\{2\}$	Optimality	$y_{\mathcal{I}}=0.75$
$\{3\}$	Infeasible	$y_{\mathcal{I}}=\infty$
$\{4\}$	Infeasible	$y_{\mathcal{I}}=\infty$

For the problem presented here, the entire search tree before pruning can be seen in Fig. 3(a). Examining level 3 of this tree we see that the search space contains four families. The order in which the nodes of the tree are considered, as well as the action taken in each case, is summarized in Table I.

The first node considered, via a call to Procedure 1, is $\emptyset$. From this procedure, Procedure 1 Step 3 is called for each size 1 subset of $\mathcal{S}=\{1,2,3,4\}$, that is the nodes $\{1\},\{2\},\{3\}$ and $\{4\}$. The first subproblem considered is $P_{\{1\}}$. Since the search space of $\mathcal{P}_{\{1\}}$, the set $\mathcal{U}^{(3)}_{\{1\}}$, contains multiple families, namely $\mathcal{U}(\{1,2,3\})$,$\mathcal{U}(\{1,2,4\})$ and $\mathcal{U}(\{1,3,4\})$, neither infeasibility now optimality apply. No suitable upper bound for the optimal value z has yet been found, hence value dominance cannot apply. Thus optimization is required. As a result, Step 3 recursively calls itself with inputs $\{1,2\},\{1,3\}$ and $\{1,4\}$, corresponding to the children of $\{1\}$. As before, the node $\{1,2\}$ requires optimization and hence recurses to nodes $\{1,2,3\}$ and

TABLE II
COSTS FOR MSCC MINIMIZATION TECHNIQUES

family	M	Removal	MCarlo	Mourad	BnB	Itrns (%)
WHad32	2	0.0313	0.0313	0.0313	0.0313	6.250
	4	0.0729	0.0729	0.0729	0.0729	0.973
	6	0.2146	0.1896	0.1479	0.1479	0.259
	8	0.2813	0.3683	0.2545	0.2321	0.089
	10	0.4243	0.3826	0.3479	0.3368	0.093
	12	0.4943	0.4706	0.4271	0.4195	0.104
	14	0.5491	0.5381	0.4887	0.4887	0.736
	16	0.6281	0.5807	0.5479	0.5469	1.637
Opp31	2	0.0345	0.0323	0.0323	0.0323	6.667
	4	0.0635	0.0546	0.0540	0.0540	2.386
	6	0.0856	0.0774	0.0865	0.0758	0.664
	8	0.1117	0.1071	0.0985	0.0985	0.367
	10	0.1479	0.1425	0.1403	0.1195	0.259
	12	0.1654	0.1608	0.1584	0.1494	0.347
	14	0.1706	0.2312	0.1707	0.1673	0.627
	16	0.1966	0.2330	0.1961	0.1961	1.850
Gold33	2	0.6816	0.5567	0.5567	0.5567	6.439
	4	0.7863	0.7815	0.7669	0.7669	0.269
	6	0.8436	0.8669	0.8331	0.8306	0.042
	8	0.8774	0.8945	0.8675	0.8675	0.011
	10	0.9047	0.9215	0.8949	0.8909	0.032
	12	0.9312	0.9342	0.9091	0.9083	0.008
	14	0.9414	0.9384	0.9215	0.9214	0.004
	16	0.9439	0.9461	0.9327	0.9327	0.002
FZC32	2	0.5939	0.5939	0.5939	0.5939	6.855
	4	0.8483	0.8785	0.8254	0.8082	1.193
	6	0.9017	0.9146	0.8906	0.8785	0.805
	8	0.9279	0.9571	0.9242	0.9189	0.153
	10	0.9500	0.9748	0.9473	0.9416	0.847
	12	0.9599	0.9910	0.9595	0.9574	0.362
	14	0.9722	1.0089	0.9706	0.9706	0.382
	16	0.9817	1.0301	0.9810	0.9810	2.443

$\{1,2,4\}$. The node $\{1,2,3\}$ is a leaf at level 3, so optimality applies, producing $y_{\{1,2,3\}} = 0.4167$. Similarly for node $\{1,2,4\}$ optimality applies and $y_{\{1,2,4\}} = 0.4167$. These results allow $\{1,2\}$ to be labeled with $y_{\{1,2\}} = \min(y_{\{1,2,3\}}, y_{\{1,2,4\}}) = 0.4167$.

Examining now $\{1,3\}$, since $\|\mathcal{U}^3_{\{1,3\}}\| = 1$ optimality applies, producing $y_{\{1,3\}} = 0.4167$. Finally, the node $\{1,4\}$ is infeasible and is hence labeled with $y_{\{1,4\}} = \infty$. These results produce $y_{\{1\}} = \min(y_{\{1,2\}}, y_{\{1,3\}}, y_{\{1,4\}}) = 0.4167$.

The next node in level 1 to consider is $\{2\}$. Since $\|\mathcal{U}^3_{\{2\}}\| = 1$, optimality applies, giving $y_{\{2\}} = 0.75$. Finally, $\mathcal{U}^3_{\mathcal{I}} = \emptyset$ for both $\mathcal{I} = \{3\}$ and $\mathcal{I} = \{4\}$ so the two nodes are infeasible and can be labeled with $y_{\{3\}} = \infty$ and $y_{\{4\}} = \infty$, respectively.

There are no further calls to Step 3 to process, so Procedure 1 can proceed to Step 4 and then terminate. The final output is thus

$$z = \min_{i=1,2,3,4} y_{\{i\}} = 0.4167. \tag{49}$$

The pruned search tree, with each node labeled appropriately, is seen in Fig. 3(b).

VIII. DESIGN EXAMPLES

A study was performed to compare the correlation performance of the branch and bound procedure to that of the Removal algorithm, a Monte Carlo algorithm similar to that of [14] and Mourad *et al.*'s approach [8].

The Removal algorithm removes sequences from the family one at a time until the desired subset size is reached. Two different versions of this algorithm, one for each problem described in Section III, were considered. The first iteratively finds the pair of sequences which yield maximum cross-correlation and removes the sequence in the pair which contributes more to the MSCC of the entire set [7]. The second, used to optimize c_{cm}, iteratively finds all sequences which achieve the maximum cross-correlation and removes that which achieves this maximum correlation the largest number of times. Whilst the Removal algorithms have the lowest computation cost of any method considered here, the subsets they produce are almost always suboptimal.

The Monte Carlo algorithm randomly generates a fixed number, in this case 1000, of subsets of size M using a uniform distribution. Of these subsets that with the smallest cost function is selected. The Monte Carlo algorithm is conceptually the simplest of the methods listed here, although it is computationally more intensive than either the Fillin or Removal methods. The random nature of its selection of subsets means there is a large uncertainty in its output. Thus there is no guarantee that its output is close to optimal, and the results can vary greatly between different executions.

The algorithm of Mourad *et al.* [8] follows a similar approach to the branch and bound procedure. The set initially contains a single sequence and sequences are added one at a time until the required set size is met. At each stage the additional sequence is the one producing the minimal incremental increase in the cost measure. By choosing all possible initial sequences, S different sets of size M are produced, the final set is that with the minimal cost. Mourad *et al.*'s algorithm differs from the branch and

TABLE III
COSTS FOR c_{cm} MINIMIZATION TECHNIQUES

family	M	Removal	MCarlo	Mourad	BnB	Itrns (%)
WHad32	2	0.0625	0.0313	0.0312	0.0312	6.452
	4	0.1563	0.0938	0.0938	0.0938	0.222
	6	0.2500	0.2188	0.1875	0.1875	0.050
	8	0.3750	0.3750	0.2500	0.2500	0.013
	10	0.5313	0.4375	0.3750	0.3750	0.013
	12	0.6250	0.6250	0.4687	0.4375	0.014
	14	0.6875	0.6875	0.5625	0.5625	0.018
	16	0.7188	0.7500	0.6562	0.6562	0.030
Opp31	2	0.0323	0.0323	0.0323	0.0323	6.897
	4	0.3135	0.0489	0.0490	0.0489	1.029
	6	0.3136	0.0848	0.0850	0.0681	0.380
	8	0.3137	0.1060	0.1064	0.1060	0.158
	10	0.3138	0.1577	0.1581	0.1133	0.067
	12	0.3139	0.1583	0.3136	0.1578	0.056
	14	0.3140	0.1754	0.3137	0.1585	0.025
	16	0.3142	0.3135	0.3138	0.3135	0.025
Gold33	2	0.2903	0.2258	0.2258	0.2258	6.629
	4	0.3226	0.2903	0.2903	0.2903	0.261
	6	0.3226	0.3226	0.3226	0.3226	0.066
	8	0.3548	0.3548	0.3226	0.3226	0.027
	10	0.3548	0.3548	0.3548	0.3548	0.019
	12	0.3548	0.3548	0.3548	0.3548	0.018
	14	0.3548	0.3548	0.3548	0.3548	0.016
	16	0.3548	0.3548	0.3548	0.3548	0.025
FZC32	2	0.1768	0.1768	0.1768	0.1768	8.669
	4	0.2882	0.2500	0.2500	0.2500	0.264
	6	0.3679	0.3536	0.3536	0.3536	0.045
	8	0.3679	0.3679	0.3536	0.3536	0.015
	10	0.5000	0.5000	0.5000	0.5000	0.006
	12	0.5000	0.5000	0.5000	0.5000	0.003
	14	0.5000	0.5000	0.5000	0.5000	0.002
	16	0.5000	0.7071	0.5000	0.5000	0.001

bound approach in that no ordering of the index sets is maintained and that only the minimal incremental cost sequence is added at each state, rather than every sequence being tested. The lack of ordering in subsets means that the same subset may be tested multiple times which, as seen below, results in a higher computational complexity than the branch and bound algorithm for small M.

The results of this investigation are shown in Tables II and III. The Removal algorithms are labeled as "Removal", Mourad *et al.*'s algorithm is labeled "Mourad", the Monte Carlo algorithm is labeled "MCarlo" and the branch and bound results are labeled as 'BnB'. Table II gives the result, $f(\mathcal{V}) = R_{\mathrm{cc}}(\mathcal{V})$, when each algorithm is applied to optimize MSCC and Table III gives the resulting cost, $f(\mathcal{V}) = c_{\mathrm{cm}}(\mathcal{V})$, when they are used to optimize maximum correlation.

These methods were applied to several different sequence families. Each family can be found as a block of rows in each table. The families were:

- The Walsh Hadamard family [17] of length 32 and size 32 (WHad32);
- The Oppermann family [3] of length 31 and size 30 with parameters $m = 1.0038, n = 1.0000$ [18] (Opp31);
- The Gold code [1] with sequence length 31 and size 33 constructed from m-sequences with taps 2 and 2, 3, 4 (Gold33);
- The FZC [2], [19] family of length 32 and size 32 (FZC32).

These families were optimized for subsets with sizes M ranging from 2 to 16. When optimizing for MSCC, there is a noticeable advantage for each family in using branch and bound over the other approaches, as seen in Table II. When maximum correlation is used as the cost function, Table III shows an improvement in the resulting cost when using the branch and bound technique compared to each of the other techniques for the Oppermann family and for the Walsh Hadamard family when $M = 12$, but no gain over Mourad *et al.*'s approach for the other families.

The number of recursive branch and bound steps required for each branch and bound MSCC optimization is shown in Fig. 4, both using a strict lower bound, $z_{\mathcal{I},\mathrm{low}}$, and using an approximate bound, $z_{\mathcal{I},\mathrm{app}}$. For the sets Gold33 and WHad32 this relationship is approximately linear on a semilog scale, however for FZC32 and Opp31 the rate of increase reduces as M increases. This is particularly noticeable when an approximate lower bound $z_{\mathcal{I},\mathrm{app}}$ is used, where the number of iterations levels after $M = 12$. The final column of Tables II and III, labeled 'Itrns (%)', gives the number of iterations required by the branch and bound algorithm when employing z_{app}, as a percentage of the size of the entire search space. They show that such an algorithm requires only a small fraction of the computation of global optimization.

A comparison of the execution times of the different methods to optimize the MSCC of the set WHad32 is shown in Fig. 5. This figure shows times for the branch and bound algorithm using both the exact and approximate bounds to determine value dominance, labeled as "BnB z_{low}" and "BnB z_{app}," respectively, as well as that for Mourad *et al.*'s approach, the Monte Carlo algorithm, and the Removal method. As noted previously, the size of the search space makes performing an exhaustive

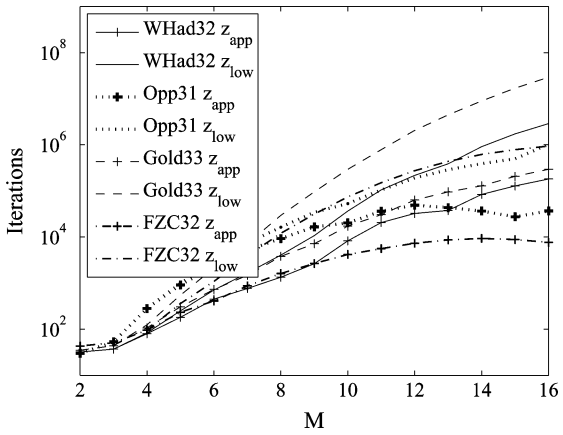

Fig. 4. The number of recursive calls to Procedure 1 required for each MSCC optimization using the branch and bound algorithm employing z_{app}.

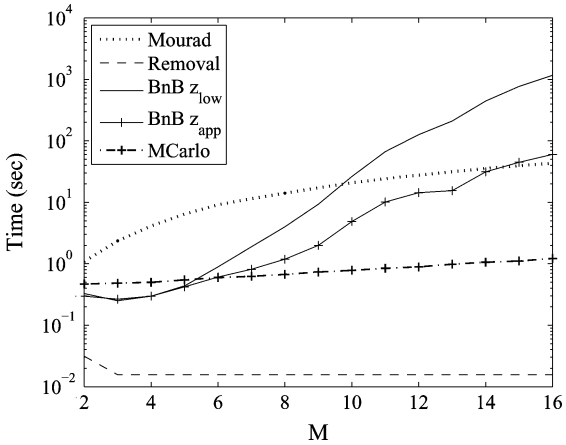

Fig. 5. Execution times for the MSCC optimization methods for WHad32.

search for comparison impractical in the best case, or impossible in the worst case. In this case, Table II shows that the number of branch and bound iterations required is generally between 2 and 3 orders of magnitude smaller than would be required for an exhaustive search. This difference would increase for a larger search space.

The platform for these tests was Matlab 7.1 on a 3.0 GHz Pentium 4 and whilst the times are platform dependent, their relative values are informative. From this graph, the branch and bound approach is faster than Mourad *et al.*'s approach for small M, although slightly slower for large M. The computational overhead of precalculating Fillin and Removal method values to produce approximate lower bounds $z_{\mathcal{I},\mathrm{app}}$ in the branch and bound approach is small enough to not be noticeable on this graph, however the reduction in total operating time compared to using a strict lower bound, $z_{\mathcal{I},\mathrm{low}}$, is evident.

IX. Conclusion

A common issue in applications using spreading sequences is the selection of which sequences from a family to employ at any given time. While this may be easily phrased as an optimization problem whose cost function is based on correlation measures of the resulting family, to date few effective methods exist to solve this problem. In this paper, two different methods to solve this problem of selecting a subset of sequences were presented.

First, a branch and bound procedure was presented. As described here the method is applicable for optimizing MSCC or maximum cross-correlation values, however the same approach may be used for any cost function. Two variations of the branch and bound approach were considered. The first variation is guaranteed to always produce the optimal subset. The second variation is more computationally efficient, but may be suboptimal in some cases. In all the cases examined here, the second approach was still optimal. The Fillin method, a computationally simple means of producing an approximation of the optimal MSCC of a subset of a given size, was also introduced. The reliability of this approximation can be improved by using the Fillin method in conjunction with the existing Removal method [7].

Results for the branch and bound procedure were produced for four different sequence families where the required subset size ranged from 2 to 16. These results were compared to three previously reported optimization methods. In contrast to the other methods used, the branch and bound procedure is guaranteed to produce the optimal result. Whilst there were some cases where previously reported methods achieved similar results to the branch and bound procedure, none of the methods tested achieved such results consistently. For small subset sizes the branch and bound approach was faster than Mourad *et al.*'s algorithm, in addition to giving smaller correlation results. With the introduction of parallel processing, the branch and bound algorithm's execution time could be further reduced.

References

[1] R. Gold, "Optimal binary sequences for spread spectrum multiplexing," *IEEE Trans. Inf. Theory*, vol. 13, no. 4, pp. 619–621, Oct. 1967.

[2] D. C. Chu, "Polyphase codes with good periodic correlation properties," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 531–532, Jul. 1972.

[3] I. Oppermann and B. S. Vucetic, "Complex spreading sequences with a wide range of correlation properties," *IEEE Trans. Commun.*, vol. 45, no. 3, pp. 365–375, Mar. 1997.

[4] M. B. Pursley, "Performance evaluation for phase-coded spread-spectrum multiple-access communication—Part I: System analysis," *IEEE Trans. Commun.*, vol. 25, no. 8, pp. 795–799, Aug. 1977.

[5] D. Mottier and D. Castelain, "A spreading sequence allocation procedure for MC-CDMA transmission systems," in *Proc. IEEE Veh. Technol. Conf.*, Boston, MA, Sep. 2000, vol. 3, pp. 1270–1275.

[6] T. M. Chan, S. Kwong, K. F. Man, and K. S. Tang, "Sequences optimization in DS/CDMA systems using genetic algorithms," in *Proc. IEEE Region 10 Int. Conf. Elect. Electron. Technol.*, Singapore, Aug. 2001, vol. 2, pp. 728–731.

[7] H. H. Dam, H.-J. Zepernick, S. Nordholm, and J. Nordberg, "Design of subsets of complex spreading sequences," in *Proc. IEEE Int. Symp. Spread Spectrum Tech. Appl.*, Prague, Czech Republic, Sep. 2002, vol. 1, pp. 194–198.

[8] A.-M. Mourad, A. Gueguen, and R. Pyndiah, "Impact of the spreading sequences on the performance of forward link MC-CDMA systems," in *Proc. IEEE Int. Symp. Spread Spectrum Tech. Appl.*, Sydney, Australia, Sep. 2004, pp. 683–687.

[9] *Optimization*, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, Eds. Amsterdam, The Netherlands: North-Holland, 1989.

[10] S. Mertens, "Exhaustive search for low-autocorrelation binary sequences," *J. Phys. A: Math. Gen.*, vol. 29, pp. 473–481, 1996.

[11] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 2001.

[12] K. H. A. Kärkkäinen, "Mean-square cross-correlation as a performance measure for department of spreading code families," in *Proc. IEEE Int. Symp. Spread Spectrum Tech. Appl.*, Tokyo, Japan, Dec. 1992, pp. 147–150.

[13] H. D. Schotten, H. Elders-Boll, and A. Busboom, "Optimization of spreading-sequences for DS-CDMA systems and frequency-selective fading channels," in *Proc. IEEE Int. Symp. Spread Spectrum Tech. Appl.*, Sun City, South Africa, Sep. 1998, vol. 1, pp. 33–37.
[14] B. J. Wysocki and T. A. Wysocki, "Modified Walsh-Hadamard sequences for DS CDMA wireless systems," *Int. J. Adapt. Control Signal Process.*, vol. 16, no. 8, pp. 589–602, Oct. 2002.
[15] Y. Shinano, M. Higaki, and R. Hirabayashi, "A generalized utility for parallel branch and bound algorithms," in *Proc. IEEE Symp. Parallel and Distrib. Process.*, San Remo, Italy, Oct. 1995, pp. 392–401.
[16] H.-J. Zepernick and A. Finger, *Pseudo Random Signal Processing, Theory and Application*. Chichester, U.K.: Wiley, 2005.
[17] P. Fan and M. Darnell, *Sequence Design for Communications Applications*, 1st ed. Somerset, U.K.: Research Studies, 1996.
[18] H. H. Dam, H.-J. Zepernick, S. Nordholm, and J. Nordberg, "Design of spreading sequences using a modified bridging method," in *Proc. IEEE Global Telecommun. Conf.*, Taipei, Taiwan, Nov. 2002, vol. 1, pp. 835–840.
[19] R. L. Frank, "Phase Coded Communication Systems," U.S. Patent 3 099 795, Jul. 1963.

Gregory Cresp received the B.Sc. degree (first class honors) in pure mathematics and the B.E. degree (first class honors) in electronic engineering from The University of Western Australia (UWA), Perth, Australia, in 2004. He received the Ph.D. degree (with distinction) from the same university in 2008.

During 2006–2007, he was a Research Assistant with the Blekinge Institute of Technology, Ronneby, Sweden. In 2008 he was an Associate Lecturer with UWA. He is currently an Electronic Engineer at MRX Technologies, Perth. His research interests include spreading sequence design and optimization methods.

Hai Huyen Dam received the B.S. degree (first class honors) and the Ph.D. degree (with distinction) from the Curtin University of Technology, Perth, Australia, in 1996 and 2001, respectively.

From 1999 to 2000, she spent one year with the Blekinge Institute of Technology, Ronneby, Sweden, as a Visiting Research Associate. From 2001 to 2007, she was with the Western Australian Telecommunications Research Institute (WATRI), Perth, as a Research Fellow and then Senior Research Fellow. Since 2006, she has been a Senior Lecturer with the Department of Mathematics and Statistics, Curtin University of Technology. Her research interests include adaptive array processing, optimization, equalization, spreading sequences design, and filter design.

Hans-Jürgen Zepernick (M'94) received the Dipl.-Ing. degree in electrical engineering from the University of Siegen, Germany, in 1987 and the Dr.-Ing. degree (with distinction) from the University of Hagen, Germany, in 1994.

From 1987 to 1989, he was with the Radio and Radar Department of Siemens AG, Munich, Germany. He is currently a Professor and the Chair of Radio Communications with the School of Engineering of the Blekinge Institute of Technology (BTH), Ronneby, Sweden. Prior to this appointment at BTH in October 2004, he held the positions of Professor of wireless communications at Curtin University of Technology, Perth, Australia; Deputy Director of the Australian Telecommunications Research Institute (ATRI); and Associate Director of the Australian Telecommunications Cooperative Research Centre (ATcrc). He was the Leader of the ATcrc's wireless program and the Leader of the center's radio transmission technology project. He is the lead author of the textbook entitled *Pseudo Random Signal Processing: Theory and Application* (Chichester, U.K.: Wiley, 2005). His research interests include radio channel characterization and modeling, sequence designs, radio transmission technologies, and mobile multimedia communications.

Dr. Zepernick served on the Board of Management of ATRI and the Executive Research Committee of the ATcrc. He has been an active participant in the European FP7 Network of Excellence *EuroFN* contributing to work packages on user-perceived quality of service.