FASTER BACKTRACKING ALGORITHMS FOR THE GENERATION OF SYMMETRY-INVARIANT PERMUTATIONS

OSCAR MORENO, JOHN RAMÍREZ, DOROTHY BOLLMAN, AND EDUSMILDO OROZCO

Received 5 March 2002

A new backtracking algorithm is developed for generating classes of permutations, that are invariant under the group G_4 of rigid motions of the square generated by reflections about the horizontal and vertical axes. Special cases give a new algorithm for generating solutions of the classical n-queens problem, as well as a new algorithm for generating Costas sequences, which are used in encoding radar and sonar signals. Parallel implementations of this latter algorithm have yielded new Costas sequences for length n, $19 \leq n \leq 24$.

1. Introduction

Backtracking is a general procedure for solving a problem by systematically generating all possible solutions. Such a process can be described by a search tree in which each node corresponds to a partial solution. Going down the tree corresponds to progress toward obtaining a complete solution. Going up the tree, that is, *backtracking*, corresponds to returning to a partial solution from which it might be hopeful to proceed forward again.

In this paper, we give a new backtracking algorithm for the generation of certain permutation matrices, that is, square matrices of 0's (or blanks) and 1's (or dots), in which each column or row contains exactly one 1. An example of such a matrix is given in Figure 2.1.

It is convenient to represent $n \times n$ permutation matrices by sequences x_i, $i = 1,2,\ldots,n$, where each x_i is the row number in which column i has its dot. Such a sequence is a permutation of the elements of $N_n = \{1,2,\ldots,n\}$, or more precisely, the sequences of images of a bijection

Journal of Applied Mathematics 2:6 (2002) 277–287
2000 Mathematics Subject Classification: 94A55, 05C30, 68W10
URL: http://dx.doi.org/10.1155/S1110757X02203022

$\pi : N_n \to N_n$. We denote the set of bijections $\pi : N_n \to N_n$ by Π_n. We will denote permutations interchangeably by either bijections π or the sequence of images $\pi(N_n) = \pi(1)\pi(2)\cdots\pi(n)$.

We develop an algorithm for generating classes of permutations of N_n, that are invariant under the group G_4 of rigid motions of the square generated by reflections about the horizontal and vertical axes. This algorithm uses backtracking, but differs from usual backtracking algorithms in two aspects.

First, our algorithm adds terms to both (not just one) sides of a sequence. An intuitive comparison between ordinary backtracking and ours is as follows. An application of ordinary backtracking would not distinguish between unwanted patterns at the beginning and at the end of a sequence. That is, a sequence could be rejected because of an unwanted pattern at the beginning, but ordinary backtracking would generate another sequence with the same pattern, say closer to the end. The idea of the algorithm presented here is that adding characters to both sides of the sequence would eliminate this redundancy, thus making the algorithm much faster.

The second source of speedup exploits invariance under the group G_4. Using this fact, it is necessary to use backtracking to generate only as few as one-fourth of the permutations of a given class and a given size.

The rest of this paper is organized as follows: in Section 2, we define the above-mentioned symmetries and develop the background necessary to present the algorithm. In Section 3, we give the algorithm and prove its correctness. In Section 4, we briefly discuss results of applying the algorithm to the generation of Costas arrays.

2. Permutation symmetries

The group of symmetries G_4 greatly reduces the size of the search space for certain sets of permutations. G_4 is a subgroup of the group of rigid motions of the square onto itself, namely, the group generated by reflections about vertical and horizontal axes, that is, reversing the columns or the rows of the permutation matrix, respectively. Thus the result of reflecting the array of Figure 2.1 about the vertical axis gives the array in Figure 2.2, and the result of reflecting it about the horizontal axis gives the array in Figure 2.3.

In the sequence representation $\sigma = \pi(N_n)$ of a permutation matrix, the reflection about the vertical axis is represented by the sequence obtained from σ by reversing its terms, and the reflection about the horizontal axis is represented by the sequence obtained from σ by replacing each term $\pi(i)$ by its complement $n+1-\pi(i)$. We denote these operations on permutation sequences by $f_1(\sigma) = \tilde{\sigma}$ and $f_2(\sigma) = -\sigma$, respectively. Clearly,

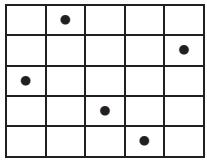

FIGURE 2.1

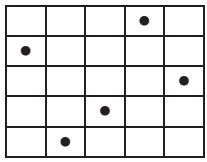

FIGURE 2.2

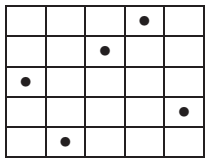

FIGURE 2.3

$G_4 = \{e, f_1, f_2, f_3 = f_1 \circ f_2\}$, where e denotes the identity operation, constitutes a group under composition $\circ$.

For any property P of permutations, that is,

$$P : \bigcup_{n \in N} \Pi_n \longrightarrow \{\text{true, false}\}, \tag{2.1}$$

where N is the set of positive integers, let U_P be the set of permutation sequences of one or more elements of N that satisfy P, and let $U_P^n = \{\sigma \in U_P \mid \sigma \text{ is of length } n\}$. We say that U_P is *symmetry-invariant* (SI) permutation if for every $\sigma = \pi(N_n) \in U_P$, we have $f_i(\sigma) \in U_P$, $i = 1,2,3$.

Examples

(1) *Costas sequences.* The condition under which a permutation $\pi \in \Pi$ is a Costas sequence can be stated in terms of the *difference triangle*. The first row of the difference triangle is formed by taking differences of adjacent terms. The second row is formed by taking differences $\pi(i+2) - \pi(i)$,

$i = 1,2,\ldots,n-2$ of distance 2 apart. In general, the jth row of the difference triangle is given by $\pi(i+j) - \pi(i)$, $i = 1,2,\ldots,n-j$. The *Costas condition* states that no row of the difference triangle should have repeated terms. The set of Costas sequences is symmetry-invariant.

(2) *n-queens sequences.* These sequences arise from the *n-queens problem*, which is the problem of placing n queens on an $n \times n$ chessboard, so that no two queens are in the same row, column, or diagonal. A permutation $\sigma = \pi(N_n)$ is an *n-queens sequence* if and only if $|\pi(i+d) - \pi(i)| \neq d$ for all $i = 1,2,\ldots,n-d$ and all $d = 1,2,\ldots,n-1$. The set of n-queens sequences is symmetry-invariant.

We call a permutation $\sigma = \pi(N_n)$ a *fixed point* if $f(\sigma) = \sigma$ for some $f \in G_4$.

Lemma 2.1. *If $\sigma = \pi(N_n)$ is fixed for some $f \in G_4$, then $f = f_3$.*

Proof. Let $\sigma = \pi(N_n)$. If $-\sigma = \sigma$, then $n+1-\pi(i) = \pi(i)$, that is, $2\pi(i) = n+1$ for all i, which contradicts that σ is a permutation. If $\tilde{\sigma} = \sigma$, then $\pi(i) = \pi(n-i+1)$ for all i, which also contradicts that σ is a permutation. Hence neither $f = f_1$ nor $f = f_2$, and so $f = f_3$.

It is important to note that not all sets of permutations have fixed points. Consider, for example, a property P for which $P(\pi)$ is true if and only if no two differences of distinct adjacent terms are equal. Such a P is called an *adjacent difference* property. An important special case is the Costas condition. □

Lemma 2.2. *If P is an adjacent difference property, then the set U_P has no fixed points.*

Proof. Suppose to the contrary that there exists a fixed point $\sigma = \pi(N_n)$ in P. Then $\sigma = -\tilde{\sigma}$, and so $\pi(m) + \pi(n-m+1) = n+1$ for all $m = 1,2,\ldots,n$. In particular, $\pi(1) + \pi(n) = n+1$ and $\pi(2) + \pi(n-1) = n+1$. Hence, $\pi(2) - \pi(1) = \pi(n) - \pi(n-1)$, which contradicts that P is an adjacent difference property.

In what follows, r denotes an arbitrary but fixed number in N_n, and q_r or q denotes the complement of r, that is, $q = n+1-r$. Now let

$$\begin{aligned}
F_n^r(P) &= \{\sigma = \pi(N_n) \in U_P^n \mid \pi(i) = r,\ \pi(j) = q,\ i \le j \wedge -\tilde{\sigma} = \sigma\},\\
A_n^r(P) &= \{\pi(N_n) \in U_P^n \mid \pi(i) = r,\ \pi(j) = q,\ 1 < i < j\\
&\qquad \wedge (n+1 > i+j \vee (n+1 = i+j \wedge \pi(m) + \pi(n-m+1) > n+1))\},\\
B_n^r(P) &= \{\pi(N_n) \in U_P^n \mid \pi(1) = r,\ \pi(n) = q \text{ and}\\
&\qquad \pi(m) + \pi(n-m+1) < n+1\},
\end{aligned} \tag{2.2}$$

where each occurrence of m denotes the least index, for which $\pi(m) + \pi(n-m+1) \neq n+1$. Finally, let

$$S_n^r(P) = A_n^r(P) \cup B_n^r(P). \tag{2.3}$$

Note that all elements of $S_n^r(P)$ have the property that r precedes q, and that the number of positions to the right of q is not less than the number of positions to the left of r.

Let

$$\begin{aligned} \tilde{S}_n^r(P) &= \{\tilde{\sigma} \mid \sigma \in S_n^r(P)\}, \\ -S_n^r(P) &= \{-\sigma \mid \sigma \in S_n^r(P)\}, \\ -\tilde{S}_n^r(P) &= \{-\tilde{\sigma} \mid \sigma \in S_n^r(P)\}. \end{aligned} \tag{2.4}$$

We will denote $S_n^r(P)$ by S_n^r when there is no ambiguity. Similarly, for $A_n^r(P)$, $B_n^r(P)$, $-S_n^r(P)$, $\tilde{S}_n^r(P)$, and $-\tilde{S}_n^r(P)$. □

Lemma 2.3. *The sets S_n^r, $\tilde{S}_n^r$, $-S_n^r$, and $-\tilde{S}_n^r$ are pairwise disjoint for any fixed P.*

Proof. First note that none of the above sets contains a fixed point. Thus, we can assume that there exists a least index m for which $\pi(m) + \pi(n+m-1) \neq n+1$. Also note that the r in any $\sigma \in S_n^r \cup -\tilde{S}_n^r$ precedes the q, whereas the r in any $\sigma \in -S_n^r \cup \tilde{S}_n^r$ follows q. Hence

$$\begin{aligned} S_n^r \cap -S_n^r &= \phi, \\ S_n^r \cap \tilde{S}_n^r &= \phi, \\ -\tilde{S}_n^r \cap -S_n^r &= \phi, \\ -\tilde{S}_n^r \cap \tilde{S}_n^r &= \phi. \end{aligned} \tag{2.5}$$

Now suppose that $\sigma = \pi(n) \in S_n^r \cap -\tilde{S}_n^r$. Since $\sigma \in S_n^r$, the number of positions to the right of q is $n - j \geq 0$. If $n - j = 0$, then $\sigma \in B_n^r$, and so $\pi(m) + \pi(n-m+1) < n+1$;

$$n+1-\pi(n-m+1) + n+1-\pi(m) > n+1, \tag{2.6}$$

which contradicts that $\sigma \in -\tilde{S}_n^r$.

Thus $n > j$, and so $\sigma \in A_n^r$. In this case, $n - j \geq i - 1 > 0$. But since $\sigma \in -\tilde{S}_n^r$, we also have that the number of positions in $-\tilde{\sigma}$ to the right of q is greater than or equal to the number of positions to the left of r. That is, $n - (n+1-i) = i - 1 \geq n+1-j-1 = n-j$. Thus, $n - j = i - 1$ or $n+1 = i+j$, and so $-\tilde{\sigma} \in A_n^r$. Hence $n+1-\pi(n-m+1) + n+1-\pi(m) > n+1$, and so

$\pi(n-m+1)+\pi(m)<n+1$, which contradicts that $\sigma \in A_n^r$. Thus,

$$S_n^r \cap -\tilde{S}_n^r = \phi. \tag{2.7}$$

Finally, suppose that $\sigma \in -S_n^r \cap \tilde{S}_n^r \neq \phi$. Then $\tilde{\sigma} \in -\tilde{S}_n^r$ and $\tilde{\sigma} \in S_n^r$, that is, $\tilde{\sigma} \in -\tilde{S}_n^r \cap S_n^r$, a contradiction. Hence,

$$-S_n^r \cap \tilde{S}_n^r = \phi. \tag{2.8}$$

□

Lemma 2.4. *The sets S_n^r, $\tilde{S}_n^r$, $-S_n^r$, and $-\tilde{S}_n^r$ are equivalent, that is, have the same number of elements.*

Proof. The function $f : S_n^r \to \tilde{S}_n^r$ defined by $f(\sigma)=\tilde{\sigma}$ is a bijection. Hence, S_n^r and $\tilde{S}_n^r$ are equivalent, written $S_n^r \equiv \tilde{S}_n^r$. Similarly, $S_n^r \equiv -S_n^r$ and $S_n^r \equiv -\tilde{S}_n^r$. Hence $S_n^r \equiv \tilde{S}_n^r \equiv -S_n^r \equiv -\tilde{S}_n^r$. □

Theorem 2.5. *For any symmetry-invariant property P,*

$$U_P^n = S_n^r(P) \cup \tilde{S}_n^r(P) \cup -S_n^r(P) \cup -\tilde{S}_n^r(P) \cup F_n^r(P) \cup -F_n^r(P). \tag{2.9}$$

Proof. Clearly,

$$S_n^r \cup \tilde{S}_n^r \cup -S_n^r \cup -\tilde{S}_n^r \cup F_n^r \cup -F_n^r \subset U_P^n. \tag{2.10}$$

Now let $\sigma=\pi(N_n)\in U_P^n$. Then either $\pi(i)=r$ precedes $\pi(j)=q$ in σ or not. Suppose it does.

Case 1. Suppose that $i=1$. If $j<n$, then $\sigma \in S_n^r \cup F_n^r$. Now suppose $j=n$. Then either $\sigma \in F_n^r$ or there exists a least index m for which $\pi(m)+\pi(n-m+1)\neq n+1$. Suppose $\sigma \notin F_n^r$. If $\pi(m)+\pi(n-m+1)<n+1$, then $\sigma \in S_n^r$. If $\pi(m)+\pi(n-m+1)>n+1$, then $n+1-\pi(m)+n+1-\pi(n-m+1)<n+1$ and so $-\tilde{\sigma} \in S_n^r$, that is, $\sigma \in -\tilde{S}_n^r$.

Case 2. Suppose that $i>1$. If $n-j>i-1$, then $\sigma \in S_n^r$. Suppose $n-j=i-1$. Then either $\sigma \in F_n^r(P)$, or there exists a least m such that $\pi(m)+\pi(n-m+1)\neq n+1$. Suppose $\sigma \notin F_n^r(P)$. If $\pi(m)+\pi(n-m+1)>n+1$, then $\sigma \in S_n^r$. If $\pi(m)+\pi(n-m+1)<n+1$, then $n+1-\pi(m)+n+1-\pi(n-m+1)>n+1$, and so $\sigma \in -S_n^r$. Now, suppose that $n-j<i-1$. The number of positions to the left of r in $-\tilde{\sigma}$ is $n+1-j-1=n-j$, and the number of positions to the right of q is $n-(n+1-i)=i-1$. Thus, $-\tilde{\sigma} \in S_n^r$, and so $\sigma \in -\tilde{S}_n^r$.

We have thus shown that for each $\sigma \in U_P^n$, where r precedes q, either $\sigma \in S_n^r$ or $\sigma \in -\tilde{S}_n^r$. Now consider $\sigma \in U_P^n$ in which q precedes r. Then either $\tilde{\sigma} \in S_n^r$ or $\tilde{\sigma} \in -\tilde{S}_n^r$. Hence either $\sigma \in \tilde{S}_n^r$ or $\sigma \in -S_n^r$. □

Corollary 2.6. *If P is an adjacent difference property, then*

$$U_P^n = S_n^r(P) \cup \tilde{S}_n^r(P) \cup -S_n^r(P) \cup -\tilde{S}_n^r(P) \tag{2.11}$$

and the cardinality of each set on the right is exactly one-fourth of the cardinality of U_P^n.

Proof. By Lemma 2.2, $F_n^r(P)$ and $-F_n^r(P)$ are empty and by Lemmas 2.3 and 2.4, the sets on the right are pairwise disjoint and equivalent. □

3. A faster algorithm for generating permutations with the SI property

An adaptation of a standard backtracking procedure [2], gives a simple algorithm for generating all permutations in Π_n satisfying a property P. This algorithm, which we denote by $A_0(P)$ or simply A_0, consists of traversing the following labeled tree $T_0 = T_0(P)$ in preorder:

(i) the root of T_0 is marked and labeled with the empty sequence;
(ii) the children of any marked node v labeled σ are labeled $\sigma s_1, \ldots, \sigma s_k$, where $s_1 < \cdots < s_k$ are the elements of N_n that do not appear in σ or in any ancestor of σ;
(iii) a node labeled σ is marked if and only if $\sigma \in U_P$. Unmarked nodes are leaves.

Clearly, T_0 has depth at most $n-1$. The set U_P^n consists of the labels of the marked leaves of depth $n-1$.

When P is an SI property, a much more efficient algorithm can be obtained by adjoining terms to both sides of the sequences, not just one side as in A_0. This new algorithm, called $A_1(P)$ or simply A_1, consists of traversing the tree T_1 in preorder, and writing the labels of marked leaves with depth $n-1$ as well as their appropriate symmetries. For any $r \in N_n$, the tree $T_1 = T_1(P)$ is defined as follows:

(a) every node of T_1 is colored either red or yellow, is either marked or unmarked, and has a label consisting of a permutation sequence of some or all members of N_n;
(b) the root of T_1 is red, marked, and has label r;
(c) if v is a red, marked node with label σ, then v has red children labeled $\sigma s_1, \sigma s_2, \ldots, \sigma s_k$, where $s_1 < s_2 < \cdots < s_k$ are the elements of N_n that do not appear in σ or in the label of any ancestor of v. If in addition σ contains q, then v has yellow children labeled $s_1\sigma, s_2\sigma, \ldots, s_k\sigma$;
(d) a yellow, marked node with label σ has yellow children that are labeled as in (c);

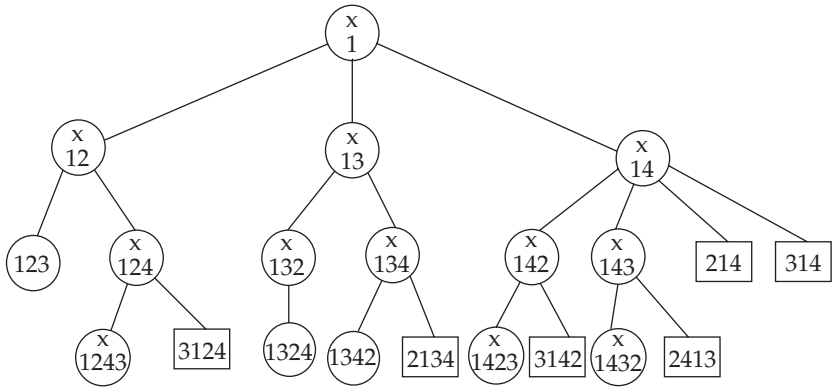

FIGURE 3.1

(e) a red node with label σs_i is marked if and only if $\sigma s_i \in U_P$ unless $i = k$ and $s_k = q$, σs_k is at depth $n-1$, and $\pi(m) + \pi(n-m+1) \geq n+1$, where $\sigma = \pi(N_n)$ and m is the least index for which $\pi(m) + \pi(n-m+1) \neq n+1$;

(f) a yellow child with label σ is marked if and only if $\sigma \in U_P$ and the number of elements to the right of q is greater than the number of elements to the left of r or $n-j = i-1$ and either $\sigma \in F_n^r$ or $\pi(m) + \pi(n-m+1) < n+1$, where $\sigma = \pi(N_n)$ and m is the least index for which $\pi(m) + \pi(n-m+1) \neq n+1$.

Suppose, for example, that $n = 4$, $r = 1$, $q = 4$, and P is the "adjacent difference" property, that is, for any $\pi \in \pi(N_n)$, $P(\pi)$ is true if and only if no two differences of distinct adjacent terms are equal. Then $T_1(P)$ is as follows (Figure 3.1), where red nodes are denoted by circles, yellow nodes by rectangles, and marked nodes are indicated by an "x" written within the circle or rectangle.

THEOREM 3.1. *For any SI property P,*

$$S_n^r \cup F_n^r = \{\sigma \mid \sigma \text{ is a label of a marked leaf of depth } n-1 \text{ of } T_1(P)\} \tag{3.1}$$

and these labels are nonrepeating.

Proof. A node v with label σ of T_1 is a leaf of depth $n-1$ if and only if there is a path from the root of T_1 to v in which the label of each node on each level k on the path is a sequence σ' of length k for which $\sigma' \in U_P$, and $\sigma = \tau_1\sigma'\tau_2$ for some τ_1 and τ_2. A red child adds an element to the right of the label of its parent, and a yellow child adds an element to the left of the label of its parent. Furthermore, each level k contains all sequences of length k that belong to $U_P \cup F_n^r$.

A yellow node can have only yellow children, and a red node can have a yellow child only when its label contains q. Letting σ_i denote the ith term added in order to form σ, we have that σ is the label of leaf v of T_1 of depth $n-1$ if and only if it is of the form

$$\sigma_n \cdots \sigma_{n-i+2}\sigma_1\sigma_2 \cdots \sigma_{n-i+1}, \tag{3.2}$$

where $\sigma_1 = r$, and $\sigma_j = q$ for some j, $i < j \leq n-i+1$, that is, if and only if for some permutation π,

$$\sigma = \pi(1)\cdots\pi(i) = r\cdots\pi(j) = q\cdots\pi(n). \tag{3.3}$$

Furthermore v is marked if and only if

(1) it is red, that is, $i = 1$ and hence either $n \neq j$ and $n-j > i-1$ or $n = j$ and $\pi(m)+\pi(n-m+1) < n+1$ where m is the least index for which $\pi(m)+\pi(n-m+1) \neq n+1$ or
(2) it is yellow, that is, $n-j > i-1$ or $n-j = i-1$ and $\pi(m)+\pi(n-m-1) > n+1$ and m is defined as before.

But these are exactly the conditions under which $\sigma \in S_n^r \cup F_n^r$ and hence equality of the two sets follows. Finally, note that for any node of T_1 with label σ, there is only one path from the root to v, and hence the marked leaves of T_1 of depth $n-1$ are distinct. □

Now algorithm $A_1(P)$ is obtained by traversing $T_1(P)$ in preorder. For each marked leaf of depth $n-1$, with label, say σ, if σ is not a fixed point, we print σ, $-\sigma$, $\tilde{\sigma}$, and $-\tilde{\sigma}$. If σ is a fixed point, we print only σ and $-\sigma$. If P is an adjacent difference property, we need not test for fixed points, but simply print σ, $-\sigma$, $\tilde{\sigma}$, and $-\tilde{\sigma}$ for each marked leaf of depth $n-1$ with label σ.

To achieve parallelism in algorithm $A_1(P)$ we use the "manager-worker" technique. The manager generates the upper portion of the tree $T_1(P)$, by generating those nodes of fixed depth d, for some d. The manager dynamically passes each of these sequences of length d to an idle worker, who in turn continues to search for sequences with property P that contain the fixed subsequence of length d. The worker returns all such sequences with property P to the manager and then waits for another subsequence of length d. The manager continues to compute subsequences of length d until no more such subsequences exist.

4. Costas sequences

The algorithm $A_1(P)$ of Section 3 has been presented in a very general setting. To specialize it to any particular class of symmetry-invariant permutations, one need only specify criteria for membership in the set U_P

Table 4.1

n	$C(n)$
19	10240
20	6464
21	3536
22	2052
23	872
24	≥ 8

for the given property P. Special cases of interest are for the Costas property and the n-queens property Q.

This last section is devoted to describing the work done in applying the algorithm to the generation of Costas sequences, which are used in encoding radar and sonar signals. In this case, it is necessary to generate only one-fourth of U_C since C is an adjacent difference property.

We have implemented $A_1(C)$ on several platforms over a period of several years, yielding new Costas sequences for various values of n. Previously, Silverman et al. [3] were able to compute all Costas sequences of size $n \leq 18$, but abandoned the search for larger sequences after predicting that the case $n = 19$ would require more than one year of computer time.

The first two authors of this work initially implemented $A_1(C)$ in OCCAM on an eight T800 transputer system mounted on an IBM PC-AT. With this (MIMD) implementation, Costas sequences were obtained for $n \leq 20$. Subsequently, they implemented it in C on an Intel Paragon, obtaining all Costas sequences for $n = 21$, $n = 22$, and $n = 23$.

The second two authors implemented $A_1(P_C)$ in C/MPI on a Super Sparc cluster of 32 workstations, verifying all of the above results. This latter implementation also yielded a new Costas sequence for size $n = 24$, namely, $X = (18,16,10,22,13,24,6,1,2,15,3,5,11,20,23,19,12,4,9,17,14,21,8,7)$. Another known Costas sequence of size 24 that results from a construction described in [1] is $Y = (6,2,4,7,20,21,3,8,18,15,14,12,5,23,17,24,10,19,9,13,1,16,11,22)$. The symmetries of the group G_4 applied to X and Y yield 6 more distinct Costas sequences. We can therefore conclude that there are at least 8 Costas sequences of size 24.

Table 4.1 summarizes the contributions of algorithm A_1 to known values of $C(n)$, the number of Costas arrays of size n.

Even though algorithm A_1 is a considerable improvement over A_0, it still has exponential computational complexity. Nevertheless, implementations of A_1 have been optimal. In each of the implementations we chose d, so that the number of nodes at depth d in T_1 is several times

greater than the number of processors available. This insures that each of the processors is kept busy. It turns out in fact that our implementations achieved speedups close to p where p is the number of processors. Furthermore, the only communications necessary are between master and slave and for fixed d, the number of these is polynomial in n. Thus, the communication to computation ratio tends to zero as $n \to \infty$.

Acknowledgment

This research was supported by grants from the Office of Naval Research (ONR) N00014-96-1-1192 and National Science Foundation (NSF) EIA9977071.

References

[1] S. Golomb and H. Taylor, *Constructions and properties of Costas arrays*, Proc. IEEE **72** (1984), 1143–1163.

[2] E. Horowitz, S. Sahni, and S. Rajasekaran, *Computer Algorithms C++*, Computer Science Press, New York, 1996.

[3] J. Silverman, V. Vickers, and J. Mooney, *On the number of Costas arrays as a function of array size*, Proc. IEEE **76** (1988), 851–853.

Oscar Moreno: Department of Mathematics and Computer Science, University of Puerto Rico, Rio Piedras, PR 00931-3355, USA

John Ramírez: The Graduate School and University Center, The City University of New York, 365 Fifth Avenue, New York, NY 10016-4309, USA

Dorothy Bollman: Department of Mathematics, University of Puerto Rico, Mayaguez, PR 00681-9018, USA
E-mail address: bollman@cs.uprm.edu

Edusmildo Orozco: Department of Mathematics, University of Puerto Rico, Mayaguez, PR 00681-9018, USA