Generating Permutations With k-Differences

Carla D. Savage*
Department of Computer Science
North Carolina State University
Raleigh, North Carolina 27695
cds@cscadm.ncsu.edu

May 14, 1989

Abstract

Given (n, k) with $n \geq k \geq 2$ and $k \neq 3$, we show how to generate all permutations of n objects (each exactly once) so that successive permutations differ in exactly k positions, as do the first and last permutations. This solution generalizes known results for the specific cases where $k = 2$ and $k = n$. When $k = 3$, we show that it is possible to generate all even (odd) permutations of n objects so that successive permutations differ in exactly 3 positions.

Keywords. permutations, Gray codes, combinatorial algorithms, Cayley graphs, Hamilton cycles

AMS(MOS) subject classifications. 05A15, 05C45

1 Introduction

The problem of generating permutations of n distinct objects is of fundamental importance both in Computer Science and in Combinatorics. Many practical problems require for their solution a sampling of random permutations or, worse, a search through all $n!$ permutations. In order for such a search to be possible, even for moderate size n, permutation generation methods must be extremely efficient. A survey of efficient techniques is presented in [24].

A problem which was initially motivated by practical concerns is that of generating permutations by *transpositions*, that is, listing all $n!$ permutations, each exactly once, in such a way that successive permutations on the list differ only by the exchange of two

*This work was supported in part by the National Science Foundation Grant No. CCR8906500

elements. This was shown to be possible in several papers, including [26], [1], [2], and [9], which are described in [24]. In fact, it is possible even if the two elements exchanged are required to be in adjacent positions ([13], [25]). It is interesting to note that in the Wells-Boothroyd-Heap algorithms, the last permutation differs from the first permutation by a transposition only when n is odd, so the scheme is not "cyclic". The Johnson-Trotter scheme for generating permutations by adjacent transpositions is cyclic and is shown in Figure 1 for $n = 1, 2, 3, 4$.

A new twist on the problem is to generate all permutations by transpositions so that every other transposition is of the elements in the first two positions. See Figure 2 for an example taken from [22]. Recently, it has been shown that this is possible even when the transpositions are required to be of adjacent positions [23].

A contrary approach to the problem is to require that permutations be listed so that each one differs from its predecessor in *every* position [20], that is, by a "derangement". The existence of such a list when $n \neq 3$ was established in [17] using Jackson's theorem [12] and a constructive solution was presented in [7]. A simpler construction, ascribed to Lynn Yarbrough, is discussed in [21] and is shown in Figure 3 for $n = 4$.

To generalize the problems of listing permutations to differ, on the one hand, in every position and, on the other hand, in only two positions, we consider the following. Given n and k satisfying $n \geq k \geq 2$, is it possible to list all permutations so that successive permutations differ in exactly k positions? In this paper, we show by construction that the answer is yes, unless $k = 3$. (We have learned that Thomas Putnam, working independently, has recently solved the same problem [19]). When $k = 3$, two permutations which differ in exactly three positions, in fact, differ by a product of two 2-cycles, so that successive permutations must have the same parity. Our motivation for studying this problem of permutation generation with k-differences is clearly not for efficient permutation generation, which would be better handled by techniques from [24]. Rather, we found the problem mathematically interesting, particularly in view of its combinatorial connections which are discussed below.

Recently, permutation generation has been studied from a combinatorial perspective, due in part to its ties with some interesting graph theoretic problems. In this context, it is of interest to specify some set S of "changes" and to then ask if it is possible to list all permutations so that each permutation differs from its predecessor on the list only by a change from the set S. More precisely, for fixed n, let $G[S]$ be the directed graph whose vertices are the $n!$ permutations of $1 \ldots n$. For permutations p and q, there will be an edge in $G[S]$ from p to q if and only if q can be obtained from p by a change specified in S. A question of interest, then, is whether or not $G[S]$ has a Hamiltonian path or cycle. For example, by results cited above, $G[S]$ is Hamiltonian when S allows only adjacent transpositions ([13], [25]) or when S allows only derangements and $n \neq 3$ ([7], [17]). As another example, it is shown in [15] that for a set S of transpositions (of specific positions), $G[S]$ is Hamiltonian if the elements of S form a basis for the symmetric group.

A well-known open problem in graph theory, due to Lovász [16], is to determine whether every undirected, connected, vertex transitive graph has a Hamilton path. In the case of permutation graphs, there are many interesting sets S for which $G[S]$ is connected, vertex transitive, and symmetric (and can therefore be regarded as undirected.) In particular, for $n \geq k \geq 2$, if S requires only that permutations differ in exactly k positions, the graph $G[S]$ is symmetric, vertex transitive, and, unless $k = 3$, connected. Thus, with our result in this paper that $G[S]$ is Hamiltonian, we exhibit a new class of graphs which will not yield a counterexample to the conjecture of Lovász.

In fact, $G[S]$ is a Cayley graph and it is an open question whether even every (undirected) Cayley graph is Hamiltonian. No general results on Cayley graphs apply to show that the k-differences graph is Hamiltonian. A related problem due to Wilf [28] is whether $G[S]$ is Hamiltonian when S allows only (1) a swap of the first two positions (i.e., $123 \ldots n$ to $213 \ldots n$) or (2) a cyclic shift one position left or right of all positions (i.e., $123 \ldots n$ to $23 \ldots n1$ or to $n123 \ldots n-1$). In work to appear in his Ph.D. thesis, Chris Compton has shown that this graph is Hamiltonian [5]. In the case of the alternating group, with generating set the 3-cycles $S = \{(1\ 2\ n), (1\ 3\ n), \ldots, (1\ n-1\ n)\}$, Gould and Roth have shown that the undirected Cayley graph is Hamiltonian [11].

Finally, we note that although the existence of a listing of permutations with k-differences for $k = n$ is guaranteed by Jackson's theorem [12] (as shown in [17] and [27]), the theorem does not apply to establish existence when $k < n - 1$. Jackson's result is that every 2-connected, d-regular graph of at most $3d$ vertices is Hamiltonian. In the case $k = n$, the degree of each vertex in $G[S]$ is the number of derangements of $1 \ldots n$, which is (see, e.g., [8], pp. 9-10)

$$n!(1 - 1/1! + 1/2! - 1/3! + \ldots + (-1)^n/n!).$$

For $n \geq 3$ this is at least

$$n!(1 - 1/1! + 1/2! - 1/3!) = n!/3.$$

So, at least the vertex degree is large enough for Jackson's theorem to apply. However, this is not the case for arbitrary k since the degree will be

$$(n!/(n-k)!)(1 - 1/1! + 1/2! - 1/3! + \ldots + (-1)^k/k!)$$

and for $n - 2 \geq k \geq 2$ this is at most

$$(n!/2)(1 - 1/1! + 1/2!) = n!/4 < n!/3.$$

2 Listing 2-Element Subsets

In [7], permutations of $1 \ldots n$ were listed by derangements using a recursive listing of the permutations of $1 \ldots n-2$ by derangements, together with a listing of the 2-element subsets of $[n]$. Our strategy is to generalize this construction and several new ideas are required.

Our technique for listing permutations of n objects with k-differences combines a recursive listing of the permutations of $n - 2$ objects with $k - 2$-differences together with a particular way of listing the 2-sets of $[n]$. As is common, we will use "2-sets" to denote "2-element subsets".

For a subset S of $[n]$, let $(1 \ldots n) \setminus S$ denote the unique permutation of the elements of $[n] \setminus S$ in which the elements appear in sorted order. For example,

$$(1 \ldots 5) \setminus \{2\} = 1345$$

$$(1 \ldots 6) \setminus \{3, 5\} = 1246.$$

For $x, y, z \in [n]$, note that although the *sets* $[n] \setminus \{x, y\}$ and $[n] \setminus \{x, z\}$ differ in only one element, the permutations $(1 \ldots n) \setminus \{x, y\}$ and $(1 \ldots n) \setminus \{x, z\}$ could differ in many positions. For example,

$$(1 \ldots 7) \setminus \{1, 7\} = 23456$$

whereas,

$$(1 \ldots 7) \setminus \{6, 7\} = 12345.$$

For the construction of this paper, we would like to list the 2-sets of $[n]$ in such a way that the following two conditions hold:

Condition 1. If $\{u, v\}$ follows $\{x, y\}$ on the list of 2-sets, then the permutations $(1 \dots n) \setminus \{u, v\}$ and $(1 \dots n) \setminus \{x, y\}$ differ in exactly one position. (In particular, this means that successive 2-sets differ in exactly one element.)

Condition 2. The first 2-set is $\{1, 2\}$ and the last is $\{3, 4\}$.

In order to satisfy these properties, we will construct a certain graph whose vertices are the 2-sets of $[n]$ and in which adjacent vertices satisfy condition (1). We then observe that this graph has a Hamilton path from $\{1, 2\}$ to $\{3, 4\}$, thereby giving a listing of 2-sets of $[n]$ satisfying conditions (1) and (2).

For $n \geq 3$ let G_n be the undirected graph whose vertices are the 2-element subsets of $[n]$ and whose edges are defined as follows. (See Figure 4.)

For $x < y$, $\{x, y\}$ is adjacent to

1. $\{x, y+1\}$, if $1 \leq x < y \leq n-1$ (vertical edges)
2. $\{x+1, y\}$, if $1 \leq x \leq y-2 \leq n-2$ (horizontal edges)
3. $\{x+1, y+1\}$, if $1 \leq x = y-1 \leq n-2$ (diagonal edges)

Lemma 1 *If $\{x, y\}$ and $\{x, z\}$ are adjacent vertices of G_n, then the permutations $(1 \dots n) \setminus \{x, y\}$ and $(1 \dots n) \setminus \{x, z\}$ differ only in one position.*

Proof. Without loss of generality, assume $y < z$. Then by definition of G_n, either

1. $z = y + 1$ (horizontal or vertical edge) or
2. $z = x + 1 = y + 2$ (diagonal edge).

In case (1), if $x < y$,

$$(1 \dots n) \setminus \{x, y\} = 1 \dots (x-1)\ (x+1) \dots (y-1)\ z\ (y+2) \dots n$$

$$(1 \dots n) \setminus \{x, z\} = 1 \dots (x-1)\ (x+1) \dots (y-1)\ y\ (y+2) \dots n$$

Similarly, if $x > y$, the sequences differ in only one position. In case (2),

$$(1 \ldots n) \setminus \{x, y\} = 1 \ldots (y-1)\ z\ (y+3) \ldots n$$

$$(1 \ldots n) \setminus \{x, z\} = 1 \ldots (y-1)\ y\ (y+3) \ldots n$$

□

Lemma 2 *For $n \geq 4$ there is always a Hamilton path in G_n which starts at $\{1, 2\}$ and ends at $\{3, 4\}$.*

Proof. Construction of such a path depends on the parity of n. See Figure 5 for details. □

There are many algorithms in the literature for generating the k-sets of an n-set. The revolving door algorithm in [18] generates all k-sets so that successive sets differ in only one element. If we used this algorithm to generate 2-sets, neither property (1) nor (2) would be satisfied.

Other algorithms ([6], [3], [10]) can in some cases generate k-sets with the *minimal change property*, that is, so that successive sets differ in exactly one element and this element has either increased or decreased by 1. For $k = 2$, this would amount to using only horizontal and vertical edges in G_n. However, it is known that there is *no* listing of k-sets with the minimal change property if and only if either n is even and k is odd or if k is one of 0, 1, n, $n-1$.

More closely related to our listing is the *strong minimal change* (SMC) *property* [6]. That is, k-sets are generated so that if a k-set is represented as a sorted sequence, successive k-sets differ in only one position. This is what property (1) requires: a listing of the $n-2$-sets of $[n]$, with the SMC property, starting with $34 \ldots n$ and ending at $1256 \ldots n$. We cannot apply directly the algorithm of [6] only because the algorithm of that paper starts with $12 \ldots n-2$ and ends with $34 \ldots n$. Note that although the sets {1, 2} and {1, 4} are adjacent in SMC order, their complements in $\{1, \ldots, 4\}$ are not. Thus, a listing of 2-sets in SMC order does not necessarily give a corresponding listing of the complementary $k-2$-sets with the same property.

3 The Basis Cases

As stated in Section 2, our technique for listing permutations of n objects with k-differences requires that it be possible to recursively list permutations of $n-2$ objects with $(k-2)$-

differences. For even values of k, the case $k = 2$ will serve as the basis of the recursion. For the case $k = 2$, any method for generating permutations by transpositions will suffice, as long as the first permutation differs from the last in only two positions. This property will be satisfied by the Johnson-Trotter scheme for listing all permutations by adjacent transpositions, which works as follows. For $n = 2$, the listing is 12, 21. For $n > 2$, make n copies of each permutation on the list for $n - 1$. In each group of n identical permutations, insert "n" into each possible position going from position n down to position 1 on odd numbered groups and from position 1 up to position n on even numbered groups. The lists for $n = 2, 3, 4$ are shown in Figure 1.

For odd k, the case $k = 3$ cannot serve as the basis since it is not possible to generate all permutations with 3-differences. Two permutations which differ in three locations differ by a 3-cycle, which is an even permutation. Thus, from any given permutation, π, any permutation obtained from π by applying a sequence of 3-cycles must have the same parity as π.

However, we show that it *is* possible to list all *even* (*odd*) permutations of n items with 3-differences. For our recursive construction in Section 4, this is sufficient. We show there how to construct a list of all permutations of n objects with 5-differences from listings of all even permutations of $n - 2$ elements with 3-differences.

We describe now a way to list all even permutations of n items (for $n \geq 3$) with 3-differences. In fact, we will require that the list start with the permutation $1234 \ldots n$ and end with $3124 \ldots n$, so the first and last elements will also differ in three positions.

If $n = 3$, the list is 123, 231, 312. For $n > 3$ and for $i = 1 \ldots n$, let group i consist of the set of even permutations of $1 \ldots n$ which have i in their first position. The list of even permutations of $1 \ldots n$ will consist of listings of the groups $1, \ldots, n$ in the following order:

$$\text{group } 1, \text{ group } n, \text{ group } n-1, \ldots, \text{ group } 4, \text{ group } 2, \text{ group } 3.$$

Within each group i, we list recursively the even permutations of $(1 \ldots n) \setminus \{i\}$ with 3-differences, and prefix each with "i". It remains to specify the first element of each group i. The first element of group 1 is $12 \ldots n$. The first element of group 3 is $32415 \ldots n$, so the list will always end with $31245 \ldots n$. For groups $n, n-1, \ldots, 4$, the first element is obtained from the last element of the preceding group by a cyclic permutation of the following three elements: 2, i and the element in position 1. Precisely, the element in position 1 moves to the location containing element 2, element 2 moves to the location containing element i and element i moves to position 1. Element 2 is acting as a "place holder" for the element in position 1. (See Figure 6.) Finally, the first element of group 2 is $23145\ldots n$ if n is divisible by 3 and $24315\ldots n$ otherwise. (See Figure 6.)

This clearly gives all even permutations of $1\ldots n$ so that it remains to verify that successive permutations differ in exactly 3 positions. This is evident within each group. Between groups $1, n, \ldots, 4$ this follows since the first permutation of each group was defined by permuting 3 positions in the last permutation of the preceding group. To check between groups 2 and 3, note that group 2 ends in either $24315\ldots n$ or $21435\ldots n$ and that each of these differs in 3 positions from $32415\ldots n$. Finally, to check between groups 4 and 2, note that group 4 always ends in $4(213)^n\ldots n$ where $(213)^n$ denotes a right cyclic shift n times of 213.

In summary, we have described a procedure which, given distinct symbols $x_1\ldots x_n$, can be used to generate all even permutations of $x_1\ldots x_n$ with 3-differences. The list begins with the permutation $x_1\ldots x_n$ and ends with the permutation $x_3x_1x_2x_4\ldots x_n$. Note that if, in every permutation on the list, the elements x_1 and x_2 were exchanged, the result would be a listing of all odd permutations of $1\ldots n$ with 3-differences, beginning with the permutation $x_2x_1x_3x_4\ldots x_n$ and ending with $x_3x_2x_1x_4\ldots x_n$.

4 The Recursive Construction

In this section we describe the recursive step of the construction. Given (n,k), where $n \geq k \geq 2$ and $k \neq 3$, the goal is to list all permutations of n items so that successive permutations differ in exactly k positions. As a preliminary, note that if it is possible to generate all permutations with k-differences so that the first and last permutations differ in k positions, then we can arrange for the first and last to differ in *any* k positions. In particular, suppose there is an algorithm, $A(x_1\ldots x_n, k)$ to list all permutations of $x_1\ldots x_n$ with k-differences, starting with $x_1\ldots x_n$, so that the first and last permutations differ exactly in the first k positions. Then for any i, $1 \leq i \leq n$, there is an algorithm $B(x_1\ldots x_n, k, i)$ which does the same, except that the first and last permutations differ exactly in the k positions *beginning with position* i, allowing wrap-around. Algorithm B works as follows: apply algorithm A to $x_ix_{i+1}\ldots x_nx_1\ldots x_{i-1}$, but then rotate each permutation generated by A to the right $i-1$ positions. See Figure 7.

Now, for the recursive construction, the list $LIST$ of permutations of $1 \ldots n$ with k-differences is composed of two parts:

$$LIST = LISTA, LISTB$$

(Following [18], a comma is used to denote the concatenation of lists.) $LISTB$ will be obtained from $LISTA$ by swapping the first two positions in every permutation on $LISTA$.

For $m = \binom{n}{2}$, let $T_1, \ldots, T_m$ be a listing of all 2-sets of $[n]$ which satisfies properties (1) and (2) of Section 2. That is, $T_1 = \{1, 2\}$, $T_m = \{3, 4\}$ and for $1 \leq j < m$, the permutations $(1 \ldots n) \setminus T_j$ and $(1 \ldots n) \setminus T_{j+1}$ differ in exactly one position.

To describe $LISTA$, first consider the case $k \geq 4$, $k \neq 5$. Corresponding to each 2-set $T_i = \{x_i, y_i\}$ will be a sublist L_i. L_i consists of a recursive listing of the permutations of $(1 \ldots n) \setminus \{x_i, y_i\}$ with $k-2$-differences, each permutation prefixed alternately with $x_i y_i$, $y_i x_i$. In order to complete the specification of L_i, two things must be determined, one of which depends on L_{i-1} and the other on T_{i+1}.

First, which of $x_i y_i$, $y_i x_i$ should be the prefix of the first permutation on L_i? This is decided so as to force the last element of L_{i-1} and the first element of L_i to differ in the first two positions. See L_1, L_2, L_3 in Figure 8 when $(n, k) = (6, 4)$. In this case, $T_1 = \{1, 2\}$, $T_2 = \{2, 3\}$, and $T_3 = \{1, 3\}$.

Second, when generating recursively the permutations of $(1 \ldots n) \setminus T_i$ with $k-2$-differences, how should the last permutation differ from the first? For $i < m$, we will do this in such a way to ensure that $\text{last}(L_i)$ and $\text{first}(L_{i+1})$ differ in exactly k positions, where $\text{first}(L)$ and $\text{last}(L)$ are used to denote the first and last elements, respectively of a list, L. By definition of the T_i, the two permutations

$$p_i = (1 \ldots n) \setminus T_i$$

and

$$p_{i+1} = (1 \ldots n) \setminus T_{i+1}$$

differ in exactly one position j (and position j of p_{i+1} contains an element of T_i.) So, assume by induction that we can generate recursively the permutations of $(1 \ldots n) \setminus T_i$ with $k-2$ differences so that the first and last permutations differ in $k-2$ positions. We would like these $k-2$ positions to include position j. To be specific, we will arrange for the first and last permutations to differ in the $k-2$ positions beginning with position j. For, if q is a permutation of $(1 \ldots n) \setminus T_i$ such that

1. q and p_i differ in $k-2$ positions and

2. j is one of the positions in which q and p_i differ,

then also q and p_{i+1} differ in exactly $k-2$ positions. It follows then that last(L_i) and first(L_{i+1}) will differ in exactly k positions.

As an example, when $(n,k) = (6,4)$, note from Figure 5 that $T_3 = \{1, 3\}$ and $T_4 = \{1, 4\}$. From Figure 8, first$(L_3) = 312456$. Thus, last(L_3) has prefix 13, forcing first(L_4) to have prefix 41. Then, first$(L_4) = 412356$. To construct all of L_3, we need to generate the permutations of $p_3 = 2456$ with 2-differences. Since $p_4 = 2356$, p_3 and p_4 differ in position 2, we must arrange for the last permutation of 2456 to differ from 2456 in the two positions beginning with position 2. The result is shown in the first column of Figure 9, which describes all of $LISTA$, as well as $LISTB$ for the case $(n,k) = (6,4)$.

In the case $k=5$, the construction differs only as follows. $LISTA$ is composed of two sublists,

$$LISTA = LISTA1, LISTA2.$$

Similar to $LISTA$, the lists $LISTA1$ and $LISTA2$ consist of sublists $L1_i$ and $L2_i$, respectively, corresponding to each 2-set $T_1, \ldots, T_m$. $L1_i$ is constructed exactly as L_i, except that since it is impossible to generate *all* permutations of $(1 \ldots n) \setminus T_i$ with $k-2 = 3$-differences, we generate only the *even* permutations. $L2_i$ is obtained from $L1_i$ as follows. If

$$(1 \ldots n) \setminus T_i = a_1 a_2 \ldots a_{n-2}$$

then $L2_i$ is obtained by swapping a_1 and a_2 in every permutation on the list $L1_i$. See Figure 10 for an example in the case where $(n,k) = (6,5)$.

The entire construction is summarized in the procedures $PERMUTE$ and $PARTIAL$ of Figures 11 and 12. These procedures should be regarded as a concise description of the list, rather than an efficient procedure for generating the list. (In fact, the algorithm can be implemented to run in time proportional to the size of the output using storage $O(n)$ instead of $n!$ storage.)

As for correctness, it follows from Section 3 and the discussion at the beginning of this section that the construction works correctly for the basis cases. For the inductive case, we assume inductively that sublists L_i and $L1_i$ are as claimed. In particular then, every permutation occurs exactly once on $LIST$. Further, these sublists were defined so that last(L_i) and first(L_{i+1}) would differ in exactly k positions for $1 \leq i < \binom{n}{2}$, and similarly for $L1_i$ and $L1_{i+1}$. It remains to show that the lists $LISTA$ and $LISTB$ (when $k \neq 5$) and the lists (when $k = 5$) $LISTA1, LISTA2, LISTB1, LISTB2$ glue together properly and that the final list is cyclic.

To show for $k \neq 5$ that last($LISTA$) and first($LISTB$) differ exactly in the first k positions, note that last($LISTA$) = last(L_m) and $T_m = \{3, 4\}$. So,

$$\text{first}(L_m) = 341256 \ldots k \ldots \text{n or } 431256 \ldots k \ldots n$$

Then if $q = \text{last}(L_m)$, recall that

1. q differs from first(L_m) exactly in the first k positions and

2. the first two positions of q contain 3 and 4 in some order.

Since first($LISTB$) $= 213456 \ldots k \ldots n$, it follows that q also differs from first($LISTB$) exactly in the first k positions. Similarly, last($LISTB$) and first($LISTA$) differ exactly in the first k positions, so that the listing is cyclic.

In the case $k = 5$, we must show that in each of the following pairs, the two permutations differ in exactly 5 positions.

1. last($LISTA1$), first($LISTA2$)

2. last($LISTB1$), first($LISTB2$)

3. last($LISTA2$), first($LISTB1$).

To show (1), first($LISTA2$) $= 124356 \ldots n$ and last($LISTA1$) = last($L1_m$). Since $T_m = \{3, 4\}$,

$$\text{first}(L1_m) = 341256 \ldots n \text{ or } 431256 \ldots n$$

so,

$$\text{last}(L1_m) = 435126 \ldots n \text{ or } 345126 \ldots n,$$

both of which differ from first($LISTA2$) exactly in the first 5 positions. Case (2) is similar.

To show (3),

$$\text{first}(LISTB1) = 213456\ldots n.$$

The last sublist of $LISTA2$ is $L2_m$ which is obtained from $L1_m$ by swapping elements 1 and 2 in every permutation. So, last($LISTA2$) = last($L1_m$) with elements 1 and 2 swapped. Then,

$$\text{last}(LISTA2) = 345216\ldots n \text{ or } 435216\ldots n,$$

both of which differ exactly in the first five positions from first($LISTB1$). Finally, to show that the list is cyclic, note that

$$\text{last}(LISTB) = 345216\ldots n \text{ or } 435216\ldots n,$$

both of which differ in the first five positions from

$$\text{first}(LISTA) = 1\ldots n.$$

5 Conclusions

Although we have been able to list permutations to differ in k positions, these positions are not always contiguous, unless $k = 2$ or $k = n$. In Figure 9, for example, successive permutations 132546 and 412356 from $LISTA$ differ in the four positions 1, 2, 4, and 5, which are not contiguous even if the last position is considered contiguous to the first. However, "differing in k contiguous positions" would also give rise to a connected, undirected, vertex transitive graph on permutations. So, it seems likely that permutations could be listed by k-contiguous differences, otherwise this would be a counterexample to the conjecture that all graphs in this class do have Hamilton paths. In any case, requiring contiguous differences would appear to require a different approach.

Acknowledgements. Thanks to Herb Wilf and the referee for helpful suggestions and to Terry Wang for comments on an earlier version of this paper.

References

[1] J. Boothroyd, "PERM (Algorithm 6)," *Computer Bulletin*, 9, No. 3 (1965), p. 104.

[2] J. Boothroyd, "Permutation of the elements of a vector (Algorithm 29)" and "Fast permutation of the elements of a vector (Algorithm 30)," *Comput. J.*, 10 (1967), pp. 310-312.

[3] M. Buck and D. Wiedemann, "Gray codes with restricted density," *Discrete Math.*, 48 (1984), pp. 163-171.

[4] P. Eades, M. Hickey, and R. C. Read, "Some Hamilton paths and a minimal change algorithm," *J. Assoc. Comput. Mach.*, 31, No. 1 (1984), pp. 19-29.

[5] C. Compton, Ph.D. Thesis, University of California, San Diego, in preparation.

[6] P. Eades and B. McKay, "An algorithm for generating subsets of fixed size with a strong minimal change property," *Inform. Process. Lett.*, 19 (1984), pp. 131-133.

[7] R. B. Eggleton and W. D. Wallis, "Problem 86: Solution I," *Math. Mag.*, 58, No. 2 (1985), pp. 12-113.

[8] M. Hall, Jr., *Combinatorial Theory, Second Edition*, John Wiley and Sons, Inc., 1986.

[9] B. R. Heap, "Permutations by interchanges," *Comput. J.*, 6 (1963), pp. 293-94.

[10] T. Hough and F. Ruskey, "An efficient implementation of the Eades, Hickey, Read adjacent interchange combination generation algorithm," *J. Combin. Math. Comb. Comput.*, 4 (1988), pp. 79-86.

[11] R. J. Gould and R. Roth, "Cayley digraphs and $(1, j, n)$ sequencings of the alternating group A_n," *Discrete Math.*, 66 (1987), pp. 91-102.

[12] B. Jackson, "Hamilton cycles in regular 2-connected graphs," *J. Comb. Theory Ser. B*, 29 (1980), pp. 27-46.

[13] S. M. Johnson, "Generation of permutations by adjacent transpositions," *Math. Comp.*, 17 (1963), pp. 282-285.

[14] J. T. Joichi and D. E. White, "Gray codes in graphs of subsets," *Discrete Math.*, 31 (1980), pp. 29-41.

[15] V. L. Kompel'makher and V. A. Liskovets, "Sequential generation of arrangements by means of a basis of transpositions," *Kibernetica*, 3 (1975), pp. 17-21.

[16] L. Lovász, Problem 11 in *Combinatorial Structures and their Applications*, Gorden and Breach, London 1970.

[17] J. Metzger, "Problem 1186," *Math. Mag.*, 58 (1985), pp. 113-114.

[18] A. Nijenhuis and H. S. Wilf, *Combinatorial Algorithms for Computers and Calculators*, Academic Press, Inc., New York 1978.

[19] T. C. Putnam, "A Gray code variant: sequencing permutations via fixed points," manuscript (1989).

[20] S. Rabinowitz, "Problem 1186," *Math. Mag.*, 57 (1984), p. 109.

[21] D. F. Rall and P. J. Slater, "Generating all permutations by graphical derangements", unpublished manuscript.

[22] F. Ruskey and G. Pruesse, "Transposition generation of the linear extensions of certain posets," to appear.

[23] F. Ruskey and C. Savage, "Hamilton cycles which extend transposition matchings in Cayley graphs of S_n," in preparation.

[24] R. Sedgewick, "Permutation generation methods," *Computing Surveys*, 9, No. 2 (1977), pp. 137-164.

[25] H. F. Trotter, "PERM (Algorithm 115)", *Comm. ACM*, 5, No. 8 (1962), pp. 434-435.

[26] M. B. Wells, "Generation of permutations by transposition", *Math. Comp.*, 15 (1961), pp. 192-195.

[27] H. S. Wilf, *Combinatorial Algorithms: An Update*, SIAM, Philadelphia, 1989.

[28] H. S. Wilf, personal communication.