

## Sorting permutations by block-interchanges

David A. Christie<sup>1</sup>

*Department of Computing Science, The University of Glasgow, Glasgow G12 8QQ, Scotland, United Kingdom*

Received 8 November 1995; revised 19 September 1996

Communicated by V. Ramachandran

---

### Abstract

Various global rearrangements of permutations, such as reversals and transpositions have recently become of interest because of their applications in genome analysis. The study of such rearrangements leads to computational problems that are of interest in their own right. In this paper we introduce an operation, called block-interchange, in which two substrings, or blocks, swap positions in the permutation. We demonstrate a polynomial-time algorithm for calculating the block-interchange distance of a permutation (i.e. the minimum number of block-interchanges required to transform the permutation to the identity). We also determine the block-interchange diameter of the symmetric group.

*Keywords:* Algorithms; Combinatorial problems; Sorting; String comparison; Permutations

---

### 1. Introduction

In classical methods of string comparison, strings may only mutate by operations that act on individual characters, e.g., insertions, deletions, and substitutions. Recently applications in molecular biology have motivated the study of larger scale mutations such as reversals [9,3], transpositions [2], and translocations [7,10], applied, in particular, to permutations rather than to general strings. Many of the resulting problems remain open as far as polynomial-time solvability or NP-completeness is concerned, though Caprara [4] has recently proved that sorting by reversals is NP-hard.

Consideration of such “global” operations on permutations leads to a number of algorithmic problems

that are of interest in their own right. For example, Aigner and West [1] have studied the problem of sorting a list by the repeated insertion of the leading element. Bounds for sorting a permutation by prefix-reversals have been obtained by Gates and Papadimitriou [6].

Global operations on permutations may be thought of as generators of the symmetric group (or a subgroup of the symmetric group). Given an arbitrary set of generators, the problem of finding a minimal sequence that generates a permutation has been shown to be NP-hard by Even and Goldreich [5]. Jerrum [8] has shown that the problem is PSPACE-complete even when the generator set contains only two generators. However when the generator set is fixed, the NP-completeness results no longer apply and in [8] some polynomial-time algorithms for sorting permutations using global operations are given.

In this paper we introduce a new, natural global operation – the *block-interchange* – which can be

---

<sup>1</sup> Supported by a postgraduate research studentship from the Engineering and Physical Sciences Research Council. Email: christie@dcs.gla.ac.uk.

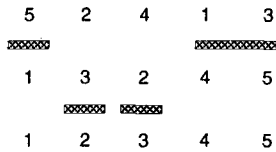


Fig. 1. An example of sorting by block-interchange.

viewed as a generalisation of a transposition. In a block-interchange two non-intersecting substrings of any length are swapped in the permutation. In a transposition the substrings must be adjacent.

Given a permutation  $\pi = [\pi_1 \pi_2 \dots \pi_n]$ , a block-interchange with parameters  $(i, j, k, l)$ , where  $1 \leq i < j \leq k < l \leq n + 1$ , is applied to  $\pi$  by exchanging the blocks  $[\pi_i \dots \pi_{j-1}]$  and  $[\pi_k \dots \pi_{l-1}]$ . Note that the special case of  $j = k$ , is a transposition.

The minimum number of block-interchanges that transform  $\pi$  into  $\sigma$  is the *block-interchange distance* between the two permutations. This value will be the same as the block-interchange distance between  $\pi \cdot \sigma^{-1}$  and the identity permutation. So finding the block-interchange distance between two permutations is equivalent to sorting a permutation using block-interchanges. The block-interchange distance of  $\pi$ ,  $bid(\pi)$ , is the length of the shortest sequence of block-interchanges that transforms  $\pi$  into the identity permutation. For example the permutation  $\pi = [5\ 2\ 4\ 1\ 3]$ , has block-interchange distance 2 as shown in Fig. 1.

If two elements are adjacent in the permutation  $\pi$ , but are not adjacent in the identity permutation then we say they form a *breakpoint*, because at some stage in the sequence of block-interchanges that sorts  $\pi$ , the two elements will have to be split apart. More precisely we define breakpoints exactly as Bafna and Pevzner [2] did for transpositions, i.e., we assume  $\pi_0 = 0$  and  $\pi_{n+1} = n + 1$ , and define a breakpoint as a value of  $i$  such that  $1 \leq i \leq n + 1$  and  $\pi_i - \pi_{i-1} \neq 1$ . Define  $b(\pi)$  to be the number of breakpoints in the permutation. Note that the identity permutation is the only permutation with no breakpoints, and a block-interchange can change  $b(\pi)$  by at most 4.

In this paper we will show that there is a polynomial-time algorithm for the evaluation of  $bid(\pi)$ . This will be achieved by first demonstrating that for all permutations, except the identity permutation, a block-interchange exists that removes at least

two breakpoints. Then it will be proved using a graph model for the permutations that an algorithm which repeatedly applies such block-interchanges sorts  $\pi$  using the fewest number of block-interchanges. We then study the permutations of  $n$  elements which require the most block-interchanges in order to sort them and show that the block-interchange diameter of the symmetric group  $S_n$  is  $\lfloor n/2 \rfloor$ .

## 2. Minimal block-interchanges

**Lemma 1.** *It is always possible to find a block-interchange that removes at least two breakpoints from a given permutation,  $\pi$ , unless  $\pi$  is the identity permutation.*

**Proof.** Since  $\pi$  is not the identity permutation, there must be at least two elements in  $\pi$  that appear in the wrong order i.e. there must be an  $x$  and a  $y$  such that  $x < y$  but  $\pi = [\dots y \dots x \dots]$ .

Now choose  $x$  to be the smallest such value and choose  $y$  to be the largest value in  $\pi$  to the left of this  $x$ . Then  $x - 1$  must be to the left of  $y$  in  $\pi$  since otherwise this would contradict the choice of  $x$ . Similarly  $y + 1$  must be to the right of  $x$  in  $\pi$ . Note that in fact  $x$  is the smallest value such that  $\pi_x \neq x$ . Hence  $\pi$  has the form

$$\pi = [1 \dots x - 1 \dots y \dots x \dots y + 1 \dots].$$

The block-interchange with parameters

$$(\pi^{-1}(x - 1) + 1, \pi^{-1}(y) + 1, \pi^{-1}(x), \pi^{-1}(y + 1))$$

transforms  $\pi$  into  $\pi'$

$$\pi' = [1 \dots x - 1\ x \dots y\ y + 1 \dots].$$

Notice that before this transformation there were breakpoints at each place where the block-interchange cut  $\pi$ . After the block-interchange there are at least two fewer breakpoints in the permutation. Hence the lemma.  $\square$

We will call the block-interchange described in the above lemma the *minimal* block-interchange. Since we can remove at most four breakpoints with any block-interchange and a minimal block-interchange

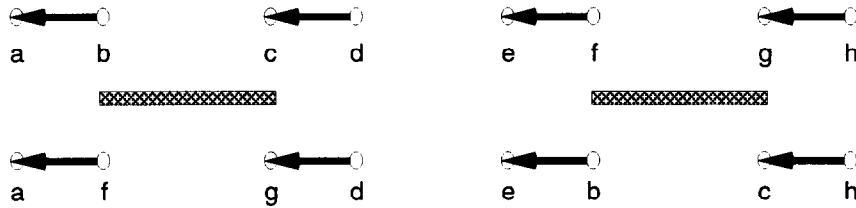


Fig. 2. The black edges that change when a block-interchange (which is not a transposition) is applied.

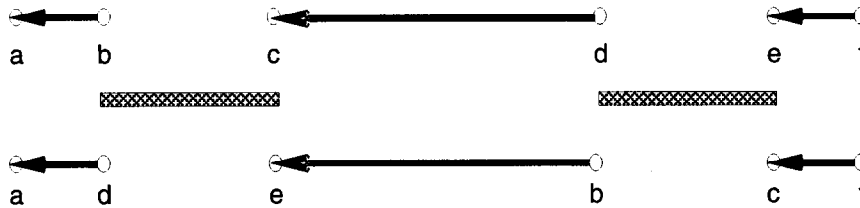


Fig. 3. The black edges that change when a block-interchange (which is a transposition) is applied.

removes at least two breakpoints, then an algorithm that repeatedly applies the minimal block-interchanges gives us a 2-approximation algorithm for block-interchange distance. However when we examine these block-interchanges on a graph model we discover that this algorithm returns the exact solution.

### 3. The graph model

The graph model we use is that of a *cycle graph* as introduced by Bafna and Pevzner [2] when dealing with transpositions. The cycle graph,  $G(\pi)$ , of a permutation  $\pi$  is a directed edge-coloured graph with vertex set  $\{0, 1, \dots, n + 1\}$  and edge set defined as follows. A grey edge connects vertex  $i$  to vertex  $i + 1$  for  $0 \leq i \leq n$ . A black edge connects vertex  $\pi_i$  to vertex  $\pi_{i-1}$  for  $1 \leq i \leq n + 1$ .

Since every incoming edge of a vertex can be uniquely paired with an outgoing edge of the other colour, we can easily completely decompose the graph into alternating cycles, and furthermore this decomposition is unique. The identity permutation is the only permutation that has  $n + 1$  alternating cycles in its graph. The number of alternating cycles in  $G(\pi)$  is denoted by  $c(\pi)$ .

A block-interchange changes only black edges in the graph. Three or four black edges are removed from the graph and replaced by new black edges as shown in Figs. 2 and 3.

**Lemma 2.** *When a minimal block-interchange is applied  $c(\pi)$  increases by two.*

**Proof.** A minimal block-interchange may change 3 or 4 black edges of  $G(\pi)$ . If it changes 3 edges then those edges must be part of one cycle. If it changes 4 edges then those edges must be part of either one or two cycles. The three possible cases are shown in Figs. 4, 5, and 6. In each of these cases  $c(\pi)$  increases by two. Note that in the figures black edges are represented by thick edges, grey edges are represented by thin edges, and paths which start and finish with a grey edge are represented by dotted thin edges.  $\square$

**Lemma 3.** *It is impossible to increase  $c(\pi)$  by more than two with a single block-interchange.*

**Proof.** A block-interchange changes at most four black edges in the graph, so the only way that we could increase  $c(\pi)$  by more than two with a single block-interchange would be if one big cycle was

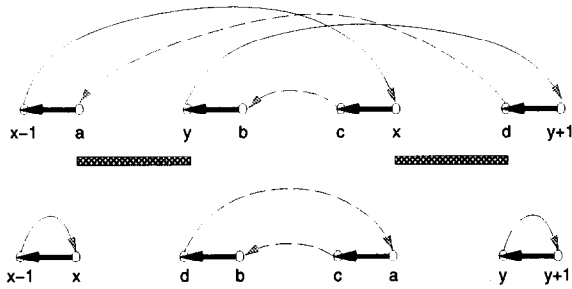


Fig. 4. How a minimal block-interchange changes  $G(\pi)$ .

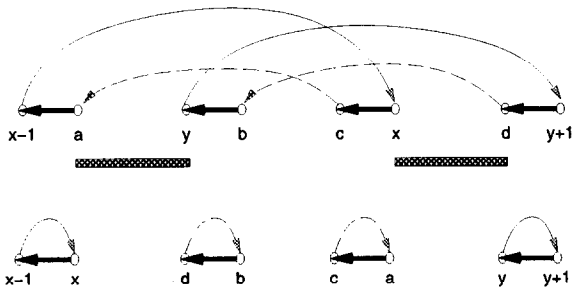


Fig. 5. How a minimal block-interchange changes  $G(\pi)$ .

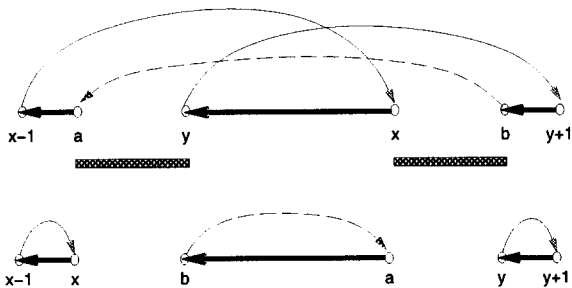


Fig. 6. How a minimal block-interchange changes  $G(\pi)$ .

broken into four cycles, but this is impossible. A block-interchange that changes black edges from four different cycles results in two cycles. By symmetry, a block-interchange that results in four cycles must have acted on black edges from two cycles (an example of this is shown in Fig. 5). So  $c(\pi)$  can be increased by at most two with a single block-interchange.  $\square$

```

initialise visited and  $\pi^{-1}$ ;
cycles := 0;
for i := 0 to n do begin
  if not visited[i] then begin
    cycles := cycles + 1;
    j := i;
    while not visited[j] do begin
      visited[j] := true;
      j := j + 1; {follow grey edge}
      j :=  $\pi[\pi^{-1}(j) - 1]$ ; {follow black edge}
    end
  end
end;
return (n + 1 - cycles)/2;

```

Fig. 7. An algorithm to calculate  $bid(\pi)$ .

```

while  $\pi$  is not sorted do begin
  locate the smallest x such that  $\pi_x \neq x$ 
  find largest value y between x - 1 and  $\pi^{-1}(x)$ 
  perform the block-interchange
    ( $x, \pi^{-1}(y), \pi^{-1}(x), \pi^{-1}(y + 1)$ )
end

```

Fig. 8. An algorithm to perform an optimal sequence of block interchanges.

#### 4. Block-interchange distance

These three lemmas lead to our main result.

**Theorem 4.** The block-interchange distance  $bid(\pi)$  of a permutation  $\pi$  is

$$bid(\pi) = \frac{1}{2}((n + 1) - c(\pi)),$$

where  $c(\pi)$  is the number of alternating cycles in the cycle graph of  $\pi$ .

**Proof.** By Lemma 2,  $bid(\pi) \leq \frac{1}{2}(c(\iota) - c(\pi))$ , and by Lemma 3,  $bid(\pi) \geq \frac{1}{2}(c(\iota) - c(\pi))$ , where  $\iota$  is the identity permutation. So  $bid(\pi) = \frac{1}{2}((n + 1) - c(\pi))$ , since  $c(\iota) = n + 1$ .  $\square$

An algorithm which calculates the block-interchange distance is shown in Fig. 7. The algorithm performs a depth-first search of  $G(\pi)$ , counting the number of cycles, and hence runs in linear time.

An algorithm which performs a sequence of block-interchanges achieving the minimum is shown in Fig. 8. This algorithm has  $O(n^2)$  complexity since

Table 1  
The number of permutations of size  $n$  that achieve  $BID$

$n$	1	2	3	4	5	6	7	8	9	10
no.	1	1	5	8	84	180	3044	8064	193248	604800

all the steps inside the while loop can be executed in linear time, and the loop itself is executed  $\lfloor n/2 \rfloor$  times at most.

## 5. Block-interchange diameter

The *block-interchange diameter*,  $BID(n)$ , of the symmetric group  $S_n$ , is the maximum value of  $bid(\pi)$  taken over all  $n$ -element permutations. When  $\pi$  is the reverse permutation of length  $n$ , i.e.  $\pi = [n\ n-1\ \dots\ 1]$ , then  $c(\pi)$  is given by the formula

$$c(\pi) = \begin{cases} 2 & \text{if } n \text{ is odd,} \\ 1 & \text{otherwise,} \end{cases}$$

i.e.  $c(\pi)$  is as small as possible (since  $c(\pi) \equiv n + 1 \pmod{2}$ ). Hence  $BID(n)$  will be achieved by the reverse permutation and is  $\lfloor n/2 \rfloor$ .

There are many other permutations which achieve this upper bound, e.g., the permutation  $[2\ 4\ 6\ 8\ 1\ 3\ 5\ 7]$  contains only one cycle in its cycle graph, and the family of permutations like it have the same value for  $c(\pi)$  as the reverse permutation. Table 1 shows how many permutations achieve the  $BID$  for small values of  $n$ .

## 6. Conclusion

Polynomial algorithms are not known (yet) for the problem of sorting permutations by transpositions and sorting by reversals has been shown to be NP-hard. By contrast we have derived a polynomial-time algorithm for sorting by block-interchanges and have also evaluated the block-interchange diameter of  $S_n$ .

## References

- [1] M. Aigner and D.B. West, Sorting by insertion of leading elements, *J. Combin. Theory Ser. A* **45** (1987) 306–309.
- [2] V. Bafna and P.A. Pevzner, Sorting by transpositions, in: *Proc. 6th Ann. ACM-SIAM Symp. on Discrete Algorithms* (1995) 614–623.
- [3] V. Bafna and P.A. Pevzner, Genome rearrangements and sorting by reversals, *SIAM J. Comput.* **25** (1996) 272–289.
- [4] A. Caprara, Sorting by reversals is difficult, April 1996.
- [5] S. Even and O. Goldreich, The minimum-length generator sequence problem is NP-hard, *J. Algorithms* **2** (1981) 311–313.
- [6] W.H. Gates and C.H. Papadimitriou, Bounds for sorting by prefix reversal, *Discrete Math.* **27** (1979) 47–57.
- [7] S. Hannenhalli, Polynomial algorithm for computing translocation distance, in: *Proc. 6th Ann. Symp. on Combinatorial Pattern Matching (CPM'95)*, Lecture Notes in Computer Science **937** (Springer, Berlin, 1995) 162–176.
- [8] M.R. Jerrum, The complexity of finding minimum-length generator sequences, *Theoret. Comput. Sci.* **36** (1985) 265–289.
- [9] J. Kececioglu and D. Sankoff, Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement, *Algorithmica* **13** (1995) 180–210.
- [10] J.D. Kececioglu and R. Ravi, Of mice and men: Algorithms for evolutionary distances between genomes with translocation, in: *Proc. 6th ACM-SIAM Symp. on Discrete Algorithms* (1995) 604–613.