Sorting by Prefix Block-Interchanges

Anthony Labarre
Laboratoire d'Informatique Gaspard Monge, Université Gustave Eiffel, LIGM (UMR 8049), CNRS, ENPC, ESIEE Paris, UPEM, 77454, Marne-la-Vallée, France
`http://igm.univ-mlv.fr/~alabarre/`
Anthony.Labarre@u-pem.fr

Abstract

We initiate the study of sorting permutations using *prefix block-interchanges*, which exchange any prefix of a permutation with another non-intersecting interval. The goal is to transform a given permutation into the identity permutation using as few such operations as possible. We give a 2-approximation algorithm for this problem, show how to obtain improved lower and upper bounds on the corresponding distance, and determine the largest possible value for that distance.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases permutations, genome rearrangements, interconnection network, sorting, edit distance, prefix block-interchange

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2020.55

Related Version `https://arxiv.org/pdf/2008.13640`

1 Introduction

The problem of transforming two sequences into one another using a specified set of operations has received a lot of attention in the last decades, with applications in computational biology as *(genome) rearrangement problems* [13] as well as interconnection network design [21]. In the context of permutations, it can be equivalently formulated as follows: given a permutation π of $[n] = \{1, 2, \ldots, n\}$ and a generating set S (also consisting of permutations of $[n]$), find a minimum-length sequence of elements from S that sorts π. The problem is known to be NP-hard in general [15] and W[1]-hard when parameterised by the length of a solution [6], but some families of operations that are important in applications lead to problems that can be solved in polynomial time (e.g. *exchanges* [17], *block-interchanges* [10] and *signed reversals* [14]), while other families yield hard problems that admit good approximations (e.g. 11/8 for *reversals* [3] and for *block-transpositions* [12]).

Several restrictions of these families have also been studied, one of which stands out in the field of interconnection network design: the so-called *prefix constraint*, which forces operations to act on a prefix of the permutation rather than on an arbitrary interval. Those restrictions were introduced as a way of reducing the size of the generated network while maintaining a low value for its *diameter*, thereby guaranteeing a low maximum communication delay [21]. The most famous example is perhaps the restriction of reversals (which reverse the order of elements along an interval) to *prefix reversals*, and the corresponding problem known as *pancake flipping*, introduced in [16] and whose complexity was only settled thirty years later [5].

As Table 1 shows (see [13] for undefined terms), although sorting problems using interval transformations are now fairly well understood, progress on the corresponding prefix sorting problems has been lacking, with only two families whose status has been settled and no approximation ratio smaller than 2 for those problems not known to be in P. As a result, while the topology of the *Cayley graph* generated by those operations might present attractive properties, efficient routing algorithms (which achieve exactly the same task as the sorting algorithms in genome rearrangements) are still needed for the network to be of practical interest.

Table 1 Complexity of some sorting problems on permutations in the unrestricted setting and under the prefix constraint.

Operation	Unrestricted	Prefix-constrained
reversal	NP-hard [7]	NP-hard [5]
signed reversal	in P [14]	open
double cut-and-join	NP-hard [8]	open
signed double cut-and-join	in P [22]	open
exchange	in P [17]	in P [1]
block-transposition	NP-hard [4]	open
block-interchange	in P [10]	open

In this work, we choose to focus on the family of block-interchanges for the following reasons:

1. Along with double cut-and-joins, they constitute one of the most general kind of operations on permutations, including both exchanges and block-transpositions as special cases;
2. Their behaviour in the unrestricted setting is understood well enough that we can hope for the corresponding prefix sorting problem to be in P;
3. Knowledge about these operations in the prefix setting is lacking and will be needed for more general studies; for instance, rearrangement problems on strings are usually NP-hard, and efficient algorithms to solve them exactly or approximately routinely rely on techniques developed for permutations [13, part II], which currently do not exist for prefix block-interchanges.

To the best of our knowledge, the only published work on prefix block-interchanges is by [9], who studied them on strings and showed that binary strings can be sorted in linear time, whereas transforming two binary strings into one another using the minimum number of prefix block-interchanges is NP-complete. Our contributions are as follows: we prove tight upper and lower bounds on the so-called *prefix block-interchange distance*; we give an approximation algorithm which we prove to be a 2-approximation with respect to two different measures; we show how to tighten those bounds; and finally, we prove that the maximum value of the distance, an important parameter in some applications [21], is $\lfloor 2n/3 \rfloor$.

2 Notation and definitions

A *permutation* is a bijective application of a set (usually $[n] = \{1, 2, \ldots, n\}$ in this work) onto itself. The *symmetric group* S_n is the set of all permutations of $[n]$ together with the usual function composition applied from right to left. We write permutations using lower case Greek letters, viewing them as sequences $\pi = \langle \pi_1\ \pi_2\ \cdots\ \pi_n \rangle$, where $\pi_i = \pi(i)$, and occasionally rely on the two-line notation to denote them. The permutation $\iota = \langle 1\ 2\ \cdots\ n \rangle$ is the *identity permutation.*

Permutations are well-known to decompose in a single way into *disjoint cycles* (up to the ordering of cycles and of elements within each cycle), leading to another notation for π based on its *disjoint cycle decomposition.* For instance, when $\pi = \langle 7\ 1\ 4\ 5\ 3\ 2\ 6 \rangle$, the disjoint cycle notation is $\pi = (1, 7, 6, 2)(3, 4, 5)$. The *conjugate* of a permutation π by a permutation σ, both in S_n, is the permutation $\pi^\sigma = \sigma\pi\sigma^{-1}$. All permutations in S_n that can be obtained from one another using this operation form a *conjugacy class* (of S_n), and have the same cycle structure.

▶ **Definition 1** ([10]). *The* block-interchange $\beta(i,j,k,\ell)$ *with* $1 \le i < j \le k < \ell \le n+1$ *is the permutation that exchanges the closed intervals determined respectively by* i *and* $j-1$ *and by* k *and* $\ell - 1$*:*

$$\left\langle \begin{array}{c} 1 \cdots i-1 \ \boxed{i \ \cdots \ j-1} \ j \ j+1 \ \cdots \ k-1 \ \boxed{k \ \cdots \ \ell-1} \ \ell \ \ell+1 \ \cdots \ n \\ 1 \cdots i-1 \ \boxed{k \ \cdots \ \ell-1} \ j \ j+1 \ \cdots \ k-1 \ \boxed{i \ \cdots \ j-1} \ \ell \ \ell+1 \ \cdots \ n \end{array} \right\rangle.$$

Block-interchanges generalise several well-studied operations: when $j = k$, the resulting operation exchanges two adjacent intervals, and is known as a *(block-)transposition* [2]; when $j = i+1$ and $\ell = k+1$, the resulting operation swaps elements in respective positions i and k, and is called an *exchange* (or *(algebraic) transposition*); finally, when $i = 1$, the resulting operation is called a *prefix block-interchange*; prefix block-transpositions and prefix exchanges are defined analogously. We study the following problem.

SORTING BY PREFIX BLOCK-INTERCHANGES (SBPBI)	
Input:	a permutation π in S_n, a number $K \in \mathbb{N}$.
Question:	is there a sequence of at most K prefix block-interchanges that sorts π?

The length of a shortest sorting sequence of prefix block-interchanges for a permutation π is its *(prefix block-interchange) distance*, which we denote $pbid(\pi)$. Distances based on other operations are defined similarly.

3 A 2-approximation based on the breakpoint graph

We give in this section a 2-approximation algorithm for SBPBI based on the *breakpoint graph*. We first use this structure in Subsection 3.1 to derive an upper bound on *pbid* and present our algorithm, then derive a lower bound in Subsection 3.2 which allows us to prove its performance guarantee. The breakpoint graph is well-known to be equivalent [18] to another structure known as the *cycle graph* [2], which allows us to use results based on either graph indifferently.

▶ **Definition 2** ([14]). *For any* π *in* S_n*, let* π' *be the permutation of* $\{0, 1, 2, \ldots, 2n+1\}$ *defined by* $\pi'_0 = 0$*,* $\pi'_{2n+1} = 2n+1$*, and* $(\pi'_{2i-1}, \pi'_{2i}) = (2\pi_i - 1, 2\pi_i)$ *for* $1 \le i \le n$*. The* breakpoint graph *of* π *is the undirected edge-bicoloured graph* $G(\pi) = (V, E_b \cup E_g)$ *whose vertex set is formed by the elements of* π' *ordered by position and whose edge set consists of:*

- $E_b = \{\{\pi'_{2i}, \pi'_{2i+1}\} \mid \forall\ 0 \le i \le n\}$*, called the set of* black *edges;*
- $E_g = \{\{2i, 2i+1\} \mid \forall\ 0 \le i \le n\}$*, called the set of* grey *edges.*

Figure 1 shows an example of a breakpoint graph. Since $G(\pi)$ is 2-regular, it decomposes in a single way into edge-disjoint cycles which alternate black and grey edges. The *length* of a cycle in $G(\pi)$ is the number of black edges it contains, and a *k-cycle* in $G(\pi)$ is a cycle of length k. We let $c(G(\pi))$ (resp. $c_k(G(\pi))$) denote the number of cycles (resp. k-cycles) in $G(\pi)$, and refer to cycles of length one as *trivial cycles*.

A crucial insight of strategies based on the breakpoint graph is the observation that the transformations that we apply never affect grey edges, whereas they "cut" black edges and replace them with new black edges. This point of view conveniently allows us to define block-interchanges in terms of the black edges on which they *act*: using the notation $b_i = \{\pi_{2i-2}, \pi_{2i-1}\}$ for a black edge, a quadruplet (b_i, b_j, b_k, b_ℓ) of black edges with $i < j \le k < \ell$ naturally defines the block-interchange $\beta(i,j,k,\ell)$ and conversely.

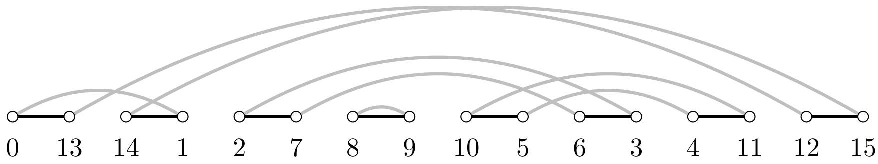


Figure 1 The breakpoint graph of ⟨7 1 4 5 3 2 6⟩.

3.1 An upper bound based on the breakpoint graph

The following quantity, defined for any π in S_n, has been shown to be a tight[1] lower bound on the prefix block-transposition distance [19] :

$$g(\pi) = \frac{n+1+c(G(\pi))}{2} - c_1(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

We prove in Theorem 5 that this quantity is also an *upper* bound on the prefix block-interchange distance. To that end, we use the following notation, based on the one introduced in [2]; for any two permutations π and σ, define:

- $\Delta c(\pi, \sigma) = c(G(\sigma)) - c(G(\pi))$,
- $\Delta c_1(\pi, \sigma) = c_1(G(\sigma)) - c_1(G(\pi))$,
- $\Delta f(\pi, \sigma) = f(\sigma) - f(\pi)$, where $f(\pi) = 0$ if π *fixes* 1 (i.e. $\pi_1 = 1$) and 1 otherwise, and
- $\Delta g(\pi, \sigma) = g(\sigma) - g(\pi)$.

These parameters allow us to obtain the following expression, which will be useful in our proofs:

$$\Delta g(\pi, \sigma) = \Delta c(\pi, \sigma)/2 - \Delta c_1(\pi, \sigma) - \Delta f(\pi, \sigma). \tag{2}$$

We start by proving in Lemma 4 the existence of a prefix block-interchange that decreases $g(\pi)$ by at least one if $\pi_1 \neq 1$. The proof uses the following structural result, where grey edges $\{\pi'_a, \pi'_b\}$ and $\{\pi'_c, \pi'_d\}$ (with $a < b$ and $c < d$) are said to *intersect* if $a < c < b < d$ or $c < a < d < b$.

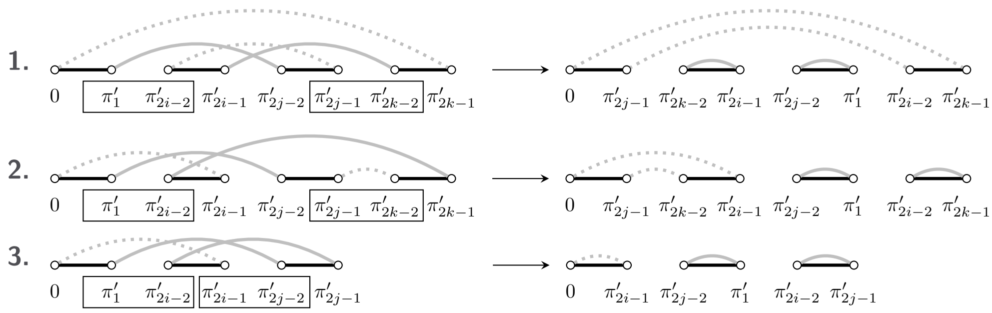
▶ **Lemma 3** ([14]). *For every permutation π, let e be a grey edge in a nontrivial cycle of $G(\pi)$; then there exists another grey edge e' in $G(\pi)$ that intersects e.*

We refer to the grey edge of $G(\pi)$ that contains π'_1 as the *first* grey edge, and to the cycle that contains 0 as the *leftmost cycle*. Our figures represent alternating subpaths (i.e., paths that alternate black and grey edges) as dotted edges; therefore, such a dotted edge might correspond to a single grey edge, or to a black edge framed by two grey edges, and so on.

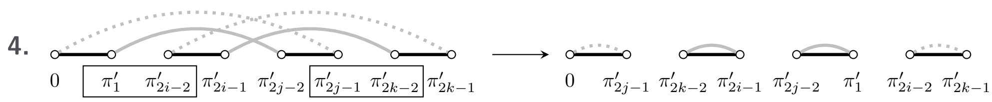
▶ **Lemma 4.** *For any π in S_n: if $\pi_1 \neq 1$, then there exists a prefix block-interchange β such that $\Delta c(\pi, \pi\beta) = 2$, $\Delta c_1(\pi, \pi\beta) \geq 2$, and $\Delta g(\pi, \pi\beta) \leq -1$.*

Proof. Lemma 3 guarantees the existence of a grey edge e' that intersects the first grey edge; moreover, the endpoints of e' ordered by position connect elements whose values are either in decreasing (case 1 below) or increasing (case 2 below) order. In both cases, if e' belongs to the leftmost cycle, then there exists a prefix block-interchange that extracts two 1-cycles (we distinguish an additional third case where e' and the first grey edge share the endpoints of a black edge):

[1] Here and in the rest of the text, "tight" means that equality is achieved by *some* but *not all* instances.



Otherwise, all edges that intersect the first grey edge belong to cycles other than the leftmost cycle. One of those edges, which belongs to some cycle C, must be in the same configuration as in case 1 above: indeed, if C contains a grey edge that takes us from the interval covered by the first grey edge $(1, y)$ to a vertex located after y, then C must also contain a grey edge that takes us back before y. Without loss of generality, we thus assume that e' is such an edge, and therefore a prefix block-interchange that extracts two new 1-cycles can also be applied:



In all four cases, we have $\Delta c(\pi, \pi\beta) = 2$ and $\Delta c_1(\pi, \pi\beta) \geq 2$. The exact value of $\Delta g(\pi, \pi\beta)$ will vary depending on the above configurations and is computed using Equation 2:

1. the leftmost cycle in $G(\pi\beta)$ is nontrivial, so $\Delta c_1(\pi, \pi\beta) = 2$, $\Delta f(\pi, \pi\beta) = 0$ and $\Delta g(\pi, \pi\beta) = -1$;
2. for exactly the same reasons as case 1, we obtain $\Delta g(\pi, \pi\beta) = -1$;
3. we have two possible subcases:
   - **a.** if the leftmost cycle in $G(\pi\beta)$ is nontrivial, then $\Delta c(\pi, \pi\beta) = 2$ and $\Delta f(\pi, \pi\beta) = 0$, so $\Delta g(\pi, \pi\beta) = -1$;
   - **b.** otherwise, $\Delta c_1(\pi, \pi\beta) = 3$ and $\Delta f(\pi, \pi\beta) = -1$, which in turn implies $\Delta g(\pi, \pi\beta) = -1$.
4. we have three possible subcases:
   - **a.** $\Delta c_1(\pi, \pi\beta) = 2$: the leftmost cycle in $G(\pi\beta)$ is nontrivial, so $\Delta f(\pi, \pi\beta) = 0$ and $\Delta g(\pi, \pi\beta) = -1$;
   - **b.** $\Delta c_1(\pi, \pi\beta) = 3$: the leftmost cycle in $G(\pi\beta)$ may or may not be trivial, so $\Delta f(\pi, \pi\beta) \leq 0$ and $\Delta g(\pi, \pi\beta) \leq -1$;
   - **c.** $\Delta c_1(\pi, \pi\beta) = 4$: the leftmost cycle in $G(\pi\beta)$ is trivial, so $\Delta f(\pi, \pi\beta) = -1$ and $\Delta g(\pi, \pi\beta) = -2$. ◀

We handle the case where $\pi_1 = 1$ in the proof of our upper bound below.

▶ **Theorem 5.** *For any π in S_n, we have $pbid(\pi) \leq g(\pi)$.*

Proof. If $\pi_1 \neq 1$, then we apply Lemma 4 to decrease $g(\pi)$ by at least 1. Otherwise, $\{\pi'_0, \pi'_1\}$ is a 1-cycle in $G(\pi)$ and $f(\pi) = 0$. Assume $\pi \neq \iota$ to avoid triviality; then $G(\pi)$ contains a nontrivial cycle, from which we select a grey edge $\{\pi'_{2i-2}, \pi'_{2j-1}\}$ with $j > i$. Applying the prefix block-interchange $\beta(1, i, i, j)$ then makes π_i and $\pi_i + 1$ contiguous in $\pi\beta$, and that pair corresponds to a new 1-cycle in $G(\pi\beta)$. On the other hand, β merges the 1-cycle $\{\pi'_0, \pi'_1\}$ in $G(\pi)$ with the cycle that contains $\{\pi'_{2i-2}, \pi'_{2j-1}\}$, so $\Delta c(\pi, \pi\beta) = 0 = \Delta c_1(\pi, \pi\beta)$, $\Delta f(\pi, \pi\beta) = 1$ and Equation 2 yields $\Delta g(\pi, \pi\beta) = 0/2 - 0 - 1 = -1$. ◀

The smallest example of a permutation for which the inequality in Theorem 5 is strict is $\pi = \langle 3\ 2\ 1 \rangle$, with $pbid(\pi) = 1 < g(\pi) = 2$. Algorithm 1 implements the strategy described in Theorem 5. We prove in the next subsection that Algorithm 1 is a 2-approximation.

Algorithm 1 ApproximateSbpbi(π).

Input: A permutation π of $[n]$.
Output: A sorting sequence of prefix block-interchanges for π.

$S \leftarrow$ empty sequence;
while $\pi \neq \iota$ **do**
  if $\pi_1 \neq 1$ **then**
    $j \leftarrow$ the position of $\pi_1 - 1$;
    if *there exists a pair* $(\pi_i, \pi_k = \pi_i + 1)$ *such that* $i \leq j \leq k$ *and the corresponding grey arc belongs to the leftmost cycle of* $G(\pi)$ **then**
      $\sigma \leftarrow \beta(1, i, j, k)$; `// Lemma 4 cases 1-3`
    else
      $i, k \leftarrow$ positions such that $i \leq j \leq k$ and $\pi_k = \pi_i - 1$;
      $\sigma \leftarrow \beta(1, i, j, k)$; `// Lemma 4 case 4`
  else `// Theorem 5`
    $i \leftarrow$ smallest index such that $\pi_{i+1} \neq \pi_i + 1$;
    $j \leftarrow$ the position of $\pi_i + 1$;
    $\sigma \leftarrow \beta(1, i, i, j)$;
  $\pi \leftarrow \pi\sigma$;
  S.append(σ);
return S;

3.2 A lower bound based on the breakpoint graph

We now prove a lower bound on *pbid* that allows us to show that Algorithm 1 is a 2-approximation for SBPBI. To that end, we use a framework introduced in [19]. The starting point is the following mapping, in which the symmetric group on $[n+1]$ is identified with the symmetric group on $\{0\} \cup [n]$ and where A_n is the subgroup of S_n formed by the set of all *even* permutations, i.e. permutations with an even number of even cycles:

$$\psi : S_n \to A_{n+1} : \pi \mapsto \overline{\pi} = (0, 1, 2, \ldots, n)(0, \pi_n, \pi_{n-1}, \ldots, \pi_1). \tag{3}$$

This mapping associates to every permutation π another permutation $\overline{\pi}$ whose disjoint cycles are in one-to-one correspondence with the cycles of $G(\pi)$. As a result, terminology based on the disjoint cycle decomposition of $\overline{\pi}$ or on the alternating cycle decomposition of $G(\pi)$ can conveniently be used indifferently, including the notation introduced at the beginning of Section 3 (e.g. $c(\overline{\pi}) = c(G(\pi))$, and therefore $\Delta c(\overline{\pi}, \overline{\pi\sigma}) = c(\overline{\pi\sigma}) - c(\overline{\pi}) = c(G(\pi\sigma)) - c(G(\pi)) = \Delta c(\pi, \pi\sigma)$). The following result will be our main tool for proving our lower bound.

▶ **Theorem 6** ([19]). *Let S be a subset of S_n whose elements are mapped by $\psi(\cdot)$ onto $S' \subseteq A_{n+1}$. Moreover, let $\mathcal{C}$ be the union of the conjugacy classes (of S_{n+1}) that intersect with S'; then for any π in S_n, any factorisation of π into t elements of S yields a factorisation of $\overline{\pi}$ into t elements of $\mathcal{C}$.*

Consequently, if we let $d_S(\sigma)$ denote the length of a shortest sorting sequence for σ consisting solely of elements from S, then Theorem 6 implies that for any π in S_n and any choice of $S \subseteq S_n$, we have $d_S(\pi) \geq d_{S'}(\overline{\pi})$. In order to use Theorem 6, we need a translation of the effect of an operation on π in terms of a transformation on $\overline{\pi}$, as well as a precise characterisation of the image of a prefix block-interchange under the mapping ψ. Both are provided, respectively, by the following results.

▶ **Lemma 7** ([19]). *For all π, σ in S_n, we have $\overline{\pi\sigma} = \overline{\pi}(\overline{\sigma}^{\pi})$.*

▶ **Lemma 8** ([19]). *For any block-interchange $\beta(i,j,k,\ell)$ in S_n, we have $\overline{\beta(i,j,k,\ell)} = (j,\ell)(i,k)$.*

As is well-known, a 2-cycle in a permutation σ containing elements from different cycles in a permutation π merges those cycles in $\pi\sigma$, while a 2-cycle in σ containing elements from the same cycle in π splits that cycle into two cycles in $\pi\sigma$. Lemma 7 and Lemma 8 therefore provide us with a very simple way of analysing the effects of a block-interchange: the effect of β on the cycles of $G(\pi)$ is the same as the effect of $\overline{\beta}^{\pi}$ on the cycles of $\overline{\pi}$, and therefore bounds on the (prefix) block-interchange distance of π can be obtained by studying the effects of pairs of 2-cycles on $\overline{\pi}$. The following lemma will be useful in restricting the number of cases in the proof of our lower bound (Theorem 11).

▶ **Lemma 9.** *For any π in S_n and any block-interchange β, we have $\Delta c(\pi, \pi\beta) \in \{-2, 0, 2\}$.*

Proof. By Lemma 8, $\overline{\beta}$ consists of two 2-cycles, each of which might split a cycle into two cycles or merge two cycles into one (Lemma 7). Combining all possible cases yields the set $\{-2, 0, 2\}$ as possible values for $\Delta c(\overline{\pi}, \overline{\pi\beta}) = \Delta c(\pi, \pi\beta)$. ◀

Finally, the following technical observation will be useful in ruling out impossible values for $\Delta f(\pi, \sigma)$, whose set of possible values is $\{-1, 0, 1\}$ when no restrictions apply.

▶ **Lemma 10.** *For any π in S_n and every prefix block-interchange β: if $\Delta c_1(\pi, \pi\beta) \geq 2$, then $\Delta f(\pi, \pi\beta) \neq 1$.*

Proof. If $\Delta c_1(\pi, \pi\beta) \geq 2$, then the new 1-cycles are obtained in one of the following ways:

1. if at least one of them is the result of a split of the leftmost cycle of $G(\pi)$, then that cycle is nontrivial and therefore $f(\pi) = 1$, thereby forbidding the value $\Delta f(\pi, \pi\beta) = 1$;
2. otherwise, all new 1-cycles are extracted from a cycle in $G(\pi)$ other than the leftmost cycle; since that cycle can only be split into at most two new cycles (Lemma 7 and Lemma 8), we have $\Delta c_1(\pi, \pi\beta) \leq 2$ in this case. Moreover, we also have $\pi_1 = 1$, otherwise the 1-cycle containing π_1 would vanish in $G(\pi\beta)$ and contradict our assumption that $\Delta c_1(\pi, \pi\beta) \geq 2$. Therefore, the value $\Delta f(\pi, \pi\beta) = 1$ is also excluded in this case. ◀

We now have everything we need to prove our lower bound on *pbid*.

▶ **Theorem 11.** *For any π in S_n, we have $pbid(\pi) \geq g(\pi)/2$.*

Proof. By Theorem 6 and Lemma 8, we have $pbid(\pi) \geq d(\overline{\pi})$, where $d(\overline{\pi})$ is the length of a shortest sorting sequence for $\overline{\pi}$ where the only nontrivial cycles of each transformation in the sequence are two 2-cycles, exactly one of which contains 1. As a result, any lower bound on $d(\overline{\pi})$ is a lower bound on $pbid(\pi)$, and therefore we only need to show that a transformation of the kind we have just described can decrease the value of $g(\pi)$ by at most 2.

Let $\overline{\beta} = (1, a)(b, c)$ be the image of a prefix block-interchange under the mapping $\psi(\cdot)$. By Lemma 9, we only need to distinguish between the following three cases; in each situation, we aim to minimise the value of $\Delta g(\overline{\pi}, \overline{\pi\beta})$.

1. If $\Delta c(\overline{\pi}, \overline{\pi\beta}) = -2$, then clearly $\Delta c_1(\overline{\pi}, \overline{\pi\beta}) \leq 0$, and Equation 2 allows us to conclude that $\Delta g(\overline{\pi}, \overline{\pi\beta}) \geq -1 - 0 - 1 = -2$.
2. If $\Delta c(\overline{\pi}, \overline{\pi\beta}) = 0$, then either 2-cycle of β merges two cycles while the other splits a cycle into two. The lengths of the involved cycles in $\overline{\pi}$ and in $\overline{\pi\beta}$ may vary, but this observation is enough to deduce that $\Delta c_1(\overline{\pi}, \overline{\pi\beta}) \leq 2$. The lowest value of $\Delta g(\overline{\pi}, \overline{\pi\beta})$ is obtained when $\Delta c_1(\overline{\pi}, \overline{\pi\beta}) = 2$, in which case Equation 2 and Lemma 10 yield $\Delta g(\overline{\pi}, \overline{\pi\beta}) \geq 0-2-0 = -2$, or when $\Delta c_1(\overline{\pi}, \overline{\pi\beta}) = 1$, in which case Equation 2 yields $\Delta g(\overline{\pi}, \overline{\pi\beta}) \geq 0 - 1 - 1 = -2$.
3. If $\Delta c(\overline{\pi}, \overline{\pi\beta}) = 2$, then both elements of β each split one cycle into two cycles. As in the previous case, the lengths of the involved cycles in $\overline{\pi}$ and in $\overline{\pi\beta}$ may vary, but this observation is enough to deduce that $\Delta c_1(\overline{\pi}, \overline{\pi\beta}) \leq 4$, and as a result $\Delta f(\overline{\pi}, \overline{\pi\beta}) \in \{-1, 0\}$ (Lemma 10). The lowest value of $\Delta g(\overline{\pi}, \overline{\pi\beta})$ is obtained in two cases:
   a. when $\Delta c_1(\overline{\pi}, \overline{\pi\beta}) = 4$, in which case the leftmost cycle of $\overline{\pi}$ splits into two 1-cycles; therefore $\Delta f(\overline{\pi}, \overline{\pi\beta}) = -1$ and Equation 2 yields $\Delta g(\overline{\pi}, \overline{\pi\beta}) \geq 1 - 4 + 1 = -2$;
   b. or when $\Delta c_1(\overline{\pi}, \overline{\pi\beta}) = 3$, in which case Equation 2 and Lemma 10 yield $\Delta g(\overline{\pi}, \overline{\pi\beta}) \geq 1 - 3 + 0 = -2$. ◀

Theorem 11 implies that Algorithm 1 is a 2-approximation for SBPBI.

4 Tightening the bounds

Although obtaining better approximation guarantees for SBPBI seems as nontrivial as for other prefix sorting problems, the bounds obtained in the previous section can be improved. We show in this section how to tighten them, and then use those improved results in Section 5 to compute the maximal value that the distance can reach.

4.1 A tighter upper bound

By Theorem 11, the largest value by which the upper bound of Theorem 5 can decrease with a single prefix block-interchange is 2. In this section, we characterise all permutations which admit such a prefix block-interchange. Other nontight permutations exist (see e.g. Proposition 14), but they do not admit such an operation as the first step of an optimal sorting sequence. As a consequence, we obtain an improved upper bound on *pbid* in Theorem 15.

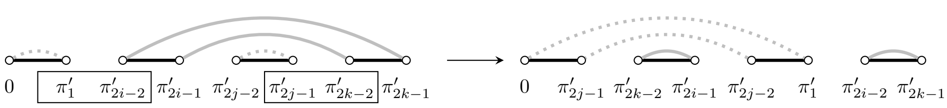
▶ **Lemma 12.** *For any π in S_n: if $G(\pi)$ contains a 2-cycle that intersects the first grey edge, then there exists a prefix block-interchange β such that $\Delta g(\pi, \pi\beta) = -2$.*

Proof. Follows from cases 4b and 4c of the proof of Lemma 4, when the cycle that contains grey edge f has length 2. ◀

Following [2], we say that a cycle C with b_i and b_k as black edges of minimum and maximum indices, respectively, *spans* a black edge b_j if $i < j < k$.

▶ **Lemma 13.** *For any π in S_n: if $G(\pi)$ contains a 2-cycle which is not the leftmost cycle and which spans a black edge that belongs to a nontrivial cycle different from the leftmost cycle, then there exists a prefix block-interchange β such that $\Delta g(\pi, \pi\beta) = -2$.*

Proof. We apply a prefix block-interchange defined by the first black edge, both black edges of the 2-cycle, and any black edge spanned by the 2-cycle:



The number of cycles does not change, so $\Delta c(\pi, \pi\beta) = 0$. Either $\pi_1 = 1$, and then $\Delta c_1(\pi, \pi\beta) = 1$ and $\Delta f(\pi, \pi\beta) = 1$; or $\pi_1 \neq 1$, and then $\Delta c_1(\pi, \pi\beta) = 2$ and $\Delta f(\pi, \pi\beta) = 0$. In both cases, Equation 2 yields $\Delta g(\pi, \pi\beta) = -2$. ◀

2-cycles other than the leftmost cycle and in a different configuration from our characterisations are still helpful. We show that even though they do not allow a prefix block-interchange that decreases $g(\cdot)$ by 2 right away, they make it possible to obtain such an operation *eventually*.

▶ **Proposition 14.** *For any π in S_n: if $G(\pi)$ contains a 2-cycle which is not the leftmost cycle, then π admits a sequence S of prefix-block interchanges that turns π into a permutation σ with $\Delta g(\pi, \sigma) = |S|$ and which admits a prefix block-interchange β such that $\Delta g(\sigma, \sigma\beta) = -2$.*

Proof. Let C denote the 2-cycle of interest. If C intersects the first grey edge or a cycle different from the leftmost cycle, then we are done (see respectively Lemma 12 and Lemma 13). Otherwise, C intersects another grey edge of the leftmost cycle, and Lemma 4 allows us to reduce $g(\pi)$ by one while reducing the length of the leftmost cycle without affecting C. Repeated applications of Lemma 4 eventually yield a permutation σ which satisfies one of the following conditions:

1. $\sigma_1 = 1$, in which case C necessarily spans a black edge that does not belong to the leftmost cycle and therefore we can apply Lemma 13;
2. $\sigma_1 \neq 1$ and C intersects another cycle than the leftmost cycle, in which case we can again apply Lemma 13; or
3. $\sigma_1 \neq 1$ and C intersects the first grey edge, in which case we can apply Lemma 12. ◀

The above results allow us to easily identify other nontight permutations (with respect to Theorem 5) whose breakpoint graph contains no 2-cycle. For instance, if the first grey edge intersects a 3-cycle C, then applying a prefix-block interchange selected according to Lemma 12 decreases the lengths of both the leftmost cycle and C, which becomes a 2-cycle and which therefore eventually allows for a prefix block-interchange that decreases $g(\cdot)$ by 2 according to Proposition 14.

The interactions between 2-cycles prevent us from simply reducing the upper bound of Theorem 5 by the number of 2-cycles in $G(\pi)$: indeed, the black edge spanned by the 2-cycle described in Lemma 13 may belong to a 2-cycle whose length will increase in the resulting permutation. Therefore, we can only conclude the following.

▶ **Theorem 15.** *For any π in S_n, we have $pbid(\pi) \leq g(\pi) - \lceil c_2^{\emptyset}(G(\pi))/2 \rceil$, where $c_2^{\emptyset}(G(\pi))$ denotes the number of 2-cycles in $G(\pi)$ excluding the leftmost cycle.*

Proof. We repeatedly apply Proposition 14 to take advantage of suitable 2-cycles. Each prefix block-interchange we use transforms a 2-cycle into two 1-cycles without affecting the other 2-cycles, except possibly in the case of Lemma 13 when the edge spanned by the 2-cycle we focus on belongs to another 2-cycle. In the worst case, every 2-cycle we try to split forces us to increase the length of a 2-cycle it intersects, hence the improvement of only $\lceil c_2^{\emptyset}(G(\pi))/2 \rceil$ over Theorem 5. ◀

Theorem 15 again yields a tight upper bound, as shown by the permutation $\langle 1\ 4\ 3\ 2 \rangle$ for which the value of the improved upper bound matches its distance.

4.2 A tighter lower bound

A trivial lower bound on *pbid* is given by the value of the block-interchange distance (denoted by $bid(\pi)$), which can be computed in $O(n)$ time thanks to the following result.

▶ **Theorem 16** ([10]). *For any π in S_n, we have $bid(\pi) = (n + 1 - c(G(\pi)))/2$.*

This lower bound often outperforms that of Theorem 11, but cases exist where the opposite holds ($\langle 1\ 4\ 3\ 2 \rangle$ is the smallest example). As we show below, it is possible to build on this trivial lower bound to obtain a much better and useful lower bound. The resulting lower bound also allows us to compute the maximum value that the prefix block-interchange can reach, a problem we address in Section 5.

▶ **Definition 17** ([14]). *Let π be a permutation. Two cycles C and D of $G(\pi)$* intersect *if C contains a grey edge e that intersects with a grey edge f of D. A* component *of $G(\pi)$ is a connected component of the intersection graph of the nontrivial cycles of $G(\pi)$.*

For instance, the breakpoint graph of Figure 1 page 4 has two components: the leftmost cycle, and the pair of intersecting 2-cycles. Let $CC(G(\pi))$ denote the number of components of $G(\pi)$. We show in Lemma 20 that prefix block-interchanges that merge components of the breakpoint graph cannot decrease the number of cycles it contains. To achieve this, we first show that if a prefix block-interchange β reduces the number of connected components of $G(\pi)$, then it cannot act on a single cycle of $G(\pi)$. This is important for the proof of Lemma 20, because some prefix block-interchanges acting on a single cycle increase the number of cycles and therefore may decrease the value of $bid(\pi)$ regardless of their effect on $pbid(\pi)$ or $g(\pi)$. The following concepts will be helpful.

▶ **Definition 18.** *For any permutation π, let $e = (e_1, e_2)$ and $f = (f_1, f_2)$ be two grey edges in $G(\pi)$, with $e_1 < e_2$ and $f_1 < f_2$. We say that e and f are* independent *if they are:*
- nested*, i.e. $e_1 < f_1 < f_2 < e_2$ (written $f \subset_\pi e$) or $f_1 < e_1 < e_2 < f_2$ (written $e \subset_\pi f$); or*
- ordered*, i.e. $e_1 < e_2 < f_1 < f_2$, in which case we say that e* precedes *f (written $e <_\pi f$), or $f_1 < f_2 < e_1 < e_2$ (i.e. f precedes e).*

Grey edges naturally define intervals in π', so we use the same notation to compare intervals, or grey edges with intervals. We will sometimes need to distinguish *proper* block-interchanges, i.e. of the form $\beta(i, j, k, \ell)$ with $j < k$, from prefix block-transpositions, which are of the form $\beta(i, j, j, \ell)$.

▶ **Lemma 19.** *For any π in S_n, let β be a prefix block-interchange with $CC(G(\pi\beta)) < CC(G(\pi))$; then β cannot act on a single cycle of $G(\pi)$.*

Proof. Let $e = (e_1, e_2)$ and $f = (f_1, f_2)$ with $e_1 < e_2$ and $f_1 < f_2$ be two grey edges of $G(\pi)$. We show that if e and f are independent in $G(\pi)$, then they remain independent in $G(\pi\beta)$. For readability, we assume that the indices of e and f correspond to positions in π rather than π'. The connections between the black edges of $G(\pi)$ on which $\beta(1, i, j, k)$ acts imply the following:
- both e_1 and f_1 lie in the interval $[1, k]$, otherwise β would not affect e or f;
- at least e or f has both endpoints in $[1, i]$, $[i, j]$ (which is empty if β is not proper) or $[j, k]$, otherwise they both intersect the cycle on which β acts and therefore $CC(G(\pi\beta)) \geq CC(G(\pi))$;

- if $x \subset_\pi I$ holds for all possible combinations of x in $\{e, f\}$ and I in $\{[1, i], [i, j], [j, k]\}$, then β trivially preserves the order of endpoints and therefore the interactions between e and f as well, so that their independence in $G(\pi)$ is preserved in $G(\pi\beta)$.

Without loss of generality, the only cases left to examine are those where $e \subset_\pi f$ and $e \subset_\pi [1, i]$, $e \subset_\pi [i, j]$, or $e \subset_\pi [j, k]$. The only two ways of making e and f intersect in $G(\pi\beta)$ are therefore either to exchange e_2 and f_2 without moving e_1 and f_1, which is impossible because β is a *prefix* block-interchange; or to exchange f_1 and e_1 without moving e_2 and f_2, which is impossible as well since β must act on the four black edges of the cycle. ◀

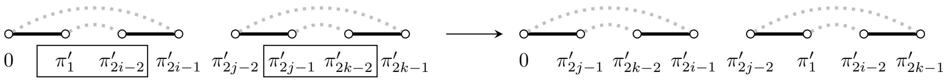
We can now prove Lemma 20.

▶ **Lemma 20.** *For any π in S_n, let β be a prefix block-interchange with $CC(G(\pi\beta)) < CC(G(\pi))$; then $\Delta c(G(\pi, \pi\beta)) \in \{-2, 0\}$.*

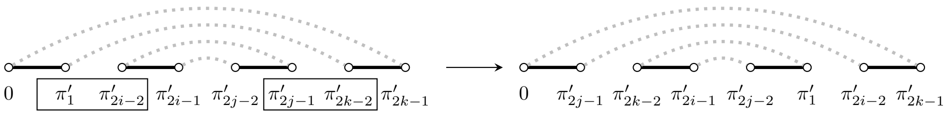
Proof. By Lemma 19, we have the following three cases to analyse:

1. if β acts on two cycles from different components, then two or three of the black edges on which β acts belong to the same cycle. In all resulting cases, we have $\Delta c(\pi, \pi\beta) = 0$; omitted cases are symmetric, and only proper block-interchanges are considered since the property we seek to prove is already known to hold for block-transpositions (see [2, Lemma 3.2 page 228]):

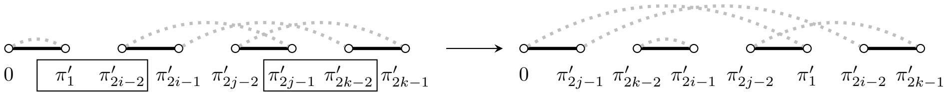
a.



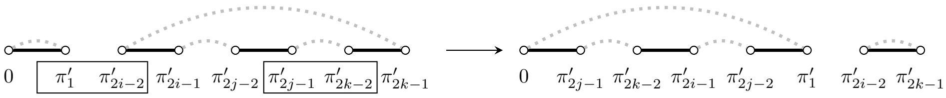
b.



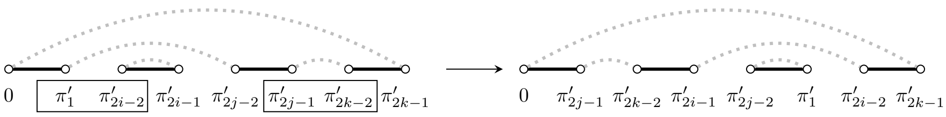
c.



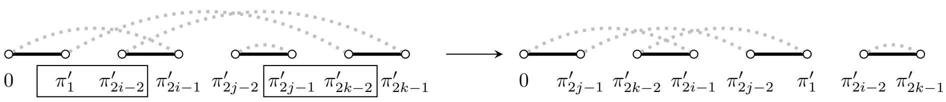
d.



e.

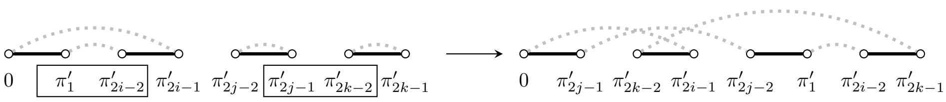


f.

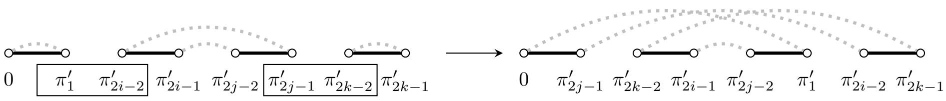


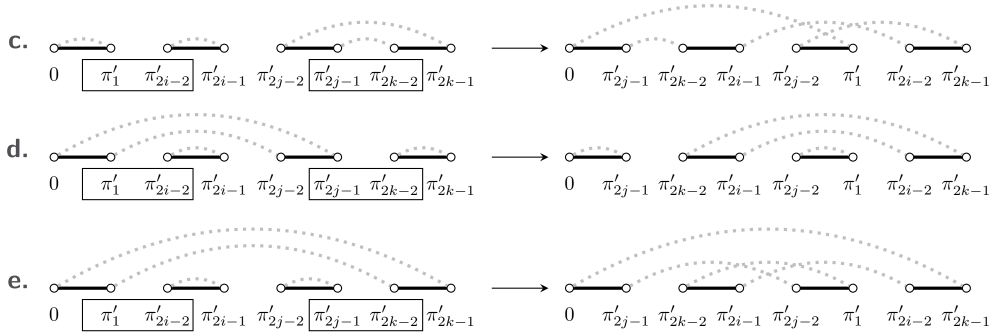
2. if β acts on three cycles, then exactly two of the black edges on which β acts belong to the same cycle. In all cases, we have $\Delta c(\pi, \pi\beta) \in \{-2, 0\}$; again, omitted cases are symmetric, and only proper block-interchanges are considered since block-transpositions acting on three cycles decrease the number of cycles by two [2, Lemma 2.1 page 227]:

a.

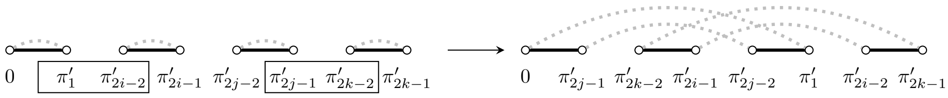


b.





3. if β acts on four cycles, then all black edges on which β acts belong to their own distinct cycle and $\Delta c(\pi, \pi\beta) = -2$:



◀

▶ **Theorem 21.** *For any π in S_n, we have* $pbid(\pi) \geq bid(\pi) + CC(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 1 & \text{otherwise.} \end{cases}$

Proof. The expression for the lower bound corresponds to the following strategy: for each component C of $G(\pi)$, sort the corresponding subpermutation if C contains 0, or use a prefix block-interchange to make it contain 0 and then sort it. Trivially, the number of steps in the sorting stage cannot be lower than the number or unrestricted block-interchanges it would require. Any other strategy would have to merge components; however, by Lemma 20, a prefix block-interchange β that merges connected components cannot increase the number of cycles, and therefore $bid(\pi\beta) \geq bid(\pi)$ for any permutation π and any such prefix block-interchange. ◀

5 The maximum value of the prefix block-interchange distance

The *diameter* of S_n is the maximum value that a distance can reach for a particular family of operations. In this section, we use our results to compute its exact value in the case of prefix block-interchanges, and show along the way that our 2-approximation algorithm based on the breakpoint graph is also a 2-approximation with respect to the following notion.

▶ **Definition 22** ([11]). *Let π be a permutation of $\{0, 1, 2, \ldots, n+1\}$ with $\pi_0 = 0$ and $\pi_{n+1} = n+1$. The pair (π_i, π_{i+1}) with $0 \leq i \leq n$ is a* breakpoint *if $i = 0$ or $\pi_{i+1} - \pi_i \neq 1$, and an* adjacency *otherwise. The number of breakpoints in a permutation π is denoted by $b(\pi)$.*

For readability, we slightly abuse notation by using $b(\pi)$ for π in S_n, with the understanding that it refers to $b(\langle 0\ \pi_1\ \pi_2\ \cdots\ \pi_n\ n+1 \rangle)$. We let $\Delta b(\pi, \sigma) = b(\sigma) - b(\pi)$, and say that a prefix block-interchange β with $\Delta b(\pi, \pi\beta) < 0$ *removes* breakpoints, or *creates* adjacencies.

▶ **Lemma 23.** *For any π in S_n and any prefix block-interchange β, we have $|\Delta b(\pi, \pi\beta)| \leq 3$.*

Proof. A prefix block-interchange β acts on at most four pairs of adjacent elements, including the pair $(0, \pi_1)$ which always counts as a breakpoint. Therefore, the number of breakpoints that β can remove or create lies in the set $\{0, 1, 2, 3\}$. ◀

Since ι is the only permutation with exactly one breakpoint, Lemma 23 immediately implies the following corollary.

▶ **Corollary 24.** *For any π in S_n : $pbid(\pi) \geq \left\lceil \frac{b(\pi)-1}{3} \right\rceil$.*

▶ **Lemma 25.** *For any π in S_n, we have $pbid(\pi) \leq 2\left\lceil \frac{b(\pi)-1}{3} \right\rceil$.*

Proof. Assume $\pi \neq \iota$ to avoid triviality, and observe that adjacencies in $\langle 0\ \pi_1\ \pi_2\ \cdots\ \pi_n\ n{+}1\rangle$ are in one-to-one correspondence with trivial cycles in $G(\pi)$ (except for the pair $(0, \pi_1)$ which by Definition 22 is always a breakpoint). If $\pi_1 \neq 1$, then Lemma 4 guarantees the existence of a prefix block-interchange β with $\Delta c_1(\pi, \pi\beta) \geq 2$ and in turn implies $\Delta b(\pi, \pi\beta) \geq 2$. If $\pi_1 = 1$, then we select β as in the proof of Theorem 5, which creates a new trivial cycle in $G(\pi\beta)$ that corresponds to a new adjacency in $\pi\beta$. Since $\pi\beta_1 \neq 1$, the previous case provides the next operation, and the number of breakpoints decreases by at least three using two prefix block-interchanges. ◀

Since $b(\pi) \leq n+1$ for all π in S_n, we immediately obtain the following.

▶ **Corollary 26.** *For any $\pi \in S_n$, we have $pbid(\pi) \leq 2n/3$.*

▶ **Theorem 27.** *The diameter of S_n under prefix block-interchanges is $\lfloor 2n/3 \rfloor$.*

Proof. The cases where $n \leq 2$ are easily verified. We build tight families of permutations for any $n \geq 3$, starting with permutations $\pi = \langle 1\ 3\ 2\rangle$, $\sigma = \langle 1\ 4\ 3\ 2\rangle$, and $\tau = \langle 1\ 3\ 2\ 5\ 4\rangle$ as base cases for the values of n, $n-1$ and $n-2$ that are multiples of 3, respectively. Theorem 21 yields $pbid(\pi) \geq 2$, $pbid(\sigma) \geq 2$ and $pbid(\tau) \geq 3$, while Corollary 26 yields $pbid(\pi) \leq 2$, $pbid(\sigma) < 3$ and $pbid(\tau) < 4$, thereby matching the lower bounds.

To obtain tight permutations for larger values of n, we concatenate the sequence $\langle n+1\ \ n+3\ \ n+2\rangle$ to π, σ or τ, and repeat the process as many times as needed. Each concatenation preserves the congruence of n and adds a new component to $G(\cdot)$ which consists of an isolated 3-cycle. The lower bound of Theorem 21 thereby increases by 2 with each concatenation, as does the upper bound of Corollary 26. As a result, a permutation with prefix block-interchange distance $\lfloor 2n/3 \rfloor$ exists for every value of n in $\mathbb{N}$. ◀

While many permutations reach the diameter when $n \not\equiv 0 \pmod 3$, the permutation $\langle 1\ 3\ 2\ 4\ 6\ 5\ \cdots\ n-2\ n\ n-1\rangle$ seems to be the only tight permutation when $n \equiv 0 \pmod 3$.

6 Conclusions and future work

We initiated in this work the study of sorting permutations by prefix block-interchanges, an operation that generalises several well-studied operations in genome rearrangements and interconnection network design. We gave tight upper and lower bounds on the corresponding distance, and derived a 2-approximation algorithm for the problem. We then showed how to obtain better bounds on the distance using a finer analysis of cycles and components of the breakpoint graph, and determined the maximum value that the distance can reach.

Several questions remain open, most notably the complexity of SBPBI, and its approximability if it turns out to be NP-complete. We note that improving the ratio of 2 will require improved lower bounds, since for all three upper bounds we have obtained (Theorem 5, Theorem 15 and Lemma 25) there are permutations whose actual distance match those bounds. A number of leads seem promising in that regard, the most obvious one being the computation of the exact value of the "special purpose distance" introduced in the proof of Theorem 11, as well as a more intricate analysis of the cycles of the breakpoint graph as well as their interactions as initiated in Section 4. Given how helpful 2-cycles are in decreasing the upper bound of Theorem 5, it would seem natural to focus on *simple permutations* (i.e. permutations whose breakpoint graph contains no cycle of length > 2). This strategy eventually led to a polynomial-time algorithm for sorting by signed reversals [14], but we do not expect such an outcome for prefix block-interchanges since the simplification process does not preserve the prefix block-interchange distance (whereas it did preserve the signed reversal distance): the smallest counterexample is $\pi = \langle 3\ 1\ 4\ 2 \rangle$, which simplifies to $\sigma = \langle 5\ 2\ 7\ 4\ 1\ 6\ 3 \rangle$, and for which $pbid(\pi) = 2 \neq pbid(\sigma) = 3$.

In a broader context, we also hope that our results and the strategies we designed to tackle SBPBI can be applied to other prefix sorting problems (for instance, a generalisation of the lower bounding strategy of Theorem 21 to any distance would be of interest). The breakpoint graph approach provides a clear strategy for unrestricted sorting problems, which, informally, usually consists in increasing the number of cycles in as few steps as possible. As our bounds show, and as has been observed for most prefix sorting problems [1, 19, 20], this no longer works under the prefix constraint since operations that *decrease* or do not affect the number of cycles can also decrease the value of our bounds. Nevertheless, bounds obtained for prefix exchanges, prefix block-transpositions, prefix block-interchanges and prefix signed reversals are all based on $g(\cdot)$, which seems to indicate common underlying features that could be taken advantage of, and possibly lead to a common framework for approximating these problems or solving them exactly.

References

1. Sheldon B. Akers, Balakrishnan Krishnamurthy, and Dov Harel. The star graph: An attractive alternative to the n-cube. In *Proceedings of the Fourth International Conference on Parallel Processing*, pages 393–400. Pennsylvania State University Press, August 1987. doi:10.5555/201173.201191.
2. Vineet Bafna and Pavel A. Pevzner. Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240 (electronic), May 1998. doi:10.1137/S089548019528280X.
3. Piotr Berman, Sridhar Hannenhalli, and Marek Karpinski. 1.375-approximation algorithm for sorting by reversals. In Rolf H. Möhring and Rajeev Raman, editors, *Proceedings of the 10th Annual European Symposium on Algorithms*, volume 2461 of *Lecture Notes in Computer Science*, pages 200–210, Rome, Italy, September 2002. Springer. doi:10.1007/3-540-45749-6_21.
4. Laurent Bulteau, Guillaume Fertin, and Irena Rusu. Sorting by transpositions is difficult. *SIAM Journal on Discrete Mathematics*, 26(3):1148–1180, 2012. doi:10.1137/110851390.
5. Laurent Bulteau, Guillaume Fertin, and Irena Rusu. Pancake flipping is hard. *Journal of Computer and System Sciences*, 81(8):1556–1574, 2015. doi:10.1016/j.jcss.2015.02.003.
6. Liming Cai, Jianer Chen, Rodney G. Downey, and Michael R. Fellows. On the parameterized complexity of short computation and factorization. *Archive for Mathematical Logic*, 36(4-5):321–337, 1997. doi:10.1007/s001530050069.
7. Alberto Caprara. Sorting permutations by reversals and Eulerian cycle decompositions. *SIAM Journal on Discrete Mathematics*, 12(1):91–110 (electronic), January 1999. doi:10.1137/S089548019731994X.
8. Xin Chen. On sorting unsigned permutations by double-cut-and-joins. *Journal of Combinatorial Optimization*, 25(3):339–351, 2013. doi:10.1007/s10878-010-9369-8.
9. Shih-Wen Chou, Chung-Han Yang, Kun-Tze Chen, and Chin Lung Lu. Prefix block-interchanges on binary strings. In William Cheng-Chung Chu, Han-Chieh Chao, and Stephen Jenn-Hwa Yang, editors, *Proceedings of the International Computer Symposium on Intelligent Systems and Applications*, volume 274 of *Frontiers in Artificial Intelligence and Applications*, pages 1960–1969, Taichung, Taiwan, December 2014. IOS Press. doi:10.3233/978-1-61499-484-8-1960.

10 David A. Christie. Sorting permutations by block-interchanges. *Information Processing Letters*, 60(4):165–169, 1996. doi:10.1016/S0020-0190(96)00155-X.

11 Zanoni Dias and João Meidanis. Sorting by prefix transpositions. In Alberto H. F. Laender and Arlindo L. Oliveira, editors, *Proceedings of the Ninth International Symposium on String Processing and Information Retrieval*, volume 2476 of *Lecture Notes in Computer Science*, pages 65–76, Lisbon, Portugal, September 2002. Springer-Verlag. doi:10.1007/3-540-45735-6_7.

12 Isaac Elias and Tzvika Hartman. A 1.375-approximation algorithm for sorting by transpositions. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(4):369–379, 2006. doi:10.1109/TCBB.2006.44.

13 Guillaume Fertin, Anthony Labarre, Irena Rusu, Eric Tannier, and Stéphane Vialette. *Combinatorics of Genome Rearrangements*. Computational Molecular Biology. The MIT Press, 2009. URL: https://mitpress.mit.edu/books/combinatorics-genome-rearrangements.

14 Sridhar Hannenhalli and Pavel A. Pevzner. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46(1):1–27, 1999. doi:10.1145/300515.300516.

15 Mark R. Jerrum. The complexity of finding minimum-length generator sequences. *Theoretical Computer Science*, 36(2-3):265–289, June 1985. doi:10.1016/0304-3975(85)90047-7.

16 D. J. Kleitman, Edvard Kramer, J. H. Conway, Stroughton Bell, and Harry Dweighter. Elementary problems: E2564-E2569. *The American Mathematical Monthly*, 82(10):1009–1010, 1975. doi:10.2307/2318260.

17 Donald E. Knuth. *Sorting and Searching*, volume 3 of *The art of Computer Programming*. Addison-Wesley, 1995.

18 Anthony Labarre. New bounds and tractable instances for the transposition distance. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(4):380–394, October 2006. doi:10.1109/TCBB.2006.56.

19 Anthony Labarre. Lower bounding edit distances between permutations. *SIAM Journal on Discrete Mathematics*, 27(3):1410–1428, 2013. doi:10.1137/13090897X.

20 Anthony Labarre and Josef Cibulka. Polynomial-time sortable stacks of burnt pancakes. *Theoretical Computer Science*, 412(8-10):695–702, March 2011. doi:10.1016/j.tcs.2010.11.004.

21 S. Lakshmivarahan, Jung-Sing Jwo, and S. K. Dhall. Symmetry in interconnection networks based on Cayley graphs of permutation groups: A survey. *Parallel Computing*, 19(4):361–407, April 1993. doi:10.1016/0167-8191(93)90054-O.

22 Sophia Yancopoulos, Oliver Attie, and Richard Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16):3340–3346, 2005. doi:10.1093/bioinformatics/bti535.