Theoretical Computer Science 412 (2011) 695–702

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Polynomial-time sortable stacks of burnt pancakes

Anthony Labarre [a,*], Josef Cibulka [b]

[a] Department of Computer Science, K. U. Leuven, Celestijnenlaan 200A - bus 2402, 3001 Heverlee, Belgium

[b] Department of Applied Mathematics, Charles University, Malostranské nám. 25, 118 00 Prague, Czech Republic

ARTICLE INFO

Article history:
Received 5 March 2009
Received in revised form 2 October 2010
Accepted 1 November 2010
Communicated by A. Apostolico

Keywords:
Algorithms
Combinatorial problems
Interconnection networks
Sorting
Permutations

ABSTRACT

Pancake flipping, a famous open problem in computer science, can be formalised as the problem of sorting a permutation of positive integers using as few *prefix reversals* as possible. In that context, a prefix reversal of length k reverses the order of the first k elements of the permutation. The *burnt* variant of pancake flipping involves permutations of *signed* integers, and reversals in that case not only reverse the order of elements but also invert their signs. Although three decades have now passed since the first works on these problems, neither their computational complexity nor the maximal number of prefix reversals needed to sort a permutation is yet known. In this work, we prove a new lower bound for sorting burnt pancakes, and show that an important class of permutations, known as "simple permutations", can be optimally sorted in polynomial time.

1. Introduction

The pancake flipping problem [1] consists in finding the minimum number of flips required to rearrange a stack of pancakes that all come in different sizes so that the smallest ends up on top and the largest lies at the bottom. The problem can be more formally stated as follows: given an ordering of $\{1, 2, \ldots, n\}$, what is the minimum number of *prefix reversals* required to sort these numbers in increasing order (where a prefix reversal of length k reverses the order of the first k elements)? A variant of the problem, known as the *burnt pancake flipping problem*, is concerned with rearranging stacks of pancakes that are burnt on one side, in such a way that the pancakes not only end up rearranged in increasing sizes but also with their burnt side down. Again, a more formal description of the problem is to sort orderings of $\{\pm 1, \pm 2, \ldots, \pm n\}$ using as few prefix *signed* reversals (which not only reverse the order of the first k elements but also invert their signs) as possible.

Gates and Papadimitriou [2] and Györi and Turán [3] proved the first results on pancake flipping three decades ago, focusing on the number of prefix reversals needed in the worst case (i.e. the maximum number of steps required to sort a stack of size n), and a tremendous amount of work (see next paragraph for more details) has since been devoted to the study of both variants of the problem. However, the computational complexity of the sorting problems or merely computing the minimum number of required steps remains open, as well as that of determining the maximum number of prefix (signed) reversals needed to sort a permutation. The best known approximation ratio is 2, both in the unsigned case (see [4]) and in the signed case (see [5], according to [4]). A very interesting and original solution to the burnt pancake flipping problem was recently proposed by Haynes et al. [6], who use bacteria to represent permutations, which eventually become antibiotic resistant when they are sorted. However, their model obviously does not yield any combinatorial or algorithmic insight on pancake flipping, and seems therefore of little theoretical help.

* Corresponding author.
E-mail address: Anthony.Labarre@cs.kuleuven.be (A. Labarre).

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.11.004

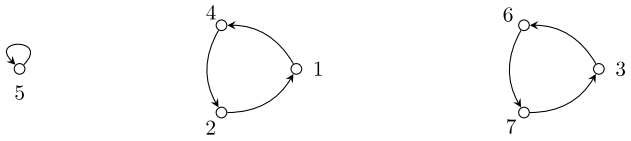

Fig. 1. The graph of the permutation $\langle 4\ 1\ 6\ 2\ 5\ 7\ 3\rangle = (1, 4, 2)(3, 6, 7)(5)$.

Although pancake flipping was introduced as a game, it is worth noting that it has since found applications in parallel computing, leading to the famous "(burnt) pancake network" topology which is the Cayley graph of a permutation group generated by prefix (signed) reversals (see [7] for a thorough survey on the use of Cayley graphs as interconnection networks). Another major application of pancake flipping, which has received considerable attention, is in the field of computational biology, where permutations model genomes and reversals correspond to actual mutations by which those genomes evolve. In that setting, reversals are no longer restricted to the prefix of the permutation: they can act on any of its segments. Interestingly enough, more is known about these seemingly more challenging versions than on the original pancake flipping problems: Caprara [8] proved that sorting unsigned permutations by arbitrary reversals was NP-hard, but quite surprisingly, Hannenhalli and Pevzner [9] proved that the signed version of this problem could be solved in polynomial time. For an extensive survey of the mathematical aspects of genome comparisons by means of large-scale mutations, known as *genome rearrangements*, see for instance [10].

In this paper, we use ideas introduced in a previous paper [11] to prove a new tight lower bound on the minimum number of prefix signed reversals required to sort a permutation, which is also referred to as their *distance*. We also examine an important class of signed permutations, known as "simple permutations", which proved crucial in solving the problem of sorting permutations by unrestricted signed reversals in polynomial time (see [9]), and give a polynomial-time algorithm for sorting these permutations optimally, as well as a formula for computing their distance in polynomial time.

2. Background

2.1. Permutations

Let us start with a quick reminder of basic notions on permutations (for more details, see e.g. [12,13]).

Definition 1. A *permutation* of $\{1, 2, \ldots, n\}$ is a bijective application of $\{1, 2, \ldots, n\}$ onto itself.

The *symmetric group* S_n is the set of all permutations of $\{1, 2, \ldots, n\}$, together with the usual function composition $\circ$, applied from right to left. We use lower case Greek letters to denote permutations, typically $\pi = \langle \pi_1\ \pi_2\ \cdots\ \pi_n\rangle$, with $\pi_i = \pi(i)$, and in particular write the *identity permutation* as $\iota = \langle 1\ 2\ \cdots\ n\rangle$.

Definition 2. The *graph* $\Gamma(\pi)$ of a permutation π has vertex set $\{1, 2, \ldots, n\}$, and contains an arc (i, j) whenever $\pi_i = j$.

As Fig. 1 shows, $\Gamma(\pi)$ decomposes in a unique 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 $\pi = \langle 4\ 1\ 6\ 2\ 5\ 7\ 3\rangle$, the disjoint cycle notation is $\pi = (1, 4, 2)(3, 6, 7)(5)$ (notice the parentheses and the commas).

The number of cycles in $\Gamma(\pi)$ is denoted by $c(\Gamma(\pi))$, and the *length* of a cycle is the number of elements it contains. A *k-cycle* in $\Gamma(\pi)$ is a cycle of length k.

Definition 3. A *signed permutation* is a permutation of $\{1, 2, \ldots, n\}$ where each element has an additional "+" or "−" sign.

The *hyperoctahedral group* $S_n^{\pm}$ is the set of all signed permutations of n elements with the usual function composition (the usual convention is that $\pi_{-i} = -\pi_i$). It is not mandatory for a signed permutation to have negative elements, so $S_n \subset S_n^{\pm}$ since each permutation in S_n can be viewed as a signed permutation without negative elements in $S_n^{\pm}$. To lighten the presentation, we will conform to the tradition of omitting "+" signs when elements are positive.

2.2. Operations on permutations

We now review a number of operations on permutations, which are themselves modelled as permutations.

Definition 4. An *exchange* $\varepsilon(i, j)$ with $1 \le i < j \le n$ is a permutation that swaps elements in positions i and j:

$$\varepsilon(i,j) = \begin{pmatrix} 1 \cdots i-1 \boxed{i} i+1 \cdots j-1 \boxed{j} j+1 \cdots n \\ 1 \cdots i-1 \boxed{j} i+1 \cdots j-1 \boxed{i} j+1 \cdots n \end{pmatrix}.$$

Definition 5. A *reversal* $\rho(i, j)$ with $1 \le i < j \le n$ is a permutation that reverses the order of elements between positions i and j:

$$\rho(i,j) = \begin{pmatrix} 1 \cdots i-1\, \underline{i\ i+1 \cdots j-1\, j}\, j+1 \cdots n \\ 1 \cdots i-1\, j\, j-1 \cdots i+1\ i\, j+1 \cdots n \end{pmatrix}.$$

Table 1
Some results on sorting permutations using various operations.

	Operation	Sorting	Distance	Best approximation
	Exchange	$O(n)$ [24]		1
	Reversal	NP-hard [8]		11/8 [25]
	Signed reversal	$O(n^{3/2})$ [20]	$O(n)$ [21]	1
Prefix	Exchange	$O(n)$ [16]		1
Prefix	Reversal	?	?	2 [4]
Prefix	Signed reversal	?	?	2 [5]

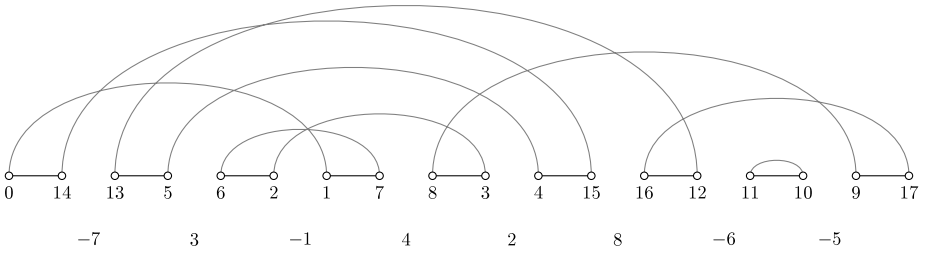

Fig. 2. The breakpoint graph of $\langle -7\ 3\ -1\ 4\ 2\ 8\ -6\ -5\rangle$.

Definition 6. A *signed reversal* $\overline{\rho}(i,j)$ with $1 \le i \le j \le n$ is a permutation that reverses both the order and the signs of elements between positions i and j:

$$\overline{\rho}(i,j) = \begin{pmatrix} 1 \cdots i-1 & \overline{i \quad i+1 \quad \cdots \quad j-1 \quad j} & j+1 \cdots n \\ 1 \cdots i-1 & -j-(j-1) \cdots -(i+1) -i & j+1 \cdots n \end{pmatrix}.$$

Each operation σ transforms a permutation π into a permutation $\pi \circ \sigma$. Setting $i = 1$ in the above definitions turns those operations into *prefix operations*, i.e. operations whose action is restricted to the initial segment of the permutation. It can be easily seen that the effect of any operation can be mimicked by at most three prefix operations of the same kind. We are interested in the following two problems on permutations.

Definition 7. Given a permutation π in $S_n^{\pm}$ and a set $X \subseteq S_n^{\pm}$ of allowed transformations, the problem of *sorting* π *by* X is that of finding a minimum-length sequence of elements of X that transforms π into ι. The *distance* of π (with respect to X) is the length of such a sequence.

The operations we have presented give rise to the exchange distance (denoted by $ed(\pi)$), the reversal distance (denoted by $rd(\pi)$) and the signed reversal distance (denoted by $srd(\pi)$), respectively, as well as the corresponding prefix variants (namely $ped(\pi)$, $prd(\pi)$ and $psrd(\pi)$). Table 1 summarises a selected portion of the current state of knowledge about these distances and the corresponding sorting problems.

2.3. The breakpoint graph

Bafna and Pevzner [14] introduced the following graph, which turned out to be an extremely useful tool to sort permutations by (possibly signed) reversals.

Definition 8. Given a signed permutation π in $S_n^{\pm}$, transform it into an unsigned permutation π' in S_{2n} by mapping π_i onto the sequence $(2\pi_i - 1, 2\pi_i)$ if $\pi_i > 0$, or $(2|\pi_i|, 2|\pi_i| - 1)$ if $\pi_i < 0$, for $1 \le i \le n$. The *breakpoint graph* of π' is the undirected bicoloured graph $BG(\pi)$ with ordered vertex set $(\pi'_0 = 0, \pi'_1, \pi'_2, \ldots, \pi'_{2n}, \pi'_{2n+1} = 2n+1)$ and whose edge set consists of:

- black edges $\{\pi'_{2i}, \pi'_{2i+1}\}$ for $0 \le i \le n$;
- grey edges $\{\pi'_{2i}, \pi'_{2i} + 1\}$ for $0 \le i \le n$.

Fig. 2 shows an example of a breakpoint graph. Since each vertex in that graph has degree two, the breakpoint graph decomposes in a single way into *alternating cycles*, i.e. cycles that alternate black and grey edges. It can be easily seen that the breakpoint graph shown in Fig. 2 decomposes into two such cycles.

Definition 9 (*[15]*)**.** The *support* of a grey edge $\{\pi'_i, \pi'_j\}$, with $i < j$, is the interval of π' delimited by i and j, endpoints included. A grey edge is *oriented* if its support contains an odd number of elements, and *nonoriented* otherwise. A cycle in $BG(\pi)$ is *oriented* if it contains an oriented edge, and *nonoriented* otherwise.

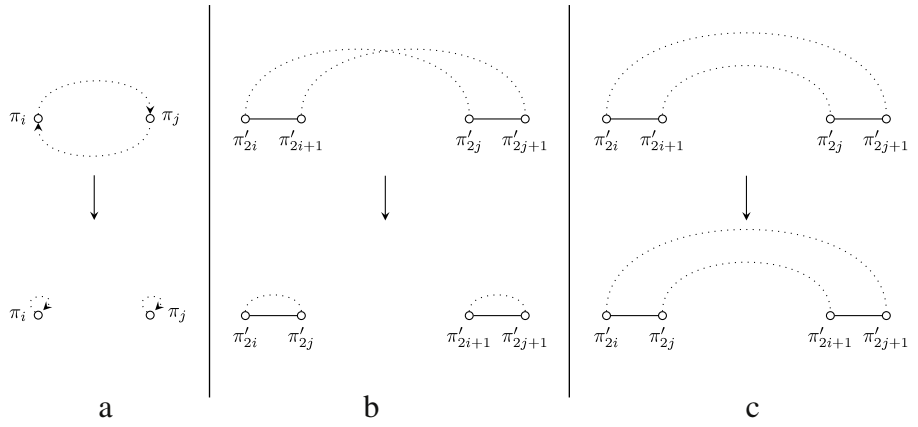

Fig. 3. (a) The effect of an exchange on a cycle of $\Gamma(\pi)$, (b) the effect of a signed reversal on an oriented cycle and (c) on a nonoriented cycle of $BG(\pi)$. Dotted lines in those drawings stand for (alternating) paths.

For example, the grey edge that connects 0 and 1 in Fig. 2 is oriented, while the one that connects 4 and 5 is nonoriented. The *length* of a cycle in a breakpoint graph is the number of black edges it contains, and a *k-cycle* is a cycle of length k. A k-cycle is called *trivial* if $k = 1$, and *nontrivial* otherwise.

Definition 10. A signed reversal $\overline{\rho}(i, j)$ is said to *act* on black edges $\{\pi'_{2i-2}, \pi'_{2i-1}\}$ and $\{\pi'_{2j}, \pi'_{2j+1}\}$ of $BG(\pi)$. Likewise, it is said to *act on one cycle* (resp. on two cycles) if both black edges on which $\overline{\rho}(i, j)$ acts belong to the same cycle (resp. to two distinct cycles) in $BG(\pi)$.

3. A new lower bound for sorting burnt pancakes

We exploit a connection between the effect of exchanges on $\Gamma(\pi)$ and that of signed reversals on $BG(\pi)$ to derive a new lower bound on the prefix signed reversal distance of any permutation. We will need the following result by Akers et al. [16] on computing the prefix exchange distance.

Theorem 1 ([16]). *For any π in S_n, we have*

$$ped(\pi) = n + c(\Gamma(\pi)) - 2c_1(\Gamma(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 2 & \text{otherwise}, \end{cases}$$

where $c_1(\Gamma(\pi))$ denotes the number of 1-cycles in $\Gamma(\pi)$.

Theorem 2. *For any π in $S_n^{\pm}$, we have*

$$psrd(\pi) \geq n + 1 + c(BG(\pi)) - 2c_1(BG(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 2 & \text{otherwise}, \end{cases} \tag{1}$$

where $c_1(BG(\pi))$ denotes the number of 1-cycles in $BG(\pi)$.

Proof. The key observation is that the action of signed reversals on the cycles of the breakpoint graph is analogous to the action of exchanges on the cycles of (the graph of) a permutation: both involve at most two distinct cycles, and can create at most one new cycle in the graph on which they act, as Fig. 3 shows.

The analogy obviously still holds under the prefix restriction, and the proof then follows from Theorem 1. □

Note that, as observed by Hannenhalli and Pevzner [9] and as shown in Fig. 3(c), it is not always possible to split a cycle in $BG(\pi)$ using a signed reversal, whereas it is always possible to split a cycle in $\Gamma(\pi)$ using an exchange (hence the lower bound instead of an equality).

4. Sorting simple permutations in polynomial time

We now turn our attention to an important class of signed permutations, which proved crucial in solving the signed version of sorting by unrestricted reversals in polynomial time (see [9]), and show how to sort those permutations by prefix signed reversals in polynomial time.

Definition 11. A signed permutation π is *simple* if $BG(\pi)$ contains only cycles of length at most 2.

Our analysis is based exclusively on simple permutations; therefore, we need to ensure that the sequences of prefix signed reversals we use will transform any given simple permutation into another simple permutation.

Definition 12. A sequence of signed reversals applied to a simple permutation π is *conservative* if it transforms π into a simple permutation σ.

We wish to stress that we only require σ to be simple: we allow intermediate permutations obtained in the process of transforming π into σ not to be simple.

Definition 13. Let $g(\pi)$ denote the right-hand side of lower bound (1); an (x, y)-*sequence* is a sequence of x prefix signed reversals transforming a permutation π into a permutation σ with $g(\pi) - g(\sigma) = y$. It is *optimal* if $x = y$.

4.1. Components of the breakpoint graph

We will need the following definitions and results of Hannenhalli and Pevzner [9].

Definition 14. Two distinct grey edges in the breakpoint graph *interleave* if their supports overlap but do not contain each other. Likewise, two distinct cycles in the breakpoint graph *interleave* if either cycle contains a grey edge that interleaves with a grey edge of the other cycle.

Definition 15. Let $H(\pi)$ be the graph whose vertices are the cycles in $BG(\pi)$ and whose edges connect two vertices if the corresponding cycles interleave. A *component* of $BG(\pi)$ is a connected component of $H(\pi)$; it is *oriented* if a vertex of that component in $H(\pi)$ corresponds to an oriented cycle in $BG(\pi)$, and *nonoriented* otherwise.

Lemma 1 ([9])**.** *A signed reversal acting on a given cycle C in $BG(\pi)$ changes the orientation of every cycle in $BG(\pi)$ that interleaves with C (i.e. it transforms every nonoriented (resp. nonoriented) cycle that interleaves with C into a nonoriented (resp. oriented) cycle).*

Lemma 2 ([9])**.** *Every grey edge of a nontrivial cycle in $BG(\pi)$ interleaves with another grey edge.*

Lemma 2 implies in particular that if π is a simple permutation, then every nontrivial nonoriented cycle in $BG(\pi)$ interleaves with another nontrivial cycle. In the following, we sometimes abuse language by saying that we sort a cycle or a component, which actually means that we transform a k-cycle or a component involving k black edges in $BG(\pi)$ into a collection of k 1-cycles.

4.2. Preliminary results

In the following, we will refer to the cycle in $BG(\pi)$ that contains the black edge $\{\pi'_0, \pi'_1\}$ as the *leftmost cycle*, and to the component that contains the leftmost cycle as the *leftmost component*.

Definition 16 ([9])**.** A signed reversal is *proper* if it increases the number of cycles in $BG(\pi)$ by one.

The following observation will be crucial.

Lemma 3. *For any simple permutation π, any minimal sequence of prefix reversals that mimics a proper reversal is both conservative and optimal.*

Proof. A proper reversal $\overline{\rho}(i, j)$ on π splits a 2-cycle into two 1-cycles; if $i = 1$, then the resulting permutation σ now fixes 1, and lower bound (1) has decreased by 1. Otherwise, the status of the first element in π and σ is the same, and there is a sequence of three prefix reversals which mimics the effect of $\overline{\rho}(i, j)$ and decreases lower bound (1) by 3. Therefore, the resulting sequence of prefix reversals is optimal, and clearly conservative. □

As a result, if $BG(\pi)$ contains an oriented component, then we can sort that component optimally in polynomial time (see [17] for more details). Therefore, the only remaining cases we need to examine are the cases where π admits no proper reversal, or equivalently where $BG(\pi)$ contains no oriented cycle, distinguishing between the case where $\pi_1 \neq 1$ and $\pi_1 = 1$.

Lemma 4. *Let π be a simple permutation; if $\pi_1 \neq 1$ and $BG(\pi)$ contains no oriented cycle, then π admits a conservative* $(1, 0)$*-sequence.*

Proof. A prefix signed reversal acting on the leftmost cycle will flip the orientation of any cycle it interleaves (Lemma 1), thereby guaranteeing the creation of at least one oriented cycle in $BG(\pi)$ and in particular transforming the leftmost component into an oriented component. Note that lower bound (1) is unaffected by that move. □

Lemma 5. *Let π be a simple permutation; if $\pi_1 = 1$ and $BG(\pi)$ contains no oriented cycle, then π admits a conservative* $(2, 2)$*-sequence.*

Proof. If $\pi_1 = 1$ and $BG(\pi)$ contains no oriented 2-cycle, then any nonoriented component $BG(\pi)$ contains can be transformed into an oriented leftmost component as follows. Pick the leftmost cycle C_1 in any nonoriented component; by Lemma 2, C_1 interleaves with another nonoriented cycle, say C_2. If C_2 contains black edges $\{\pi'_{2j-2}, \pi'_{2j-1}\}$ and $\{\pi'_{2\ell-2}, \pi'_{2\ell-1}\}$, then applying $\overline{\rho}(1, j-1)$ followed by $\overline{\rho}(1, \ell-1)$ transforms π into another simple permutation with an oriented leftmost component, as Fig. 4 shows.

Neither the number of cycles nor the number of 1-cycles in the breakpoint graph is affected, but the resulting permutation no longer fixes 1, so lower bound (1) decreases by 2. □

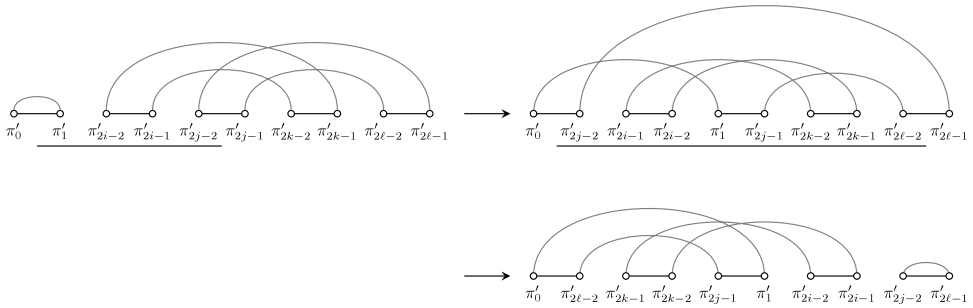

Fig. 4. An optimal conservative sequence of prefix signed reversals that transforms a nonoriented component into an oriented leftmost component.

4.3. Computing psrd for simple permutations

We have proved the existence of optimal conservative sequences for every simple permutation, except in the case where $\pi_1 \neq 1$ and the leftmost component is nonoriented. In this section, we prove that the strategy proposed in Lemma 4 is optimal (Lemma 8), and derive a formula for computing the prefix signed reversal distance of simple permutations (Theorem 3).

We will refer to prefix signed reversals that act on two nontrivial cycles of the breakpoint graph as *merging moves*, and to those that split the leftmost cycle into two cycles, at least one of which is trivial, as *splitting moves*. It can be easily seen from Theorem 2 that any sequence of prefix signed reversals that is to outperform the strategy proposed in Lemma 4 must consist solely of these types of moves, and eventually lead to an oriented leftmost 2-cycle, a step that must precede the creation of two new trivial cycles. Therefore, our proof will consist in showing that this strategy will fail to orient the leftmost component.

We have already defined orientation for grey edges (Definition 9 page 697). We will also need an analogous definition for black edges by Tannier and Sagot [18].

Definition 17 ([18])**.** A black edge $\{\pi'_{2i}, \pi'_{2i+1}\}$ in $BG(\pi)$ is *oriented* if π_i and π_{i+1} have opposite signs, and *nonoriented* otherwise.

Definition 18 ([9])**.** The smallest interval that contains the leftmost and rightmost elements of a given component $\mathscr{C}$ in $BG(\pi)$ is denoted by

$$Extent(\mathscr{C}) = \left[\min_{C\in\mathscr{C}} \min_{\pi'_i\in C} i,\ \max_{C\in\mathscr{C}} \max_{\pi'_j\in C} j\right].$$

Components can be ordered by inclusion based on their extent. We will be interested mainly in minimal components, defined below.

Definition 19 ([9])**.** A component $\mathscr{C}$ of $BG(\pi)$ is *minimal* if every cycle C with $support(C) \subseteq Extent(\mathscr{C})$ belongs to $\mathscr{C}$.

Graphically speaking, a minimal component is an "innermost" one. Tannier and Sagot [18] note that $BG(\pi^{-1})$ can be obtained from $BG(\pi)$ by exchanging positions and elements in π' as well as edge colours (i.e. black (resp. grey) edges in $BG(\pi)$ become grey (resp. black) edges in $BG(\pi^{-1})$). As a consequence, cycles in $BG(\pi)$ and in $BG(\pi^{-1})$ are in one-to-one correspondence, and we denote C^{-1} the cycle in $BG(\pi^{-1})$ onto which a given cycle C in $BG(\pi)$ is mapped. We show below that this mapping extends to whole components of the breakpoint graph.

Lemma 6. *A cycle C belongs to a component $\mathscr{C}$ in $BG(\pi)$ if and only if C^{-1} belongs to $\mathscr{C}^{-1}$ in $BG(\pi^{-1})$.*

Proof. We examine components by their inclusion order, starting with minimal ones and removing them as we go to proceed by induction.

We refer to the pairs $\{\pi'_{2i-1}, \pi'_{2i}\}$ for $1 \leq i \leq n$ as *white edges*, which form the same set in both $BG(\pi)$ and $BG(\pi^{-1})$. The alternating path made of white and grey (resp. black) edges in $BG(\pi)$ will be referred to as the *WG-path* (resp. *WB-path*), which when starting with the leftmost vertex of $BG(\pi)$ visits the vertices of $BG(\pi)$ in the natural order (resp. in the order in which they appear in π').

We now show that a minimal component $\mathscr{C}$ in $BG(\pi)$ corresponds to a minimal component $\mathscr{C}^{-1}$ in $BG(\pi^{-1})$. If $Extent(\mathscr{C}) = [i, j]$, then vertices $\{\pi'_{2i}, \ldots, \pi'_{2j-1}\}$ induce a sub-path of the WG-path in $BG(\pi)$, which is mapped onto a WB-path in $BG(\pi^{-1})$ and implies that cycles of $\mathscr{C}^{-1}$ do not interleave with cycles outside $\mathscr{C}^{-1}$. To see that the cycles we obtain in that way belong to the same connected component in $BG(\pi^{-1})$, assume on the contrary that some cycles were mapped onto an additional minimal component, say $\mathscr{C}'$, in $BG(\pi^{-1})$. By the above argument, cycles in $\mathscr{C}'^{-1}$ do not interleave with other cycles in $BG((\pi^{-1})^{-1}) = BG(\pi)$, a contradiction. □

The following result will also be useful.

Lemma 7. *A component $\mathscr{C}$ in $BG(\pi)$ is oriented if and only if it contains an oriented black edge.*

Proof. As in the proof of Lemma 6, we examine components by their inclusion order, starting with minimal ones and removing them as we go to proceed by induction. Recall (Definition 15 page 699) that $\mathscr{C}$ is oriented if it contains an oriented grey edge, which corresponds to a pair of elements of opposite signs in π. If $\mathscr{C}$ is minimal, then it contains two elements of opposite signs in π if and only if it contains a pair of *adjacent* elements of opposite sign, which themselves correspond to an oriented black edge (Definition 17).

As observed by Bergeron et al. [19], the elements of π that "frame" $\mathscr{C}$ (i.e. $\pi_{i/2}$ and $\pi_{(j+1)/2}$, if $Extent(\mathscr{C}) = [i, j]$) have the same sign. We can then remove $\mathscr{C}$ from $BG(\pi)$, renumber the elements appropriately, and handle the remaining components in the same way. □

Since oriented black (resp. grey) edges in $BG(\pi)$ become oriented grey (resp. black) edges in $BG(\pi^{-1})$, Lemma 7 implies that the orientation of components in $BG(\pi)$ is also preserved in $BG(\pi^{-1})$.

Lemma 8. *Let π be a simple permutation which does not fix 1 and whose leftmost component is nonoriented. If a sequence of merging and splitting moves transforms π into a permutation σ whose leftmost cycle is a 2-cycle, then σ is simple and the leftmost component of $BG(\sigma)$ is nonoriented.*

Proof. Splitting moves extract 1-cycles from the leftmost cycle, while merging moves merge it with 2-cycles, so clearly σ is simple. To prove that its leftmost component is nonoriented, we first show that the sets of black edges in $BG(\pi)$ and in $BG(\sigma)$ differ only in the black edges that belonged to 2-cycles in $BG(\pi)$ and became 1-cycles in $BG(\sigma)$.

Grey edges in $BG(\sigma^{-1})$ connect pairs of elements that appear at the same positions as in $BG(\pi^{-1})$, except for edges that originally belonged to 2-cycles that were turned into 1-cycles. Each 2-cycle in $BG(\sigma^{-1})$ corresponds to a 2-cycle in $BG(\pi^{-1})$ on the same quadruple of positions (since by the transformation described right after Definition 19, black and grey edges are exchanged in the process of transforming $BG(\pi)$ into $BG(\pi^{-1})$ and conversely). The orientation of any 2-cycle in $BG(\sigma^{-1})$ is the same as the orientation of the corresponding 2-cycle in $BG(\pi^{-1})$, and there is no new pair of interleaving 2-cycles in $BG(\sigma^{-1})$. This together with Lemma 7 implies that the leftmost component of $BG(\sigma)$ is nonoriented. □

We now have everything we need to prove a formula for computing the prefix signed reversal distance of simple permutations.

Theorem 3. *For every simple permutation π in $S_n^{\pm}$, we have:*

$$psrd(\pi) = n + 1 + c(BG(\pi)) - 2c_1(BG(\pi)) + t(\pi) - \begin{cases} 0 & \text{if } \pi_1 = 1 \\ 2 & \text{otherwise} \end{cases},$$

where $t(\pi) = 1$ if $\pi_1 \neq 1$ and the leftmost component of $BG(\pi)$ is nonoriented, and 0 otherwise.

Proof. The upper bound follows from the fact that there exists an optimal conservative sequence for dealing with every simple permutation, except when $\pi_1 \neq 1$ and $BG(\pi)$ contains no oriented cycle; however, a single prefix signed reversal turns the leftmost component into an oriented component (Lemma 4). This situation cannot occur more than once in the sorting process, since once the leftmost component has been sorted, either the resulting permutation is ι or we can sort every remaining oriented component of $BG(\pi)$ optimally — or create a leftmost oriented component in $BG(\pi)$ if no oriented component exists (Lemma 5).

Finally, if $\pi_1 \neq 1$ and the leftmost component of $BG(\pi)$ is nonoriented, then no sorting sequence can outperform the strategy proposed in Lemma 4 (see Lemma 8), which implies the desired lower bound and completes the proof. □

4.4. The sorting algorithm

Algorithm 4.1 outlines how to sort simple permutations by prefix signed reversals in polynomial time. Tannier et al. [17] cover step 3 in details (they sort oriented components using arbitrary signed reversals, but as we have seen these can be mimicked by optimal sequences of prefix signed reversals (Lemma 3)), while step 5 can be achieved either by applying a reversal on the leftmost cycle, if $\mathscr{D}$ is the leftmost component (Lemma 4), or by applying the 2-move sequence proposed in Lemma 5. The algorithm can be implemented so as to run in $O(n^{3/2})$ time (see [17,20]), while the distance can be computed in $O(n)$ time (see [21]).

5. Conclusions

We proved a new lower bound on the minimum number of prefix signed reversals needed to sort any signed permutation of n elements, whereas the exact computation of that number remains an open problem. Using this lower bound, we were able to show that an important class of permutations, known as "simple permutations", could be sorted in polynomial time, and proposed both a sorting algorithm and a formula for computing the minimum number of required operations.

Hannenhalli and Pevzner [9] proved that every permutation π could be transformed into a simple permutation $\tilde{\pi}$ in such a way that $srd(\pi) = srd(\tilde{\pi})$ (see [22] for a $O(n)$ time algorithm for transforming π into $\tilde{\pi}$, and a $O(n \log n)$ time algorithm for recovering the original permutation). Unfortunately, the transformation does not preserve the prefix signed

Algorithm 4.1 SIMPLEBURNTPANCAKEFLIPPING(π)

Input: a simple permutation π
Output: the identity permutation

while $\pi \neq \iota$ **do**
  if $BG(\pi)$ contains an oriented component $\mathcal{C}$ **then**
    sort $\mathcal{C}$;
  else
    orient any nonoriented component $\mathcal{D}$;
  end if
end while

reversal distance, as shown by the following counter-example: if $\pi = \langle 2\ 1 \rangle$, then the corresponding simple permutation is $\tilde{\pi} = \langle 3\ 2\ 1 \rangle$, but it can be verified that $psrd(\pi) = 3$ and $psrd(\tilde{\pi}) = 5$. Therefore, Algorithm 4.1 cannot immediately be used to sort an arbitrary permutation, but since every sequence of signed reversals on a simple permutation can be used to sort the original permutation [9], we have $psrd(\pi) \leq psrd(\tilde{\pi})$. Moreover, we believe that our contributions should be useful for designing improved approximation or exact algorithms for solving the burnt pancake flipping problem, as well as for getting insight into its computational complexity. The unsigned version of sorting by prefix reversals (i.e. the original pancake flipping problem) may also benefit from our results, since both variants are strongly connected (see [23] for more details).

Acknowledgements

The authors would like to thank the anonymous referee, whose helpful comments undoubtedly contributed to a substantial improvement of this paper. This work has been partially supported by ERC Starting Grant 240186 'MiGraNT'.

Josef Cibulka was supported by the project 1M0545 of the Ministry of Education of the Czech Republic and by the Czech Science Foundation under the contract no. 201/09/H057.

References

[1] H. Dweighter, Elementary problems and solutions, American Mathematical Monthly 82 (10) (1975) 296. problem E2569.
[2] W.H. Gates, C.H. Papadimitriou, Bounds for sorting by prefix reversal, Discrete Mathematics (ISSN: 0012-365X) 27 (1) (1979) 47–57.
[3] E. Györi, G. Turán, Stack of pancakes, Studia Scientiarum Mathematicarum Hungarica 13 (1978) 133–137.
[4] J. Fischer, S. W. Ginzinger, A 2-Approximation Algorithm for Sorting by Prefix Reversals, in: G.S. Brodal, S. Leonardi (Eds.), Proceedings of the Thirteenth Annual European Symposium on Algorithms, ESA, in: Lecture Notes in Computer Science, vol. 3669, Springer-Verlag, 2005, pp. 415–425.
[5] D.S. Cohen, M. Blum, On the Problem of Sorting Burnt Pancakes, Discrete Applied Mathematics 61 (2) (1995) 105–120.
[6] K. Haynes, M. Broderick, A. Brown, T. Butner, J. Dickson, W.L. Harden, L. Heard, E. Jessen, K. Malloy, B. Ogden, S. Rosemond, S. Simpson, E. Zwack, A.M. Campbell, T. Eckdahl, L. Heyer, J. Poet, Engineering bacteria to solve the Burnt Pancake Problem, Journal of Biological Engineering (ISSN: 1754-1611) 2 (1) (2008) 8. doi:10.1186/1754-1611-2-8. URL http://www.jbioleng.org/content/2/1/8.
[7] S. Lakshmivarahan, J.-S. Jwo, S. K. Dhall, Symmetry in interconnection networks based on Cayley graphs of permutation groups: a survey, Parallel Computing 19 (4) (1993) 361–407.
[8] A. Caprara, Sorting permutations by reversals and Eulerian cycle decompositions, SIAM Journal on Discrete Mathematics (ISSN: 1095-7146) 12 (1) (1999) 91–110 (electronic).
[9] S. Hannenhalli, P.A. Pevzner, Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals, Journal of the ACM (ISSN: 0004-5411) 46 (1) (1999) 1–27.
[10] G. Fertin, A. Labarre, I. Rusu, E. Tannier, S. Vialette, Combinatorics of genome rearrangements, in: Computational Molecular Biology, The MIT Press, 2009.
[11] A. Labarre, Edit Distances and Factorisations of Even Permutations, in: D. Halperin, K. Mehlhorn (Eds.), Proceedings of the Sixteenth Annual European Symposium on Algorithms, ESA, in: Lecture Notes in Computer Science, vol. 5193, Springer-Verlag, ISBN: 978-3-540-87743-1, 2008, pp. 635–646.
[12] A. Björner, F. Brenti, Combinatorial descriptions, in: Combinatorics of Coxeter Groups, in: Graduate Texts in Mathematics, vol. 231, Springer-Verlag, 2005 (Chapter 8).
[13] H. Wielandt, Finite Permutation Groups, Academic Press, New York, 1964, Translated from German by R. Bercov.
[14] V. Bafna, P.A. Pevzner, Genome rearrangements and sorting by reversals, in: Proceedings of the Thirty-Fourth Annual Symposium on Foundations of Computer Science, FOCS, ACM/SIAM, Palo Alto, Los Alamitos, CA, ISBN: 0-89871-349-8, 1993, pp. 148–157.
[15] A. Bergeron, A very elementary presentation of the Hannenhalli–Pevzner theory, Discrete Applied Mathematics (ISSN: 0166-218X) 146 (2) (2005) 134–145.
[16] S.B. Akers, B. Krishnamurthy, D. Harel, The star graph: an attractive alternative to the n-cube, in: Proceedings of the Fourth International Conference on Parallel Processing, ICPP, Pennsylvania State University Press, 1987, pp. 393–400.
[17] E. Tannier, A. Bergeron, M.-F. Sagot, Advances on sorting by reversals, Discrete Applied Mathematics 155 (6–7) (2007) 881–888.
[18] E. Tannier, M.-F. Sagot, Sorting by reversals in subquadratic time, in: Proceedings of the Fifteenth Annual Symposium on Combinatorial Pattern Matching, CPM, in: Lecture Notes in Computer Science, vol. 3109, Springer-Verlag, ISBN: 3-540-22341-X, 2004, pp. 1–13.
[19] A. Bergeron, S. Heber, J. Stoye, Common intervals and sorting by reversals: a marriage of necessity, Bioinformatics 18 (Suppl. 2) (2002) S54–S63.
[20] Y. Han, Improving the efficiency of sorting by reversals, in: Proceedings of The 2006 International Conference on Bioinformatics and Computational Biology, CSREA Press, Las Vegas, Nevada, USA, ISBN: 1-60132-002-7, 2006, p. 4.
[21] D.A. Bader, B.M.E. Moret, M. Yan, A linear-time algorithm for computing inversion distance between signed permutations with an experimental study, Journal of Computational Biology 8 (5) (2001) 483–491.
[22] S. Gog, M. Bader, Fast algorithms for transforming back and forth between a signed permutation and its equivalent simple permutation, Journal of Computational Biology 15 (8) (2008) 1–13.
[23] S. Hannenhalli, P.A. Pevzner, To cut... or not to cut (applications of comparative physical maps in molecular evolution), in: Proceedings of the Seventh Annual ACM–SIAM Symposium on Discrete Algorithms, SODA, ACM, Atlanta, Georgia, 1996, pp. 304–313.
[24] D.E. Knuth, Sorting and Searching, in: The Art of Computer Programming, vol. 3, Addison-Wesley, ISBN: 0-201-03803-X, 1995.
[25] P. Berman, S. Hannenhalli, M. Karpinski, 1.375-approximation algorithm for sorting by reversals, in: Proceedings of the Tenth Annual European Symposium on Algorithms, ESA, in: Lecture Notes in Computer Science, vol. 2461, Springer-Verlag, Rome, Italy, 2002, pp. 200–210.