Approximation Algorithms for Sorting by Signed Short Reversals

Gustavo Rodrigues Galvão
Institute of Computing
University of Campinas - Unicamp
Av. Albert Einstein, 1251 – Cidade Universitária
CEP 13083-852, Campinas, São Paulo, Brazil
ggalvao@ic.unicamp.br

Zanoni Dias
Institute of Computing
University of Campinas - Unicamp
Av. Albert Einstein, 1251 – Cidade Universitária
CEP 13083-852, Campinas, São Paulo, Brazil
zanoni@ic.unicamp.br

ABSTRACT

During evolution, global mutations may modify the gene order in a genome. Such mutations are commonly referred to as rearrangement events. One of the most frequent rearrangement events observed in genomes are reversals, which are responsible for reversing the order and orientation of a sequence of genes. The problem of sorting by reversals, that is, the problem of computing the most parsimonious reversal scenario to transform one genome into another, is a well-studied problem that finds application in comparative genomics. There is a number of works concerning this problem in the literature, but these works generally do not take into account the length of the reversals. Since it has been observed that short reversals are prevalent in the evolution of some species, recent efforts have been made to address this issue algorithmically. In this paper, we add to these efforts by introducing the problem of sorting by signed short reversals and by presenting three approximation algorithms for solving it. Although the worst-case approximation ratios of these algorithms are high, we show that their expected approximation ratios for sorting a random equiprobable signed permutation are much lower. Moreover, we present experimental results on small signed permutations which indicate that the worst-case approximation ratios of these algorithms may be better than those we have been able to prove.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics; G.2.1 [**Combinatorics**]: Combinatorial algorithms

General Terms

Algorithm

Keywords

Genome Rearrangement, Short Reversals, Approximation Algorithms

BCB'14, September 20–23, 2014, Newport Beach, CA, USA.

http://dx.doi.org/10.1145/2649387.2649413.

1. INTRODUCTION

In genome rearrangements, one estimates the evolutionary distance between two species by finding the rearrangement distance between their genomes, which is the length of the shortest sequence of rearrangement events that transforms one genome into the other. Assuming genomes consist of a single linear chromosome, share the same set of genes, and contain no duplicated genes, we can represent them as permutations of integers where each integer corresponds to a gene. If the orientation of the genes is known, then each integer has a sign, $+$ or $-$, indicating its orientation and the permutation is called a signed permutation. Otherwise, the permutation is called an unsigned permutation. Thus, taking the target genome as the identity permutation ($1\ 2\ \ldots\ n$), the problem of finding the rearrangement distance between two genomes can be modeled as the combinatorial problem of transforming a permutation into the identity permutation using a minimum number of rearrangement events. This problem is commonly referred to as permutation sorting problem.

Depending on the rearrangement events allowed to sort a permutation, we have a different variant of the permutation sorting problem. One of the most studied variants is the so-called problem of sorting by reversals, which is the variant of the permutation sorting problem that allows reversals only. A reversal is a rearrangement event that reverses the order of the elements (genes) on a certain portion of the permutation (genome) and flips their signs (orientation) if the permutation is signed. For this reason, it is common to refer to a reversal as an unsigned (signed) reversal to indicate that it acts on an unsigned (signed) permutation. The minimum number of reversals necessary to sort a permutation is called the reversal distance of this permutation.

The problem of sorting an unsigned permutation by reversals is a NP-hard problem [7]. It was introduced by Watterson *et al.* [23], who presented a simple heuristic for solving it. Kececioglu and Sankoff [14] presented the first constant-factor approximation algorithm, a 2-approximation algorithm. The best known result is due to Berman, Hannenhalli and Karpinski [6], who presented a 1.375-approximation algorithm. The problem of sorting a signed permutation by reversals was introduced by Bafna and Pevzner [4], who presented a 1.5-approximation algorithm for solving it. Hannenhalli and Pevzner [11] presented the first polynomial algorithm for this problem, which was further improved by Tannier, Bergeron and Sagot [22] to run in subquadratic time. Barder, Moret and Yan [3] showed how to compute the reversal distance (without actually sorting) of a signed permutation in linear time.

One drawback of the problem of sorting by reversals is that it does not take into account the length of the reversals. This is a relevant issue because, as pointed out by Sankoff [19], separate set of observations concerns the prevalence and significance of short reversals (*i.e.* reversals involving one or a few genes) in the evolution of microbial genomes [8], bacterial genomes [15], and lower eukaryotes genomes [16, 20]. For this reason, different approaches have been proposed recently to address this issue, from which we highlight two: Reversal weighting and reversal bounding.

The reversal weighting approach consists in assigning weights to reversals according to their length. Pinter and Skiena [18] introduced a non-unit cost model based on the reversed sequence length of unsigned permutations. They proposed an algorithm that takes a monotonic function of length into account as an optimization criterion and proved that such an algorithm has a guaranteed approximation ratio of $O(\lg^2 n)$. Also for unsigned permutations, Bender *et al.* [5] proposed approximation algorithms for a class of cost functions $f(l) = l^\alpha$, where l is the length of a reversal and α is a positive real constant. When $\alpha = 1$, they guarantee a $O(\lg n)$ approximation ratio. When $\alpha \geq 2$, they guarantee a 2-approximation ratio. Some of the results on unsigned permutations were extended to signed permutations by Swidan *et al.* [21]. More specifically, they were able to guarantee the same approximation ratio $O(\lg n)$ for signed permutations when $\alpha = 1$. Recently, Arruda, Dias, and Dias [1, 2] have presented heuristics for sorting both signed and unsigned permutations by length-weighted reversals.

The reversal bounding approach consists in restricting the reversals based on their length. Jerrum [13] proved that the problem of sorting an unsigned permutation by reversals of length 2 is solvable in polynomial time. Later, Heath and Vergara [12] considered the problem of sorting an unsigned permutation by reversals of length at most 3 and presented the best known solution for it, a 2-approximation algorithm. They called this problem as the problem of sorting by short swaps, but it is also known as the problem of sorting by (unsigned) short reversals. Recently, Egri-Nagy *et al.* [9] have proved that the problem of sorting an unsigned circular permutation by reversals of length 2 is solvable in polynomial time.

Finally, we remark that Nguyen, Ngo, and Nguyen [17] proposed a mixed approach, that is, they introduced a problem which consists of three inputs: An unsigned permutation, a cost function f on the length of the reversals, and a positive integer k. The aim is to sort the permutation applying a minimum-cost series of reversals whose lengths do not exceed k. They presented a $(2\lg^2 n + \lg n)$-approximation algorithm for solving this problem considering the same class of cost functions considered by Bender *et al.* [5].

In this paper, we follow the reversal bounding approach and introduce the problem of sorting a signed permutation by reversals of length at most 3. This problem is a natural extension of the problem of sorting by unsigned short reversals introduced by Heath and Vergara [12] since signed permutations constitute a more biologically relevant model for genomes. We present three approximation algorithms for this problem: One 12-approximation algorithm and two 9-approximation algorithms. Although the worst-case approximation ratios of these algorithms are high, we show that the expected approximation ratios of these algorithms for sorting a random equiprobable signed permutation are much lower. Moreover, we present experimental results on small signed permutations which suggest that the worst-case approximation ratios of these algorithms may be lower than those we have been able to prove.

The rest of this paper is organized as follows. Section 2 presents the basic definitions and notations used in this paper. Section 3 presents the three approximation algorithms we have developed. Section 4 presents the experimental results obtained by running these algorithms on signed permutations with up to 10 elements. Finally, Section 5 concludes the paper.

2. PRELIMINARIES

An *unsigned permutation* π is a bijection of $\{1, 2, \ldots, n\}$ onto itself. A classical notation used in combinatorics for denoting an unsigned permutation π is the two-row notation

$$\pi = \begin{pmatrix} 1 & 2 & \ldots & n \\ \pi_1 & \pi_2 & \ldots & \pi_n \end{pmatrix},$$

$\pi_i \in \{1, 2, \ldots, n\}$ for $1 \leq i \leq n$. It indicates that $\pi(1) = \pi_1$, $\pi(2) = \pi_2, \ldots, \pi(n) = \pi_n$. The notation used in genome rearrangement literature, which is the one we will adopt, is the one-row notation $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$. We say that π has size n. The set formed by all unsigned permutations of size n is denoted by S_n.

A *signed permutation* π is a bijection of $\{-n, \ldots, -2, -1, 1, 2, \ldots, n\}$ onto itself that satisfies $\pi(-i) = -\pi(i)$ for all $i \in \{1, 2, \ldots, n\}$. The two-row notation for a signed permutation is

$$\pi = \begin{pmatrix} -n & \ldots & -2 & -1 & 1 & 2 & \ldots & n \\ -\pi_n & \ldots & -\pi_2 & -\pi_1 & \pi_1 & \pi_2 & \ldots & \pi_n \end{pmatrix},$$

$\pi_i \in \{1, 2, \ldots, n\}$ for $1 \leq i \leq n$. We will also adopt the one-row notation $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$ for representing signed permutations (we drop the mapping of the negative elements since $\pi(-i) = -\pi(i)$ for all $i \in \{1, 2, \ldots, n\}$). By abuse of notation, we say that π has size n. The set formed by all signed permutations of size n is denoted by $S_n^\pm$.

A *signed reversal* $\rho(i, j)$, $1 \leq i \leq j \leq n$, is an operation that transforms a signed permutation $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_{i-1}\ \underline{\pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1}\ \pi_j}\ \pi_{j+1} \ldots \pi_n)$ into the signed permutation $\pi \cdot \rho(i, j) = (\pi_1\ \pi_2 \ldots \pi_{i-1}\ \underline{-\pi_j\ -\pi_{j-1}\ \ldots\ -\pi_{i+1}\ -\pi_i}\ \pi_{j+1} \ldots \pi_n)$. A signed reversal $\rho(i, j)$ is called a *signed k-reversal* if $k = j - i + 1$. A signed k-reversal is called *short* if $k \leq 3$. The problem of sorting by signed short reversals consists in finding the minimum number of signed short reversals that transform a permutation $\pi \in S_n^\pm$ into the *identity permutation* $\iota_n = (1\ 2\ \ldots\ n)$. This number is known as the *signed short reversal distance* of a permutation π and it is denoted by $d(\pi)$.

The number of negative elements of a signed permutation π is denoted by $Neg(\pi)$. The change in the number of negative elements of a signed permutation π due to a signed short reversal ρ is denoted by $\Delta Neg(\pi, \rho)$, that is, $\Delta Neg(\pi, \rho) = Neg(\pi) - Neg(\pi \cdot \rho)$.

LEMMA 1. $\Delta Neg(\pi, \rho) \leq 3$.

PROOF. A signed short reversal can flip the sign of at most three elements of a signed permutation, therefore the lemma follows. □

We say that a pair of elements (π_i, π_j) of a signed permutation π is an *inversion* if $i < j$ and $|\pi_i| > |\pi_j|$. The number of inversions in a signed permutation π is denoted by $Inv(\pi)$.

EXAMPLE 1. *Let $\pi = (2\ 1\ {-6}\ 3\ {-7}\ {-5}\ 4)$ be a signed permutation. We have that the pairs of elements $(2, 1)$, $(-6, 3)$, $(-6, -5)$, $(-6, 4)$, $(-7, -5)$, $(-7, 4)$, and $(-5, 4)$ are inversions.*

LEMMA 2. *Let π be a signed permutation. If $Inv(\pi) > 0$, then there exists a signed 2-reversal that removes an inversion of π.*

PROOF. We say that an element π_e of a signed permutation π is out of position if $|\pi_e| \neq e$. Note that there must exist elements out of position in π if $Inv(\pi) > 0$. Let π_i be the first (from left to right) element out of position in π and let $|\pi_i| = j$ (it is not hard to see that $j > i$). Then, we have that there exists an element π_k, $k > i$, such that $|\pi_k| = i$. Let $\pi_i\ \pi_{i+1}\ \ldots\ \pi_l$ be the greatest sequence of contiguous elements starting at π_i such that $|\pi_i| < |\pi_{i+1}| < \cdots < |\pi_l|$ (note that $l < k$ because $|\pi_i| > |\pi_k|$). Then, we have that the pair of elements (π_l, π_{l+1}) is an inversion and the signed 2-reversal $\rho(l, l+1)$ removes it. $\square$

The change in the number of inversions in a signed permutation π due to a signed short reversal ρ is denoted by $\Delta Inv(\pi, \rho)$, that is, $\Delta Inv(\pi, \rho) = Inv(\pi) - Inv(\pi \cdot \rho)$.

LEMMA 3. *$\Delta Inv(\pi, \rho) \leq 3$.*

PROOF. A signed short reversal $\rho(i, j)$ can remove at most three inversions of a signed permutation π, precisely when $j = i + 2$ and $|\pi_i| > |\pi_{i+1}| > |\pi_{i+2}|$, therefore the lemma follows. $\square$

For each element π_i of a signed permutation π, we define a *vector* $v(\pi_i)$ whose length is given by $|v(\pi_i)| = ||\pi_i| - i|$. If $|v(\pi_i)| > 0$, the vector $v(\pi_i)$ has a direction indicated by the sign of $|\pi_i| - i$. A vector diagram V_π of π is the set of vectors of the elements of π. The sum of the lengths of all the vectors in V_π is denoted by $|V_\pi|$.

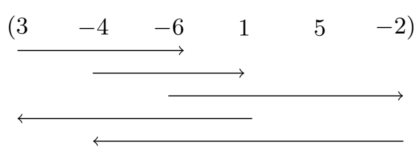

Figure 1: Vector diagram of the signed permutation $(3\ {-4}\ {-6}\ 1\ 5\ {-2})$.

The change in the sum of the lengths of all the vectors in V_π due to a signed short reversal ρ is denoted by $\Delta|V_\pi|(\pi, \rho)$, that is, $\Delta|V_\pi|(\pi, \rho) = |V_\pi| - |V_{\pi\cdot\rho}|$.

LEMMA 4. *$\Delta|V_\pi|(\pi, \rho) \leq 4$.*

PROOF. A signed short reversal $\rho(i, j)$ can decrease $|V_\pi|$ by at most 4, precisely when $|\pi_i| - i \geq 2$ and $|\pi_j| - j \leq -2$, therefore the lemma follows. $\square$

Two elements π_i and π_j, $i < j$, of a signed permutation π are said to be *vector-opposite* if the vectors $v(\pi_i)$ and $v(\pi_j)$ differ in direction, $|v(\pi_i)| \geq j - i$, and $|v(\pi_j)| \geq j - i$. Besides, they are said to be *m-vector-opposite* if $j - i = m$. Note that m specifies the distance between vector-opposite elements.

EXAMPLE 2. *Let $\pi = (3\ {-4}\ {-6}\ 1\ 5\ {-2})$ be a signed permutation. We have that elements -4 and 1 are 2-vector-opposite and elements 6 and -2 are 3-vector-opposite. Note that elements 3 and -2 are not vector-opposite although their vectors differ in direction.*

Heath and Vergara [12] have also defined vector-opposite elements and vector diagrams in the context of unsigned permutations and they have proved some properties about them. The two lemmas below correspond to these properties adapted to the context of signed permutations.

LEMMA 5. *Let π be a signed permutation. If $Inv(\pi) > 0$, then π contains at least one pair of vector-opposite elements.*

PROOF. Analogous to the proof of Lemma 5 of [12]. $\square$

LEMMA 6. *Let $\pi \in S_n^{\pm}$ be a signed permutation such that $Inv(\pi) > 0$ and let π_i and π_j be m-vector-opposite elements. Moreover, let $\pi' \in S_n^{\pm}$ be a signed permutation such that $\pi'_i = \pi_j$, $\pi'_j = \pi_i$, and $\pi'_k = \pi_k$ for all $k \notin \{i, j\}$. Then, $|V_\pi| - |V_{\pi'}| = 2m$.*

PROOF. We have that

$$\begin{aligned} |V_\pi| - |V_{\pi'}| &= \textstyle\sum_{k=1}^{n}(|v(\pi_k)| - |v(\pi'_k)|) \\ &= |v(\pi_i)| - |v(\pi'_i)| + |v(\pi_j)| - |v(\pi'_j)| \\ &= m + m \\ &= 2m, \end{aligned}$$

therefore the lemma follows. $\square$

3. APPROXIMATION ALGORITHMS

In this section, we present the three approximation algorithms we have developed for the problem of sorting by signed short reversals. One of these algorithms, which has an approximation ratio of 12, is a greedy algorithm based on a concept we called *inversion potential*. This algorithm is presented in Section 3.1. The other two algorithms, which have an approximation ratio of 9, are greedy algorithms based on a concept we called *vector potential*. These algorithms are presented in Section 3.2.

3.1 Algorithm based on the inversion potential

The identity permutation is the only signed permutation that has neither inversions nor negative elements, therefore a trivial algorithm for sorting a signed permutation $\pi \in S_n^{\pm}$ is to perform signed 2-reversals on the inversions until the permutation has no inversions (Lemma 2 guarantees that it is possible) and then to perform signed 1-reversals on the negative elements until the permutation has no negative elements. It is not hard to see that this algorithm would perform at most $Inv(\pi) + n$ signed short reversals for sorting π. Moreover, in the specific cases where $Inv(\pi) = 0$, the algorithm would perform exactly $Neg(\pi)$ signed short reversals for sorting π. According to lemmas 1 and 3, we have that $d(\pi) \geq \max\{\frac{Inv(\pi)}{3}, \frac{Neg(\pi)}{3}\}$. This means that, in the cases where $Inv(\pi) = 0$, the approximation ratio of the trivial algorithm is 3. On the other hand, in the cases where $Inv(\pi) > 0$, the approximation ratio of the trivial algorithm can be as bad as $3 + \frac{3n}{Inv(\pi)}$.

Let π be a signed permutation. We define the *inversion potential* of π as $InvPot(\pi) = 3Inv(\pi) + Neg(\pi)$. The change in the inversion potential of π due to a signed short reversal ρ is denoted by $\Delta InvPot(\pi, \rho)$, that is, $\Delta InvPot(\pi, \rho) = InvPot(\pi) - InvPot(\pi \cdot \rho) = 3\Delta Inv(\pi, \rho) + \Delta Neg(\pi, \rho)$.

Lemma 7. $\Delta InvPot(\pi, \rho) \leq 12$.

Proof. The lemma follows directly from the fact that, as shown by lemmas 1 and 3, $\Delta \mathrm{Neg}(\pi, \rho) \leq 3$ and $\Delta \mathrm{Inv}(\pi, \rho) \leq 3$. □

It is not hard to see that the only signed permutation π for which $InvPot(\pi) = 0$ is the identity permutation, therefore we propose the greedy algorithm described below (Algorithm 1). While Lemma 8 shows that Algorithm 1 sorts any given signed permutation $\pi \in S_n^{\pm}$ in $O(n^3)$ time, Theorem 1 shows that it is a 12-approximation algorithm.

Algorithm 1: Greedy algorithm based on the inversion potential.

Data: A permutation $\pi \in S_n^{\pm}$.
Result: Number of signed short reversals applied for sorting π.

```
d ← 0;
while π ≠ ι_n do
    Let ρ(i, j) be the signed short reversal such that
    ΔInvPot(π, ρ) is maximum;
    π ← π · ρ(i, j);
    d ← d + 1;
end
return d;
```

Theorem 1. *Algorithm 1 is a 12-approximation algorithm.*

Proof. Consider this two cases:

a) $Neg(\pi) > 0$. In this case, it is possible to apply a signed 1-reversal ρ that flips the sign of a negative element, therefore $\Delta InvPot(\pi, \rho) = 1$.

b) $Neg(\pi) = 0$. In this case, we have that $Inv(\pi) > 0$, otherwise π would be sorted. According to Lemma 2, it is possible to apply a signed 2-reversal on an inversion of π, but the elements that form the inversion would become negative, therefore $\Delta InvPot(\pi, \rho) = 3 - 2 = 1$.

Thus, in the worst case, it is possible to decrease the inversion potential of a signed permutation π by at least 1 unit. According to Lemma 7, we have that $d(\pi) \geq \frac{InvPot(\pi)}{12}$, therefore the lemma follows. □

Lemma 8. *Algorithm 1 sorts any given signed permutation $\pi \in S_n^{\pm}$ in $O(n^3)$ time.*

Proof. To show that Algorithm 1 sorts π, it suffices to show that the while loop terminates. From the proof of Theorem 1, we can conclude that every signed short reversal applied by Algorithm 1 causes a strict decrease in $InvPot(\pi)$. Therefore, it eventually becomes zero (precisely when $\pi = \iota_n$) and it follows that the while loop terminates.

For the time complexity, let $InvPot_{max}(n)$ be the maximum value of the inversion potential of a signed permutation considering all signed permutation of size n. From the proof of Theorem 1, we can conclude that the while loop can iterate at most $InvPot_{max}(n)$ times. Since a signed permutation $\pi \in S_n^{\pm}$ can have at most $\binom{n}{2}$ inversions and n negative elements, we have that $InvPot_{max}(n) = 3\binom{n}{2} + n$. Moreover, at each iteration, Algorithm 1 has to scan $2n - 3$ signed short reversals. Therefore, it follows that Algorithm 1 runs in $O(n^3)$ time. □

We close this section by noting that there is a large class of signed permutations for which the approximation ratio of Algorithm 1 is much lower than the worst-case approximation ratio (Lemma 9). Moreover, based on the fact that the expected number of inversions in a random equiprobable signed permutation $\pi \in S_n^{\pm}$ is $\frac{n^2-n}{4}$ (Lemma 10), we can conclude that the expected approximation ratio of Algorithm 1 for sorting a random equiprobable signed permutation is much lower than the worst-case approximation ratio (Theorem 2).

Lemma 9. *Let $A_1(\pi)$ be the number of signed short reversals applied by Algorithm 1 for sorting a signed permutation $\pi \in S_n^{\pm}$. We have that $\frac{A_1(\pi)}{d(\pi)} \leq 4$ when $Inv(\pi) \geq 3n$.*

Proof. We can conclude, from the proof of Theorem 1, that Algorithm 1 behaves like the trivial algorithm in the worst case, therefore $\frac{A_1(\pi)}{d(\pi)} \leq 3 + \frac{3n}{Inv(\pi)}$. Then, assuming $Inv(\pi) \geq 3n$, we have that $\frac{A_2(\pi)}{d(\pi)} \leq 4$ and the lemma follows. □

Lemma 10. *Let $E(Inv(\pi))$ be the expected number of inversions in a random equiprobable signed permutation $\pi \in S_n^{\pm}$. Then, $E(Inv(\pi)) = \frac{n^2-n}{4}$.*

Proof. Let $X_{i,j}$ be the indicator random variable for the event that the pair (π_i, π_j) is an inversion. It is not hard to see that the probability of a pair of elements in random equiprobable signed permutation be an inversion is $\frac{1}{2}$, therefore the expected value of $X_{i,j}$, $E(X_{i,j})$, is $\frac{1}{2}$. Since the total number of inversions in π, $Inv(\pi)$, equals the sum of all the $X_{i,j}$'s, we have that

$$\begin{aligned}
Inv(\pi) &= \textstyle\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{i,j} \\
E(Inv(\pi)) &= E(\textstyle\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{i,j}) \\
&= \textstyle\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E(X_{i,j}) \\
&= \textstyle\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{2} \\
&= \frac{n^2-n}{4},
\end{aligned}$$

therefore the lemma follows. □

Theorem 2. *The expected approximation ratio of Algorithm 1 for sorting a random equiprobable signed permutation $\pi \in S_n^{\pm}$ is no greater than 4 when $n \geq 13$.*

Proof. According to Lemma 9, we have that the approximation ratio of Algorithm 1 for sorting a given signed permutation $\sigma \in S_n^{\pm}$ is no greater than 4 when $Inv(\sigma) \geq 3n$. Since we know that the expected number of inversions in a random equiprobable signed permutation $\pi \in S_n^{\pm}$ is $\frac{n^2-n}{4}$ (Lemma 10), we can conclude that the expected approximation ratio of Algorithm 1 for sorting π is no greater than 4 if $\frac{n^2-n}{4} \geq 3n$. This inequality holds when $n \geq 13$, therefore the theorem follows. □

3.2 Algorithms based on the vector potential

Heath and Vergara [12] presented a 2-approximation algorithm for sorting by short swaps (or unsigned short reversals) based on vector-opposite elements. At each iteration, the algorithm selects a pair of vector-opposite elements and swaps their positions applying short swaps. Since the identity permutation is the only one which has no vector-opposite elements (considering just unsigned permutations), the algorithm stops when the permutation is sorted. In the worst case, each short swap applied by the algorithm decreases the sum of the lengths of all the vectors by 2 on average.

It is possible to adapt Heath and Vergara's algorithm to sort a signed permutation $\pi \in S_n^{\pm}$ as follows: Perform signed short reversals analogously to the short swaps performed by Heath and Vergara's algorithm until the signed permutation has no vector-opposite elements and then perform signed 1-reversals on the negative elements until the permutation has no negative elements. Such an algorithm would perform at most $\frac{|V_\pi|}{2} + n$ signed short reversals for sorting π. According to lemmas 1 and 4, we have that $d(\pi) \geq \max\{\frac{|V_\pi|}{4}, \frac{Neg(\pi)}{3}\}$. This means that, in the cases where $Inv(\pi) = 0$, the approximation ratio of the adapted algorithm is 3. On the other hand, in the cases where $Inv(\pi) > 0$, the approximation ratio of the adapted algorithm can be as bad as $2 + \frac{4n}{|V_\pi|}$.

Let π be a signed permutation. We define the *vector potential* of π as $VecPot(\pi) = \frac{3}{2}|V_\pi| + Neg(\pi)$. The change in the vector potential of π due to a signed short reversal ρ is denoted by $\Delta VecPot(\pi, \rho)$, that is, $\Delta VecPot(\pi, \rho) = VecPot(\pi) - VecPot(\pi \cdot \rho) = \frac{3}{2}\Delta|V_\pi| + \Delta Neg(\pi)$.

LEMMA 11. *$\Delta VecPot(\pi, \rho) \leq 9$.*

PROOF. The lemma follows directly from the fact that, as shown by lemmas 1 and 4, $\Delta \mathrm{Neg}(\pi, \rho) \leq 3$ and $\Delta|V_\pi|(\pi, \rho) \leq 4$. □

It is not hard to see that the only signed permutation π for which $VecPot(\pi) = 0$ is the identity permutation, therefore we propose two greedy algorithms (algorithms 2 and 3) that try to apply signed short reversals that cause the best average net decrease in $VecPot(\pi)$. Basically, they select m-vector-opposite elements such that m is maximum and swap their positions applying signed short reversals. The reason why they select m-vector-opposite elements such that m is maximum is that the greater the value of m, the greater is the average net decrease in $VecPot(\pi)$ (this will become more clear in the proofs of theorems 3 and 4). The difference of these algorithms lies in the way they swap the positions of vector-opposite elements: Algorithm 2 does it preserving the signs of the elements, while Algorithm 3 does not. This is why Algorithm 3 removes the negative elements of the signed permutation only after it has "removed" all the vector-opposite elements. Theorems 3 and 4 show that both algorithms are 9-approximation algorithms, while lemmas 14 and 15 show that they sort any given signed permutation $\pi \in S_n^{\pm}$ in $O(n^4)$ time.

Algorithm 2: Greedy algorithm based on the vector potential (preserving signs).

Data: A permutation $\pi \in S_n^{\pm}$.
Result: Number of signed short reversals applied for sorting π.

```
d ← 0;
Apply signed 1-reversals on π until it has no negative
  elements and update d accordingly;
while |V_π| > 0 do
    Let π_i and π_j be m-vector opposite elements such
      that m is maximum;
    Apply signed short reversals on π such as described
      by Lemma 13;
    if m mod 4 ≠ 0 then
        d ← d + 4⌈m/4⌉ − 1;
    else
        d ← d + m;
    end
end
return d;
```

Algorithm 3: Greedy algorithm based on the vector potential (not preserving signs).

Data: A permutation $\pi \in S_n^{\pm}$.
Result: Number of signed short reversals applied for sorting π.

```
d ← 0;
while |V_π| > 0 do
    Let π_i and π_j be m-vector opposite elements such
      that m is maximum;
    Apply signed short reversals on π such as described
      by Lemma 12;
    d ← d + 2⌈m/2⌉ − 1;
end
Apply signed 1-reversals on π until it has no negative
  elements and update d accordingly;
return d;
```

LEMMA 12. *Let $\pi \in S_n^{\pm}$ be a signed permutation such that $Inv(\pi) > 0$ and let π_i and π_j be m-vector-opposite elements. It is possible to transform π into $\pi' \in S_n^{\pm}$ such that $|\pi'_i| = |\pi_j|$, $|\pi'_j| = |\pi_i|$, and $|\pi'_k| = |\pi_k|$ for all $k \notin \{i, j\}$ applying $2\lceil \frac{m}{2} \rceil - 1$ signed short reversals.*

PROOF. We have two cases to consider:

a) m is even. In this case, we have that $\pi' = (((((\pi \cdot \rho(i, i+2)) \cdot \rho(i+2, i+4)) \cdots \rho(j-4, j-2)) \cdot \rho(j-2, j)) \cdot \rho(j-4, j-2)) \cdots \rho(i, i+2)$. Therefore, to transform π into π', we have to apply $m-1$ signed 3-reversals.

b) m is odd. In this case, we have that $\pi' = (((((\pi \cdot \rho(i, i+2)) \cdot \rho(i+2, i+4)) \cdots \rho(j-3, j-1)) \cdot \rho(j-1, j)) \cdot \rho(j-3, j-1)) \cdots \rho(i, i+2)$. Therefore, to transform π into π', we have to apply $m-1$ signed 3-reversals and 1 signed 2-reversal, totalizing m signed short reversals.

Since in both cases we can transform π into π' applying $2\lceil \frac{m}{2} \rceil - 1$ signed short reversals, the lemma follows. □

LEMMA 13. *Let $\pi \in S_n^{\pm}$ be a signed permutation such that $Inv(\pi) > 0$ and let π_i and π_j be m-vector-opposite elements. It is possible to transform π into $\pi' \in S_n^{\pm}$ such that $\pi'_i = \pi_j$, $\pi'_j = \pi_i$, and $\pi'_k = \pi_k$ for all $k \notin \{i, j\}$ applying d signed short reversals, where*

$$d = \begin{cases} 4\lceil \frac{m}{4} \rceil - 1 & \textit{if } m \bmod 4 \neq 0 \\ m & \textit{otherwise.} \end{cases}$$

PROOF. We have four cases to consider:

a) $m \bmod 4 = 0$. In this case, we can transform π into a signed permutation $\pi'' \in S_n^{\pm}$ such that $\pi''_i = \pi_j$, $\pi''_j = \pi_i$, $\pi''_{j-1} = -\pi_{j-1}$, and $\pi''_k = \pi_k$ for all $k \notin \{i, j-1, j\}$ applying the sequence of signed short reversals shown in the case a) of Lemma 12. Then, we can transform π'' into π' applying the signed 1-reversal $\rho(j-1, j-1)$. Therefore, to transform π into π', we have to apply $m-1$ signed 3-reversals and 1 signed 1-reversal, totalizing m signed short reversals.

b) $m \bmod 4 = 1$. In this case, we can transform π into a signed permutation $\pi'' \in S_n^{\pm}$ such that $\pi''_i = -\pi_j$, $\pi''_j = -\pi_i$, and $\pi''_k = \pi_k$ for all $k \notin \{i, j\}$ applying the sequence of signed short reversals shown in the case b) of Lemma 12. Then, we can transform π'' into π' applying the signed 1-reversals $\rho(i, i)$ and $\rho(j, j)$. Therefore, to transform π into π', we have to apply $m - 1$ signed 3-reversals, 1 signed 2-reversal, and 2 signed 1-reversals, totalizing $m + 2$ signed short reversals.

c) $m \bmod 4 = 2$. In this case, we can transform π into π' applying a sequence of signed short reversals very similar to the sequence shown in the case a) of Lemma 12, but instead of applying the signed 3-reversal $\rho(j-2, j)$, we apply the signed 2-reversals $\rho(j-2, j-1)$, $\rho(j-1, j)$, and $\rho(j-2, j-1)$. Therefore, to transform π into π', we have to apply $m-2$ signed 3-reversals and 3 signed 2-reversals, totalizing $m+1$ signed short reversals.

d) $m \bmod 4 = 3$. In this case, we can transform π into π' applying the sequence of signed short reversals shown in the case b) of Lemma 12. Therefore, to transform π into π', we have to apply m signed short reversals.

Since we can transform π into π' applying m signed short reversals when $m \bmod 4 = 0$ and we can transform π into π' applying $4\lceil \frac{m}{4} \rceil - 1$ signed short reversals when $m \bmod 4 \neq 0$, the lemma follows. $\square$

THEOREM 3. *Algorithm 2 is a 9-approximation algorithm.*

PROOF. Consider this two cases:

a) $Neg(\pi) > 0$. In this case, it is possible to apply a signed 1-reversal ρ that flips the sign of a negative element, therefore $\Delta VecPot(\pi, \rho) = 1$.

b) $Neg(\pi) = 0$. In this case, we have that $Inv(\pi) > 0$, otherwise π would be sorted. Let π_i and π_j be the m-vector-opposite elements chosen by Algorithm 2. After applying the signed short reversals described by Lemma 13, we have that $\Delta|V_\pi| = 2m$ (Lemma 6) and $\Delta Neg(\pi) = 0$, therefore $\Delta VecPot(\pi, \rho) = 3m$. Thus, in the worst case, we have that the average decrease of the vector potential is $\min\{\frac{3m}{m}, \frac{3m}{m+1}, \frac{3m}{m+2}\} = 1$, precisely when $m = 1$.

Thus, in the worst case, each signed short reversal applied by Algorithm 2 decreases the vector potential of a signed permutation π by at least 1 unit on average. According to Lemma 11, we have that $d(\pi) \geq \frac{VecPot(\pi)}{9}$, therefore the lemma follows. $\square$

LEMMA 14. *Algorithm 2 sorts any given signed permutation $\pi \in S_n^{\pm}$ in $O(n^4)$ time.*

PROOF. After line 2 executes, we have that $Neg(\pi) = 0$, therefore it suffices to show that the while loop terminates in order to prove that Algorithm 2 sorts π. It is not hard to see that, by swapping vector-opposite elements, Algorithm 2 causes a strict decrease in $|V_\pi|$. Therefore, it eventually becomes zero (precisely when $\pi = \iota_n$) and it follows that the while loop terminates.

For the time complexity, we have that the while loop can iterate $O(n^2)$ times since $|V_\pi| \leq n^2$ for all $\pi \in S_n^{\pm}$. Moreover, at each iteration of the while loop, Algorithm 2 has to scan $O(n^2)$ vector-opposite elements and has to apply $O(n)$ signed short reversals. Therefore, it follows that the while loop runs in $O(n^4)$ time. Since line 2 runs in $O(n)$ time, we can conclude that Algorithm 2 runs in $O(n^4)$ time and the lemma follows. $\square$

THEOREM 4. *Algorithm 3 is a 9-approximation algorithm.*

PROOF. Consider this two cases:

a) $|V_\pi| = 0$. In this case, we have that $Neg(\pi) > 0$, otherwise π would be sorted, so it is possible to apply a signed 1-reversal ρ that flips the sign of a negative element, therefore $\Delta VecPot(\pi, \rho) = 1$.

b) $|V_\pi| > 0$. Let π_i and π_j be the m-vector-opposite elements chosen by Algorithm 3. After applying the signed short reversals described by Lemma 12, we have that $\Delta|V_\pi| = 2m$ (Lemma 6) and

$$\Delta Neg(\pi) = \begin{cases} -1 & \text{if } m \bmod 4 = 0 \\ -2 & \text{if } m \bmod 4 = 1 \\ -3 & \text{if } m \bmod 4 = 2 \\ 0 & \text{if } m \bmod 4 = 3 \end{cases}$$

in the worst case. To see why $\Delta Neg(\pi)$ can assume these values, we have to analyze four cases:

a) $m \bmod 4 = 0$. This case has been analyzed in the case a) of Lemma 13 and it is not hard to see that, in the worst case, $\Delta Neg(\pi) = -1$.

b) $m \bmod 4 = 1$. This case has been analyzed in the case b) of Lemma 13 and it is not hard to see that, in the worst case, $\Delta Neg(\pi) = -2$.

c) $m \bmod 4 = 2$. In this case, the sequence of signed short reversals described in the case a) of Lemma 12 transform π into π' such that $\pi'_i = -\pi_j$, $\pi'_j = -\pi_i$, $\pi'_{j-1} = -\pi_{j-1}$, and $\pi'_k = \pi_k$ for all $k \notin \{i, j-1, j\}$. Therefore, in the worst case, we have that $\Delta Neg(\pi) = -3$.

d) $m \bmod 4 = 3$. This case has been analyzed in the case d) of Lemma 13 and it is not hard to see that $\Delta Neg(\pi) = 0$.

Thus, in the worst case, we have that the average decrease of the vector potential is $\min\{\frac{3m-1}{m-1}, \frac{3m-2}{m}, \frac{3m-3}{m-1}, \frac{3m}{m}\} = 1$, precisely when $m = 1$ (note that the denominator equals $m-1$ iff m is even).

It means that each signed short reversal applied by Algorithm 3 decreases the vector potential of a signed permutation π by at least 1 unit on average. According to Lemma 11, we have that $d(\pi) \geq \frac{VecPot(\pi)}{9}$, therefore the lemma follows. $\square$

LEMMA 15. *Algorithm 3 sorts any given signed permutation $\pi \in S_n^{\pm}$ in $O(n^4)$ time.*

PROOF. Analogous to the proof of Lemma 14. $\square$

Similarly to what we have done in the previous section, we close this section by noting that there is a large class of signed permutations for which the approximation ratios of algorithms 2 and 3 are much lower than their respective worst-case approximation ratios (lemmas 16 and 17). Moreover, based on the fact that the expected value of $|V_\pi|$ of a random equiprobable signed permutation $\pi \in S_n^{\pm}$ is $\frac{n^2-1}{3}$ (Lemma 19), we can conclude that the expected approximation ratios of algorithms 2 and 3 for sorting a random equiprobable signed permutation are much lower than the respective worst-case approximation ratios (theorems 5 and 6 respectively).

LEMMA 16. *Let $A_2(\pi)$ be the number of signed short reversals applied by Algorithm 2 for sorting a signed permutation $\pi \in S_n^{\pm}$. We have that $\frac{A_2(\pi)}{d(\pi)} \leq 7$ when $|V_\pi| \geq 4Neg(\pi)$.*

PROOF. Algorithm 2 performs at most $\frac{3|V_\pi|}{2} + Neg(\pi)$ signed short reversals for sorting π: Firstly it performs $Neg(\pi)$ signed short reversal to eliminate the negative elements of π, then it decreases $|V_\pi|$ by at least 2 units after applying at most 3 signed short reversals until π is sorted. Since $d(\pi) \geq \frac{|V_\pi|}{4}$ (Lemma 4), we have that $\frac{A_2(\pi)}{d(\pi)} \leq 6 + \frac{4Neg(\pi)}{|V_\pi|}$. Thus, assuming $|V_\pi| \geq 4Neg(\pi)$, we can conclude that $\frac{A_2(\pi)}{d(\pi)} \leq 7$ and the lemma follows. $\square$

LEMMA 17. *Let $A_3(\pi)$ be the number of signed short reversals applied by Algorithm 3 for sorting a signed permutation $\pi \in S_n^{\pm}$. We have that $\frac{A_3(\pi)}{d(\pi)} \leq 3$ when $|V_\pi| \geq 4n$.*

PROOF. We can conclude from the proof of Theorem 4 that Algorithm 3 behaves like the adapted algorithm of Heath and Vergara in the worst case, therefore $\frac{A_3(\pi)}{d(\pi)} \leq 2 + \frac{4n}{|V_\pi|}$. Then, assuming $|V_\pi| \geq 4n$, we have that $\frac{A_3(\pi)}{d(\pi)} \leq 3$ and the lemma follows. $\square$

LEMMA 18. *Let $\pi \in S_n^{\pm}$ be a signed permutation and let $Pr(|v(\pi_i)| = j)$ be the probability that $|v(\pi_i)| = j$. We have that $\sum_{i=1}^n Pr(|v(\pi_i)| = j) = \frac{2(n-j)}{n}$ for $1 \leq j \leq n-1$.*

PROOF. Since $|S_n^{\pm}| = n!2^n$ and there are $(n-1)!2^n$ signed permutations for which $|\pi_i| = k$, $1 \leq k \leq n$, we have that

$$Pr(|v(\pi_i)| = j) = \begin{cases} \frac{1}{n} & \text{if } j = 0 \\ \frac{2}{n} & \text{if } i+j \leq n \text{ and } i-j \geq 1 \\ \frac{1}{n} & \text{if } i+j \leq n \text{ or } i-j \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

for $0 \leq j \leq n-1$. Then, to evaluate $\sum_{i=1}^n Pr(|v(\pi_i)| = j)$ for a given j, let us consider these two cases:

(i) $1 \leq j < \frac{n}{2}$. In this case, we have that

$$Pr(|v(\pi_i)| = j) = \begin{cases} \frac{1}{n} & \text{if } 1 \leq i \leq j \\ \frac{1}{n} & \text{if } n-j+1 \leq i \leq n \\ \frac{2}{n} & \text{otherwise.} \end{cases}$$

Therefore, we can conclude that $\sum_{i=1}^n Pr(|v(\pi_i)| = j) = \frac{j}{n} + \frac{j}{n} + \frac{2(n-2j)}{n} = \frac{2(n-j)}{n}$.

(ii) $\frac{n}{2} \leq j \leq n$. In this case, we have that

$$Pr(|v(\pi_i)| = j) = \begin{cases} \frac{1}{n} & \text{if } 1 \leq i \leq n-j \\ \frac{1}{n} & \text{if } j+1 \leq i \leq n \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, we can conclude that $\sum_{i=1}^n Pr(|v(\pi_i)| = j) = \frac{n-j}{n} + \frac{n-j}{n} = \frac{2(n-j)}{n}$.

Thus, in both cases, we have that $\sum_{i=1}^n Pr(|v(\pi_i)| = j) = \frac{2(n-j)}{n}$, and the lemma follows. $\square$

LEMMA 19. *Let $\pi \in S_n^{\pm}$ be a random equiprobable signed permutation and let $E(|V_\pi|)$ be the expected value of $|V_\pi|$. Then, $E(|V_\pi|) = \frac{n^2-1}{3}$.*

PROOF. Given that the expected value of $|v(\pi_i)|$, $E(|v(\pi_i)|)$, is $\sum_{j=0}^{n-1} jPr(|v(\pi_i)| = j) = \sum_{j=1}^{n-1} jPr(|v(\pi_i)| = j)$, we have that

$$\begin{aligned} |V_\pi| &= \textstyle\sum_{i=1}^n |v(\pi_i)| \\ E(|V_\pi|) &= E(\textstyle\sum_{i=1}^n |v(\pi_i)|) \\ &= \textstyle\sum_{i=1}^n E(|v(\pi_i)|) \\ &= \textstyle\sum_{i=1}^n \sum_{j=1}^{n-1} jPr(|v(\pi_i)| = j) \\ &= \textstyle\sum_{j=1}^{n-1} j \sum_{i=1}^n Pr(|v(\pi_i)| = j) \\ &= \textstyle\sum_{j=1}^{n-1} j \frac{2(n-j)}{n} \\ &= 2\textstyle\sum_{j=1}^{n-1} j - \frac{2}{n} \sum_{j=1}^{n-1} j^2 \\ &= 2(\frac{n^2-n}{2}) - \frac{2}{n}(\frac{(n-1)n(2n-1)}{6}) \\ &= n^2 - n - \frac{2n^2-3n+1}{3} \\ &= \frac{n^2-1}{3}, \end{aligned}$$

therefore the lemma follows. $\square$

THEOREM 5. *The expected approximation ratio of Algorithm 2 for sorting a random equiprobable signed permutation $\pi \in S_n^{\pm}$ is no greater than 7 when $n \geq 7$.*

PROOF. According to Lemma 16, we have that the approximation ratio of Algorithm 2 for sorting a given signed permutation $\sigma \in S_n^{\pm}$ is no greater than 7 when $|V_\sigma| \geq 4Neg(\sigma)$. Since we know that the expected value of $|V_\pi|$ of a random equiprobable signed permutation $\pi \in S_n^{\pm}$ is $\frac{n^2-1}{3}$ (Lemma 19) and that the expected value of $Neg(\pi)$ is $\frac{n}{2}$ (each element of a signed permutation has equal probability of being negative or positive), we can conclude that the expected approximation ratio of Algorithm 2 for sorting π is no greater than 7 if $\frac{n^2-1}{3} \geq 2n$. This inequality holds when $n \geq 7$, therefore the theorem follows. $\square$

THEOREM 6. *The expected approximation ratio of Algorithm 3 for sorting a random equiprobable signed permutation $\pi \in S_n^{\pm}$ is no greater than 3 when $n \geq 13$.*

PROOF. According to Lemma 17, we have that the approximation ratio of Algorithm 3 for sorting a given signed permutation $\sigma \in S_n^{\pm}$ is no greater than 3 when $|V_\sigma| \geq 4n$. Since we know that the expected value of $|V_\pi|$ of a random equiprobable signed permutation $\pi \in S_n^{\pm}$ is $\frac{n^2-1}{3}$ (Lemma 19), we can conclude that the expected approximation ratio of Algorithm 3 for sorting π is no greater than 3 if $\frac{n^2-1}{3} \geq 4n$. This inequality holds when $n \geq 13$, therefore the theorem follows. □

4. EXPERIMENTAL RESULTS

We have implemented algorithms 1, 2, and 3 and we have audited them using GRAAu [10]. The audit consists of comparing the distance computed by an algorithm with $d(\pi)$ for every $\pi \in S_n^{\pm}$, $1 \leq n \leq 10$. The results are presented in tables 1, 2, and 3, where n is the size of the permutations, *Avg. Ratio* is the average of the ratios between the distance outputted by the algorithm and the signed short reversal distance, *Max. Ratio* is the greatest ratio among all the ratios between the distance outputted by the algorithm and the signed short reversal distance, and *Equals* is the percentage of distances outputted by the algorithm that is equal to the signed short reversal distance.

Table 1: Results obtained from the audit of the implementation of Algorithm 1.

n	Avg. Ratio	Max. Ratio	Equals
1	1.00	1.00	100.00%
2	1.00	1.00	100.00%
3	1.20	2.50	68.75%
4	1.33	3.50	42.97%
5	1.41	3.50	22.81%
6	1.47	4.33	10.53%
7	1.51	4.33	4.35%
8	1.54	4.33	1.59%
9	1.56	4.33	0.52%
10	1.58	4.60	0.16%

Table 2: Results obtained from the audit of the implementation of Algorithm 2.

n	Avg. Ratio	Max. Ratio	Equals
1	1.00	1.00	100.00%
2	1.75	5.00	62.50%
3	2.23	6.00	22.92%
4	2.33	6.00	7.03%
5	2.37	6.00	1.88%
6	2.38	6.00	0.41%
7	2.39	6.00	0.07%
8	2.40	6.00	0.01%
9	2.40	6.00	0.00%
10	2.40	6.00	0.00%

Table 3: Results obtained from the audit of the implementation of Algorithm 3.

n	Avg. Ratio	Max. Ratio	Equals
1	1.00	1.00	100.00%
2	1.00	1.00	100.00%
3	1.13	2.50	77.08%
4	1.18	3.00	60.16%
5	1.25	3.00	39.45%
6	1.30	3.00	23.38%
7	1.36	3.00	12.17%
8	1.40	3.00	5.62%
9	1.43	3.00	2.33%
10	1.47	3.00	0.87%

As Figure 2 illustrates, Algorithm 3 had the best performance, while Algorithm 2 had the worst performance. We think the main reason why Algorithm 2 has performed so poorly is that it applies a considerable number of unnecessary signed short reversals. For instance, Algorithm 2 applies the sequence of signed short reversals $\rho(1, 1)$, $\rho(2, 2)$, $\rho(1, 2)$, $\rho(1, 1)$, and $\rho(2, 2)$ to sort the signed permutation $(-2\ -1)$, but it is possible to sort this permutation applying just one signed short reversal, namely $\rho(1, 2)$.

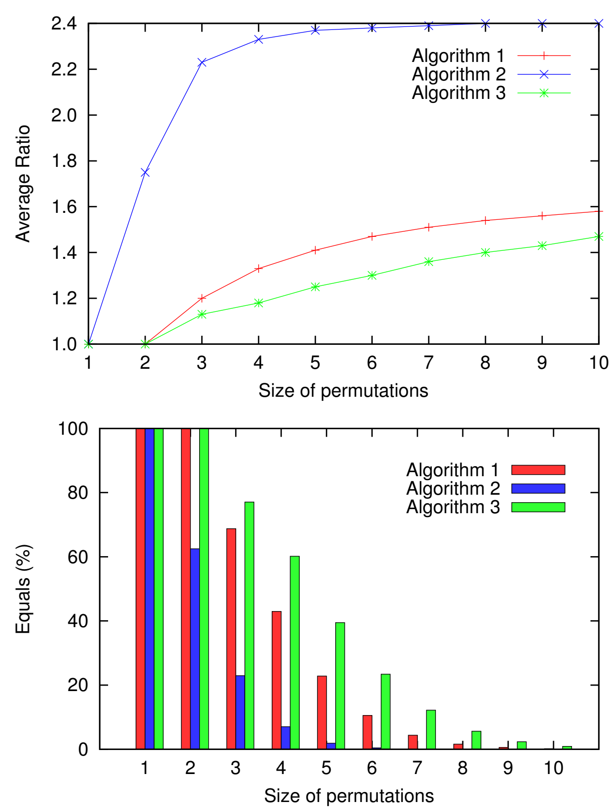

Figure 2: Comparison of algorithms 1, 2, and 3 based on the results provided by GRAAu.

As indicated by theorems 2, 5, and 6, the average approximation ratios (*Avg. Ratio*) observed for the three algorithms are much lower than the theoretical approximation ratios proved in theorems 1, 3, and 4. In fact, even the practical worst-case approximation ratios (*Max. Ratio*) are much lower than the theoretical approximation ratios, what we believe to be a good indication that the theoretical ratios are not tight.

Besides providing the *Max. Ratio*, GRAAu also provides up to 50 permutations for which the algorithms achieved this ratio. These permutations can be used to provide lower bounds on the worst-case approximation ratios of algorithms 1, 2, and 3. This is precisely what lemmas 20, 21, and 22 do.

Lemma 20. *The worst-case approximation ratio of Algorithm 1 is at least* $\frac{23}{5} = 4.6$.

Proof. Let $\pi = (-3\ -2\ +10\ +1\ -5\ -4\ -7\ -6\ -9\ -8)$ be a signed permutation. We have that $A_1(\pi) = 23$ because Algorithm 1 applies the sequence of signed short reversals $\rho(1, 2)$, $\rho(5, 6)$, $\rho(7, 8)$, $\rho(9, 10)$, $\rho(3, 4)$, $\rho(2, 3)$, $\rho(4, 5)$, $\rho(1, 2)$, $\rho(5, 6)$, $\rho(6, 7)$, $\rho(7, 8)$, $\rho(8, 9)$, $\rho(9, 10)$, $\rho(1, 1)$, $\rho(2, 2)$, $\rho(3, 3)$, $\rho(4, 4)$, $\rho(5, 5)$, $\rho(6, 6)$, $\rho(7, 7)$, $\rho(8, 8)$, $\rho(9, 9)$, and $\rho(10, 10)$ for sorting π. On the other hand, we have that $d(\pi) \leq 5$ because the sequence of signed short reversals $\rho(3, 4)$, $\rho(4, 6)$, $\rho(6, 8)$, $\rho(8, 10)$, and $\rho(1, 3)$ sorts π, therefore the lemma follows. $\square$

Lemma 21. *The worst-case approximation ratio of Algorithm 2 is at least* 6.

Proof. Let $\pi = (-3\ -2\ -1)$ be a signed permutation. We have that $A_2(\pi) = 6$ because Algorithm 2 applies the sequence of signed short reversals $\rho(1, 1)$, $\rho(2, 2)$, $\rho(3, 3)$, $\rho(1, 2)$, $\rho(2, 3)$, and $\rho(1, 2)$ for sorting π. On the other hand, we have that $d(\pi) = 1$ because the signed short reversal $\rho(1, 3)$ sorts π, therefore the lemma follows. $\square$

Lemma 22. *The worst-case approximation ratio of Algorithm 3 is at least* 3.

Proof. Let $\pi = (+3\ +4\ -1\ -2)$ be a signed permutation. We have that $A_3(\pi) = 6$ because Algorithm 3 applies the sequence of signed short reversals $\rho(2, 4)$, $\rho(1, 3)$, $\rho(1, 1)$, $\rho(2, 2)$, $\rho(3, 3)$, and $\rho(4, 4)$ for sorting π. On the other hand, we have that $d(\pi) = 2$ because the sequence of signed short reversals $\rho(1, 3)$ and $\rho(2, 4)$ sorts π, therefore the lemma follows. $\square$

5. CONCLUSIONS AND FUTURE WORK

In this work, we have introduced the problem of sorting by signed short reversals. We have taken the first steps toward exploring this problem by presenting three approximation algorithms for solving it. Although the worst-case approximation ratios of these algorithms are high, we have shown that the expected approximation ratios of these algorithms for sorting a random equiprobable signed permutation are much lower. Moreover, the experimental results on small signed permutations indicated that the worst-case approximation ratios of these algorithms may be lower than those we have been able to prove. In particular, we believe that the approximation ratio of Algorithm 3 is 3. We intend to keep working on the problem of sorting by signed short reversals and try to develop approximation algorithms with better approximation ratios. Besides, it would be interesting to find a polynomial time solution for this problem.

6. ACKNOWLEDGMENTS

This work was made possible by a PhD Fellowship from CNPq to GRG (number 142260/2014-2) and by project fundings from CNPq to ZD (numbers 306730/2012-0, 477692/2012-5, and 483370/2013-4). The authors also thank the Center for Computational Engineering and Sciences at Unicamp for financial support through the FAPESP/CEPID Grant 2013/08293-7. FAPESP and CNPq are Brazilian research funding agencies. We also acknowledge the "Laboratório Multiusuário de Bioinformática da Embrapa" for the use of computational resources.

7. REFERENCES

[1] T. S. Arruda, U. Dias, and Z. Dias. Heuristics for the sorting by length-weighted inversion problem. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics (BCB'2013)*, pages 498–507, Washington DC, USA, 2013. ACM.

[2] T. S. Arruda, U. Dias, and Z. Dias. Heuristics for the sorting by length-weighted inversions problem on signed permutations. In *Proceedings of the First International Conference on Algorithms for Computational Biology (AlCoB'2014)*, Lecture Notes in Computer Science, Tarragona, Spain, 2014. Springer-Verlag. To appear.

[3] D. Bader, B. Moret, and M. Yan. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *Journal of Computational Biology*, 8(5):483–491, 2001.

[4] V. Bafna and P. A. Pevzner. Genome rearrangements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.

[5] M. A. Bender, D. Ge, S. He, H. Hu, R. Y. Pinter, S. Skiena, and F. Swidan. Improved bounds on sorting by length-weighted reversals. *Journal of Computer and System Sciences*, 74(5):744–774, 2008.

[6] P. Berman, S. Hannenhalli, and M. Karpinski. 1.375-approximation algorithm for sorting by reversals. In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA'2002)*, volume 2461 of *Lecture Notes in Computer Science*, pages 200–210, Rome, Italy, 2002. Springer-Verlag.

[7] A. Caprara. Sorting permutations by reversals and eulerian cycle decompositions. *SIAM Journal on Discrete Mathematics*, 12(1):91–110, 1999.

[8] D. Dalevi, N. Eriksen, K. Eriksson, and S. Andersson. Genome comparison: The number of evolutionary events separating *C. pneumoniae* and *C. trachomatis*. Technical report, University of Uppsala, 2000.

[9] A. Egri-Nagy, V. Gebhardt, M. M. Tanaka, and A. R. Francis. Group-theoretic models of the inversion process in bacterial genomes. *Journal of Mathematical Biology*, pages 1–23, 2013.

[10] G. R. Galvão and Z. Dias. GRAAu: Genome Rearrangement Algorithm Auditor. In *Proceedings of the 4th International Conference on Bioinformatics and Computational Biology (BICoB'2012)*, pages 97–101, Las Vegas, Nevada, USA, 2012. Curran Associates, Inc.

[11] S. Hannenhalli and P. A. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting

signed permutations by reversals. *Journal of the ACM*, 46(1):1–27, 1999.
[12] L. S. Heath and J. P. C. Vergara. Sorting by short swaps. *Journal of Computational Biology*, 10(5):775–789, 2003.
[13] M. Jerrum. The complexity of finding minimum-length generator sequences. *Theoretical Computer Science*, 36:265–289, 1985.
[14] J. D. Kececioglu and D. Sankoff. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, 13(1-2):80–110, 1995.
[15] J. F. Lefebvre, N. El-Mabrouk, E. Tillier, and D. Sankoff. Detection and validation of single gene inversions. *Bioinformatics*, 19(suppl 1):i190–i196, 2003.
[16] A. McLysaght, C. Seoighe, and K. H. Wolfe. High frequency of inversions during eukaryote gene order evolution. In D. Sankoff and J. Nadeau, editors, *Comparative Genomics*, volume 1 of *Computational Biology*, pages 47–58. Springer Netherlands, 2000.
[17] T. C. Nguyen, H. T. Ngo, and N. B. Nguyen. Sorting by restricted-length-weighted reversals. *Genomics Proteomics Bioinformatics*, 3(2):120–127, 2005.
[18] R. Y. Pinter and S. Skiena. Genomic sorting with length-weighted reversals. *Genome Informatics*, 13:103–111, 2002.
[19] D. Sankoff. Short inversions and conserved gene cluster. *Bioinformatics*, 18(10):1305, 2002.
[20] C. Seoighe, N. Federspiel, T. Jones, N. Hansen, V. Bivolarovic, R. Surzycki, R. Tamse, C. Komp, L. Huizar, R. W. Davis, S. Scherer, E. Tait, D. J. Shaw, D. Harris, L. Murphy, K. Oliver, K. Taylor, M. A. Rajandream, B. G. Barrell, and K. H. Wolfe. Prevalence of small inversions in yeast gene order evolution. *Proceedings of the National Academy of Sciences USA*, 97(26):14433–14437, 2000.
[21] F. Swidan, M. A. Bender, D. Ge, S. He, H. Hu, and R. Y. Pinter. Sorting by length-weighted reversals: Dealing with signs and circularity. In S. Sahinalp, S. Muthukrishnan, and U. Dogrusoz, editors, *Combinatorial Pattern Matching*, volume 3109 of *Lecture Notes in Computer Science*, pages 32–46. Springer Berlin Heidelberg, 2004.
[22] E. Tannier, A. Bergeron, and M. F. Sagot. Advances on sorting by reversals. *Discrete Applied Mathematics*, 155(6-7):881–888, 2007.
[23] G. A. Watterson, W. J. Ewens, T. E. Hall, and A. Morgan. The chromosome inversion problem. *Journal of Theoretical Biology*, 99(1):1–7, 1982.