Sorting by Reversals and Transpositions with Proportion Restriction

Klairton Lima Brito[1](✉), Alexsandro Oliveira Alexandrino[1], Andre Rodrigues Oliveira[1], Ulisses Dias[2], and Zanoni Dias[1]

[1] Institute of Computing, University of Campinas, Campinas, Brazil
{klairton,alexsandro,andrero,zanoni}@ic.unicamp.br
[2] School of Technology ,University of Campinas, Limeira, Brazil
ulisses@ft.unicamp.br

Abstract. In the field of comparative genomics, one way of comparing two genomes is through the analysis of how they distinguish themselves based on a set of mutations called rearrangement events. When considering that genomes undergo different types of rearrangements, it can be assumed that some events are more common than others. To model this assumption one can assign different weights to different events, where more common events tend to cost less than others. However, this approach, called weighted, does not guarantee that the rearrangement assumed to be the most frequent will be also the most frequently returned by proposed algorithms. To overcome this issue, we investigate a new problem where we seek the shortest sequence of rearrangement events able to transform one genome into the other, with a restriction regarding the proportion between the events returned. Here we consider two rearrangement events: reversal, that inverts the order and the orientation of the genes inside a segment of the genome, and transposition, that moves a segment of the genome to another position. We present an approximation algorithm applicable to any desired proportion, for both scenarios where the orientation of the genes is known or unknown. We also show an improved (asymptotic) approximation algorithm for the case where the gene orientation is known.

Keywords: Rearrangement events · Proportion restriction · Approximation algorithm

1 Introduction

When comparing two genomes, one of the main goals is to determine the sequence of mutations that occurred during the evolutionary process capable of transforming a genome into another. In comparative genomics, we estimate this sequence through genome rearrangements, evolutionary events (mutations) affecting a large sequence of the genome.

Two genomes G_1 and G_2 can be computationally represented as the sequence of labels assigned to their shared genes (or shared blocks of genes). Labels are

J. C. Setubal and W. M. Silva (Eds.): BSB 2020, LNBI 12558, pp. 117–128, 2020.
https://doi.org/10.1007/978-3-030-65775-8_11

usually integer numbers. In addition, we associate a positive or negative sign in each of the numbers, reflecting the orientation of that gene (or block) inside of the genomes. Assuming that the genomes do not contain duplicated genes, this representation results in a signed permutation, when the orientation of the genes is known, and in an unsigned permutation otherwise. One of the genomes can be seen as the identity permutation, in which the elements are in ascending order, so problems dealing with genome rearrangements are usually treated as sorting problems, in which the goal is to transform a given permutation into the identity.

Two of the most studied genome rearrangements in the literature are the reversal, that inverts the order and the orientation of the genes inside a segment of the genome, and transposition, that moves a segment of the genome to another position. The Sorting by Reversals problem has an exact polynomial algorithm for signed permutations [5] but it is NP-hard for unsigned permutations [4].

The Sorting by Transpositions problem is NP-hard [3]. When we allow the use of reversals and transpositions, and assuming that both events occur with the same frequency (unweighted approach), we have the Sorting by Reversals and Transpositions (SbRT) problem that is NP-hard on signed and unsigned permutations [7].

In the weighted approach each type of event has an associated cost, and the goal is to find a sequence of rearrangement events that transforms one genome into another minimizing the sum of the costs. Oliveira *et al.* [7] showed that Sorting by Weighted Reversals and Transpositions (SbWRT) problem is NP-hard on signed and unsigned permutations when the ratio between the cost of a transposition and the cost of a reversal is less than or equal to 1.5. Oliveira *et al.* [8] developed a 1.5-approximation algorithm for SbWRT on signed permutations considering costs 2 and 3 for reversals and transpositions, respectively.

The problem with weighted approaches is that they do not guarantee that lower cost rearrangements, i.e., assumed to be most frequent, will be the most frequently used by the algorithms. To overcome this issue we propose and investigate the Sorting by Reversals and Transpositions with Proportion Restriction problem on signed and unsigned permutations. In this problem, we seek a sorting sequence with an additional constraint in which the ratio between the number of reversals and the size of the sequence must be greater than or equal to a given parameter $k \in [0..1]$. We provide an algorithm that guarantees an approximation for any value of k on signed and unsigned permutations. We also show an asymptotic algorithm for the signed case with an improved approximation factor.

This manuscript is organized as follows. Section 2 provides definitions used throughout the paper. Section 3 presents an approximation algorithm for the signed and unsigned cases. Section 4 presents an asymptotic approximation algorithm for the signed case with an improved approximation factor. Section 5 concludes the paper.

2 Basic Definitions

This section formally presents the definitions used in the genome rearrangement problems. Given two genomes $\mathcal{G}_1$ and $\mathcal{G}_2$, each synteny block (common block of genes between the two genomes) is represented by an integer that also has a positive or negative sign to indicate its orientation, if known. Therefore, each genome is a permutation of integers. We assume that one of them is represented by the identity permutation $\iota_n = (+1\ +2\ \dots\ +n)$ and the other is represented by a signed (or unsigned) permutation $\pi = (\pi_1\ \pi_2\ \dots\ \pi_n)$.

We define a rearrangement model $\mathcal{M}$ as the set of rearrangement events allowed to compute the distance. Given a rearrangement model $\mathcal{M}$ and a permutation π, the rearrangement distance $d(\pi)$ is the minimum number of rearrangements of $\mathcal{M}$ that sorts π (i.e., that transforms π into ι). The goal of the Sorting by Genome Rearrangements problems consists in finding such distance and the sequence that reflects it.

In this work, we will assume that $\mathcal{M}$ contains both reversals and transpositions. Let us formally define these events.

Definition 1. *Given a signed permutation $\pi = (\pi_1\ \dots\ \pi_n)$, a reversal $\rho(i,j)$, with $1 \leq i \leq j \leq n$, transforms π in the permutation $\pi \cdot \rho(i,j) = (\pi_1\ \dots\ \pi_{i-1}\ \underline{-\pi_j\ \dots\ -\pi_i}\ \pi_{j+1}\ \dots\ \pi_n)$.*

Definition 2. *Given an unsigned permutation $\pi = (\pi_1\ \dots\ \pi_n)$, a reversal $\rho(i,j)$, with $1 \leq i < j \leq n$, transforms π in the permutation $\pi \cdot \rho(i,j) = (\pi_1\ \dots\ \pi_{i-1}\ \underline{\pi_j\ \dots\ \pi_i}\ \pi_{j+1}\ \dots\ \pi_n)$.*

Definition 3. *Given a permutation $\pi = (\pi_1\ \dots\ \pi_n)$, a transposition $\tau(i,j,k)$, with $1 \leq i < j < k \leq n+1$, applied to π transforms it in the permutation $\pi \cdot \tau(i,j,k) = (\pi_1\ \dots\ \pi_{i-1}\ \underline{\pi_j\ \dots\ \pi_{k-1}}\ \underline{\pi_i\ \dots\ \pi_{j-1}}\ \pi_k\ \dots\ \pi_n)$. The effect of a transposition is the same on signed and unsigned permutations.*

The following definition helps us to formally define the problem of sorting by reversals and transpositions with a constraint on the number of reversals used in the sorting sequence.

Definition 4. *Given a sequence of reversals and transpositions S, let $|S|$ denote the number of events in S and let $|S_\rho|$ denote the number of reversals in S.*

Sorting by Reversals and Transpositions with Proportion Restriction (SbRTwPR)

Input: A permutation π, that can be signed or unsigned, and a rational number $k \in [0..1]$.

Task: Find the shortest sequence S of reversals and transpositions that turns π into ι, such that $\frac{|S_\rho|}{|S|} \geq k$.

Note that when $k = 1$ the SbRTwPR problem becomes the Sorting by Reversals problem on signed [5] and unsigned [4] permutations. Moreover, when $k = 0$ we have the Sorting by Reversals and Transpositions problem on signed [10] and unsigned [9] permutations.

Example 1 shows an optimal solution S for $\pi = (-1\ {+4}\ {-8}\ {+3}\ {+5}\ {+2}\ {-7}\ {-6})$ considering the SbRT and the SbWRT problems (SbWRT using costs 2 for reversals and 3 for transpositions). Note that half of the operations in S are reversals and half are transpositions, even using a higher cost for transpositions.

Example 1.

$$\begin{aligned}
& & \pi &= & &(-1\ {+4}\ {-8}\ {+3}\ {+5}\ {+2}\ {-7}\ {-6})\\
\pi^1 &= & \pi \cdot \rho(1,5) &= & &(\underline{-5\ {-3}\ {+8}\ {-4}\ {+1}}\ {+2}\ {-7}\ {-6})\\
\pi^2 &= & \pi^1 \cdot \tau(2,4,9) &= & &(-5\ \underline{-4\ {+1}\ {+2}\ {-7}\ {-6}}\ \underline{-3\ {+8}})\\
\pi^3 &= & \pi^2 \cdot \tau(1,3,7) &= & &(\underline{+1\ {+2}\ {-7}\ {-6}}\ \underline{-5\ {-4}}\ {-3}\ {+8})\\
\pi^4 &= & \pi^3 \cdot \rho(3,7) &= & &(+1\ {+2}\ \underline{+3\ {+4}\ {+5}\ {+6}\ {+7}}\ {+8})\\
& & S &= & &\{\rho(1,5), \tau(2,4,9), \tau(1,3,7), \rho(3,7)\}
\end{aligned}$$

Example 2 shows an optimal solution S' for the same signed permutation π considering the SbRTwPR problem, adopting $k = 0.6$ (i.e., at least 60% of the operations in S must be reversals). Compared with Example 1, the sequence S' has only one more operation than S, while ensuring the minimum proportion of reversals and using both reversals and transpositions.

Example 2.

$$\begin{aligned}
& & \pi &= & &(-1\ {+4}\ {-8}\ {+3}\ {+5}\ {+2}\ {-7}\ {-6})\\
\pi^1 &= & \pi \cdot \rho(2,8) &= & &(-1\ \underline{+6\ {+7}\ {-2}\ {-5}\ {-3}\ {+8}\ {-4}})\\
\pi^2 &= & \pi^1 \cdot \rho(2,4) &= & &(-1\ \underline{+2\ {-7}\ {-6}}\ {-5}\ {-3}\ {+8}\ {-4})\\
\pi^3 &= & \pi^2 \cdot \tau(6,8,9) &= & &(-1\ {+2}\ {-7}\ {-6}\ {-5}\ \underline{-4}\ \underline{-3\ {+8}})\\
\pi^4 &= & \pi^3 \cdot \rho(1,1) &= & &(\underline{+1}\ {+2}\ {-7}\ {-6}\ {-5}\ {-4}\ {-3}\ {+8})\\
\pi^5 &= & \pi^4 \cdot \rho(3,7) &= & &(+1\ {+2}\ \underline{+3\ {+4}\ {+5}\ {+6}\ {+7}}\ {+8})\\
& & S' &= & &\{\rho(2,8), \rho(2,4), \tau(6,8,9), \rho(1,1), \rho(3,7)\}
\end{aligned}$$

In the following, we present breakpoints and the cycle graph, both widely used to obtain bounds for the distance and to develop algorithms.

2.1 Breakpoints

Given a permutation $\pi = (\pi_1\ \dots\ \pi_n)$, we extend π by adding the elements $\pi_0 = 0$ and $\pi_{n+1} = n + 1$, with these elements having positive signs when considering signed permutations. We observe that these elements are not affected by rearrangement events. From now on, we work on extended permutations.

Definition 5. *For an unsigned permutation π, a pair of elements π_i and π_{i+1}, with $0 \leq i \leq n$, is a breakpoint if $|\pi_{i+1} - \pi_i| \neq 1$.*

The number of breakpoints in a permutation π is denoted by $b(\pi)$. Given an operation γ, let $\Delta b(\pi, \gamma) = b(\pi) - b(\pi \cdot \gamma)$, that is, $\Delta b(\pi, \gamma)$ denotes the change in the number of breakpoints after applying γ to π.

Remark 1. The identity permutation ι is the only permutation with $b(\pi) = 0$.

2.2 Cycle Graph

For a signed permutation π, we define the cycle graph $G(\pi) = (V, E)$, such that $V = \{+\pi_0, -\pi_1, +\pi_1, -\pi_2, +\pi_2, \ldots, -\pi_n, +\pi_n, -\pi_{n+1}\}$ and $E = E_b \cup E_g$, where $E_b = \{(-\pi_i, +\pi_{i-1}) \,|\, 1 \leq i \leq n+1\}$ and $E_g = \{(+(i-1), -i) \,|\, 1 \leq i \leq n+1\}$. We say that E_b is the set of black edges and E_g is the set of gray edges.

Note that each vertex is incident to two edges (a gray edge and a black edge) and, so, there exists a unique decomposition of edges in cycles. The size of a cycle $C \in G(\pi)$ is the number of black edges in C. A cycle C is trivial if it has size 1. If C has size less than or equal to 3, then C is called short and, otherwise, C is called long. The identity permutation ι_n is the only one with a cycle graph containing $n + 1$ cycles, which are all trivial.

The number of cycles in $G(\pi)$ is denoted by $c(\pi)$. Given an operation γ, let $\Delta c(\pi, \gamma) = c(\pi \cdot \gamma) - c(\pi)$, that is, $\Delta c(\pi, \gamma)$ denotes the change in the number of cycles after applying γ to π.

The cycle graph $G(\pi)$ is drawn in a way to highlight characteristics of the permutation, as shown in Fig. 1. In this representation, we draw the vertices in a horizontal line, from left to right, following the order $+\pi_0, -\pi_1, +\pi_1, \ldots, -\pi_n, +\pi_n, -\pi_{n+1}$. The black edges are horizontal lines and the gray edges are arcs.

For $1 \leq i \leq n+1$, the black edge $(-\pi_i, +\pi_{i-1})$ is labeled as i. We represent a cycle C by the sequence of labels of its black edges following the order they are traversed, assuming that the first black edge is the one with highest label (rightmost black edge of C) and it is traversed from right to left. Assuming this representation, if a black edge is traversed from left to right we add a minus sign to its label (the first black is always positive since it is traversed from right to left by convention).

Two black edges of a cycle C are divergent if their labels have different signs, and convergent otherwise. A cycle C is divergent if at least one pair of black edges of C are divergent, and it is convergent otherwise.

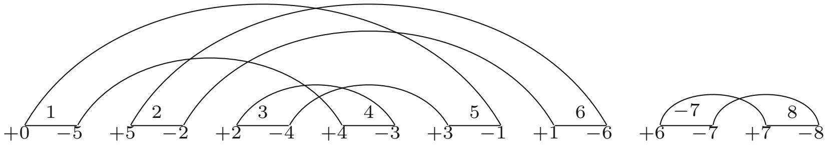


Fig. 1. Cycle Graph for $\pi = (+5 \ +2 \ +4 \ +3 \ +1 \ +6 \ -7)$. In this cycle graph, we have the cycles $C_1 = (5, 3, 4, 1)$, $C_2 = (6, 2)$, and $C_3 = (8, -7)$.

We also classify convergent cycles as oriented and non-oriented. A cycle $C = (c_1, c_2, \ldots, c_k)$ is non-oriented if $c_i > c_{i+1}$, for all $1 \leq i < k$. Otherwise, we say that C is oriented.

Two cycles $C = (c_1, c_2, \ldots, c_k)$ and $D = (d_1, d_2, \ldots, d_k)$ are interleaving if either $|c_1| > |d_1| > |c_2| > |d_2| > \ldots > |c_k| > |d_k|$ or $|d_1| > |c_1| > |d_2| > |c_2| > \ldots > |d_k| > |c_k|$.

Let g_1 be a gray edge adjacent to black edges with labels x_1 and y_1, such that $|x_1| < |y_1|$ and let g_2 be a gray edge adjacent to black edges with labels x_2 and y_2, such that $|x_2| < |y_2|$. We say that two gray edges g_1 and g_2 intersect if $|x_1| < |x_2| \leq |y_1| < |y_2|$. Two cycles C and D intersect if an edge from C intersect with an edge from D.

An open gate is a gray edge from a cycle C that does not intersect with any other gray edge from C. An open gate g_1 from C is closed if another gray edge (which is not from C) intersects with g_1. All open gates of $G(\pi)$ must be closed [8].

In the example of Fig. 1, the cycle $C_1 = (5, 3, 4, 1)$ is convergent and oriented, the cycle $C_2 = (6, 2)$ is convergent and non-oriented, and the cycle $C_3 = (8, -7)$ is divergent. The gray edge from C_1 adjacent to black edges 1 and 4 intersects with the gray edge from C_2 adjacent to black edges 2 and 6, so the cycles C_1 and C_2 intersect.

3 Approximation Algorithms

In this section, we present approximation algorithms considering both unsigned and signed permutations.

3.1 Unsigned Case

Here we present an approximation algorithm with a factor of $3 - k$ based on breakpoints for SbRTwPR on unsigned permutations.

Lemma 1 (Kececioglu and Sankoff [6]). *For any reversal ρ, $\Delta b(\pi, \rho) \leq 2$.*

Lemma 2 (Walter *et al.* [10]). *For any transposition τ, $\Delta b(\pi, \tau) \leq 3$.*

Lemma 3. *Given an instance (π, k) for* SbRTwPR *on unsigned permutations, and an optimal sequence of events S, the average number of breakpoints decreased by an operation in S is less than or equal to $3 - k$.*

Proof. Since $|S|$ is an optimal sequence for the instance (π, k), we have that at least $|S|k$ operations present in S are reversals. By Lemmas 1 and 2, we have that a reversal can remove up to two breakpoints while a transposition can remove up to three. Let $\phi b(S)$ denote the average number of breakpoints decreased by an operation in S, we have that

$$\phi b(S) \leq \frac{(2|S|k) + (3|S|(1-k))}{|S|} = 2k + 3(1-k) = 3 - k. \qquad \square$$

Theorem 1. *Given an instance* (π, k) *for* SbRTwPR *on unsigned permutations, we have that* $d_k(\pi) \geq \frac{b(\pi)}{3-k}$.

Proof. Since $b(\pi)$ breakpoints must be removed in order to turn the permutation π into ι and, by Lemma 3, up to $3 - k$ breakpoints are removed per operation on average, the theorem follows. □

Theorem 2 (Kececioglu and Sankoff [6]). *It is possible to turn an unsigned permutation* π *into* ι *using at most* $b(\pi)$ *reversals.*

Theorem 3. SbRTwPR *is approximable by a factor of* $3 - k$ *on unsigned permutations.*

Proof. By Theorem 2, we can turn any unsigned permutation π into ι using at most $b(\pi)$ reversals. Since we use only reversals, the constraint $\frac{|S_\rho|}{|S|} \geq k$ is not violated. By the lower bound showed in Theorem 1, we have $\frac{b(\pi)}{\frac{b(\pi)}{3-k}} = 3 - k$. □

In order to avoid solutions for the problem consisting exclusively of reversals, we propose the Algorithm 1. This algorithm guarantees the same approximation factor for the problem and tends to provide solutions in which the ratio between the number of reversals and the size of the sorting sequence is close to k.

Algorithm 1: An approximation algorithm for SbRTwPR on unsigned permutations.

```
Input: An unsigned permutation $\pi$ and a value $k$
Output: A sequence of reversals and transpositions that sorts $\pi$
Let $S \leftarrow \{\}$
while $\pi \neq \iota$ do
    if $\frac{|S_\rho|}{|S|+1} \geq k$ and there is a transposition $\tau$ such that $\Delta b(\pi, \tau) \geq 1$ then
        Apply $\tau$ in $\pi$
        Append $\tau$ to $S$
    else
        Let $S'$ be a sequence of reversals that decreases, on average, one or more
        breakpoints per operation [6]
        Apply $S'$ in $\pi$
        Append $S'$ to $S$
return $S$
```

Note that a transposition τ is only applied if two constraints are fulfilled: (i) $\frac{|S_\rho|}{|S|+1} \geq k$, this ensures that the sorting sequence will comply with the main restriction of the problem that $\frac{|S_\rho|}{|S|} \geq k$. (ii) $\Delta b(\pi, \tau) \geq 1$, this constraint ensures that the sorting sequence will contain a maximum of $b(\pi)$ operations, since every reversal sequence removes, on average, one or more breakpoints per operation. Since Algorithm 1 removes one or more breakpoints by iteration, it guarantees that the permutation π will be sorted. In addition, no more than $b(\pi)$ operations will be used to sort π, maintaining the approximation factor of $3 - k$. Since each operation (reversal or transposition) can be found in linear time and $|S| \leq b(\pi) \leq n + 1$, the running time of Algorithm 1 is $\mathcal{O}(n^2)$.

3.2 Signed Case

Here we present an approximation algorithm with a factor of $3 - \frac{3k}{2}$ based on the cycle graph for SbRTwPR on signed permutations.

Lemma 4 (Hannenhalli and Pevzner [5]**).** *For any reversal ρ, $\Delta c(\pi, \rho) \leq 1$.*

Lemma 5 (Bafna and Pevzner [1]**).** *For any transposition τ, $\Delta c(\pi, \tau) \leq 2$.*

Lemma 6. *Given an instance (π, k) for* SbRTwPR *on signed permutations, and an optimal sequence of events S, the average number of cycles increased by an operation in S is less than or equal to $2 - k$.*

Proof. Since $|S|$ is an optimal sequence for the instance (π, k), we have that at least $|S|k$ operations in S sequence are reversals. By Lemmas 4 and 5, we have that a reversal creates at most one new cycle, while a transposition creates at most two new cycles. Let $\phi c(S)$ denote the average number of cycles increased by an operation in S, we have that:

$$\phi c(S) \leq \frac{(1|S|k) + (2|S|(1-k))}{|S|} = 1k + 2(1-k) = 2 - k. \qquad \square$$

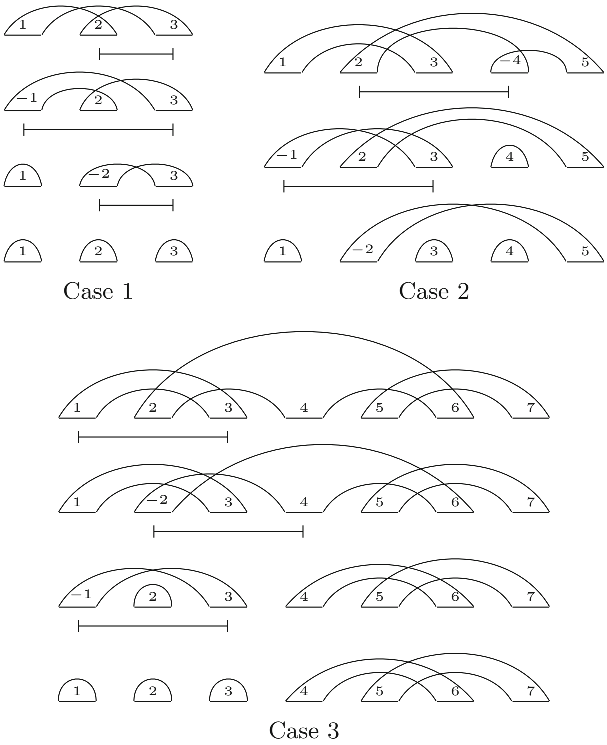
Theorem 4. *Given an instance (π, k) for* SbRTwPR *on signed permutations, we have that $d_k(\pi) \geq \frac{n+1-c(\pi)}{2-k}$.*

Proof. Since $(n + 1) - c(\pi)$ new cycles must be created in order to turn the permutation π into ι and, by Lemma 6, up to $2 - k$ new cycles are created per operation on average, the theorem follows. □

Theorem 5. *Given a signed permutation π, there exists a sequence of reversals S that transforms π into ι such that the average number of cycles increased by any reversal in S is greater than or equal to $2/3$.*

Proof. If at any stage $G(\pi)$ has a divergent cycle C, then there exists a reversal applied to C that increases the number of cycles by one unit [10]. Otherwise, $G(\pi)$ has only convergent cycles, and one of the following is true [8]:

- there exists a long oriented cycle (Fig. 2, Case 1);
- there exists a short cycle C whose open gates are closed by another non-trivial cycle D (Fig. 2, Case 2);
- there exists a long non-oriented cycle C whose open gates are closed by one or more non-trivial cycles (Fig. 2, Case 3).



Case 1

Case 2

Case 3

Fig. 2. Operations applied in each case of Theorem 5.

If $G(\pi)$ has an oriented long cycle C, then we can apply a reversal on its black edges in such a way that it turns C into a divergent cycle C'. Since C' is long, we can apply at least two reversals on C' that increase the number of cycles by one unit each (Fig. 2, Case 1).

In the other two cases we can turn the cycle C into an oriented cycle C' by applying one reversal to a cycle D that closes an open gate from C. If C' is short, we can break it into two trivial cycles with a reversal, and this second reversal turns D into a divergent cycle D', which guarantees that we can apply a third reversal to D' that increases the number of cycles by one (Fig. 2, Case 2). If C' is long, then we can apply at least two reversals that increase the number of cycles by one unit each (Fig. 2, Case 3).

In the three cases above we applied three reversals that increased the number of cycles by two, and the theorem follows. □

Theorem 6. SbRTwPR *is approximable by a factor of* $3 - \frac{3k}{2}$ *on signed permutations.*

Proof. By Theorem 5, we can turn any signed permutation π into ι using at most $\frac{3(n+1-c(\pi))}{2}$ reversals. Since we use only reversals, the constraint $\frac{|S_\rho|}{|S|} \geq k$ is not violated. By the lower bound showed in Theorem 4, we have:

$$\frac{\frac{3(n+1-c(\pi))}{2}}{\frac{n+1-c(\pi)}{2-k}} = 3 - \frac{3k}{2}.$$

□

Note that in order to avoid a solution composed exclusively of reversals, the approach used in Algorithm 1 can be adapted to be applied in this case as well. In Sect. 4, we will present an asymptotic approximation algorithm with an improved approximation factor for the signed case.

4 Asymptotic Approximation for the Signed Case

In this section we show an asymptotic algorithm for SbRTwPR on signed permutations, where $k \in [0,1]$ with an approximation factor of $(\frac{2-k}{1-\frac{k}{3}})$.

Definition 6. *Let $\mathcal{A}_\rho$ be an algorithm that sorts a permutation using only signed reversals and guarantees a ratio of 2/3 of cycles increased by applied reversals (Theorem 5), and let $\mathcal{A}_\rho(\pi)$ represents the sequence of reversals returned by the algorithm that sorts π.*

Now consider Algorithm 2.

Algorithm 2: An approximation algorithm for SbRTwPR on signed permutations.

```
Input: A signed permutation $\pi$ and a value $k$
Output: A sequence of reversals and transpositions that sorts $\pi$
Let $S \leftarrow \{\}$
while $|\mathcal{A}_\rho(\pi)| > k(|S| + |\mathcal{A}_\rho(\pi)|)$ do
    if $G(\pi)$ has a divergent cycle then
        Let $\rho$ be a reversal that increases one cycle in $G(\pi)$
        Apply $\rho$ in $\pi$
        Append $\rho$ to $S$
    else
        Let $S'$ be a sequence of at most two transpositions that increases two
        cycles in $G(\pi)$ [2, Theorem 3.4]
        Apply $S'$ in $\pi$
        Append $S'$ to $S$
Apply $\mathcal{A}_\rho(\pi)$ in $\pi$
Append $\mathcal{A}_\rho(\pi)$ to $S$
if $|S_\rho| < k|S|$ then
    Replace the last two transpositions of $S$ with six reversals [8]
return $S$
```

Lemma 7. *Given a signed permutation π, Algorithm 2 sorts π using at most $(n + 1 - c(\pi))/(1 - k/3) + 4$ operations.*

Proof. Let $S = (S_1, \dots, S_{|S|})$ be the sorting sequence generated by the algorithm without considering the substitution of transpositions by reversals applied in line 14. Let S' be the subsequence of operations applied in the while loop of lines 2 to 10. Each operation in S' increases the number of cycles by at least one unit, and each operation in $S \setminus S'$ (that is, the operations applied outside the while loop) increases on average in $2/3$ the number of cycles. By the condition of line 2, we have that $|S'| \geq (1 - k)|S|$ and, therefore, the average increase in the number of cycles in S is at least $\frac{(1-k)|S| + k|S|2/3}{|S|} = 1 - k/3$. Since these operations increase at most $n + 1 - c(\pi)$ cycles, we have that $|S| \leq \frac{n+1-c(\pi)}{1-k/3}$. In the final sequence, we may increase four operations by replacing the last two transpositions with six reversals (only if necessary). Therefore, the size of this sequence is at most $\frac{n+1-c(\pi)}{1-k/3} + 4$.

Theorem 7. *Algorithm 2 is a $\frac{2-k}{1-k/3}$-asymptotic approximation algorithm for* SbRTwPR.

Proof. Since the algorithm only adds transpositions while the condition of line 2 is satisfied and at most two transpositions are added in the sorting sequence in one iteration, we guarantee that $|S_\rho| \geq k$ by replacing the last two transpositions by reversals. By Lemma 7 and Theorem 4, the sequence S returned by Algorithm 2 satisfies $|S| \leq \frac{n+1-c(\pi)}{1-k/3} + 4 \leq \frac{2-k}{1-k/3} d_k(\pi) + 4$. Therefore, it is a $\frac{2-k}{1-k/3}$-asymptotic approximation algorithm for SbRTwPR. □

5 Conclusion

We investigated the Sorting by Reversals and Transpositions with Proportion Restriction problem and presented an approximation algorithm with a factor of $3 - k$ for unsigned permutations, and an approximation and an asymptotic approximation algorithm with factors $3 - \frac{3k}{2}$ and $\frac{2-k}{1-\frac{k}{3}}$ for signed permutations, respectively.

As future work, we intend to test the proposed algorithms and develop heuristics for the problems. Another interesting research line would be to investigate the complexity of the problems when $0 < k < 1$.

Acknowledgments. This work was supported by the National Council of Technological and Scientific Development, CNPq (grants 400487/2016-0, 140272/2020-8, and 425340/2016-3), the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and the São Paulo Research Foundation, FAPESP (grants 2013/08293-7, 2015/11937-9, 2017/12646-3, and 2019/27331-3).

References

1. Bafna, V., Pevzner, P.A.: Sorting permutations by transpositions. In: Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1995), pp. 614–623. Society for Industrial and Applied Mathematics, Philadelphia (1995)
2. Bafna, V., Pevzner, P.A.: Sorting by transpositions. SIAM J. Discret. Math. **11**(2), 224–240 (1998)
3. Bulteau, L., Fertin, G., Rusu, I.: Sorting by transpositions is difficult. SIAM J. Discret. Math. **26**(3), 1148–1180 (2012)
4. Caprara, A.: Sorting permutations by reversals and Eulerian cycle decompositions. SIAM J. Discret. Math. **12**(1), 91–110 (1999)
5. Hannenhalli, S., Pevzner, P.A.: Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. J. ACM **46**(1), 1–27 (1999)
6. Kececioglu, J.D., Sankoff, D.: Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. Algorithmica **13**, 180–210 (1995)
7. Oliveira, A.R., Brito, K.L., Dias, U., Dias, Z.: On the complexity of sorting by reversals and transpositions problems. J. Comput. Biol. **26**, 1223–1229 (2019)
8. Oliveira, A.R., Brito, K.L., Dias, Z., Dias, U.: Sorting by weighted reversals and transpositions. J. Comput. Biol. **26**, 420–431 (2019)
9. Rahman, A., Shatabda, S., Hasan, M.: An approximation algorithm for sorting by reversals and transpositions. J. Discret. Algorithms **6**(3), 449–457 (2008)
10. Walter, M.E.M.T., Dias, Z., Meidanis, J.: Reversal and transposition distance of linear chromosomes. In: Proceedings of the 5th International Symposium on String Processing and Information Retrieval (SPIRE 1998), pp. 96–102. IEEE Computer Society, Los Alamitos (1998)