Journal of Universal Computer Science, vol. 26, no. 9 (2020), 1076-1094
submitted: 17/1/20, accepted: 24/6/20, appeared: 28/9/20

On the Complexity of Some Variations of Sorting by Transpositions

Alexsandro Oliveira Alexandrino
(Institute of Computing, University of Campinas
Campinas, Brazil
alexsandro@ic.unicamp.br)

Andre Rodrigues Oliveira
(Institute of Computing, University of Campinas
Campinas, Brazil
andrero@ic.unicamp.br)

Ulisses Dias
(School of Technology, University of Campinas
Limeira, São Paulo, Brazil
ulisses@ft.unicamp.br)

Zanoni Dias
(Institute of Computing, University of Campinas
Campinas, Brazil
zanoni@ic.unicamp.br)

Abstract: One of the main challenges in Computational Biology is to find the evolutionary distance between two organisms. In the field of comparative genomics, one way to estimate such distance is to find a minimum cost sequence of rearrangements (large scale mutations) needed to transform one genome into another, which is called the rearrangement distance. In the past decades, these problems were studied considering many types of rearrangements (such as reversals, transpositions, transreversals, and revrevs) and considering the same weight for all rearrangements, or different weights depending on the types of rearrangements. The complexity of the problems involving reversals, transpositions, and both rearrangements is known, even though the hardness proof for the problem combining reversals and transpositions was recently given. In this paper, we enhance the knowledge for these problems by proving that models involving transpositions alongside reversals, transreversals, and revrevs are NP-hard, considering weights w_1 for reversals and w_2 for the other rearrangements such that $w_2/w_1 \leq 1.5$. In addition, we address a cost function related to the number of fragmentations caused by a rearrangement, proving that the problem of finding a minimum cost sorting sequence, considering the fragmentation cost function with some restrictions, is NP-hard for transpositions and the combination of reversals and transpositions.

Key Words: Genome Rearrangements, Weighted Rearrangements, Transpositions
Category: F.2.0, G.2.1.

1 Introduction

A *Genome Rearrangement* is a large scale mutation which changes the position and the orientation of conserved regions in a genome. In the field of comparative genomics, one common approach to estimate the evolutionary distance is to formulate it as the minimum number of rearrangements that transforms a genome into another, which is called *rearrangement distance*.

In comparative genomics, a genome is represented as an ordered sequence of conserved blocks (high similarity regions) and, depending on the genomic information available, different mathematical models can be used. Considering that a genome has no repeated conserved blocks, we can model the genome as a permutation, where each element represents a conserved block. If the orientation of the genes is known, we use signed permutations to indicate the orientation of the elements. If the orientation is unknown, we use unsigned permutations. When using this representation, the problem of finding the rearrangement distance between two genomes is equivalent to the problem of finding the sorting by rearrangements distance of a permutation [Kececioglu and Sankoff, 1995], which is the minimum number of operations needed to transform this permutation into a permutation where each element is positive (or unsigned) and in ascending order.

A *rearrangement model* defines the set of rearrangements allowed to compute the distance. Two of the most studied rearrangements are reversals, which inverts a segment of the genome, and transpositions, which swaps the position of two adjacent segments of the genome. Previous works focused on the problems of Sorting by Reversals, Transpositions, and both rearrangements. [Hannenhalli and Pevzner, 1995] presented a polynomial algorithm for Sorting Signed Permutations by Reversals. For the unsigned case, [Caprara, 1999] showed that the problem is NP-hard. Since transpositions do not change the sign of the elements, when considering only transpositions we have the problem of Sorting Unsigned Permutations by Transpositions, which was also proved to be NP-hard [Bulteau et al., 2012]. Despite having approximation algorithms proposed since the late 1990's [Walter et al., 1998], the complexity of the problems of Sorting (Signed or Unsigned) Permutations by Reversals and Transpositions had been unknown until recently, when [Oliveira et al., 2019] presented a proof that these problems are NP-hard. A block-interchange is an operation that swaps any two segments of the genome without changing the orientation of the elements. The problems of Sorting Unsigned Permutations by Block-Interchanges and Sorting Signed Permutations by Reversals and Block-Interchanges are solvable in polynomial time [Christie, 1996, Mira and Meidanis, 2007].

Other rearrangement operations are transreversals and revrevs. Given two adjacent segments of a genome, a transreversal is an operation that swaps these two segments and inverts the elements of one of these segments, while a revrev is an operation that inverts the elements for each of these two segments. Although the complexity of the problems involving these operations was unknown, many approximation algorithms were presented in the literature [Fertin et al., 2009]. [Gu et al., 1999] presented a 2-approximation algorithm for signed permutations considering reversals, transpositions, and transreversals. [Lin and Xue, 2001] added the revrev operation to the model and gave a 1.75-approximation algorithm. For the model containing transpositions, transreversals, and revrevs, the best result for signed permutations is a 1.5-approximation algorithm [Hartman and Sharan, 2005]. [Lou and Zhu, 2010] presented a 2.25-approximation algorithm for Sorting Unsigned Permutations by Reversals, Transpositions, and Transreversals.

The traditional approach for the genome rearrangements problems is to consider that every rearrangement has the same cost and, thus, the sorting distance consists in finding a minimum length sorting sequence of rearrangements. The weighted approach was motivated by the observation that some rearrangements are more likely to occur than others [Bader and Ohlebusch, 2007, Blanchette et al., 1996]. In a weighted approach, each rearrangement has an associated cost and the goal is to find a minimum-cost sorting sequence of rearrangements. We use w_1 to represent the weight of reversals and w_2 to represent the weights of transpositions, transreversals, and revrevs. For values of w_1 and w_2 such that $1 \leq w_2/w_1 \leq 2$, [Bader and Ohlebusch, 2007] gave a 1.5-approximation algorithm for the model containing reversals, transpositions, and transreversals on signed permutations. For the same problem and considering $w_2/w_1 = 2$, [Eriksen, 2002] presented a 7/6-approximation and a polynomial-time approximation scheme. We show that the problems of Sorting (Signed or Unsigned) Permutations by Rearrangements are NP-hard for rearrangement models that include transpositions or the combination of reversals and transpositions alongside transreversals and revrevs, considering that $w_2/w_1 \leq 1.5$.

[Alexandrino et al., 2018] introduced a new cost function equal to the number of fragmentations (i.e., breaks of adjacent elements) caused by a rearrangement, and they presented approximation algorithms for models containing reversals and transpositions, considering this cost function. [Alexandrino et al., 2020] also considered fragmentation-weighted operations, presenting better approximation algorithms for some permutation classes. In this approach, prefix and suffix operations, which respectively modify the beginning and the end of the genome, cause less fragmentation in the genome and, so, they cost less. Using a parsimony criterion, the problem is modeled so that the number of fragmentations in the genome is minimized during the sorting process. We show that the problems of Sorting (Signed or Unsigned) Permutations by Transpositions, or by Reversals and Transpositions, are NP-hard considering the fragmentation cost function proposed by [Alexandrino et al., 2018, Alexandrino et al., 2020]. We also show that these problems are NP-hard for other combinations of weights related to the number of fragmentations caused by the operations.

This work is organized as follows. Section 2 presents definitions and notations related to the problems. Section 3 shows hardness proofs for the models containing transreversals and revrevs. Section 4 shows hardness proofs for the fragmentation-weighted problems containing transpositions. At last, Section 5 presents final considerations and future work.

2 Definitions

Considering the case where a genome does not have repeated genes, a genome $\mathcal{G}$ is modeled as a permutation whose elements represent conserved blocks. If the orientation of the genes is known, $\mathcal{G}$ is represented as a signed permutation, and $\mathcal{G}$ is represented as an unsigned permutation otherwise. In this case, the problem of finding the rearrangement distance between two genomes $\mathcal{G}_1$ and $\mathcal{G}_2$ is equivalent to the problem of finding the sorting rearrangement distance of a permutation [Fertin et al., 2009].

A *signed permutation* is represented as $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$, such that $\pi_i \in \{-n, \ldots, -1, +1, \ldots, +n\}$ and $|\pi_i| \neq |\pi_j| \leftrightarrow i \neq j$, for all i and j. An *unsigned permutation* is also represented as $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$, but $\pi_i \in \{1, 2, \ldots, n\}$ and $\pi_i \neq \pi_j \leftrightarrow i \neq j$, for all i and j. The *identity permutation*, $\iota = (1\ 2\ \ldots\ n)$, is the sorted permutation, and it is the target of the sorting problems. For signed permutations, we have $\iota = (+1\ +2\ \ldots\ +n)$. The *reverse permutation* is defined as $\eta = (n\ (n-1)\ \ldots\ 1)$, for unsigned permutations, and $\bar{\eta} = (-n\ -(n-1)\ \ldots\ 1)$, for signed permutations.

A *rearrangement model* $\mathcal{M}$ is the set of allowed operations in a rearrangement problem. Considering the unweighted approach, given a rearrangement model $\mathcal{M}$ and a permutation π, the sorting distance $d_{\mathcal{M}}(\pi)$ is equal to the minimum number of rearrangements from $\mathcal{M}$ that sorts the permutation π.

Considering a weighted function $w : \mathcal{M} \to \mathbb{R}$, given a rearrangement $\mathcal{M}$ and a permutation π, the sorting distance $d_{\mathcal{M}}(\pi)$ is equal to $\sum_{i=1}^{\ell} w(\beta_i)$ such that $\beta_i \in \mathcal{M}$, for $1 \leq i \leq \ell$, $\pi \cdot \beta_1 \cdot \ldots \cdot \beta_\ell = \iota$, and $\sum_{i=1}^{\ell} w(\beta_i)$ is minimum. For a sequence of rearrangements $S = \beta_1, \beta_2, \ldots, \beta_\ell$, we have that $w(S) = \sum_{i=1}^{\ell} w(\beta_i)$. The unweighted approach is equivalent to using a unitary weight for all rearrangements.

A *reversal* is a rearrangement which inverts a segment of the genome and, when applied to a signed permutation, flips the sign of the elements in this segment. A *transposition* is a rearrangement that exchanges the position of two adjacent segments of the genome. Next, we formally define these operations.

Definition 1. Considering an unsigned permutation π, a reversal $\rho(i, j)$, with $1 \leq i < j \leq n$, is an operation that when applied to π transforms it in the permutation $\pi \cdot \rho(i,j) = (\pi_1 \ \ldots \ \pi_{i-1} \ \underline{\pi_j \ \pi_{j-1} \ \ldots \ \pi_{i+1} \ \pi_i} \ \pi_{j+1} \ \ldots \ \pi_n)$.

Definition 2. Considering a signed permutation π, a reversal $\bar{\rho}(i,j)$, with $1 \leq i \leq j \leq n$, is an operation that when applied to π transforms it in the permutation $\pi \cdot \bar{\rho}(i,j) = (\pi_1 \ \ldots \ \pi_{i-1} \ \underline{-\pi_j \ -\pi_{j-1} \ \ldots \ -\pi_{i+1} \ -\pi_i} \ \pi_{j+1} \ \ldots \ \pi_n)$.

Definition 3. A transposition $\tau(i,j,k)$, with $1 \leq i < j < k \leq n+1$, is an operation that when applied to π transforms it in the permutation $\pi \cdot \tau(i,j,k) = (\pi_1 \ \ldots \ \pi_{i-1} \ \underline{\pi_j \ \ldots \ \pi_{k-1}} \ \underline{\pi_i \ \ldots \ \pi_{j-1}} \ \pi_k \ \ldots \ \pi_n)$.

For two adjacent segments A and B, a *transreversal* is a rearrangement that inverts the elements of A (type one) or B (type two) and exchanges the position of A and B. Also, when applied to a signed permutation, it flips the sign of the elements in the inverted segment. A *revrev* inverts each of two adjacent segments and, when applied to a signed permutation, flips the sign of the elements affected.

Definition 4. Considering an unsigned permutation π, a transreversal Type 1 $\rho\tau_1(i,j,k)$ and a transreversal Type 2 $\rho\tau_2(i,j,k)$, with $1 \leq i < j < k \leq n+1$, are operations that transform a permutation π in the following way:

$$\pi \cdot \rho\tau_1(i,j,k) = (\ldots \ \pi_{i-1} \ \underline{\pi_j \ \ldots \ \pi_{k-1}} \ \underline{\pi_{j-1} \ \ldots \ \pi_i} \ \pi_k \ \ldots),$$
$$\pi \cdot \rho\tau_2(i,j,k) = (\ldots \ \pi_{i-1} \ \underline{\pi_{k-1} \ \ldots \ \pi_j} \ \underline{\pi_i \ \ldots \ \pi_{j-1}} \ \pi_k \ \ldots).$$

Definition 5. Considering a signed permutation π, a transreversal Type 1 $\bar{\rho\tau}_1(i,j,k)$ and a transreversal Type 2 $\bar{\rho\tau}_2(i,j,k)$, with $1 \leq i < j < k \leq n+1$, are operations that transform a permutation π in the following way:

$$\pi \cdot \bar{\rho\tau}_1(i,j,k) = (\ldots \ \pi_{i-1} \ \underline{\pi_j \ \ldots \ \pi_{k-1}} \ \underline{-\pi_{j-1} \ \ldots \ -\pi_i} \ \pi_k \ \ldots),$$
$$\pi \cdot \bar{\rho\tau}_2(i,j,k) = (\ldots \ \pi_{i-1} \ \underline{-\pi_{k-1} \ \ldots \ -\pi_j} \ \underline{\pi_i \ \ldots \ \pi_{j-1}} \ \pi_k \ \ldots).$$

Definition 6. Considering an unsigned permutation π, a revrev $\rho\rho(i,j,k)$, with $1 \leq i < j < k \leq n+1$, is an operation that when applied to π transforms it in the permutation $\pi \cdot \rho\rho(i,j,k) = (\pi_1 \ \ldots \ \pi_{i-1} \ \underline{\pi_{j-1} \ \ldots \ \pi_i} \ \underline{\pi_{k-1} \ \ldots \ \pi_j} \ \pi_k \ \ldots \ \pi_n)$.

Definition 7. Considering a signed permutation π, a revrev $\bar{\rho\rho}(i,j,k)$, with $1 \leq i < j < k \leq n+1$, is an operation that when applied to π transforms it in the permutation $\pi \cdot \bar{\rho\rho}(i,j,k) = (\pi_1 \ \ldots \ \pi_{i-1} \ \underline{-\pi_{j-1} \ \ldots \ -\pi_i} \ \underline{-\pi_{k-1} \ \ldots \ -\pi_j} \ \pi_k \ \ldots \ \pi_n)$.

We use $\rho\tau$ or $\bar{\rho\tau}$ to denote both types of transreversals generically.

2.1 Breakpoints

The proofs presented in the next sections use the concept of breakpoints in a permutation. This concept is widely used in algorithms for sorting permutations by rearrangements [Fertin et al., 2009]. The definition of breakpoints depends on the rearrangement model considered and whether the permutation is signed or unsigned. First, we define what an extended permutation is.

Definition 8. Given a permutation π, we extend π by adding elements $\pi_0 = 0$ and $\pi_{n+1} = n + 1$. These elements are positive when considering signed permutations. Furthermore, these elements are never affected by a rearrangement.

Next, we define the two types of breakpoints.

Definition 9. An unsigned reversal breakpoint exists between a pair of consecutive elements (π_i, π_{i+1}) if $|\pi_{i+1} - \pi_i| \neq 1$, for $0 \leq i \leq n$.

Example 1. For the unsigned permutation $\pi = (0\ 4\ 3\ 5\ 1\ 2\ 6\ 7)$ (which is in the extended form), we have the following breakpoints from Definition 9 (represented by the symbol $\circ$):

$$\pi = (0 \circ 4\ 3 \circ 5 \circ 1\ 2 \circ 6\ 7).$$

Definition 10. A transposition breakpoint (also called signed reversal breakpoint) exists between a pair of consecutive elements (π_i, π_{i+1}) if $\pi_{i+1} - \pi_i \neq 1$, for $0 \leq i \leq n$.

Example 2. For the unsigned permutation $\pi = (0\ 4\ 3\ 5\ 1\ 2\ 6\ 7)$ (which is in the extended form), we have the following breakpoints from Definition 10 (represented by the symbol $\circ$):

$$\pi = (0 \circ 4 \circ 3 \circ 5 \circ 1\ 2 \circ 6\ 7).$$

Example 3. For the signed permutation $\pi = (+0\ -4\ -3\ +5\ +1\ +2\ -6\ +7)$ (which is in the extended form), we have the following breakpoints from Definition 10 (represented by the symbol $\circ$):

$$\pi = (+0 \circ -4\ -3 \circ +5 \circ +1\ +2 \circ -6 \circ +7).$$

We use the number of breakpoints in a permutation as an indicator of how far this permutation is from the identity permutation. When considering signed permutations, all models use signed reversal breakpoints (Definition 10) and, in this way, only the signed identity permutation has zero breakpoints. Considering unsigned permutations, the reverse permutation $\eta = (n\ (n-1)\ \ldots\ 1)$ has only 2 unsigned reversal breakpoints (Definition 9) and it has $n + 1$ transposition breakpoints (Definition 10). The permutation η can be transformed into ι using only one reversal or at most two operations from the models that have transpositions alongside transreversals or revrevs, but when using only transpositions it needs a sequence of size $\Theta(n)$ [Meidanis et al., 2000]. Therefore, transposition breakpoints (Definition 10) are used for the model considering only transpositions and the other models for unsigned permutations use unsigned reversal breakpoints (Definition 9).

Definition 11. Given a model $\mathcal{M}$ and a permutation π, let $b_{\mathcal{M}}(\pi)$ denote the number of breakpoints in π.

Definition 12. Given a model $\mathcal{M}$, a rearrangement $\beta \in \mathcal{M}$, and a permutation π, let $\Delta b_{\mathcal{M}}(\pi, \beta) = b_{\mathcal{M}}(\pi) - b_{\mathcal{M}}(\pi \cdot \beta)$ denote the variation in the number of breakpoints after applying β to π.

We note that the identity permutation is the only one without breakpoints of definitions 9 and 10.

Definition 13. Considering a type of breakpoint, a *strip* is a maximal sequence of elements without breakpoints between consecutive elements in the sequence.

For unsigned permutations, a strip $(\pi_i\ \pi_{i+1}\ \ldots\ \pi_j)$, with $0 \leq i < j \leq n+1$, is called *increasing* if $\pi_{k+1} > \pi_k$ for all $i \leq k < j$; otherwise the strip is called *decreasing*. A *singleton* is a strip of length one. A singleton is called increasing if it is equal to (π_0) or (π_{n+1}), and it is called decreasing otherwise. Note that the elements π_0 and π_{n+1} always belong to increasing strips. For signed permutations, a strip is called *positive* if its elements are positive, and *negative* otherwise.

2.2 Sorting Permutations by Transpositions

The hardness proofs presented in sections 3 and 4 rely on reductions from the following decision problem:

Definition 14. *SB3T* Problem: Given an unsigned permutation π, decide if it is possible to sort π with a sequence of $b_\tau(\pi)/3$ transpositions.

[Bulteau et al., 2012] proved that *SB3T* is NP-hard by reducing from the Boolean Satisfiability (*SAT*) problem.

3 Models with Transreversal and Revrev

In this section, we prove that the problems with the following models are NP-hard for signed and unsigned permutations:

- $\mathcal{M}_1 = \{\tau, \rho\tau\}$ or $\bar{\mathcal{M}}_1 = \{\tau, \bar{\rho}\tau\}$: Transpositions and Transreversals;
- $\mathcal{M}_2 = \{\rho, \tau, \rho\tau\}$ or $\bar{\mathcal{M}}_2 = \{\bar{\rho}, \tau, \bar{\rho}\tau\}$: Reversals, Transpositions, and Transreversals;
- $\mathcal{M}_3 = \{\tau, \rho\rho\}$ or $\bar{\mathcal{M}}_3 = \{\tau, \bar{\rho}\rho\}$: Transpositions and Revrevs;
- $\mathcal{M}_4 = \{\rho, \tau, \rho\rho\}$ or $\bar{\mathcal{M}}_4 = \{\bar{\rho}, \tau, \bar{\rho}\rho\}$: Reversals, Transpositions, and Revrevs;

- $\mathcal{M}_5 = \{\tau, \rho\tau, \rho\rho\}$ or $\bar{\mathcal{M}}_5 = \{\tau, \bar{\rho}\tau, \bar{\rho}\rho\}$: Transpositions, Transreversals, and Revrevs.
- $\mathcal{M}_6 = \{\rho, \tau, \rho\tau, \rho\rho\}$ or $\bar{\mathcal{M}}_6 = \{\bar{\rho}, \tau, \bar{\rho}\tau, \bar{\rho}\rho\}$: Reversals, Transpositions, Transreversals, and Revrevs.

Besides that, we use costs w_1 for all reversals and w_2 for all transpositions, transreversals, and revrevs. When $w_1 = w_2$, this problem is equivalent to the unweighted approach. If a model does not contain reversals, we consider that $w_1 = \infty$.

Definition 15. *WSR* Problem: Given a rearrangement model $\mathcal{M}$, weights w_1 and w_2, a permutation π, and a value k, decide if it is possible to sort π with a sequence of rearrangements S, such that $w(S) \leq k$ and every rearrangement of S is in $\mathcal{M}$, that is, $d_{\mathcal{M}}(\pi) \leq k$.

Now, we present lower bounds for the sorting distance and bounds for the variation in the number of breakpoints for some permutation families.

Lemma 16. *For any signed permutation π, the weights w_1 and w_2, and model $\mathcal{M} \in \{\bar{\mathcal{M}}_1, \bar{\mathcal{M}}_2, \bar{\mathcal{M}}_3, \bar{\mathcal{M}}_4, \bar{\mathcal{M}}_5, \bar{\mathcal{M}}_6\}$, we have that*

$$d_{\mathcal{M}}(\pi) \geq \min\left\{\frac{w_1}{2}, \frac{w_2}{3}\right\} b_{\mathcal{M}}(\pi).$$

Proof. Since a reversal breaks the permutation on two positions, we have that $-2 \leq \Delta b_{\mathcal{M}}(\pi, \rho) \leq 2$ and $-2 \leq \Delta b_{\mathcal{M}}(\pi, \bar{\rho}) \leq 2$ for any reversal. If an operation β is a transposition, transreversal, or revrev, we have that $-3 \leq \Delta b_{\mathcal{M}}(\pi, \beta) \leq 3$, since these operations break the permutation on three positions.

The identity permutation is the only one without breakpoints and, consequently, a sorting sequence removes $b_{\mathcal{M}}(\pi)$ breakpoints. The minimum cost to remove a breakpoint is equal to $\min\left\{\frac{w_1}{2}, \frac{w_2}{3}\right\}$. Therefore, we conclude that any sorting sequence has cost greater than or equal to $\min\left\{\frac{w_1}{2}, \frac{w_2}{3}\right\} b_{\mathcal{M}}(\pi)$. □

Lemma 17. *For any signed permutation π such that π has only positive strips:*

- $\Delta b_{\bar{\rho}}(\pi, \bar{\rho}) \leq 0$, *for any reversal $\bar{\rho}$;*
- $\Delta b_{\bar{\rho}\tau}(\pi, \bar{\rho}\tau) \leq 1$, *for any transreversal $\bar{\rho}\tau$;*
- $\Delta b_{\bar{\rho}\rho}(\pi, \bar{\rho}\rho) \leq 1$, *for any revrev $\bar{\rho}\rho$.*

Proof. Consider a reversal $\bar{\rho}$ and let $\pi' = \pi \cdot \bar{\rho} = (\pi_1 \ \dots \ \pi_{i-1} \ \underline{-\pi_j \ \dots \ -\pi_i} \ \pi_{j+1} \ \dots \ \pi_n)$, with $1 \leq i \leq j \leq n$. Suppose for the sake of contradiction that $\Delta b_{\bar{\rho}}(\pi, \bar{\rho}) > 0$, which indicates that (i) $(\pi_{i-1}, -\pi_j)$ is not a breakpoint or (ii) $(-\pi_i, \pi_{j+1})$ is not a breakpoint. If $(\pi_{i-1}, -\pi_j)$ is not a breakpoint, then π_{i-1} and

$-\pi_j$ must have the same sign. In the same way, if $(-\pi_i, \pi_{j+1})$ is not a breakpoint, then $-\pi_i$ and π_{j+1} must have the same sign, which contradicts the fact that π has only positive elements. Therefore, $\Delta b_{\bar{\rho}}(\pi, \bar{\rho}) \leq 0$.

Consider a transreversal Type 1 $\bar{\rho}\tau_1$ and let $\pi' = \pi \cdot \bar{\rho}\tau_1 = (\pi_1 \ \dots \ \pi_{i-1} \ \underline{\pi_j \ \dots \ \pi_{k-1}} \ \underline{-\pi_{j-1} \ \dots \ -\pi_i} \ \pi_k \ \dots \ \pi_n)$, with $1 \leq i < j < k \leq n+1$. Using a similar argument to the one used for reversals: the pairs $(\pi_{k-1}, -\pi_{j-1})$ and $(-\pi_i, \pi_k)$ must be breakpoints and only the pair (π_{i-1}, π_j) may not be a breakpoint. Therefore, $\Delta b_{\bar{\rho}\tau}(\pi, \bar{\rho}\tau_1) \leq 1$. We use an analogous argument for a transreversal Type 2.

Consider a revrev $\bar{\rho}\rho$ and let $\pi' = \pi \cdot \bar{\rho}\rho = (\pi_1 \ \dots \ \pi_{i-1} \ \underline{-\pi_{j-1} \ \dots \ -\pi_i} \ \underline{-\pi_{k-1} \ \dots \ -\pi_j} \ \pi_k \ \dots \ \pi_n)$, with $1 \leq i < j < k \leq n+1$. The pairs $(\pi_{i-1}, -\pi_{j-1})$ and $(-\pi_j, \pi_k)$ must be breakpoints and only the pair $(-\pi_i, -\pi_{k-1})$ may not be a breakpoint. Therefore, $\Delta b_{\bar{\rho}\rho}(\pi, \bar{\rho}\rho) \leq 1$. □

Theorem 18. *Considering* $\mathcal{M} = \{\bar{\mathcal{M}}_1, \bar{\mathcal{M}}_2, \bar{\mathcal{M}}_3, \bar{\mathcal{M}}_4, \bar{\mathcal{M}}_5, \bar{\mathcal{M}}_6\}$ *and* $w_2/w_1 \leq 1.5$, *WSR is NP-hard.*

Proof. Consider $\mathcal{M} = \bar{\mathcal{M}}_6$. The proof is similar for the other models, since our strategy is to show that if an instance is satisfied, then only transpositions are used to sort the permutation and, therefore, a similar argument can be used for the models since they have a subset of the operations allowed in $\bar{\mathcal{M}}_6$.

We now present a reduction from the *SB3T* problem to *WSR*. Given an instance $\pi = (\pi_1 \ \dots \ \pi_n)$ for *SB3T*, we construct the instance $(\mathcal{M}, w_1, w_2, \pi', k)$ for *WSR*, where π' is the signed permutation $(+\pi_1 \ +\pi_2 \ \dots \ +\pi_n)$ and $k = w_2 b_\tau(\pi)/3$.

Next, we show that the instance π is sorted by $b_\tau(\pi)/3$ transpositions if and only if $d_{\mathcal{M}}(\pi') \leq w_2 b_\tau(\pi)/3$.

$(\rightarrow)$ If π is sorted by a sequence S of length $b_\tau(\pi)/3$, then S also sorts π', since π' has only positive elements, and $w(S) = w_2 b_\tau(\pi)/3$ (note that S only has transpositions and each transposition has cost w_2).

$(\leftarrow)$ If π' is sorted by a sequence S of cost less than or equal to $w_2 b_\tau(\pi)/3$, then we claim that S has only transpositions and, therefore, S also sorts π and it has length $b_\tau(\pi)/3$.

Note that the minimum cost to remove a breakpoint in π' is equal to $\min\{w_1/2, w_2/3\} = w_2/3$, since $w_2/w_1 \leq 1.5$. Since π' has only positive elements, $\pi_{i+1} - \pi_i = 1$ if and only if $\pi'_{i+1} - \pi'_i = 1$, so $b_\tau(\pi) = b_{\mathcal{M}}(\pi')$. Therefore, the lower bound of Lemma 16 becomes $w_2 b_{\mathcal{M}}(\pi')/3 = w_2 b_\tau(\pi)/3$.

We have that $w(S)$ is equal to the lower bound $w_2 b_\tau(\pi)/3$ and, consequently, every rearrangement of S has to remove exactly $w' \times 3/w_2$ breakpoints, where w' is the rearrangement cost. Note that a sorting sequence removes $b_\tau(\pi)/3$ breakpoints. Now, suppose that S has an operation that is not a transposition. Let β be the first non transposition in S to be applied. Note that before β is applied, all strips in the permutation are positive since transpositions do not change the sign of elements. By Lemma 17, a reversal does not remove breakpoints and a transreversal or revrev remove at most one breakpoint in permutations that have only positive strips, which contradicts the fact that every rearrangement of S removes $w' \times 3/w_2$ breakpoints. Therefore, S has only transpositions. Also, since $w(S) = w_2 b_\tau(\pi)/3$, we conclude that S has length $b_\tau(\pi)/3$. $\square$

Lemma 19. *For any unsigned permutation π, the weights w_1 and w_2, and model $\mathcal{M} \in \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5, \mathcal{M}_6\}$, we have that*

$$d_{\mathcal{M}}(\pi) \geq \min\left\{\frac{w_1}{2}, \frac{w_2}{3}\right\} b_{\mathcal{M}}(\pi).$$

Proof. Similar to the proof of Lemma 16. $\square$

Lemma 20. *For any unsigned permutation π such that π has only increasing strips,*

- *$\Delta b_\rho(\pi, \rho) \leq 0$, for any reversal ρ;*
- *$\Delta b_{\rho\tau}(\pi, \rho\tau) \leq 1$, for any transreversal $\rho\tau$;*
- *$\Delta b_{\rho\rho}(\pi, \rho\rho) \leq 1$, for any revrev $\rho\rho$.*

Proof. Consider a reversal ρ and let $\pi' = \pi \cdot \rho = (\pi_1 \ \dots\ \pi_{i-1}\ \underline{\pi_j \ \dots\ \pi_i}\ \pi_{j+1} \ \dots\ \pi_n)$, with $1 \leq i \leq j \leq n$. Suppose for the sake of contradiction that $\Delta b_\rho(\pi, \rho) > 0$, which indicates that either (i) (π'_{i-1}, π'_i) is not a breakpoint or (ii) (π'_j, π'_{j+1}) is not a breakpoint. Note that the strips in $\pi'_i, \dots, \pi'_j$ are all decreasing, since π has only increasing strips. Let $S = (\pi_{i'}, \dots, \pi_{i-1})$ be the strip containing the element π_{i-1} in π. If (π'_{i-1}, π'_i) is not a breakpoint, then the strip S becomes equal to $S' = (\pi_{i'}, \dots, \pi_{i-1}, \pi_j, \dots, \pi_{j'})$ in π'. Since π has only increasing strips, we have that S has at least two elements or $\pi_{i-1} = \pi_0$. So, the strip S' must be an increasing strip, which contradicts the fact that the strips in $\pi'_i, \dots, \pi'_j$ are all decreasing. We reach a similar contradiction if (π'_j, π'_{j+1}) is not a breakpoint. Therefore, $\Delta b_\rho(\pi, \rho) \leq 0$.

Consider a transreversal Type 1 $\rho\tau_1$ and let $\pi' = \pi \cdot \rho\tau_1(i, j, k) = (\pi_1 \ \dots\ \pi_{i-1}\ \underline{\pi_j \ \dots\ \pi_{k-1}}\ \underline{\pi_{j-1} \ \dots\ \pi_i}\ \pi_k \ \dots\ \pi_n)$. Using a similar argument, the pairs (π_{k-1}, π_{j-1}) and (π_i, π_k) are breakpoints and only the pair (π_{i-1}, π_j) may not be a breakpoint. Therefore, $\Delta b_{\rho\tau}(\pi, \rho\tau_1) \leq 1$. We use an analogous argument for a transreversal Type 2.

Consider a revrev $\rho\rho$ and let $\pi' = \pi \cdot \rho\rho = (\pi_1 \ \dots\ \pi_{i-1}\ \underline{\pi_{j-1} \ \dots\ \pi_i}\ \underline{\pi_{k-1} \ \dots\ \pi_j}\ \pi_k \ \dots\ \pi_n)$, with $1 \leq i < j < k \leq n+1$. The pairs (π_{i-1}, π_{j-1}) and (π_j, π_k) must be breakpoints and only the pair (π_i, π_{k-1}) may not be a breakpoint. Therefore, $\Delta b_{\rho\rho}(\pi, \rho\rho) \leq 1$. $\square$

Theorem 21. *Considering* $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5, \mathcal{M}_6\}$ *and* $w_2/w_1 \leq 1.5$*, WSR is NP-hard.*

Proof. Consider $\mathcal{M} = \mathcal{M}_6$. The proof is similar for the other models, since our strategy is to show that if an instance is satisfied, then only transpositions are used to sort the permutation and, therefore, a similar argument can be used for the models since they have a subset of the operations allowed in $\mathcal{M}_6$.

We also present a reduction from the *SB3T* problem to *WSR*. Given an instance $\pi = (\pi_1 \ \dots \ \pi_n)$ for *SB3T*, we construct the instance $(\mathcal{M}, w_1, w_2, \pi', k)$ for *WSR*, where $k = w_2 b_\tau(\pi)/3$ and π' is a permutation with $2n$ elements such that $\pi'_{2i-1} = 2\pi_i - 1$ and $\pi'_{2i} = 2\pi_i$, with $1 \leq i \leq n$.

Next, we show that the instance π is sorted by $b_\tau(\pi)/3$ transpositions if and only if $d_\mathcal{M}(\pi') \leq w_2 b_\tau(\pi)/3$.

$(\rightarrow)$ If π is sorted by a sequence S of length $b_\tau(\pi)/3$, then we construct the sorting sequence S' such that, for every transposition $\tau(i, j, k)$ in S, we add the transposition $\tau(2i-1, 2j-1, 2k-1)$ in S', since every element of π was mapped into two consecutive elements of π'. In this way, $w(S') = w_2 b_\tau(\pi)/3$.

$(\leftarrow)$ If π' is sorted by a sequence S' of cost less than or equal to $w_2 b_\tau(\pi)/3$, then we claim that there exists the sequence S such that S has only transpositions, it sorts π, and it has length $b_\tau(\pi)/3$.

Note that since π'_{2i-1} and π'_{2i} are consecutive elements, the pairs (π'_{2i-1}, π'_{2i}) are not breakpoints, for any $1 \leq i \leq n$. Also, for $0 \leq i \leq n$:

- if $\pi_{i+1} - \pi_i = 1$, then $\pi'_{2i+1} - \pi'_{2i} = 2\pi_{i+1} - 1 - 2\pi_i = 1$;
- if $\pi_{i+1} - \pi_i > 1$, then $\pi'_{2i+1} - \pi'_{2i} = 2\pi_{i+1} - 1 - 2\pi_i = 2(\pi_{i+1} - \pi_i) - 1 > 1$;
- if $\pi_{i+1} - \pi_i < 1$, then $\pi'_{2i+1} - \pi'_{2i} = 2\pi_{i+1} - 1 - 2\pi_i = 2(\pi_{i+1} - \pi_i) - 1 < 1$.

In this way, $b_\tau(\pi) = b_\mathcal{M}(\pi')$. Note that the minimum cost to remove a breakpoint in π' is equal to $\min\{w_1/2, w_2/3\} = w_2/3$, since $w_2/w_1 \leq 1.5$. So, the lower bound of Lemma 19 becomes $w_2 b_\mathcal{M}(\pi')/3 = w_2 b_\tau(\pi)/3$. We have that $w(S)$ is equal to the lower bound $w_2 b_\tau(\pi)/3$ and, consequently, every rearrangement of S' has to remove exactly $w' \times 3/w_2$ breakpoints, where w' is the rearrangement cost.

Suppose that S' has an operation that is not a transposition and let β be the first non transposition in S' to be applied. Note that before β is applied, all strips of the permutation are increasing, since transpositions that remove 3 breakpoints do not inverse increasing strips. By Lemma 20, β does not remove $w' \times 3/w_2$ breakpoints, where w' is the rearrangement cost, which is a contradiction. Therefore, S' has only transpositions.

Since $w(S') = w_2 b_\tau(\pi)/3$, the sequence S' has $b_\tau(\pi)/3$ transpositions. Note that the transpositions of S' do not break the pairs (π'_{2i-1}, π'_{2i}), for $1 \leq i \leq n$. Consider $S' = \tau'_1, \tau'_2, \dots, \tau'_{b_\tau(\pi)/3}$. Now, we construct the sorting sequence

$S = \tau_1, \tau_2, \ldots, \tau_{b_\tau(\pi)/3}$ for π, such that $\tau_x = \tau((i+1)/2, (j+1)/2, (k+1)/2)$ for $\tau'_x = \tau(i, j, k)$, with $1 \leq x \leq b_\tau(\pi)/3$. □

4 Fragmentation-Weighted Rearrangements

To introduce the fragmentation cost function, we first formally define prefix, suffix, and complete rearrangements.

Definition 22. A *prefix reversal* is a reversal $\rho(1, j)$ or $\bar{\rho}(1, j)$, with $1 \leq j \leq n$. A *suffix reversal* is a reversal $\rho(i, n)$ or $\bar{\rho}(i, n)$, with $1 \leq i \leq n$. Furthermore, a reversal $\rho(i, j)$ or $\bar{\rho}(i, j)$ is *complete* when $i = 1$ and $j = n$.

Definition 23. A *prefix transposition* is a transposition $\tau(1, j, k)$, with $1 < j < k \leq n+1$. A *suffix transposition* is a transposition $\tau(i, j, n+1)$, with $1 \leq i < j \leq n$. Furthermore, a transposition $\tau(i, j, k)$ is called *complete* if $i = 1$ and $k = n + 1$.

For any pair of consecutive positions $(i, i+1)$ of a permutation π, with $1 \leq i < n$, a rearrangement β causes *fragmentation* between $(i, i+1)$ if π_i and π_{i+1} are not adjacent in $\pi \cdot \beta$.

Definition 24. Formally, the fragmentation cost function $f : \mathcal{M} \to \mathbb{R}$, where $\mathcal{M}$ is a rearrangement model, is defined as

$$f(\rho(i,j)) = \begin{cases} f^0, \text{ if } i = 1 \text{ and } j = n \\ f^1, \text{ if } i = 1 \text{ and } j < n \\ f^1, \text{ if } i > 1 \text{ and } j = n \\ f^2, \text{ if } i > 1 \text{ and } j < n, \end{cases} \tag{1}$$

$$f(\tau(i,j,k)) = \begin{cases} f^1, \text{ if } i = 1 \text{ and } k = n+1 \\ f^2, \text{ if } i = 1 \text{ and } k < n+1 \\ f^2, \text{ if } i > 1 \text{ and } k = n+1 \\ f^3, \text{ if } i > 1 \text{ and } k < n+1, \end{cases} \tag{2}$$

where f^0 is a non-negative constant and f^1, f^2, and f^3 are positive constants. The superscript number in these constants indicates the number of fragmentations caused by the operation receiving that weight. We note that $f(\rho(i, j)) = f(\bar{\rho}(i, j))$, for all i and j.

[Alexandrino et al., 2018] considered that $f^0 = 0$, $f^1 = 1$, $f^2 = 2$, and $f^3 = 3$. In our hardness proofs, we consider the following conditions: $f^3/f^2 \leq 1.5$ and $f^3/f^1 \leq 3$. Note that the cost function considered by [Alexandrino et al., 2018] satisfies these conditions. In the next sections, even when omitted, we consider these conditions to be always true.

Given a rearrangement model $\mathcal{M}$, the fragmentation sort distance of a permutation π is denoted by $d^f_{\mathcal{M}}(\pi)$.

4.1 Fragmentation Breakpoints

Now, we present the definitions of fragmentation breakpoints, which are similar to the definitions presented in Section 2.1, but they do not use the extended permutation.

Definition 25. An unsigned reversal fragmentation breakpoint exists between a pair of consecutive elements (π_i, π_{i+1}) if $|\pi_{i+1} - \pi_i| \neq 1$, for $1 \leq i < n$.

Example 4. For the unsigned permutation $\pi = (4\ 3\ 5\ 1\ 2\ 6)$, we have the following breakpoints from Definition 25 (represented by the symbol $\circ$):

$$\pi = (4\ 3 \circ 5 \circ 1\ 2 \circ 6).$$

Definition 26. A transposition fragmentation breakpoint (also called signed reversal fragmentation breakpoint) exists between a pair of consecutive elements (π_i, π_{i+1}) if $\pi_{i+1} - \pi_i \neq 1$, for $1 \leq i < n$.

Example 5. For the unsigned permutation $\pi = (4\ 3\ 5\ 1\ 2\ 6)$, we have the following breakpoints from Definition 26 (represented by the symbol $\circ$):

$$\pi = (4 \circ 3 \circ 5 \circ 1\ 2 \circ 6).$$

Example 6. For the signed permutation $\pi = (-4\ -3\ +5\ +1\ +2\ -6)$, we have the following breakpoints from Definition 26 (represented by the symbol $\circ$):

$$\pi = (-4\ -3 \circ +5 \circ +1\ +2 \circ -6).$$

Given a rearrangement model $\mathcal{M}$, the number of fragmentation breakpoints in a permutation π is denoted by $b^f_{\mathcal{M}}(\pi)$ and the change in the number of fragmentation breakpoints caused by a rearrangement β is denoted by $\Delta b^f_{\mathcal{M}}(\pi, \beta)$. The strips are defined in a similar way.

For unsigned permutations, only the identity permutation ι has no transposition fragmentation breakpoints (Definition 26) and only the identity permutation ι and the reverse permutation η have no unsigned reversal fragmentation breakpoints (Definition 25). Note that a complete reversal does not remove breakpoints.

For signed permutations, only the identity permutation ι and the reverse permutation $\bar{\eta}$ have no signed reversal fragmentation breakpoints (Definition 26). Note that, for signed permutation, we only use signed reversal fragmentation breakpoints (Definition 26).

Lemma 27 [Alexandrino et al., 2018]. *For any permutation π and transposition τ, the maximum number of breakpoints from definitions 25 or 26 removed by τ is equal to the number of fragmentations caused by this operation.*

Lemma 28 [Alexandrino et al., 2018]. *For any unsigned permutation π and reversal ρ, the maximum number of breakpoints from definition 25 removed by ρ is equal to the number of fragmentations caused by this operation.*

Lemma 29 [Alexandrino et al., 2018]. *For any signed permutation π and reversal $\bar{\rho}$, the maximum number of breakpoints from definition 26 removed by $\bar{\rho}$ is equal to the number of fragmentations caused by this operation.*

Lemma 30. *For any unsigned permutation π, we have*

$$d^f_\tau(\pi) \geq \frac{f^3 b^f_\tau(\pi)}{3}, \text{ and}$$
$$d^f_{\{\rho,\tau\}}(\pi) \geq \frac{f^3 b^f_{\{\rho,\tau\}}(\pi)}{3}.$$

Proof. Consider the model with only transpositions. The identity permutation has no breakpoints and, therefore, a sorting sequence must remove all breakpoints from π. From Lemma 29, the ratio between cost and breakpoints removed is at least $min(f^3/3, f^2/2, f^1/1)$. By the restriction on the weights of the transpositions, we have that $min(f^3/3, f^2/2, f^1/1) = f^3/3$. Therefore, to remove all breakpoints, a sorting sequence has cost of at least $f^3 b^f_\tau(\pi)/3$. When considering reversals, the minimum ratio between cost and breakpoints removed is also $f^3/3$ and the proof is similar. □

Lemma 31. *For any signed permutation π, we have*

$$d^f_{\{\bar{\rho},\tau\}}(\pi) \geq \frac{f^3 b^f_{\{\bar{\rho},\tau\}}(\pi)}{3}.$$

Proof. Similar to the proof of Lemma 30. □

4.2 Hardness Proofs

Now, we prove that the problems with the following models are NP-hard considering the fragmentation cost function:

- $\mathcal{M}^f_1 = \{\tau\}$: Transpositions on Unsigned Permutations;
- $\mathcal{M}^f_2 = \{\bar{\rho},\tau\}$: Reversals and Transpositions on Signed Permutations;
- $\mathcal{M}^f_3 = \{\rho,\tau\}$: Reversals and Transpositions on Unsigned Permutations.

Definition 32. *FWSR* Problem: Given a rearrangement model $\mathcal{M}$, a permutation π, and a value k, decide if it is possible to sort π with a sequence of rearrangements S, such that $f(S) \leq k$ and every rearrangement of S is in $\mathcal{M}$, that is, $d^f_{\mathcal{M}}(\pi) \leq k$.

Lemma 33. *For any unsigned permutation π, such that $\pi_1 = 1$ and $\pi_n = n$, we have that $\Delta b^f_\tau(\pi, \tau)$ is less than the number of fragmentations caused by τ, if τ is a prefix, suffix, or complete transposition.*

Proof. Consider a prefix transposition $\tau(1, j, k)$, with $1 < j < k \leq n+1$, and let $\pi' = \pi \cdot \tau(1, j, k) = (\underline{\pi_j \ \ldots \ \pi_{k-1}}\ \underline{\pi_1 \ \ldots \ \pi_{j-1}}\ \pi_k \ \ldots \ \pi_n)$. Note that the number of fragmentations caused by $\tau(1, j, k)$ is equal to 2. The pair (π_{k-1}, π_1) must be a breakpoint (Definition 26) in π', since $\pi_1 = 1$ and $\pi_1 - \pi_{k-1} < 0$. Since only the pair (π_{j-1}, π_k) may not be a breakpoint, we have that $\Delta b^f_\tau(\pi, \tau(1, j, k)) < 2$. Note that fragmentation breakpoints do not consider the extended permutation and the pair (π_0, π_1) is not considered.

The proof is similar when considering suffix or complete transpositions. □

Considering the model $\mathcal{M}^f_1 = \{\tau\}$, we define the problem of deciding if a permutation can be sorted with a sequence of cost equal to the the lower bound from Lemma 30.

Definition 34. *FWST* Problem: Given a permutation π, decide if it is possible to sort π with a sequence of transpositions S, such that $f(S) = f^3 b^f_\tau(\pi)/3$, that is, $d^f_{\mathcal{M}^f_1}(\pi) = f^3 b^f_\tau(\pi)/3$.

Theorem 35. *The FWST problem is NP-hard.*

Proof. Given an instance π for *SB3T*, we construct the instance π' for *FWST*, where $\pi' = (1\ (\pi_1 + 1)\ (\pi_2 + 1)\ \ldots\ (\pi_n + 1)\ n + 2)$.

Note that, for $1 \leq i \leq n+2$, $\pi'_i = \pi_{i-1} + 1$ and, so, $\pi'_{i+1} - \pi'_i = (\pi_i + 1) - (\pi_{i-1} + 1) = \pi_i - \pi_{i-1}$. Therefore, the pair (π_i, π_{i+1}) is a breakpoint (Definition 10) if and only if (π'_{i+1}, π'_{i+2}) is a breakpoint (Definition 26), for $0 \leq i \leq n$, and $b_\tau(\pi) = b^f_\tau(\pi')$. Observe that the Definition 26 does not consider the extended permutation.

Next, we show that the instance π is sorted by $b_\tau(\pi)/3$ transpositions if and only if $d^f_\tau(\pi') = f^3 b^f_\tau(\pi')/3$.

$(\rightarrow)$ If π is sorted by a sequence S of length $b_\tau(\pi)/3$, then we construct a sorting sequence S' for π' by mapping each transposition $\tau(i, j, k)$ in S into the transposition $\tau(i+1, j+1, k+1)$ in S'. Note that each transposition in S' has fragmentation cost of f^3 and so $f(S') = f^3 b^f_\tau(\pi')/3$.

$(\leftarrow)$ If π' is sorted by a sequence S' such that $f(S') = f^3 b^f_\tau(\pi')/3$, then we claim that all transpositions of S' have cost f^3 and, consequently, there exists sequence S of length $b_\tau(\pi)/3$ which sorts π.

Suppose for the sake of contradiction that S' has a prefix, suffix, or complete transposition. Consider $S' = \tau'_1, \tau'_2, \ldots, \tau'_m$. Since $f(S')$ is exactly the lower bound of Lemma 30, we have that for every transposition τ' in S', the number of breakpoints removed by τ' must be equal to the number of fragmentations caused by τ'. Let τ'_i be the first prefix, suffix, or complete transposition of S' applied to the permutation. Before τ'_i is applied, we have that the first and the last elements of the permutation are in the correct position and, according to Lemma 33, the variation in the number of breakpoints caused by τ' is less than the number of fragmentations caused by it, which is a contradiction. Therefore, all transpositions of S' have cost f^3 and $|S'| = b_\tau(\pi)/3$.

Now, we construct the sorting sequence $S = \tau_1, \tau_2, \dots, \tau_{b_\tau(\pi)/3}$ for π, such that $\tau_x = \tau(i-1, j-1, k-1)$ for $\tau'_x(i,j,k)$, with $1 \leq x \leq b_\tau(\pi)/3$. $\square$

Corollary 36. *Considering the model $\mathcal{M}_1^f = \{\tau\}$, the FWSR problem is NP-hard.*

Now, we present hardness proofs for the *FWSR* problem, considering the models $\mathcal{M}_2^f = \{\bar{\rho}, \tau\}$ and $\mathcal{M}_3^f = \{\rho, \tau\}$, using a reduction from the *FWST* problem.

Lemma 37 [Alexandrino et al., 2018]. *For any sequence S with more than one complete reversal, there exists a sequence S' such that S and S' have the same effect and S' has at most one complete reversal, which is the last rearrangement of S', if it exists.*

Lemma 38. *For any signed permutation π, such that all strips of π are positive, we have that $\Delta b^f_{\mathcal{M}_2^f}(\pi, \bar{\rho})$ is less than the number of fragmentations caused by $\bar{\rho}$, if $\bar{\rho}$ is a prefix or suffix reversal.*

Proof. Using a similar argument to the one used in Lemma 17, we know that a reversal does not remove breakpoints of π and the results follows. $\square$

Theorem 39. *Considering the model $\mathcal{M}_2^f = \{\bar{\rho}, \tau\}$, the FWSR problem is NP-hard.*

Proof. Given an instance π for the *FWST* problem, we construct the instance $(\mathcal{M}_2^f, \pi', k)$ for *FWSR*, where $\pi' = (+\pi_1 \ \dots \ +\pi_n)$ and $k = f^3 b^f_\tau(\pi)/3$.

Note that $b^f_\tau(\pi) = b^f_{\mathcal{M}_2^f}(\pi')$, since the definition of breakpoint is the same and all elements of π' are positive.

Next, we show that $d^f_\tau(\pi) = f^3 b^f_\tau(\pi)/3$ if, and only if, $d^f_{\mathcal{M}_2^f}(\pi) = f^3 b^f_\tau(\pi)/3$.

($\rightarrow$) If π is sorted by a sequence of transpositions S of cost $f^3 b^f_\tau(\pi)/3$, then S also sorts π' with the same cost.

($\leftarrow$) If π' is sorted by a sequence S of cost $f^3 b^f_\tau(\pi)/3$, then we claim that S has only transpositions and also sorts π with cost $f^3 b^f_\tau(\pi)/3$.

Consider, without loss of generality, that S has at most one complete reversal, which is the last rearrangement of S, if it exists (Lemma 37).

Since $f(S)$ is exactly the lower bound of Lemma 30, we have that for every operation β in S, the number of breakpoints removed by β must be equal to the number of fragmentations caused by it. Suppose for the sake of contradiction that S has a reversal that is not a complete reversal. Let $\bar{\rho}$ be the first reversal of S to be applied. Before $\bar{\rho}$ is applied, the permutation has only positive strips and, according to Lemma 38, this rearrangement does not remove breakpoints, which is a contradiction. Therefore, S has only transpositions, except for the last rearrangement that may be a complete reversal. Suppose that the last rearrangement β' in S is a complete reversal. Since the other rearrangements of S are transpositions, all elements of the permutation are positive before applying β', and a complete reversal would turn every element into a negative element, which contradicts the fact that S is a sorting sequence for π'. Therefore, S has only transpositions and also sorts π with cost $f^3 b_\tau^f(\pi)/3$. □

Lemma 40. *For any unsigned permutation π, such that all strips of π are increasing, we have that $\Delta b^f_{\mathcal{M}_3^f}(\pi, \rho)$ is less than the number of fragmentations caused by ρ, if ρ is a prefix or suffix reversal.*

Proof. Using a similar argument to the one used in Lemma 20, we know that a reversal does not remove breakpoints of π and the results follows. □

Note that, for any unsigned permutation π in which all strips of π are increasing, applying a transposition τ in π, such that $\Delta b^f_{\mathcal{M}_3^f}(\pi, \tau)$ is maximum, does not turn any increasing strip into a decreasing strip. However, a complete reversal applied to π turns all increasing strips into decreasing strips.

Theorem 41. *Considering the model $\mathcal{M}_3^f = \{\rho, \tau\}$, the FWSR problem is NP-hard.*

Proof. Similar to the proof of theorems 21 and 39, using Lemma 40. □

5 Conclusions

We showed that the problems of Sorting (Signed or Unsigned) Permutations by Rearrangements are NP-hard for twelve rearrangement models, which include transpositions alongside reversals, transreversals, and revrevs, considering that a reversal has cost w_1, the other rearrangements have cost w_2 and $w_2/w_1 \leq 1.5$. Moreover, we presented hardness proofs for the problems of Sorting Permutations by Fragmentation-Weighted Rearrangements, considering transpositions and the combination of reversals and transpositions, for some combinations of weights.

The complexity of the first set of problems remains open when $w_2/w_1 > 1.5$, and the complexity of the fragmentation-weighted problems remains open for the model with only reversals. Another future work direction is to study the hardness of approximation for the optimization problems of genome rearrangements, including models with only transpositions and transpositions alongside reversals.

Acknowledgments

This work was supported by the National Council of Technological and Scientific Development, CNPq (grants 400487/2016-0 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

[Alexandrino et al., 2018] Alexandrino, A. O., Lintzmayer, C. N., and Dias, Z.: "Approximation Algorithms for Sorting Permutations by Fragmentation-Weighted Operations"; In Algorithms for Computational Biology, volume 10849, pages 53–64. Springer International Publishing, Heidelberg, Germany. 2018.

[Alexandrino et al., 2020] Alexandrino, A. O., Lintzmayer, C. N., and Dias, Z.: "Sorting Permutations by Fragmentation-Weighted Operations"; Journal of Bioinformatics and Computational Biology, 18, 2 (2020), 2050006.

[Bader and Ohlebusch, 2007] Bader, M. and Ohlebusch, E.: "Sorting by Weighted Reversals, Transpositions, and Inverted Transpositions"; Journal of Computational Biology, 14, 5 (2007), 615–636.

[Blanchette et al., 1996] Blanchette, M., Kunisawa, T., and Sankoff, D.: "Parametric Genome Rearrangement"; Gene, 172, 1 (1996), GC11–GC17.

[Bulteau et al., 2012] Bulteau, L., Fertin, G., and Rusu, I.: "Sorting by Transpositions is Difficult"; SIAM Journal on Discrete Mathematics, 26, 3 (2012), 1148–1180.

[Caprara, 1999] Caprara, A.: "Sorting Permutations by Reversals and Eulerian Cycle Decompositions"; SIAM Journal on Discrete Mathematics, 12, 1 (1999), 91–110.

[Christie, 1996] Christie, D. A.: "Sorting Permutations by Block-Interchanges"; Information Processing Letters, 60, 4 (1996), 165–169.

[Eriksen, 2002] Eriksen, N.: "(1+ϵ)-Approximation of Sorting by Reversals and Transpositions"; Theoretical Computer Science, 289, 1 (2002), 517–529.

[Fertin et al., 2009] Fertin, G., Labarre, A., Rusu, I., Tannier, É., and Vialette, S.: " Combinatorics of Genome Rearrangements"; Computational Molecular Biology. The MIT Press, London, England (2009).

[Gu et al., 1999] Gu, Q.-P., Peng, S., and Sudborough, I. H.: "A 2-Approximation Algorithm for Genome Rearrangements by Reversals and Transpositions"; Theoretical Computer Science, 210, 2 (1999): 327–339.

[Hannenhalli and Pevzner, 1995] Hannenhalli, S. and Pevzner, P. A.: "Transforming Men into Mice (Polynomial Algorithm for Genomic Distance Problem)"; In Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'1995), pages 581–592. IEEE Computer Society Press, Washington, DC, USA. 1995.

[Hartman and Sharan, 2005] Hartman, T. and Sharan, R.: "A 1.5-Approximation Algorithm for Sorting by Transpositions and Transreversals"; Journal of Computer and System Sciences, 70, 3 (2005), 300–320.

[Kececioglu and Sankoff, 1995] Kececioglu, J. D. and Sankoff, D.: "Exact and Approximation Algorithms for Sorting by Reversals, with Application to Genome Rearrangement"; Algorithmica, 13 (1995), 180–210.

[Lin and Xue, 2001] Lin, G.-H. and Xue, G.: "Signed Genome Rearrangement by Reversals and Transpositions: Models and Approximations"; Theoretical Computer Science, 259, 1-2 (2001), 513–531.
[Lou and Zhu, 2010] Lou, X.-W. and Zhu, D.-M.: "Sorting Unsigned Permutations by Weighted Reversals, Transpositions, and Transreversals"; Journal of Computer Science and Technology, 25, 4 (2010), 853–863.
[Meidanis et al., 2000] Meidanis, J., Walter, M. E. M. T., and Dias, Z.: "A Lower Bound on the Reversal and Transposition Diameter"; Technical Report IC-00-16, Institute of Computing, University of Campinas (2000).
[Mira and Meidanis, 2007] Mira, C. and Meidanis, J.: "Sorting by block-interchanges and signed reversals"; In Fourth International Conference on Information Technology (ITNG'07), pages 670–676. IEEE Computer Society, Los Alamitos, CA, USA. 2007.
[Oliveira et al., 2019] Oliveira, A. R., Brito, K. L., Dias, U., and Dias, Z.: "On the Complexity of Sorting by Reversals and Transpositions Problems"; Journal of Computational Biology, 26, 11 (2019), 1223–1229.
[Walter et al., 1998] Walter, M. E. M. T., Dias, Z., and Meidanis, J.: "Reversal and Transposition Distance of Linear Chromosomes"; In Proceedings of the 5th International Symposium on String Processing and Information Retrieval (SPIRE'1998), pages 96–102. IEEE Computer Society, Los Alamitos, CA, USA. 1998.