

Sorting Signed Permutations by Inverse Tandem Duplication Random Losses

Tom Hartmann, Max Bannach, and Martin Middendorf *

January 18, 2019

Abstract

Gene order evolution of unichromosomal genomes, for example mitochondrial genomes, has been modelled mostly by four major types of genome rearrangements: inversions, transpositions, inverse transpositions, and tandem duplication random losses. Generalizing models that include all those rearrangements while admitting computational tractability are rare. In this paper we study such a rearrangement model, namely the inverse tandem duplication random loss (iTDRL) model, where an iTDRL duplicates and inverts a continuous segment of a gene order followed by the random loss of one of the redundant copies of each gene. The iTDRL rearrangement has currently been proposed by several authors suggesting it to be a possible mechanism of mitochondrial gene order evolution. We initiate the algorithmic study of this new model of genome rearrangement on signed permutations by proving that a shortest rearrangement scenario that transforms one given gene order into another given gene order can be obtained in quasilinear time. Furthermore, we show that the length of such a scenario, i. e., the minimum number of iTDRLs in the transformation, can be computed in linear time.

1 Introduction

Genome rearrangements are evolutionary events that change the content or the arrangement of the genes on the chromosomes. When compared to other

*TH, MM are associated with the Swarm Intelligence and Complex Systems Group, Faculty of Mathematics and Computer Science, University of Leipzig, Augustusplatz 10, D-04109 Leipzig, Germany. E-mail: {thartmann,middendorf}@informatik.uni-leipzig.de
MB is associated with the Institute for Theoretical Computer Science, Universität zu Lübeck, Germany. E-mail: bannach@tcs.uni-luebeck.de

evolutionary events, e. g., mutations of single nucleotides, genome rearrangements are assumed to occur rarely [1]. By using this and the fact that closely related species often share the same gene order [2] valuable information about phylogenetic relationships of organisms can be inferred by analyzing the differences in gene orders of contemporary species. Such differences in gene orders have been shown to be an effective indicator of evolutionary relations for various animal groups, for instance for Annelida [3] and Brachyura [4]. Two central optimization problems in gene order analysis are the *sorting problem* and the *distance problem*. The former aims to obtain the shortest scenario of rearrangements that transforms one given gene order into another one; and the later asks for the minimum number of rearrangements required by such a transformation. The shortest rearrangement scenario is of particular importance as it is often assumed to be close to reality and more likely than a scenario with a larger number of rearrangements [1].

Mitochondrial gene orders are a particularly fruitful source for phylogenetic investigations because their gene orders are known for a large number of species [5]. The relevant genome rearrangements for mitochondria are inversions (a segment of contiguous genes is reversed and the orientation of each affected gene is flipped), transpositions (two consecutive segments of contiguous genes are exchanged), inverse transpositions (a transposition where one transposed segment is inverted) and tandem duplication random losses (a duplication of a contiguous segment of genes, followed by the random loss of one copy of each of the duplicated genes; henceforth denoted TDRL) [6, 7, 8].

The computational complexities of the sorting problem and the distance problem depend on the types of the considered rearrangements and on whether the gene strandedness of the given gene orders is considered (oriented gene orders) or not (non-oriented gene orders). For example, both problems can be solved efficiently for TDRLs [9, 10] and on oriented gene orders for inversions (e. g., [11, 12, 13]). In contrast, the sorting problem with inversions for non-oriented gene orders [14, 15] and the sorting problem with transpositions [16] are both NP-hard. Several approximation algorithms have been proposed for these cases, e. g., in [17, 18, 19] for sorting with inversions and in [20, 21] for sorting with transpositions.

A generalization of the transposition rearrangement is the TDRL. The TDRL rearrangement was initially studied in [9] for linear gene orders, which are gene orders that are organized in a linear structure. In their work, the authors presented a cost value of α^ℓ to weight a single TDRL that effects $\ell \in \mathbb{N}$ genes while $\alpha \geq 1$ is a constant. The authors presented polynomial time algorithms that solve the sorting problem (and therefore the distance problem) for the cases $\alpha = 1$ and $\alpha \geq 2$. For the case $\alpha = 1$ it is sufficient to consider only TDRLs that effect the whole gene order. It was shown in [10]

that the distance problem and the sorting problem can be solved in polynomial time for circular gene orders, which are gene orders that are organized in a circular structure. In [5] the set of all sorting TDRLs, which are TDRLs that reduce the distance of one gene order towards another given gene order, has been investigated. Bouvel and Rossin studied in [22] the minimum number of TDRLs (each effecting only ℓ genes) that are necessary and sufficient to obtain any gene order from any other gene order. They further proved that all gene orders that can be obtained after a given number of TDRLs define classes of pattern-avoiding permutations (which were further analyzed in [23]). A TDRL variant called mirror TDRL, where the duplicated segment of genes is reversed and the orientation is unchanged, has been investigated in terms of pattern-avoiding permutations in [24]. The authors proved that the sorting problem with mirror TDRLs can be solved in polynomial time. It is worth to mention that a tandem duplication variant has been suggested in which the subsequent loss is not completely random, but depends on the gene orientation or the transcript structure [25].

In order to compute realistic scenarios of rearrangements, several authors considered combinations of different types of rearrangements. The sorting problem with transpositions and inversions has been investigated for non-oriented gene orders with exact algorithms having an exponential runtime (e. g., [26, 27]) and with a machine learning approach [28]. For oriented gene orders approximation algorithms have been designed (e. g., [29, 30, 31]). It is known that the sorting problem can be solved in polynomial time on instances in which the number of genes that are effected by a transposition or an inversion is never greater than two [32]. Walter et al. presented in [33] an approximation algorithm that solves the distance problem with transpositions and inversions for oriented gene orders. When considering transpositions, inversions, and inverse transpositions approximation algorithms (e. g., [34, 35]) and exact exponential-runtime algorithms (see [36]) have been presented. There are various exact exponential-time algorithms and heuristics that solve the sorting problem for oriented gene orders with all four major types of mitochondrial rearrangements, see for instance [37, 38, 39].

As mentioned above, combined genome rearrangement models that include transpositions are typically hard problems where known exact algorithms have an exponential worst case behaviour. Therefore, Yancopoulos et al. [40] and Bergeron et al. [41] suggested the double cut and join genome rearrangement (DCJ), which cuts a (potentially multichromosomal) gene order at two different positions and rejoins the resulting fragments. The DCJ model has the benefit that it allows to include all four major types of unichromosomal genome rearrangements (and also other rearrangements that are common in multichromosomal gene orders) while simplifying the computa-

tional complexity for both, the sorting problem and the distance problem. Furthermore, the DCJ rearrangement allows the coexistence of multiple chromosomes, which may be linear or circular in the genomes. While such genome structures are considered to be aberrant for mitochondria, there are some cases that indicate that cut and join rearrangement mechanisms might be present in the mitochondrial genome of some animal groups [7]. Since the usual mitochondrial genomes are organized in a single circular chromosome [7], there is a need for a tractable genome rearrangement model that includes all major types of rearrangements of mitochondria and excludes the coexistence of multiple chromosomes.

In this work we suggest such a genome rearrangement model that generalizes all major mitochondrial rearrangements and we show that the corresponding sorting problem, as well as the distance problem, can be solved in polynomial time. More precisely, we study the inverse tandem duplication random loss rearrangement (iTDRL). This rearrangement duplicates and inverts a segment of continuous genes of a gene order followed by the random loss of one of the redundant copies of each gene. Evidence for an iTDRL as evolutionary mechanism has been found in mitochondrial gene order comparisons on the walking stick *Ramulus hainanense* [42], the tongue sole *Cynoglossus semilaevis* [43], and the flatfish *Crossorhombus azureus* [44]. The iTDRL rearrangement is also motivated by the fact that inverted duplications often occur in the control region of Insecta mitochondrial genomes [45]. Despite the fact that there is some evidence for the iTDRL rearrangement, it is still not entirely clear whether the iTDRL rearrangement corresponds to a specific molecular mechanism or the composition of two subsequent rearrangement events, e. g., a TDRL followed by an inversion. An iTDRL can always be interpreted as the composition of two subsequent rearrangement events, e. g., a TDRL followed by an inversion. Therefore, further biological evidence is still necessary to determine that the iTDRL rearrangement corresponds to a specific molecular mechanism. In this paper, we initiate the algorithmic study of the iTDRL rearrangement by showing that the sorting problem with iTDRLs for oriented gene orders can be solved in quasilinear time and that the corresponding distance problem can be solved in linear time with respect to the number of genes of the given gene order. The iTDRL rearrangement provides the benefit that it can mimic all major mitochondrial rearrangements: i) an inversion and an inverse transposition can each be represented by at most two iTDRLs and ii) a transposition and a TDRL can each be represented by two iTDRLs. Therefore, the distance problem with iTDRLs provides bounds on the distance problem with all four major mitochondrial rearrangements. Besides of being computationally tractable, iTDRLs are also motivated by the fact that inverted duplications often occur in the control region of insecta

mitochondrial genomes [45].

The paper is organized as follows. Section 2 provides basic definitions and notations. The number of iTDRLs that are necessary and sufficient to obtain a permutation having a specific number of maximum increasing substrings from the identity permutation is investigated in Section 3. Computational results for the distance problem and the sorting problem with iTDRLs are given in Section 4. This section also presents a quasilinear time algorithm for solving the sorting problem for an arbitrary gene order. In Section 5 we discuss that the distance problem with iTDRLs provides bounds on the distance problem with all four major mitochondrial rearrangements. The behaviour of these bounds are further investigated on artificial gene order data in Section 6. A conclusion is given in Section 7.

2 Preliminaries

A *signed permutation* π of length $n \in \mathbb{N}$, denoted by $\pi = (\pi(1) \dots \pi(n))$, is a bijection $\pi: [-n:n] \setminus \{0\} \rightarrow [-n:n] \setminus \{0\}$ such that $\pi(-i) = -\pi(i)$ for all $i \in [-n:n] \setminus \{0\}$. The length of π is denoted by $|\pi|$. Signed permutations are used as a formal model for gene orders in which each element represents a gene and the sign represents its orientation. When the context is clear, a signed permutation is called permutation and the $+$ sign of an element is omitted. The set of all signed permutations of length n is denoted by $s\mathcal{P}_n$. Observe that a signed permutation $\pi = (\pi(-n) \dots \pi(-1) \pi(1) \dots \pi(n))$ can always be represented by $\pi = (\pi(1) \dots \pi(n))$, as we have $\pi(-i) = -\pi(i)$ by definition. The identity permutation $(1 \ 2 \dots n)$ is denoted by ι .

For a permutation $\pi = (\pi(1) \dots \pi(n))$ the corresponding permutation in which the order and the sign of all elements is reversed is defined as permutation $\bar{\pi}$ with $\bar{\pi}(i) = -\pi(n+1-i)$ for all $i \in [1:n]$. Note that $\bar{\pi}$ is uniquely defined for every $\pi \in s\mathcal{P}_n$. Figure 1 illustrates an example of $\bar{\pi}$.

A *subsequence* of $\pi = (\pi(1) \dots \pi(n))$ is a sequence $\pi(i_1)\pi(i_2) \dots \pi(i_k)$ with $1 \leq i_1 < i_2 < \dots < i_k \leq n$. When all elements in a subsequence S of π appear consecutively, then S is called a *substring* of π . We refer to the set of all signed elements of a subsequence S with $\mathcal{E}(S)$ and denote the first (last) element of S with f_S (resp. l_S). A substring $S = \pi(i) \dots \pi(k)$ (of π with $1 \leq i \leq k \leq n$) is called *increasing* if either $i = k$ or $\pi(j) < \pi(j+1)$ for all $j \in [i:k-1]$. An increasing substring is called *maximal* when it cannot be extended into a longer increasing substring. The set of all maximal increasing substrings of a permutation π is denoted by \mathcal{S}_π , and we refer to the number of such objects with $|\mathcal{S}_\pi|$. The notation of maximal increasing substrings is crucial for the development of our algorithmic framework, as it allows the

following structural decomposition of signed permutations:

Definition 1 (Maximal Increasing Substring Decomposition). *Let $\pi \in s\mathcal{P}_n$ be a permutation. The maximal increasing substring decomposition of π is the unique list of pairwise disjoint maximal increasing substrings $\tau_1\tau_2 \dots \tau_{|\mathcal{S}_\pi|}$ of π such that $\pi(1) \dots \pi(n) = \tau_1\tau_2 \dots \tau_{|\mathcal{S}_\pi|}$ and for all $1 \leq j \leq |\mathcal{S}_\pi|$ holds that $|\tau_j| \geq 1$.*

For an example of a maximal increasing substring decomposition consider the permutation $\pi = (1\ 2\ -3\ 4\ -5\ 6\ 7\ 9\ 8)$. The set of maximal increasing substrings of π is $\mathcal{S}_\pi = \{1\ 2, -3\ 4, -5\ 6\ 7\ 9, 8\}$ and the maximal increasing substring decomposition of π is $\tau_1\tau_2\tau_3\tau_4$ with $\tau_1 = 1\ 2$, $\tau_2 = -3\ 4$, $\tau_3 = -5\ 6\ 7\ 9$, and $\tau_4 = 8$. Other examples of the maximal increasing substring decomposition are illustrated in Figure 1 and Figure 3.

A convenient way to work with signed permutations is to represent them as strings over the alphabet of integers. We can then represent a permutation as a concatenation of (character) disjoint substrings. This also holds for reversing the order and the sign of every element of a substring of a permutation, i. e., if $S = \pi(i_1) \dots \pi(i_k)$ is a substring of π , then \bar{S} is the sequence for which $\bar{\pi}(i_j) = -\pi(i_k + i_1 - i_j)$ for all $j \in [1:k]$. Let S_1 and S_2 be two substrings of a permutation π such that $\mathcal{E}(S_1) \cap \mathcal{E}(S_2) = \emptyset$. By $S_1 \oplus S_2$ we denote the sequence that is created by sorting the elements S_1 and S_2 increasingly. Figure 1 illustrates the given definitions that are related to permutations and substrings.

As we study signed permutations to model biological structures, we shall model the biological processes that modify these structures in terms of signed permutations as well.

Definition 2 (iTDR). *An inverse tandem duplication random loss $\rho: s\mathcal{P}_n \rightarrow s\mathcal{P}_n$ is an mapping that processes an input π by taking two subsequences L and R (of π) with $\mathcal{E}(L) \cap \mathcal{E}(R) = \emptyset$ and $\mathcal{E}(L) \cup \mathcal{E}(R) = \mathcal{E}(\pi)$ that are outputted as $\rho \circ \pi = L\bar{R}$ or as $\rho \circ \pi = \bar{L}R$. In the first case ρ is called right iTDR (riTDR) and in the second case it is called left iTDR (liTDR). We record, in slight abuse of notation, such mappings as $\rho = (r, \mathcal{E}(L), \mathcal{E}(R))$ for an riTDR and $\rho = (\ell, \mathcal{E}(L), \mathcal{E}(R))$ for an liTDR, and we define the set of all iTDRs for permutations of length $n \in \mathbb{N}$ as \mathcal{R}_n .*

From a biological point of view an iTDR can be seen as first applying a reversed tandem duplication to π , i. e., $\bar{\pi}$ is placed adjacently to the left (resp. right) of π resulting in a duplicated intermediate $\bar{\pi}\pi$ (resp. $\pi\bar{\pi}$), and to subsequently obtain a new permutation by random loss of one copy of every duplicated element. See Figure 2 for illustrations of this process. Note

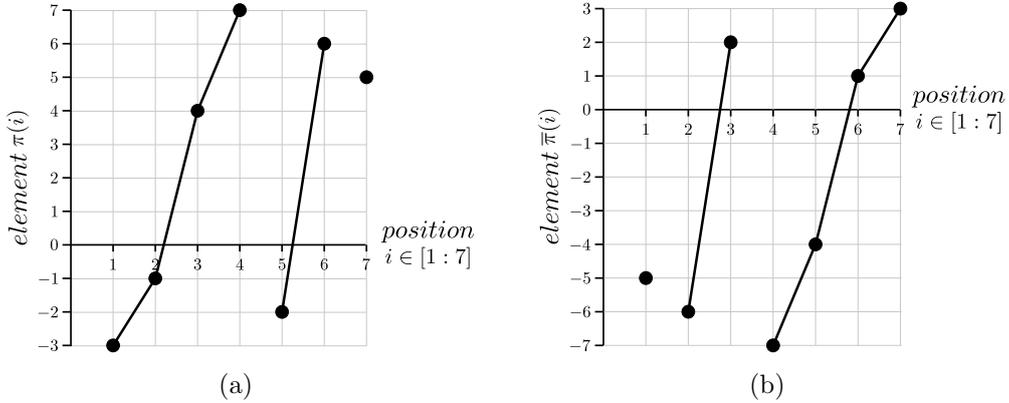


Figure 1: Illustration of $\pi = (-3 -1 4 7 -2 6 5)$ (a) and $\bar{\pi} = (-5 -6 2 -7 -4 1 3)$ (b). Every dot represents an element of the corresponding permutation. Maximal increasing substrings are illustrated by continuous lines. The maximal increasing substring decomposition of π is $\pi = \tau_1\tau_2\tau_3$ with $\tau_1 = -3 -1 4 7$, $\tau_2 = -2 6$, and $\tau_3 = 5$. Consequently, $\mathcal{S}_\pi = \{\tau_1, \tau_2, \tau_3\}$ and $|\mathcal{S}_\pi| = 3$. The substring $-1 4$ of π is increasing. While the sequence $-5 2 -4 3$ is a subsequence of $\bar{\pi}$, the sequences $S_1 = -6 2$ and $S_2 = -7 -4 1 3$ are substrings of $\bar{\pi}$. For S_1 and S_2 it holds that $S_1 \oplus S_2 = -7 -6 -4 1 2 3$. Intriguingly, $\bar{\pi}$ is point-symmetrically to π .

that for a given permutation an iTDRL can be represented by a signed permutation and therefore the composition of two functions f and g is denoted by $f \circ g$, i. e., $(f \circ g)(x) := f(g(x))$. It is also worth to mention that we have $\rho_r \circ \pi = \rho_\ell \circ \pi = \bar{\pi}$ for the iTDRLs $\rho_r = (r, \emptyset, \mathcal{E}(\pi))$ and $\rho_\ell = (\ell, \mathcal{E}(\pi), \emptyset)$.

3 Structural Characterization of Permutations Generated by Repeated Application of iTDRLs

In this section we characterize the structure of permutations that can be generated by sequentially applying k iTDRLs to the identity permutation ι with respect to the number of maximal increasing substrings. A lower bound on the (minimal) number of iTDRLs that are necessary to produce a permutation with a certain number of maximal increasing substrings is given in Subsection 3.1. A corresponding upper bound is given in Subsection 3.2. From the lower and upper bound we derive the main theorem in Subsection 3.3. The insights gained in this section are an important component to solve the sorting problem (and the distance problem) for iTDRLs in Section 4.

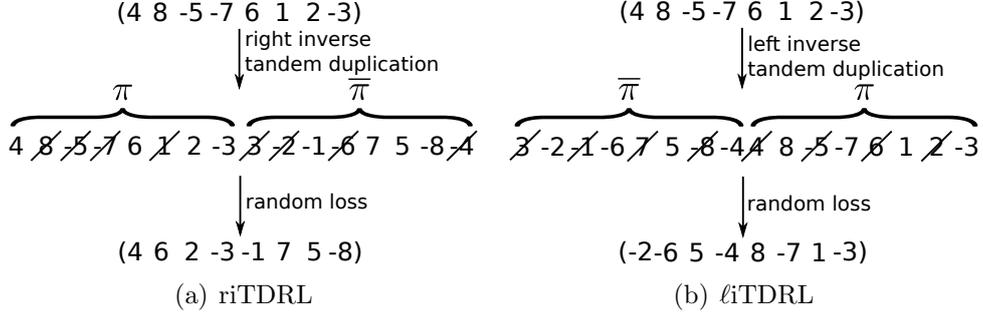


Figure 2: Application of the riTDRL $\rho = (r, \{-3, 2, 4, 6\}, \{-7, -5, 1, 8\})$ (a) and the liTDRL $\rho = (\ell, \{2, 4, 5, 6\}, \{-7, -3, 1, 8\})$ (b) to $\pi = (4\ 8\ -5\ -7\ 6\ 1\ 2\ -3)$ resulting into $(4\ 6\ 2\ -3\ -1\ 7\ 5\ -8)$ and $(-2\ -6\ 5\ -4\ 8\ -7\ 1\ -3)$, respectively.

3.1 The Lower Bound

This subsection provides a lower bound on the number of required iTDRs to generate a permutation from ι that has a certain number of maximal increasing substrings as formulated in the following proposition.

Proposition 1. *For a permutation $\pi \in s\mathcal{P}_n$ that has been obtained from $\iota \in s\mathcal{P}_n$ by the application of $k \in \mathbb{N}$ iTDRs it holds that either $|\mathcal{S}_\pi| \leq 2^{k-1}$ or $|\mathcal{S}_\pi| = 2^{k-1} + 1$ and $\ell_\pi < 0 < f_\pi$.*

For the proof of Proposition 1 we need the following four lemmas.

Lemma 1. *Let π be a signed permutation of length n . Then $S \in \mathcal{S}_\pi$ if and only if $\overline{S} \in \mathcal{S}_{\overline{\pi}}$.*

Proof. Let $\pi \in s\mathcal{P}_n$ and let $S = \pi(i) \dots \pi(j) \in \mathcal{S}_\pi$ with $1 \leq i \leq j \leq n$. The fact that $S \in \mathcal{S}_\pi$ implies that S is an increasing substring of π , i. e., it holds that $\pi(k) < \pi(k+1)$ for all $i \leq k < j$. Since S is maximal it cannot be extended to the left or the right, i. e., either $i = 1$ (resp. $j = n$) or if $i > 1$ (resp. $j < n$) then $\pi(i-1) > \pi(i)$ (resp. $\pi(j) > \pi(j+1)$). By the definition of $\overline{\pi}$ it holds that $\overline{S} = -\pi(j) \ -\pi(j-1) \ \dots \ -\pi(i)$ is a substring of $\overline{\pi}$. From $\pi(k) < \pi(k+1)$ for all $i \leq k < j$ it follows $-\pi(k+1) < -\pi(k)$, hence \overline{S} is an increasing substring of $\overline{\pi}$. If $i = 1$ and $j = n$ it directly follows that \overline{S} cannot be extended in the respective direction. Hence, consider that $i > 1$ or $j < n$. Consequently, $\pi(i-1) > \pi(i)$ or $\pi(j) > \pi(j+1)$ holds, which implies that $-\pi(i-1) < -\pi(i)$ or $-\pi(j) < -\pi(j+1)$, respectively. Thus, \overline{S} is a maximal increasing substring of $\overline{\pi}$. Consequently, $S \in \mathcal{S}_\pi$ implies $\overline{S} \in \mathcal{S}_{\overline{\pi}}$.

The other direction, i. e., if $\overline{S} \in \mathcal{S}_{\overline{\pi}}$ then $S \in \mathcal{S}_{\pi}$, follows from this implication and the fact that $\overline{\overline{\pi}} = \pi$ and $\overline{\overline{S}} = S$. \square

Lemma 1 shows that a substring S of a permutation π is maximal increasing if and only if its reversed substring \overline{S} is maximal increasing in $\overline{\pi}$. The following corollary is an immediate consequence of Lemma 1. It shows that the number of maximal increasing substrings of π and $\overline{\pi}$ coincide.

Corollary 1. *For a signed permutation π of length n and its reversed permutation $\overline{\pi}$ it holds that $|\mathcal{S}_{\pi}| = |\mathcal{S}_{\overline{\pi}}|$.*

Proof. By Lemma 1 it holds that $S \in \mathcal{S}_{\pi}$ if and only if $\overline{S} \in \mathcal{S}_{\overline{\pi}}$. Hence, $\mathcal{S}_{\pi} = \{S_1, \dots, S_{|\mathcal{S}_{\pi}|}\}$ holds if and only if $\mathcal{S}_{\overline{\pi}} = \{\overline{S_1}, \dots, \overline{S_{|\mathcal{S}_{\pi}|}}\}$ holds. Consequently, $|\mathcal{S}_{\pi}| = |\mathcal{S}_{\overline{\pi}}|$. \square

Consider a string of integers S and a substring S' of S . The following lemma proves that the number of maximal increasing substrings of S is always at least the number of maximal increasing substrings of S' .

Lemma 2. *Let S be a string of integers with the maximal increasing substring decomposition $S = S_1 \dots S_{|\mathcal{S}_S|}$ and let S' be a subsequence of S . The following equation holds:*

$$|\mathcal{S}_{S'}| \leq |\{i \in [1: |\mathcal{S}_S|] : \mathcal{E}(S') \cap \mathcal{E}(S_i) \neq \emptyset\}| \leq |\mathcal{S}_S|.$$

Proof. Let $S' = S'_1 \dots S'_{|\mathcal{S}_{S'}|}$ be the maximal increasing substring decomposition of S' . The fact that every maximal increasing substring of S' contains at least one element, and the fact that S' is a subsequence of S (i. e., $\mathcal{E}(S') \subset \mathcal{E}(S)$), ensures that for every maximal increasing substring S'_i of S' with $i \in [1: |\mathcal{S}_{S'}|]$ we have $1 = |\mathcal{S}_{S'_i}| \leq |\{j \in [1: |\mathcal{S}_S|] : \mathcal{E}(S'_i) \cap \mathcal{E}(S_j) \neq \emptyset\}|$. Consequently, $|\mathcal{S}_{S'}| = \sum_{S'_i \in \mathcal{S}_{S'}} |\mathcal{S}_{S'_i}| \leq \sum_{S'_i \in \mathcal{S}_{S'}} |\{j \in [1: |\mathcal{S}_S|] : \mathcal{E}(S'_i) \cap \mathcal{E}(S_j) \neq \emptyset\}| = |\{i \in [1: |\mathcal{S}_S|] : \mathcal{E}(S') \cap \mathcal{E}(S_i) \neq \emptyset\}|$ gives the left equation. The right equation follows by the fact that $|\mathcal{S}_S|$ is the maximum of the set $|\{i \in [1: |\mathcal{S}_S|] : \mathcal{E}(S') \cap \mathcal{E}(S_i) \neq \emptyset\}|$. \square

The following lemma shows that the sum of the number of maximal increasing substrings over a set of signed permutations is always at least the number of maximal increasing substrings of a concatenation of subsequences of the given set of permutations.

Lemma 3. *Let $\pi_1, \dots, \pi_k \in \mathcal{SP}_n$ and let S_i be a subsequence of π_i for $i \in [1: k]$ such that $\mathcal{E}(S_i) \cap \mathcal{E}(S_j) = \emptyset$ for all $1 \leq i < j \leq k$. The following equation*

holds:

$$|\mathcal{S}_{S_1 S_2 \dots S_k}| = \sum_{i=1}^k |\mathcal{S}_{S_i}| - K \leq \sum_{i=1}^k |\mathcal{S}_{S_i}| \leq \sum_{i=1}^k |\mathcal{S}_{\pi_i}|,$$

where $K = |\{j \in [1 : k-1] : \ell_{S_j} < f_{S_{j+1}}\}|$.

Proof. By Lemma 2 we have $|\mathcal{S}_{S_i}| \leq |\mathcal{S}_{\pi_i}|$ for all $i \in [1 : k]$ and, thus, the last two equations hold. With $\tau_1^i \dots \tau_{|\mathcal{S}_{S_i}|}^i$ we denote the maximal increasing substring decomposition of S_i for all $i \in [1 : k]$ and it holds that $\mathcal{S}_{S_i} = \{\tau_1^i, \dots, \tau_{|\mathcal{S}_{S_i}|}^i\}$. Now observe that all “internal” maximum increasing substrings of a S_i are present in $S_1 S_2 \dots S_k$ as well, i. e., for all S_i it holds that $\tau_k^i \in \mathcal{S}_{S_1 S_2 \dots S_k}$ for all $k \in [2 : |\mathcal{S}_{S_i}| - 1]$. Further, we have $\tau_1^1 \in \mathcal{S}_{S_1 S_2 \dots S_k}$ and $\tau_{|\mathcal{S}_{S_k}|}^k \in \mathcal{S}_{S_1 S_2 \dots S_k}$ since the first and the last maximum increasing substring cannot be extended to the left and the right, respectively. Since $\mathcal{E}(S_x) \cap \mathcal{E}(S_y) = \emptyset$ for all $x, y \in [1 : k]$ with $x \neq y$, we have either $\ell_{S_i} > f_{S_{i+1}}$ or $\ell_{S_i} < f_{S_{i+1}}$ for every $i \in [1 : k-1]$. If $\ell_{S_i} > f_{S_{i+1}}$ then $\tau_{|\mathcal{S}_{S_i}|}^i$ cannot be extended to the right and τ_1^{i+1} cannot be extended to the left. Hence, $\tau_{|\mathcal{S}_{S_i}|}^i$ and τ_1^{i+1} are counted separately in $\mathcal{S}_{S_1 S_2 \dots S_k}$ as they are in $\sum_{i=1}^k |\mathcal{S}_{S_i}| - K$. If $\ell_{S_i} < f_{S_{i+1}}$ then $\tau_{|\mathcal{S}_{S_i}|}^i \tau_1^{i+1}$ forms an increasing substring in $S_1 S_2 \dots S_k$. Consequently, while $\tau_{|\mathcal{S}_{S_i}|}^i$ and τ_1^{i+1} are both counted separately in $\sum_{i=1}^k |\mathcal{S}_{S_i}|$, only one string, i. e., $\tau_{|\mathcal{S}_{S_i}|}^i \tau_1^{i+1}$, is counted in $\mathcal{S}_{S_1 S_2 \dots S_k}$. Observe that this case is counted in K , which (in this case) reduces $\sum_{i=1}^k |\mathcal{S}_{S_i}|$ by one. Altogether, the first equation of the lemma follows. \square

The next and final lemma proves that the application of an iTDRL to a permutation π always results in a permutation that has less maximal increasing substrings than twice the number of maximal increasing substrings of π .

Lemma 4. *Let π be a signed permutation of length n with $\ell_\pi < 0 < f_\pi$. Then for every iTDRL $\rho \in \mathcal{R}_n$ holds $|\mathcal{S}_{\rho \circ \pi}| \leq 2|\mathcal{S}_\pi| - 1$. Further, if $|\mathcal{S}_{\rho \circ \pi}| = 2|\mathcal{S}_\pi| - 1$ then $\ell_{\rho \circ \pi} < 0 < f_{\rho \circ \pi}$.*

Proof. Let $\pi \in s\mathcal{P}_n$ with $\ell_\pi < 0 < f_\pi$ and let $\rho \in \mathcal{R}_n$. Then $\rho \circ \pi$ can be written as $\rho \circ \pi = \tau \tau'$ (resp. $\rho \circ \pi = \tau' \tau$), where τ is a subsequence of π and τ' is a subsequence of $\bar{\pi}$, if ρ is a riTDRL (resp. liTDRL). Corollary 1 implies that $|\mathcal{S}_\pi| = |\mathcal{S}_{\bar{\pi}}|$ and by Lemma 2 it holds that $|\mathcal{S}_\tau| \leq |\mathcal{S}_\pi|$ and $|\mathcal{S}_{\tau'}| \leq |\mathcal{S}_{\bar{\pi}}|$. Then, Lemma 3 implies $|\mathcal{S}_{\tau \tau'}| = |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| - K_1 \leq |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| \leq 2|\mathcal{S}_\pi|$ and $|\mathcal{S}_{\tau' \tau}| = |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| - K_2 \leq |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| \leq 2|\mathcal{S}_\pi|$, where $K_1 = 1$ (resp. $K_2 = 1$) if $\ell_\tau < 0 < f_{\tau'}$ (resp. $\ell_{\tau'} < 0 < f_\tau$) and $K_1 = 0$ (resp. $K_2 = 0$) otherwise.

Hence, if $|\mathcal{S}_\tau| \leq |\mathcal{S}_\pi| - 1$ and $|\mathcal{S}_{\tau'}| \leq |\mathcal{S}_\pi| - 1$, then $|\mathcal{S}_{\tau\tau'}| \leq 2|\mathcal{S}_\pi| - 2 < 2|\mathcal{S}_\pi| - 1$ and $|\mathcal{S}_{\tau'\tau}| \leq 2|\mathcal{S}_\pi| - 2 < 2|\mathcal{S}_\pi| - 1$. Consequently, it remains to consider the cases where $|\mathcal{S}_\tau|, |\mathcal{S}_{\tau'}| \geq |\mathcal{S}_\pi| - 1$ and at least one of $|\mathcal{S}_\tau|$ or $|\mathcal{S}_{\tau'}|$ is $|\mathcal{S}_\pi|$. More precisely it remains to consider the following cases: i) $|\mathcal{S}_\tau| = |\mathcal{S}_{\tau'}| = |\mathcal{S}_\pi|$, ii) $|\mathcal{S}_\tau| = |\mathcal{S}_\pi| - 1$ and $|\mathcal{S}_{\tau'}| = |\mathcal{S}_\pi|$, or iii) $|\mathcal{S}_\tau| = |\mathcal{S}_\pi|$ and $|\mathcal{S}_{\tau'}| = |\mathcal{S}_\pi| - 1$. Let $\pi = \pi_1 \dots \pi_{|\mathcal{S}_\pi|}$ be the maximal increasing substring decomposition of π . Then $0 < f_\pi$ (resp. $\ell_\pi < 0$) implies that π_1 (resp. $\pi_{|\mathcal{S}_\pi|}$) contains only positive (resp. negative) elements. Hence, Lemma 1 implies that $\bar{\pi}_1$ (resp. $\bar{\pi}_{|\mathcal{S}_\pi|}$) contains only positive (resp. negative) elements, where $\bar{\pi} = \bar{\pi}_1 \dots \bar{\pi}_{|\mathcal{S}_\pi|}$ is the maximal increasing substring decomposition of $\bar{\pi}$. In the following we prove the statement in the cases (i) and (ii). The proof for Case (iii) is similar to Case (ii).

Case (i): By Lemma 2 it holds that $\mathcal{E}(\tau) \cap \mathcal{E}(\pi_i) \neq \emptyset$ and $\mathcal{E}(\tau') \cap \mathcal{E}(\bar{\pi}_i) \neq \emptyset$ for all $i \in [1:|\mathcal{S}_\pi|]$. Hence, since $\mathcal{E}(\tau) \cap \mathcal{E}(\pi_1) \neq \emptyset$ (resp. $\mathcal{E}(\tau) \cap \mathcal{E}(\pi_{|\mathcal{S}_\pi|}) \neq \emptyset$) it holds that $f_\tau > 0$ (resp. $\ell_\tau < 0$). Analogously, $\mathcal{E}(\tau') \cap \mathcal{E}(\bar{\pi}_1) \neq \emptyset$ (resp. $\mathcal{E}(\tau') \cap \mathcal{E}(\bar{\pi}_{|\mathcal{S}_\pi|}) \neq \emptyset$) implies that $f_{\tau'} > 0$ (resp. $\ell_{\tau'} < 0$). Hence, $\ell_\tau < 0 < f_{\tau'}$ (resp. $\ell_{\tau'} < 0 < f_\tau$) if ρ is an riTDRL (resp. liTDRL). Consequently, by Lemma 3 it holds that $|\mathcal{S}_{\tau\tau'}| = |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| - K_1 = |\mathcal{S}_\pi| + |\mathcal{S}_\pi| - 1 = 2|\mathcal{S}_\pi| - 1$ and $|\mathcal{S}_{\tau'\tau}| = |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| - K_2 = |\mathcal{S}_\pi| + |\mathcal{S}_\pi| - 1 = 2|\mathcal{S}_\pi| - 1$. Hence, $|\mathcal{S}_{\rho\circ\pi}| = 2|\mathcal{S}_\pi| - 1$ and $\ell_{\rho\circ\pi} < 0 < f_{\rho\circ\pi}$.

Case (ii): By Lemma 2 it holds that $\mathcal{E}(\tau') \cap \mathcal{E}(\bar{\pi}_i) \neq \emptyset$ for all $i \in [1:|\mathcal{S}_\pi|]$. Hence, since $\mathcal{E}(\tau') \cap \mathcal{E}(\bar{\pi}_1) \neq \emptyset$ (resp. $\mathcal{E}(\tau') \cap \mathcal{E}(\bar{\pi}_{|\mathcal{S}_\pi|}) \neq \emptyset$) it holds that $f_{\tau'} > 0$ (resp. $\ell_{\tau'} < 0$). By Lemma 2 holds that there exist an $i \in [1:|\mathcal{S}_\pi|]$ with $\mathcal{E}(\tau) \cap \mathcal{E}(\pi_i) = \emptyset$ and for all $j \in [1:|\mathcal{S}_\pi|] \setminus \{i\}$ we have $\mathcal{E}(\tau) \cap \mathcal{E}(\pi_j) \neq \emptyset$.

- Consider first that $i = 1$, then $\mathcal{E}(\pi_{|\mathcal{S}_\pi|}) \cap \mathcal{E}(\tau) \neq \emptyset$, hence $\ell_\tau < 0$. Consequently, $\ell_\tau < 0 < f_{\tau'}$ and we have $|\mathcal{S}_{\tau\tau'}| = |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| - K_1 = |\mathcal{S}_\pi| - 1 + |\mathcal{S}_\pi| - 1 = 2|\mathcal{S}_\pi| - 2 < 2|\mathcal{S}_\pi| - 1$ by Lemma 3. Analogously, Lemma 3 yields $|\mathcal{S}_{\tau'\tau}| = |\mathcal{S}_{\tau'}| + |\mathcal{S}_\tau| - K_2 \leq |\mathcal{S}_\pi| - 1 + |\mathcal{S}_\pi| = 2|\mathcal{S}_\pi| - 1$ with $\ell_{\rho\circ\pi} = \ell_{\tau'\tau} = \ell_\tau < 0 < f_{\tau'} = f_{\tau'\tau} = f_{\rho\circ\pi}$.
- Consider now that $i = |\mathcal{S}_\pi|$. Then, by Lemma 2 $\mathcal{E}(\pi_1) \cap \mathcal{E}(\tau) \neq \emptyset$, hence $f_\tau > 0$. Consequently, $\ell_{\tau'} < 0 < f_\tau$ and by applying Lemma 3 again we get $|\mathcal{S}_{\tau'\tau}| = |\mathcal{S}_{\tau'}| + |\mathcal{S}_\tau| - K_2 = |\mathcal{S}_\pi| + |\mathcal{S}_\pi| - 1 - 1 = 2|\mathcal{S}_\pi| - 2 < 2|\mathcal{S}_\pi| - 1$. Analogously, Lemma 3 also provides $|\mathcal{S}_{\tau\tau'}| = |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| - K_1 \leq |\mathcal{S}_\pi| - 1 + |\mathcal{S}_\pi| = 2|\mathcal{S}_\pi| - 1$ with $\ell_{\rho\circ\pi} = \ell_{\tau\tau'} = \ell_{\tau'} < 0 < f_\tau = f_{\tau\tau'} = f_{\rho\circ\pi}$.
- Finally, consider $i \in [2:|\mathcal{S}_\pi| - 1]$. Then, by Lemma 2 $\mathcal{E}(\pi_1) \cap \mathcal{E}(\tau) \neq \emptyset$ and $\mathcal{E}(\pi_{|\mathcal{S}_\pi|}) \cap \mathcal{E}(\tau) \neq \emptyset$, hence $f_\tau > 0$ and $\ell_\tau < 0$ holds. Thus,

$\ell_\tau < 0 < f_{\tau'}$ and $\ell_{\tau'} < 0 < f_\tau$ is implied. As before, using Lemma 3 yields $|\mathcal{S}_{\tau\tau'}| = |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| - K_1 = |\mathcal{S}_\pi| - 1 + |\mathcal{S}_\pi| - 1 = 2|\mathcal{S}_\pi| - 2 < 2|\mathcal{S}_\pi| - 1$ and $|\mathcal{S}_{\tau'\tau}| = |\mathcal{S}_{\tau'}| + |\mathcal{S}_\tau| - K_2 = |\mathcal{S}_\pi| + |\mathcal{S}_\pi| - 1 - 1 = 2|\mathcal{S}_\pi| - 2 < 2|\mathcal{S}_\pi| - 1$.

Altogether, either $|\mathcal{S}_{\rho\circ\pi}| < 2|\mathcal{S}_\pi| - 1$ or $|\mathcal{S}_{\rho\circ\pi}| = 2|\mathcal{S}_\pi| - 1$ and $\ell_{\rho\circ\pi} < 0 < f_{\rho\circ\pi}$ holds, which proves the statement. \square

Given Lemma 1 to Lemma 4, we are now ready to prove the lower bound that is stated in Proposition 1.

Proof of Proposition 1. We prove the proposition by induction on k . First consider the case $k = 1$. The application of a single iTDRL to ι yields a permutation with at most $2^{1-1} + 1 = 2$ maximal increasing substrings. This can be seen by the following argumentation that considers π' to be obtained by applying a single iTDRL $\rho \in \mathcal{R}_n$ (i. e., ρ is a riTDRL or a liTDRL) to ι . By the definition of an iTDRL, π' can be written as $\pi' = \tau\tau'$ (resp. $\pi' = \tau'\tau$), where τ is a subsequence of ι and τ' is a subsequence of $\bar{\iota}$, in the case that ρ is a riTDRL (resp. liTDRL). Corollary 1 implies $|\mathcal{S}_\iota| = |\mathcal{S}_{\bar{\iota}}|$ and by Lemma 2 it holds that $|\mathcal{S}_\tau| \leq |\mathcal{S}_\iota|$ and $|\mathcal{S}_{\tau'}| \leq |\mathcal{S}_{\bar{\iota}}|$. Certainly, $|\mathcal{S}_\iota| = 1$. Thus, by Lemma 3 $|\mathcal{S}_{\pi'}| = |\mathcal{S}_{\tau\tau'}| \leq |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| \leq 2$ (resp. $|\mathcal{S}_{\pi'}| = |\mathcal{S}_{\tau'\tau}| \leq |\mathcal{S}_{\tau'}| + |\mathcal{S}_\tau| \leq 2$) if ρ is a riTDRL (resp. liTDRL). Consequently, if one of τ or τ' is empty, then $|\mathcal{S}_{\pi'}| \leq 1$. If τ and τ' are not empty, then $|\mathcal{S}_\tau| = 1$ and $|\mathcal{S}_{\tau'}| = 1$, and since ι contains only positive elements all elements of τ (resp. τ') are positive (resp. negative). Thus, $\ell_{\tau'}, f_{\tau'} < 0$ and $\ell_\tau, f_\tau > 0$. Since $\ell_{\tau'} < f_\tau$ it follows by Lemma 3 that $|\mathcal{S}_{\pi'}| = |\mathcal{S}_{\tau'\tau}| = |\mathcal{S}_{\tau'}| + |\mathcal{S}_\tau| - 1 \leq 2 - 1 = 1$. Additionally, $\ell_{\pi'} < 0 < f_{\pi'}$ holds for the case that ρ is an riTDRL. Altogether, the statement holds for $k = 1$.

For the induction step, assume that σ is a permutation that has been obtained from ι by the application of $k-1$ iTDRLs and let π be a permutation obtained from σ by the application of a single iTDRL ρ . We can write $\pi = \tau\tau'$ (resp. $\pi = \tau'\tau$), where τ is a subsequence of σ and τ' is a subsequence of $\bar{\sigma}$, when ρ is an riTDRL (resp. liTDRL). Corollary 1 implies that $|\mathcal{S}_\sigma| = |\mathcal{S}_{\bar{\sigma}}|$ and by Lemma 2 holds that $|\mathcal{S}_\tau| \leq |\mathcal{S}_\sigma|$ and $|\mathcal{S}_{\tau'}| \leq |\mathcal{S}_{\bar{\sigma}}|$. Then, Lemma 3 implies $|\mathcal{S}_{\tau\tau'}| = |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| - K_1 \leq |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| \leq 2|\mathcal{S}_\sigma|$ and $|\mathcal{S}_{\tau'\tau}| = |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| - K_2 \leq |\mathcal{S}_\tau| + |\mathcal{S}_{\tau'}| \leq 2|\mathcal{S}_\sigma|$, where $K_1 = 1$ (resp. $K_2 = 1$) if $\ell_\tau < 0 < f_{\tau'}$ (resp. $\ell_{\tau'} < 0 < f_\tau$) and $K_1 = 0$ (resp. $K_2 = 0$) otherwise. By the induction hypothesis $|\mathcal{S}_\sigma| \leq 2^{k-2} + 1$ and if $|\mathcal{S}_\sigma| = 2^{k-2} + 1$ then $\ell_\sigma < 0 < f_\sigma$. Therefore, $|\mathcal{S}_\tau| \leq 2^{k-2} + 1$ and $|\mathcal{S}_{\tau'}| \leq 2^{k-2} + 1$. Hence, if $|\mathcal{S}_\tau| \leq 2^{k-2}$ and $|\mathcal{S}_{\tau'}| \leq 2^{k-2}$, then $|\mathcal{S}_{\tau\tau'}| \leq 2^{k-1}$ and $|\mathcal{S}_{\tau'\tau}| \leq 2^{k-1}$. Consequently, it remains to consider the cases where $|\mathcal{S}_\tau|, |\mathcal{S}_{\tau'}| \geq 2^{k-2}$ and at least one of $|\mathcal{S}_\tau|$ or $|\mathcal{S}_{\tau'}|$ is $2^{k-2} + 1$. Note that this implies that $|\mathcal{S}_\sigma| = 2^{k-2} + 1$ and (by the

induction hypothesis) it holds that $\ell_\sigma < 0 < f_\sigma$. Lemma 4 now implies that $|\mathcal{S}_\pi| \leq 2|\mathcal{S}_\sigma| - 1 = 2(2^{k-2} + 1) - 1 = 2^{k-1} + 1$. Further, Lemma 4 implies that $|\mathcal{S}_\pi| = 2^{k-1} + 1$ implies $\ell_\pi < 0 < f_\pi$.

Altogether, we have either $|\mathcal{S}_\pi| < 2^{k-1} + 1$, or $|\mathcal{S}_\pi| = 2^{k-1} + 1$ and $\ell_\pi < 0 < f_\pi$, which proves the statement. \square

3.2 The Upper Bound

In this subsection we show an upper bound on the minimum number of iTDRLs that have to be applied to ι in order to produce a permutation with a certain number of maximal increasing substrings as formulated in the following proposition.

Proposition 2. *Let $\pi \in s\mathcal{P}_n \setminus \{\iota\}$ such that $|\mathcal{S}_\pi| = 2^{k-1} + 1$ and $\ell_\pi < 0 < f_\pi$ or $|\mathcal{S}_\pi| \leq 2^{k-1}$ for a $k \in \mathbb{N}$. Then π can be obtained by applying k iTDRLs to ι .*

The main idea to prove Proposition 2 is to perform an induction over k and then, given a permutation π , apply a “reversed” iTDRL on π that divides the number of maximal increasing substrings of π by two. Essentially, this reversed operation “sorts” the second (first) half of π which was created by an riTDRL (resp. an liTDRL) back into the first (resp. second) half.

In the following we define two transformations $T_i: s\mathcal{P}_n \rightarrow s\mathcal{P}_n$, $i \in [1:2]$, to construct a permutation $T_i(\pi)$ from π which has the property that there always exist an iTDRL ρ such that $\rho \circ T_i(\pi) = \pi$. With other words, these transformations are inverted iTDRL operations (which is shown in Lemma 5). Depending on whether the transformation is the inverse of an riTDRL or an liTDRL we define a different transformation T_1 or T_2 (and in each case we also distinguish if the number of maximal increasing substrings of π is even or odd). For both constructions consider a signed permutation π of length n with the maximal substring decomposition $\pi = \pi_1 \dots \pi_{|\mathcal{S}_\pi|}$.

- 1) If $|\mathcal{S}_\pi|$ is even, then $T_1(\pi) := \tau_1 \tau_2 \dots \tau_{|\mathcal{S}_\pi|/2-1} \tau_{|\mathcal{S}_\pi|/2}$, where $\tau_1 = \pi_1 \oplus \overline{\pi_{|\mathcal{S}_\pi|}}$, $\tau_2 = \pi_2 \oplus \overline{\pi_{|\mathcal{S}_\pi|-1}}$, \dots , $\tau_{|\mathcal{S}_\pi|/2-1} = \pi_{|\mathcal{S}_\pi|/2-1} \oplus \overline{\pi_{|\mathcal{S}_\pi|/2+2}}$, and $\tau_{|\mathcal{S}_\pi|/2} = \pi_{|\mathcal{S}_\pi|/2} \oplus \overline{\pi_{|\mathcal{S}_\pi|/2+1}}$. If $|\mathcal{S}_\pi|$ is odd, then $T_1(\pi) := \tau_1 \tau_2 \dots \tau_{\lfloor |\mathcal{S}_\pi|/2 \rfloor} \tau_{\lceil |\mathcal{S}_\pi|/2 \rceil}$, where $\tau_1 = \pi_1 \oplus \overline{\pi_{|\mathcal{S}_\pi|}}$, $\tau_2 = \pi_2 \oplus \overline{\pi_{|\mathcal{S}_\pi|-1}}$, \dots , $\tau_{\lfloor |\mathcal{S}_\pi|/2 \rfloor} = \pi_{\lfloor |\mathcal{S}_\pi|/2 \rfloor} \oplus \overline{\pi_{\lfloor |\mathcal{S}_\pi|/2 \rfloor + 1}}$, and $\tau_{\lceil |\mathcal{S}_\pi|/2 \rceil} = \nu_{\lceil |\mathcal{S}_\pi|/2 \rceil} \oplus \overline{\kappa_{\lceil |\mathcal{S}_\pi|/2 \rceil}}$ with $\nu_{\lceil |\mathcal{S}_\pi|/2 \rceil}$ (resp. $\kappa_{\lceil |\mathcal{S}_\pi|/2 \rceil}$) being the smallest substring of $\pi_{\lceil |\mathcal{S}_\pi|/2 \rceil}$ that contains all its negative (resp. positive) elements.
- 2) If $|\mathcal{S}_\pi|$ is even, then $T_2(\pi) := \tau_1 \tau_2 \dots \tau_{|\mathcal{S}_\pi|/2-1} \tau_{|\mathcal{S}_\pi|/2}$, where $\tau_1 = \pi_{|\mathcal{S}_\pi|/2+1} \oplus \overline{\pi_{|\mathcal{S}_\pi|/2}}$, $\tau_2 = \pi_{|\mathcal{S}_\pi|/2+2} \oplus \overline{\pi_{|\mathcal{S}_\pi|/2-1}}$, \dots , $\tau_{|\mathcal{S}_\pi|/2-1} = \pi_{|\mathcal{S}_\pi|-1} \oplus \overline{\pi_2}$, and

$\tau_{|\mathcal{S}_\pi|/2} = \pi_{|\mathcal{S}_\pi|} \oplus \overline{\pi_1}$. If $|\mathcal{S}_\pi|$ is odd, then $T_2(\pi) := \tau_1 \tau_2 \dots \tau_{\lfloor |\mathcal{S}_\pi|/2 \rfloor} \tau_{\lceil |\mathcal{S}_\pi|/2 \rceil}$, where $\tau_1 = \kappa_{\lfloor |\mathcal{S}_\pi|/2 \rfloor} \oplus \overline{v_{\lfloor |\mathcal{S}_\pi|/2 \rfloor}}$, $\tau_2 = \pi_{\lfloor |\mathcal{S}_\pi|/2 \rfloor + 1} \oplus \overline{\pi_{\lfloor |\mathcal{S}_\pi|/2 \rfloor}}$, \dots , $\tau_{\lfloor |\mathcal{S}_\pi|/2 \rfloor} = \pi_{|\mathcal{S}_\pi| - 1} \oplus \overline{\pi_2}$, and $\tau_{\lceil |\mathcal{S}_\pi|/2 \rceil} = \pi_{|\mathcal{S}_\pi|} \oplus \overline{\pi_1}$ with $\kappa_{\lfloor |\mathcal{S}_\pi|/2 \rfloor}$ (resp. $v_{\lfloor |\mathcal{S}_\pi|/2 \rfloor}$) being the smallest substring of $\pi_{\lfloor |\mathcal{S}_\pi|/2 \rfloor}$ that contains all its positive (resp. negative) elements.

See Figure 3 for examples of both transformations. We need the following four lemmas in order to prove Proposition 2.

The first lemma shows that the transformation T_1 (resp. T_2) is an inverted riTDRL (resp. liTDRL) operation.

Lemma 5. *For every $\pi \in s\mathcal{P}_n$ the following statements are true:*

- 1) *There exist an riTDRL $\rho \in \mathcal{R}_n$ such that $\rho \circ T_1(\pi) = \pi$.*
- 2) *There exist an liTDRL $\rho \in \mathcal{R}_n$ such that $\rho \circ T_2(\pi) = \pi$.*

Proof. Let $\pi \in s\mathcal{P}_n$ and let $\pi = \pi_1 \pi_2 \dots \pi_{|\mathcal{S}_\pi|}$ be the maximal increasing substring decomposition of π . For the sake of a clear argument, we will only consider the case that $|\mathcal{S}_\pi|$ is even. The proof for the odd case is fully analogues. Then $T_1(\pi) = \tau_1 \tau_2 \dots \tau_{|\mathcal{S}_\pi|/2 - 1} \tau_{|\mathcal{S}_\pi|/2}$. We therefore have that $S_1 = \pi_1 \pi_2 \dots \pi_{|\mathcal{S}_\pi|/2 - 1} \pi_{|\mathcal{S}_\pi|/2}$ and $S'_1 = \overline{\pi_{|\mathcal{S}_\pi|}} \overline{\pi_{|\mathcal{S}_\pi| - 1}} \dots \overline{\pi_{|\mathcal{S}_\pi|/2 + 2}} \overline{\pi_{|\mathcal{S}_\pi|/2 + 1}}$ are disjoint subsequences of $T_1(\pi)$.

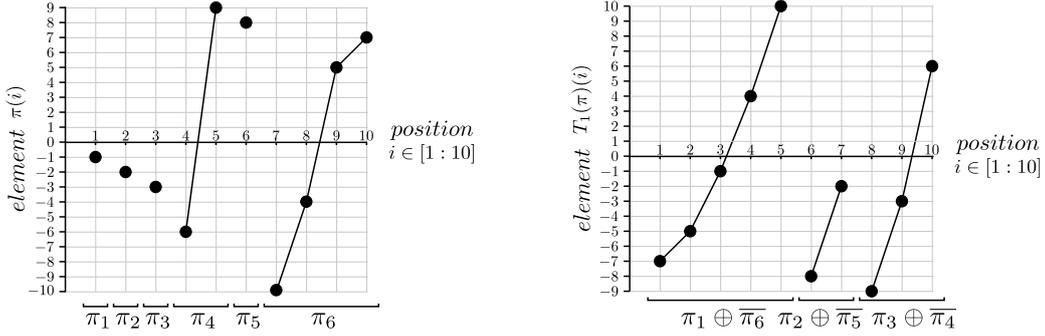
Furthermore, we have $T_2(\pi) = \tau_1 \tau_2 \dots \tau_{|\mathcal{S}_\pi|/2 - 1} \tau_{|\mathcal{S}_\pi|/2}$. Now consider the sequence $S_2 = \overline{\pi_{|\mathcal{S}_\pi|/2}} \overline{\pi_{|\mathcal{S}_\pi|/2 - 1}} \dots \overline{\pi_2} \overline{\pi_1}$ and $S'_2 = \pi_{|\mathcal{S}_\pi|/2 + 1} \pi_{|\mathcal{S}_\pi|/2 + 2} \dots \pi_{|\mathcal{S}_\pi| - 1} \pi_{|\mathcal{S}_\pi|}$ and observe that they are disjoint subsequences of $T_2(\pi)$.

Note that $\pi = S_i \overline{S'_i}$ holds (in the respective case $i \in [1 : 2]$) and $S_i, \overline{S'_i}$ are disjoint subsequences of $T_i(\pi)$ that together include all elements of $T_i(\pi)$. Hence, for the riTDRL (resp. liTDRL) $\rho = (r, \mathcal{E}(S_1), \mathcal{E}(S'_1))$ (resp. $\rho = (\ell, \mathcal{E}(S_1), \mathcal{E}(S'_1))$) it holds that $\rho \circ T_1(\pi) = \pi$ (resp. $\rho \circ T_2(\pi) = \pi$). \square

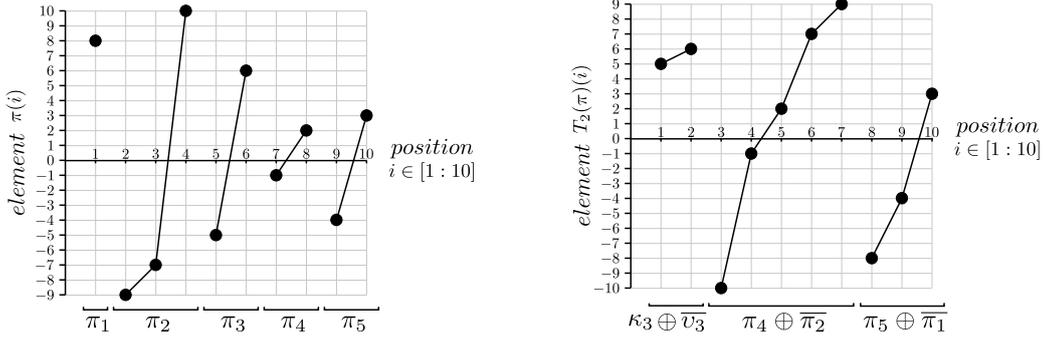
The transformations $T_1(\pi)$ and $T_2(\pi)$ have been designed such that the following lemma holds.

Lemma 6. *Let π be a signed permutation of length n . For each of the permutations $T_i(\pi)$, $i \in [1 : 2]$, the given respective decomposition into strings $\tau_1 \tau_2 \dots \tau_t$ (where t is as in the respective case) is a maximal increasing substring decomposition.*

Proof. Let $\pi \in s\mathcal{P}_n$ and let $\pi = \pi_1 \dots \pi_{|\mathcal{S}_\pi|}$ be the maximal increasing substring decomposition of π . For the proof it is enough to show that for each $j \in [1 : t - 1]$ it holds that the last element of τ_j is larger than the first element of τ_{j+1} , i. e., $\ell_{\tau_j} > f_{\tau_{j+1}}$. In the following we describe the proof for the case that $i = 1$ (in the case that $i = 2$ the proof can be done analogously).



(a) Case $T_1(\pi)$



(b) Case $T_2(\pi)$

Figure 3: Examples of the transformation T_1 (a) and T_2 (b) that is applied to $\pi = (-1 -2 -3 -6 9 8 -10 -4 5 7)$ (a) and $\pi = (8 -9 -7 10 -5 6 -1 2 -4 3)$ (b). The transformation $T_i(\pi)$ with $i \in [1:2]$ is shown on the right in the respective subfigure. The notation is as in Figure 1. In addition, κ_3 (resp. v_3) is the smallest substring of π_3 that contains all its positive (resp. negative) elements, i. e., $\kappa_3 = 6$ and $v_3 = -5$. For every permutation that is illustrated, its maximal increasing substring decomposition is shown on the bottom of a permutation.

In this case the lemma has to be shown for $T_1(\pi) := \tau_1\tau_2\dots\tau_{t-1}\tau_t$, where $t = |\mathcal{S}_\pi|/2$ (resp. $t = \lceil |\mathcal{S}_\pi|/2 \rceil$) in the case that $|\mathcal{S}_\pi|$ is even (resp. odd). Since all elements of two sequences X and Y are sorted increasingly in $X \oplus Y$ it follows that every τ_j with $j \in [1:t]$ is an increasing substring. By the construction of $T_1(\pi)$ it holds that $\tau_j = \pi_j \oplus \overline{\pi_{t+1-j}}$ for $j \in [1:t]$. Hence, a τ_j always contains all elements of π_j . Consequently, $\ell_{\tau_j} \geq \ell_{\pi_j}$ and $f_{\tau_j} \leq f_{\pi_j}$ holds for all $j \in [1:t]$. Now the fact that π_j and π_{j+1} are two maximal increasing substrings, i. e., $\ell_{\pi_j} > f_{\pi_{j+1}}$ for $j \in [1:|\mathcal{S}_\pi| - 1]$, implies $\ell_{\tau_j} \geq \ell_{\pi_j} > f_{\pi_{j+1}} \geq f_{\tau_{j+1}}$ for all $j \in [1:t - 1]$. Therefore, τ_1, \dots, τ_t are maximal, which proves the statement. \square

The following lemma shows that the application of the transformation T_1 (resp. T_2) to a permutation π results in a permutation that has half as many maximal increasing substrings as π .

Lemma 7. *Let $\pi \in s\mathcal{P}_n$ with $|\mathcal{S}_\pi| > 1$. Then $|\mathcal{S}_{T_i(\pi)}| = \lceil |\mathcal{S}_\pi|/2 \rceil$ holds for all $i \in [1:2]$.*

Proof. Let $\pi \in s\mathcal{P}_n$ with $|\mathcal{S}_\pi| > 1$. Consider the case that $|\mathcal{S}_\pi|$ is even. By the construction of $T_i(\pi) = \tau_1 \dots \tau_{|\mathcal{S}_{T_i(\pi)}|}$ with $i = 1, 2$ it holds that two maximal increasing substrings of π always form a new increasing substring in $T_i(\pi)$, hence $|\mathcal{S}_{T_i(\pi)}| \leq |\mathcal{S}_\pi|/2$. By Lemma 6 it holds that every τ_i of $T_i(\pi)$ is also maximal, and hence $|\mathcal{S}_{T_i(\pi)}| \geq |\mathcal{S}_\pi|/2$. Altogether, $|\mathcal{S}_{T_i(\pi)}| = |\mathcal{S}_\pi|/2$ if $|\mathcal{S}_\pi|$ is even.

Now consider that $|\mathcal{S}_\pi|$ is odd. By the construction of $T_i(\pi) = \tau_1 \dots \tau_{|\mathcal{S}_{T_i(\pi)}|}$ with $i = 1, 2$ it holds that $\tau_1, \dots, \tau_{|\mathcal{S}_{T_i(\pi)}|-1}$ (resp. $\tau_2, \dots, \tau_{|\mathcal{S}_{T_i(\pi)}|}$) of $T_i(\pi)$ are always formed by two maximal increasing substrings of π and $\tau_{|\mathcal{S}_{T_i(\pi)}|}$ (resp. τ_1) is formed by one maximal increasing substring of π if $i = 1$ (resp. $i = 2$). Hence, $|\mathcal{S}_{T_i(\pi)}| \leq \lceil |\mathcal{S}_\pi|/2 \rceil$. By Lemma 6 it holds that every τ_i of $T_i(\pi)$ is also maximal, hence $|\mathcal{S}_{T_i(\pi)}| \geq \lceil |\mathcal{S}_\pi|/2 \rceil$. Altogether, $|\mathcal{S}_{T_i(\pi)}| = \lceil |\mathcal{S}_\pi|/2 \rceil$ if $|\mathcal{S}_\pi|$ is odd. \square

Consider a signed permutation π with at least two maximal increasing substrings such that the first (last) element of π is positive (resp. negative). The following lemma proves that the application of the transformation T_i , $i \in [1:2]$, to π preserve this structure, i. e., the first (last) element of $T_i(\pi)$ is positive (resp. negative).

Lemma 8. *Let π be a signed permutation of length n with $|\mathcal{S}_\pi| > 1$, $|\mathcal{S}_\pi|$ odd, and $\ell_\pi < 0 < f_\pi$. Then it holds that $\ell_{T_i(\pi)} < 0 < f_{T_i(\pi)}$ for $i = 1, 2$.*

Proof. Let $\pi \in s\mathcal{P}_n$ and let $\pi = \pi_1 \dots \pi_{|\mathcal{S}_\pi|}$ be the maximal increasing substring decomposition of π with $|\mathcal{S}_\pi| > 1$ and $|\mathcal{S}_\pi|$ is odd. The fact that $0 < f_\pi$

(resp. $\ell_\pi < 0$) implies that π_1 (resp. $\pi_{|\mathcal{S}_\pi|}$) contains only positive (resp. negative) elements. Consequently, $\overline{\pi_1}$ (resp. $\overline{\pi_{|\mathcal{S}_\pi|}}$) contains only negative (resp. positive) elements. Therefore, $\pi_1 \oplus \overline{\pi_{|\mathcal{S}_\pi|}}$ (that is the leftmost maximal increasing substring in $T_1(\pi)$) contains only positive elements. Analogously, $\pi_{|\mathcal{S}_\pi|} \oplus \overline{\pi_1}$ (the rightmost maximal increasing substring in $T_2(\pi)$) contains only negative elements and, thus, we have $0 < f_{T_1(\pi)}$ and $\ell_{T_2(\pi)} < 0$. By definition we have that the rightmost (resp. leftmost) maximal increasing substring of $T_1(\pi)$ (resp. $T_2(\pi)$) is $v_{[\mathcal{S}_\pi]} \oplus \overline{\kappa_{[\mathcal{S}_\pi]}}$ (resp. $\kappa_{[\mathcal{S}_\pi]} \oplus \overline{v_{[\mathcal{S}_\pi]}}$). Since $\kappa_{[\mathcal{S}_\pi]}$ (resp. $v_{[\mathcal{S}_\pi]}$) contains only positive (resp. negative) elements, it holds that $\overline{\kappa_{[\mathcal{S}_\pi]}}$ (resp. $\overline{v_{[\mathcal{S}_\pi]}}$) contains only negative (resp. positive) elements. Therefore, $v_{[\mathcal{S}_\pi]} \oplus \overline{\kappa_{[\mathcal{S}_\pi]}}$ (resp. $\kappa_{[\mathcal{S}_\pi]} \oplus \overline{v_{[\mathcal{S}_\pi]}}$) contains only negative (resp. positive) elements, thus $\ell_{T_1(\pi)} < 0$ and $0 < f_{T_2(\pi)}$. \square

With Lemma 5 to Lemma 8 at hand, we are finally ready to proof the claim of Proposition 2.

Proof of Proposition 2. Let $\pi \in s\mathcal{P}_n \setminus \{\iota\}$. We prove the statement by induction on k . For the base case assume $k = 1$. There exists the two cases: i) $|\mathcal{S}_\pi| \leq 2^{1-1} = 2^0 = 1$ or ii) $|\mathcal{S}_\pi| = 2^{1-1} + 1 = 2^0 + 1 = 2$ and $\ell_\pi < 0 < f_\pi$. Consider Case (i). Since every permutation has at least one maximal increasing substring we have $|\mathcal{S}_\pi| = 1$. Since π is unequal to ι it follows that it can be written as $\pi = (\pi(1) \dots \pi(t) \pi(t+1) \dots \pi(n))$, where $\pi(1) < \dots < \pi(t) < 0 < \pi(t+1) < \dots < \pi(n)$ and $t \in [1 : n-1]$. Hence, for ℓ iTDRL $\rho = (\ell, \{-\pi(1), \dots, -\pi(t)\}, \{\pi(t+1), \dots, \pi(n)\})$ it holds that $\rho \circ \iota = \pi$. Now consider Case (ii). Since $0 < f_\pi$ (resp. $\ell_\pi < 0$) it holds that the left (resp. right) maximal increasing substring of π contains only positive (resp. negative) elements. Hence, π can be written as $\pi = (\pi(1) \dots \pi(t) \pi(t+1) \dots \pi(n))$, where $\pi(t+1) < \dots < \pi(n) < 0 < \pi(1) < \dots < \pi(t)$ and $t \in [1 : n-1]$. Then for the r iTDRL $\rho = (r, \{\pi(1), \dots, \pi(t)\}, \{-\pi(t+1), \dots, -\pi(n)\})$ we have $\rho \circ \iota = \pi$.

For the induction step let $k > 1$. It is enough to consider the following two cases: i) $|\mathcal{S}_\pi| \leq 2^{k-1}$ or ii) $|\mathcal{S}_\pi| = 2^{k-1} + 1$ and $\ell_\pi < 0 < f_\pi$. Lemma 7 implies $|\mathcal{S}_{T_i(\pi)}| \leq 2^{k-2}$ if Case (i) holds and $|\mathcal{S}_{T_i(\pi)}| \leq \lceil (2^{k-1} + 1)/2 \rceil = 2^{k-2} + 1$ if Case (ii) holds. Further, Lemma 8 ensures that $\ell_{T_i(\pi)} < 0 < f_{T_i(\pi)}$ if Case (ii) holds. By Lemma 5, π can be obtained by applying a single r iTDRL (resp. a ℓ iTDRL) to $T_1(\pi)$ (resp. $T_2(\pi)$) and by the induction hypothesis $T_i(\pi)$ with $i = 1, 2$ can be obtained by applying $k-1$ iTDRLs to ι . Hence, π can be obtained by applying k iTDRLs to ι . \square

Remark 1. Observe that the proof of Proposition 2 shows that the permutation π in Proposition 2 can always be obtained from ι by a single iTDRL (i. e., r iTDRL or ℓ iTDRL) followed by $k-1$ r iTDRLs.

3.3 The Main Theorem

The following theorem characterizes permutations π that have a certain number of maximal increasing substrings with respect to the number of iTDRLs that are necessary and sufficient to obtain π from ι .

Theorem 1. *Let $\pi \in s\mathcal{P}_n \setminus \{\iota\}$ be such that either $|\mathcal{S}_\pi| = 2^{k-2} + 1$ and $\ell_\pi > 0$ or $f_\pi < 0$, $2^{k-2} + 1 < |\mathcal{S}_\pi| \leq 2^{k-1}$, or $|\mathcal{S}_\pi| = 2^{k-1} + 1$ and $\ell_\pi < 0 < f_\pi$ for a $k \in \mathbb{N}$. Then k iTDRLs are necessary and sufficient in order to obtain π from ι .*

Proof. Let $\pi \in s\mathcal{P}_n \setminus \{\iota\}$ be such that either $2^{k-2} + 1 = |\mathcal{S}_\pi|$ and $\ell_\pi > 0$ or $f_\pi < 0$, $2^{k-2} + 1 < |\mathcal{S}_\pi| \leq 2^{k-1}$, or $|\mathcal{S}_\pi| = 2^{k-1} + 1$ and $\ell_\pi < 0 < f_\pi$ for a $k \in \mathbb{N}$. Proposition 1 shows that at least k iTDRLs are necessary to obtain π from ι . Proposition 2 shows that k iTDRLs are sufficient to obtain π from ι . Altogether, the theorem follows. \square

4 Solving the Distance and Sorting Problem for iTDRLs

This section considers the distance problem and the sorting problem for signed permutations with respect to iTDRLs.

Definition 3 (Distance Problem). *The distance problem for a permutation $\pi \in s\mathcal{P}_n$ with respect to iTDRLs aims to find the minimum number of iTDRLs that are required to produce π from ι , i.e., $d(\iota, \pi) := \arg \min_{k \in \mathbb{N}_0} \{\exists \rho_1, \dots, \rho_k \in \mathcal{R}_n : \rho_k \circ \dots \circ \rho_1 \circ \iota = \pi\}$. The sought minimum $d(\iota, \pi)$ is called distance between ι and π , or (iTDR) distance if the context is clear.*

Observe that Definition 3 indirectly covers the distance between arbitrary permutations $\pi, \sigma \in s\mathcal{P}_n$ as $d(\pi, \sigma) = d(\pi^{-1} \circ \pi, \pi^{-1} \circ \sigma) = d(\iota, \pi^{-1} \circ \sigma)$, where π^{-1} is the unique inverse permutation of π defined by $\pi^{-1}(e) := i$ if and only if $\pi(i) = e$ for all $i, e \in [-n : n] \setminus \{0\}$.

In the following lemma, we make use of Theorem 1 to give a closed formula for the iTDR distance.

Lemma 9. *The iTDR distance of ι to any signed permutation $\pi \in s\mathcal{P}_n \setminus \{\iota\}$ is given by*

$$d(\iota, \pi) = \begin{cases} \log_2(|\mathcal{S}_\pi| - 1) + 1 & \text{if } \exists k \in \mathbb{N} \text{ with } |\mathcal{S}_\pi| = 2^{k-1} + 1 \\ & \text{and } \ell_\pi < 0 < f_\pi; \\ \lceil \log_2 |\mathcal{S}_\pi| \rceil + 1 & \text{else.} \end{cases}$$

Proof. Let $\pi \in s\mathcal{P}_n \setminus \{\iota\}$. Then either $|\mathcal{S}_\pi| = 2^{k-2} + 1$ and $\ell_\pi > 0$ or $f_\pi < 0$, $2^{k-2} + 1 < |\mathcal{S}_\pi| \leq 2^{k-1}$, or $|\mathcal{S}_\pi| = 2^{k-1} + 1$ and $\ell_\pi < 0 < f_\pi$ for a $k \in \mathbb{N}$. Hence, either $2^{k-2} < |\mathcal{S}_\pi| \leq 2^{k-1}$ or $|\mathcal{S}_\pi| = 2^{k-1} + 1$ and $\ell_\pi < 0 < f_\pi$ holds. By Theorem 1 we have that k iTDRLs are necessary and sufficient to obtain π from ι , i. e., $d(\iota, \pi) = k$. Consider first $2^{k-2} < |\mathcal{S}_\pi| \leq 2^{k-1}$. This implies $k - 2 < \log_2 |\mathcal{S}_\pi| \leq k - 1$ and hence $k - 1 \leq \lceil \log_2 |\mathcal{S}_\pi| \rceil \leq k - 1$. Consequently, it holds that $d(\iota, \pi) = k = \lceil \log_2 |\mathcal{S}_\pi| \rceil + 1$. Now consider $|\mathcal{S}_\pi| = 2^{k-1} + 1$. Then $|\mathcal{S}_\pi| - 1 = 2^{k-1}$ implies $\log_2(|\mathcal{S}_\pi| - 1) = k - 1$. Consequently, $d(\iota, \pi) = k = \log_2(|\mathcal{S}_\pi| - 1) + 1$. \square

Note that Lemma 9 implies that the distance of a signed permutation π of length n can be computed by calculating the number of its maximal increasing substrings which can be done in time $\mathcal{O}(n)$.

Motivated by the tractability of the distance problem, the following sections study the sorting problem of signed permutations with respect to iTDRLs, i. e., one aims to find a minimum length sequence of iTDRLs that transforms one given permutation into another given permutation. The relevance of this problem is motivated by its biological connection: a minimum length sequence of rearrangements can be interpreted as a shortest path of mutations effecting the (possibly mitochondrial) gene order of the considered species that are represented by signed permutations. Hence, solving the sorting problem efficiently allows to trace back gene order evolution under the considered model to a limited extent.

Definition 4 (Sorting Problem). *The sorting problem for a signed permutation π with respect to iTDRLs aims to find a minimum length sequence of iTDRLs that transforms ι into π .*

Similarly to the distance problem, Definition 4 indirectly covers the problem to find a shortest sequence $\rho_1 = (d_1, L_1, R_1), \dots, \rho_k = (d_k, L_k, R_k) \in \mathcal{R}_n$ of iTDRLs between two arbitrary permutations $\pi, \sigma \in s\mathcal{P}_n$ (i. e., $\rho_k \circ \dots \circ \rho_1 \circ \pi = \sigma$), since such a sequence can be obtained by firstly finding $\rho'_k \circ \dots \circ \rho'_1 \circ \iota = \pi^{-1} \circ \sigma$ with $\rho'_i = (d'_i, L'_i, R'_i) \in \mathcal{R}_n$ and subsequently obtaining the sought iTDRLs ρ_1, \dots, ρ_k by $d_1 = d'_1$, $L_i = \{\pi(e) : e \in L'_i\}$, and $R_i = \{\pi(e) : e \in R'_i\}$ for all $i \in [1 : k]$.

In the following we present an algorithm that solves the sorting problem with respect to iTDRLs. Recall that by Remark 1 there is always a solution of the considered sorting problem that contains at most one ℓ iTDRL. Therefore, the following algorithm computes for a given permutation $\pi \in s\mathcal{P}_n$ a sequence $\rho_1, \dots, \rho_{d(\iota, \pi)} \in \mathcal{R}_n$ of iTDRLs such that $\rho_{d(\iota, \pi)} \circ \dots \circ \rho_1 \circ \iota = \pi$ and either $\rho_1, \dots, \rho_{d(\iota, \pi)}$ are riTDRLs or ρ_1 is a ℓ iTDRL and $\rho_2, \dots, \rho_{d(\iota, \pi)}$ are riTDRLs. The pseudo code of the algorithm can be found in Algorithm 1.

The main idea is to iteratively apply a “reversed” iTDRL transformation T_1 or (once in the last step) T_2 to the given permutation π to obtain a permutation $T_i(\pi)$ that has at most half as many maximal increasing substrings as π (see Lemma 7). By that process a minimum length sequence S of transformations (of T_1 or T_2) is obtained in reversed order, i. e., S transforms π into ι . Subsequently, the sought sequence of iTDRLs transforming ι into π is obtained by computing the inverting iTDRL for every transformation in S and reversing the relative order of all computed iTDRLs. Since Algorithm 1 uses exactly $d(\iota, \pi)$ iTDRLs to construct the sought sequence of iTDRLs, it solves the sorting problem for a given permutation π with respect to iTDRLs exactly.

Let $\pi \in s\mathcal{P}_n$. The case $\pi = \iota$ (i. e., the sorting sequence of iTDRLs is empty) is handled in lines 1-2. If otherwise $\pi \neq \iota$ then $\rho_{d(\iota, \pi)}, \dots, \rho_1 \in \mathcal{R}_n$ are iteratively computed in the lines 3-19. By Lemma 9 either $d(\iota, \pi) = \lceil \log_2 |\mathcal{S}_\pi| \rceil + 1$ or $d(\iota, \pi) = \log_2(|\mathcal{S}_\pi| - 1) + 1$ and both cases are handled in lines 3-6. For every $j \in [d(\iota, \pi) : 1]$ the maximal increasing substring composition of π is calculated in Line 8 and – depending on j and whether $|\mathcal{S}_\pi|$ is even or odd – either T_1 or T_2 is applied to π in lines 13-19 or lines 9-12. More precisely, if $j = 1$ and $f_\pi < 0$ (i. e., π is exactly one maximal increasing substring that contains negative and possibly positive elements) then T_2 is applied to π in Line 10 and in Line 11 the corresponding ℓ iTDRL (which exists by Lemma 5) is computed. Otherwise, i. e., either $j > 1$ or $j = 1$ and $f_\pi > 0$, the permutation π is substituted by $T_1(\pi)$ and the corresponding riTDRL $\rho_j \in \mathcal{R}_n$ is constructed as defined in the proof of Lemma 5. This iterative procedure gives $\rho_1, \dots, \rho_{d(\iota, \pi)} \in \mathcal{R}_n$ which are returned in Line 20.

For a runtime analysis of Algorithm 1 consider $\pi \in s\mathcal{P}_n$. Certainly, the check whether $\pi = \iota$ (Line 1), the computation of $d(\iota, \pi)$ (lines 3-6), the computation of the maximal increasing substring decomposition (Line 8), the construction of $T_1(\pi)$ and $T_2(\pi)$ (lines 14, 17 and Line 10), and the construction of ρ_j (lines 11, 15, 18) can be done in time $\mathcal{O}(n)$. Therefore, lines 8-19 are executed in time $\mathcal{O}(n)$ and they are executed at most $\lceil \log_2 |\mathcal{S}_\pi| \rceil + 1$ times. Since $|\mathcal{S}_\pi| \leq n$ it follows that Algorithm 1 has a runtime in $\mathcal{O}(n \log_2 n)$.

Algorithm 1 is implemented in C++ and it is freely available on <http://pacosy.informatik.uni-leipzig.de/spitdrl>.

Algorithm 1: Pseudo code of sorting by iTDRLs

Data: $\pi \in s\mathcal{P}_n$
Result: $\rho_1, \dots, \rho_k \in \mathcal{R}_n$ such that $\rho_k \circ \dots \circ \rho_1 \circ \iota = \pi$

```

1 if  $\pi == \iota$  then
2   | return  $\emptyset$ ;
3 if  $\exists h \in \mathbb{N}_0: |\mathcal{S}_\pi| = 2^h + 1$  and  $\ell_\pi < 0 < f_\pi$  then
4   |  $k = \log_2(|\mathcal{S}_\pi| - 1) + 1$ ;
5 else
6   |  $k = \lceil \log_2 |\mathcal{S}_\pi| \rceil + 1$ ;
7 for  $j \leftarrow k, \dots, 1$  do
8   |  $\pi = \pi_1 \dots \pi_{|\mathcal{S}_\pi|}$ ;
9   | if  $j == 1$  and  $f_\pi < 0$  then // Application  $T_2$ 
10  |   |  $\pi \leftarrow \overline{v_1} \oplus \kappa_1 = T_2(\pi)$ ;
11  |   |  $\rho_j = (\ell, \mathcal{E}(\overline{v_1}), \mathcal{E}(\kappa_1))$ ;
12  |   | continue;
13  | if  $|\mathcal{S}_\pi|$  is even then // Application  $T_1$ 
14  |   |  $\pi \leftarrow \pi_1 \oplus \overline{\pi_{|\mathcal{S}_\pi|}} \dots \pi_{|\mathcal{S}_\pi|/2} \oplus \overline{\pi_{|\mathcal{S}_\pi|/2+1}} = T_1(\pi)$ ;
15  |   |  $\rho_j = (r, \mathcal{E}(\pi_1 \dots \pi_{|\mathcal{S}_\pi|/2}), \mathcal{E}(\overline{\pi_{|\mathcal{S}_\pi|}} \dots \overline{\pi_{|\mathcal{S}_\pi|/2+1}}))$ ;
16  | else
17  |   |  $\pi \leftarrow \pi_1 \oplus \overline{\pi_{|\mathcal{S}_\pi|}} \dots v_{\lceil |\mathcal{S}_\pi|/2 \rceil} \oplus \overline{\kappa_{\lceil |\mathcal{S}_\pi|/2 \rceil}} = T_1(\pi)$ ;
18  |   |  $\rho_j = (r, \mathcal{E}(\pi_1 \dots v_{\lceil |\mathcal{S}_\pi|/2 \rceil}), \mathcal{E}(\overline{\pi_{|\mathcal{S}_\pi|}} \dots \overline{\kappa_{\lceil |\mathcal{S}_\pi|/2 \rceil}}))$ ;
19  |   | continue;
20 return  $\rho_1, \dots, \rho_k$ ;

```

5 Bounding the Distance Problem with respect to Mitochondrial Rearrangements

In the following, the four types of rearrangement operations that are relevant for mitochondrial gene orders evolution [7] are defined formally. Let $\pi \in s\mathcal{P}_n$, and let L and R be two character disjoint subsequences of π such that $\mathcal{E}(L) \cup \mathcal{E}(R) = \mathcal{E}(\pi)$. Further, let X and Y be two consecutive substrings of π , i. e., X, Y and either XY or YX are substrings of π . The *inversion* $\rho_I(X)$ applied to π reverses the order and it toggles the sign of every element of X . The *inverse transposition* $\rho_{iT}(X, Y)$ applied to π exchanges the order of X and Y and, in addition, it reverses the order and toggles the sign of every element in X . The *TDRL* $\rho_{TDRL}(L, R)$ applied to π duplicates π in tandem, followed by the loss of all elements of L (resp. R) in the left (resp. right) copy of the duplicated intermediate. A *transposition* $\rho_T(X, Y)$ applied to π swaps the order of X and Y . It is not hard to see that a transposition is a special case of the TDRL rearrangement. When all these four types of rearrangements are considered, we henceforth use the term *4-type rearrangements* for the sake of brevity. According to the definition of the distance with respect to iTDRL, the minimum number of 4-type rearrangements needed to transform one given signed permutation into another given signed permutation is called *4-type rearrangement distance*.

Up to now, it is still unknown whether the distance problem and the sorting problem (for a permutation) with respect to the 4-type rearrangements can be solved in polynomial time. However, the solutions obtained by solving the respective problem for the same permutation with respect to the iTDRL rearrangement can be used to obtain approximate solutions. The fundamental idea is to realize the connection between iTDRLs and 4-type rearrangements: 1) every iTDRL can be mimicked by at least one and at most two 4-type rearrangements, and 2) every 4-type rearrangement can be mimicked by at least one and at most two iTDRLs. On one hand every iTDRL has either the same effect as an inversion (see Figure 4(a)) or an inverse transposition (see Figure 4(b)), or it can be mimicked by both, a TDRL followed by an inversion, as well as a TDRL followed by an inverse transposition (see Figure 4(c)). On the other hand, an inversion and an inverse transposition have either the same effect as an iTDRL, or they can be mimicked by two iTDRLs, see Figure 4(a) and Figure 4(b), respectively. A TDRL (and therefore also transposition) can always be mimicked by two iTDRLs (see Figure 4(d)).

The fact that every iTDRL can be mimicked with at most two 4-type rearrangements implies that the 4-type rearrangement distance is less than

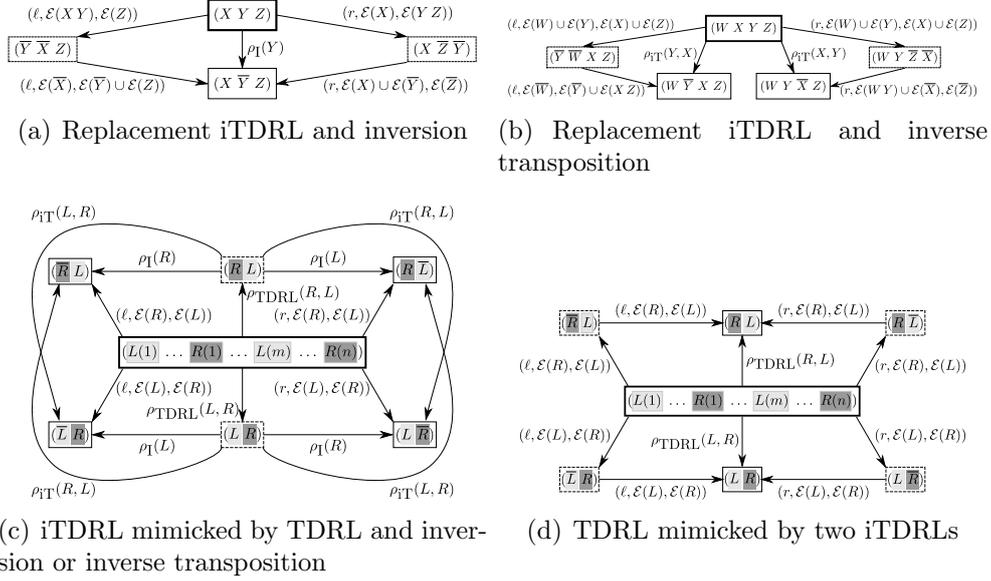


Figure 4: Sequences of 4-type rearrangements (resp. iTDRLs) that mimic an iTDRL (resp. 4-type rearrangement). Rearrangements are illustrated by black arrows. A permutation is framed by a thick (resp. thin) continuous black square if it is the input (resp. output) of a rearrangement that is mimicked. An intermediate permutation of a sequence of rearrangements is framed by a dashed black square. Intervals of a permutation are denoted by W , X , Y , Z , L , and R , where $L = L(1) \dots L(m)$ and $R = R(1) \dots R(n)$. (a) The inversion $\rho_I(Y)$ can be mimicked by $(\ell, \mathcal{E}(XY), \mathcal{E}(Z))$ (resp. $(r, \mathcal{E}(X), \mathcal{E}(YZ))$), followed by $(\ell, \mathcal{E}(\bar{X}), \mathcal{E}(\bar{Y}) \cup \mathcal{E}(Z))$ (resp. $(r, \mathcal{E}(X) \cup \mathcal{E}(\bar{Y}), \mathcal{E}(\bar{Z}))$). Observe that if X (resp. Z) is empty, then the second iTDRL is an identity mapping, hence the first iTDRL and the inversion have the same effect. (b) The application of $\rho_{iT}(Y, X)$ (resp. $\rho_{iT}(X, Y)$) can be mimicked by the application of $(\ell, \mathcal{E}(W) \cup \mathcal{E}(Y), \mathcal{E}(X) \cup \mathcal{E}(Z))$ (resp. $(r, \mathcal{E}(W) \cup \mathcal{E}(Y), \mathcal{E}(X) \cup \mathcal{E}(Z))$), followed by $(\ell, \mathcal{E}(\bar{W}), \mathcal{E}(\bar{Y}) \cup \mathcal{E}(XZ))$ (resp. $(r, \mathcal{E}(WY) \cup \mathcal{E}(\bar{X}), \mathcal{E}(\bar{Z}))$). Observe that if W (resp. Z) is empty, then the second iTDRL in the respective sequence is the identity mapping, hence the inverse transposition has the same effect as the first iTDRL. (c) The iTDRL $(d, \mathcal{E}(L), \mathcal{E}(R))$ (resp. $(d, \mathcal{E}(R), \mathcal{E}(L))$) can be replaced by applying the TDRL $\rho_{TDRL}(L, R)$ (resp. $\rho_{TDRL}(R, L)$), followed by the inversion $\rho_I(R)$ if $d = r$, or $\rho_I(L)$ if $d = \ell$ (resp. $\rho_I(R)$ if $d = \ell$ or $\rho_I(L)$ if $d = r$). Alternatively, the iTDRL $(d, \mathcal{E}(L), \mathcal{E}(R))$ (resp. $(d, \mathcal{E}(R), \mathcal{E}(L))$) can be mimicked by applying the TDRL $\rho_{TDRL}(R, L)$ (resp. $\rho_{TDRL}(L, R)$), followed by the inverse transposition $\rho_{iT}(L, R)$ if $d = \ell$, or $\rho_{iT}(R, L)$ if $d = r$ (resp. $\rho_{iT}(R, L)$ if $d = \ell$, or $\rho_{iT}(L, R)$ if $d = r$). (d) The TDRL $\rho_{TDRL}(L, R)$ (resp. $\rho_{TDRL}(R, L)$) can be mimicked by iteratively applying two times $(d, \mathcal{E}(L), \mathcal{E}(R))$ (resp. $(d, \mathcal{E}(R), \mathcal{E}(L))$), where $d \in \{\ell, r\}$.

twice the iTDRL distance, i. e., $d_{4\text{-type}}(\pi, \sigma) \leq 2d(\pi, \sigma)$, where $\pi, \sigma \in s\mathcal{P}_n$, and $d_{4\text{-type}}(\pi, \sigma)$ denotes the 4-type rearrangement distance for π and σ . In addition, the fact that every 4-type rearrangement can be mimicked by at most two iTDRLs implies $d(\pi, \sigma) \leq 2d_{4\text{-type}}(\pi, \sigma)$. Combining both inequations gives the following bounds on the 4-type rearrangement distance:

$$\frac{d(\pi, \sigma)}{2} \leq d_{4\text{-type}}(\pi, \sigma) \leq 2d(\pi, \sigma). \quad (1)$$

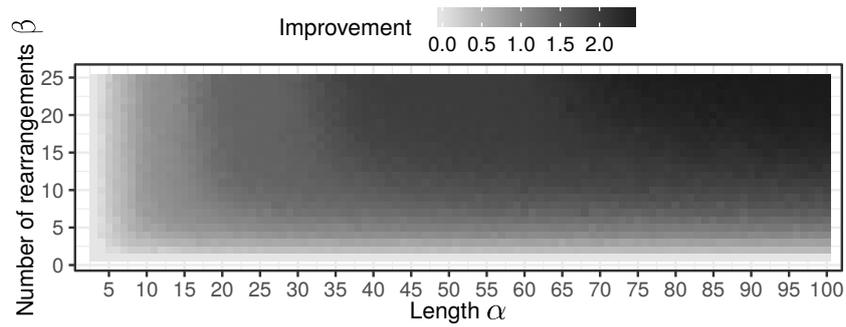
Consequently, the iTDRL distance is a 2-approximation for the 4-type rearrangement distance. In addition, an approximated sequence of 4-type rearrangements sorting π to σ can also be obtained by replacing every iTDRL by either an inversion, an inverse transposition, a TDRL and an inversion, or a TDRL and an inverse transposition as explained above.

6 Experiments

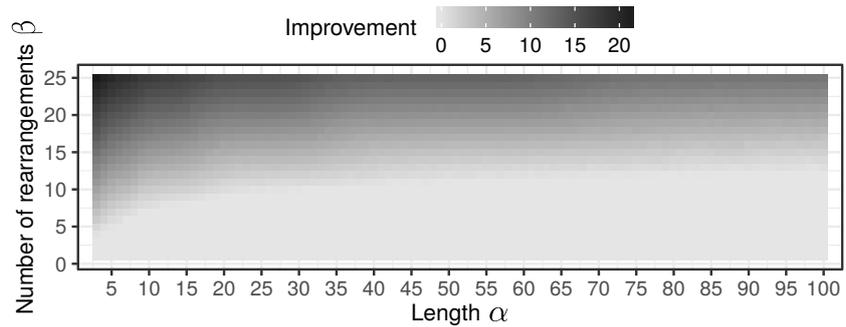
In this section we present experimental results for Algorithm 1 on sets of randomly generated benchmark problems. The aim of the experiment is to investigate the usefulness of the bounds on the 4-type rearrangement distance that are implied by the iTDRL distance.

Algorithm 1 has been applied to 2425 sets of benchmark problems that have been generated as explained in the following. To obtain a test permutation of the benchmark set $\mathcal{B}_{\alpha, \beta}$, exactly $\beta \in [1:25]$ 4-type rearrangements were applied by starting with the identity permutation ι of length $\alpha \in [3:100]$. For each of the β rearrangements, the type was first chosen uniformly at random, then a uniformly at random chosen rearrangement of the type that has been determined was applied. A resulting test permutation was included to the benchmark set $\mathcal{B}_{\alpha, \beta}$ only in the case that it was not equal to the identity permutation. For every combination of α and β this procedure was repeated until every benchmark set $\mathcal{B}_{\alpha, \beta}$ contains 100 signed permutations. Observe that by construction for each permutation $\pi \in \mathcal{B}_{\alpha, \beta}$ it holds that $1 \leq d_{4\text{-type}}(\iota, \pi) \leq \beta$.

Recall that for every permutation $\pi \in \mathcal{B}_{\alpha, \beta}$ the bounds $d(\iota, \pi)/2 \leq d_{4\text{-type}}(\iota, \pi) \leq 2d(\iota, \pi)$ are implied by the iTDRL distance. We refer to these bounds as iTDRL bounds. To investigate the contribution of the iTDRL bounds on the 4-type rearrangement distance, the following *improvement values* for the lower bound and the upper bound were computed: For every set $\mathcal{B}_{\alpha, \beta}$ the improvement value of the lower (upper) bound is defined as $lb_{\alpha, \beta} := d(\iota, \pi)/2 - 1$ if $d(\iota, \pi)/2 \geq 1$ and 0 otherwise (resp. $ub_{\alpha, \beta} := \beta - 2d(\iota, \pi)$ if $\beta \geq 2d(\iota, \pi)$ and 0 otherwise). Hence, a large (small)



(a) Improvement of lower bound



(b) Improvement of upper bound

Figure 5: Average improvement values for the lower bound (a) and the upper bound (b) for each benchmark set $\mathcal{B}_{\alpha,\beta}$ with $\alpha \in [3:100]$ and $\beta \in [1:25]$. Each tile represents a benchmark set and its grey value the corresponding improvement value.

improvement value implies that bounds obtained by the construction of the benchmark set are less (resp. more) improved by the iTDRL bounds.

Figure 5(a) illustrates the average improvement values of the lower bound of all benchmark sets. The average improvement value $lb_{\alpha,\beta}$ varies from 0 for $\mathcal{B}_{3,1}$ to 2.49 in $\mathcal{B}_{100,25}$. The figure shows that $lb_{\alpha,25}$ increases continuously for increasing permutation length α . This effect is reduced as β decreases such that the average improvement value of the lower bound is constant 0 for all benchmark sets with $\beta = 1$. The figure also shows that the difference $lb_{\alpha,25} - lb_{\alpha,1}$ increases with increasing α . Both effects follow from the fact that longer permutations are able to contain more maximal increasing substrings, which enables a larger iTDRL distance resulting in a larger lower bound.

The average improvement values for the upper bound are illustrated in Figure 5(b). The figure shows that $ub_{\alpha,\beta} = 0$ for all benchmark sets $\mathcal{B}_{\alpha,\beta}$ with approximately $\alpha \leq 2 \log_2 \beta$. For the benchmark sets with approximately $\alpha > 2 \log_2 \beta$ it can be seen that the improvement values increase as β increases. Both effects follow from the fact that the iTDRL distance (and therefore the upper bound it implies) grows logarithmically with respect to the size of the permutation, while the upper bound on the 4-type rearrangement distance β that is obtained by the construction grows linear. The results reflect that the upper bound β is very loose. This is reasonable, since Formula (1) implies that the 4-type rearrangement distance grows logarithmically with respect to the length of the permutation and not linear as β .

Both figures show that the benefit of the iTDRL bounds on the 4-type rearrangement distance is negligible for permutations that have a small size or that are separated by less rearrangements. However, if longer permutations are separated by a greater number of rearrangements, the bounds become more effective.

7 Conclusion

In this work the problem of computing the minimum number of iTDRL rearrangements (and a corresponding shortest scenario) that are necessary to transform one given gene order into another given gene order has initially been studied. It has been shown that the minimum number of iTDRLs needed for such a transformation can be computed in linear time, and that a corresponding scenario can be obtained in quasilinear time. Using the benefit that every type of major mitochondrial rearrangement (resp. every iTDRL) can be mimicked by at most two iTDRLs (resp. major mitochondrial rearrangements) and the fact that the distance problem with respect to iTDRLs is computationally tractable, we have shown that an upper bound and a lower

bound on the minimum number of inversions, (inverse) and a lower bound on the minimum number of inversions, (inverse) transpositions, and TDRs that are necessary to transform one given gene order into another given gene order can be computed efficiently. The effectiveness of these bounds has been investigated on simulated signed permutations.

References

- [1] G. Fertin, A. Labarre, I. Rusu, E. Tannier, and S. Vialette, *Combinatorics of genome rearrangements*. MIT press, 2009.
- [2] J. L. Boore, “Animal mitochondrial genomes,” *Nucleic Acids Research*, vol. 27, no. 8, pp. 1767–1780, 1999.
- [3] A. Weigert, A. Golombek, M. Gerth, F. Schwarz, T. H. Struck, and C. Bleidorn, “Evolution of mitochondrial gene order in annelida,” *Molecular Phylogenetics and Evolution*, vol. 94, pp. 196–206, 2016.
- [4] A. Basso, M. Babbucci, M. Pauletto, E. Riginella, T. Patarnello, and E. Negrisolo, “The highly rearranged mitochondrial genomes of the crabs *maja crispata* and *maja squinado* (majidae) and gene order evolution in brachyura,” *Scientific reports*, vol. 7, no. 1, p. 4096, 2017.
- [5] M. Bernt, K.-Y. Chen, M.-C. Chen, A.-C. Chu, D. Merkle, H.-L. Wang, K.-M. Chao, and M. Middendorf, “Finding all sorting tandem duplication random loss operations,” *Journal of Discrete Algorithms*, vol. 9, no. 1, pp. 32–48, 2011.
- [6] M. Al Arab, M. Bernt, C. H. zu Siederdissen, K. Tout, and P. F. Stadler, “Partially local three-way alignments and the sequence signatures of mitochondrial genome rearrangements,” *Algorithms for Molecular Biology*, vol. 12, no. 1, p. 22, 2017.
- [7] M. Bernt, A. Braband, B. Schierwater, and P. F. Stadler, “Genetic aspects of mitochondrial genome evolution,” *Molecular Phylogenetics and Evolution*, vol. 69, no. 2, pp. 328–338, 2013.
- [8] D. San Mauro, D. J. Gower, R. Zardoya, and M. Wilkinson, “A hotspot of gene order rearrangement by tandem duplication and random loss in the vertebrate mitochondrial genome,” *Molecular Biology and Evolution*, vol. 23, no. 1, pp. 227–234, 2005.

- [9] K. Chaudhuri, K. Chen, R. Mihaescu, and S. Rao, “On the tandem duplication-random loss model of genome rearrangement,” in *Proc. 17th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA '06)*, pp. 564 – 570, 2006.
- [10] T. Hartmann, A.-C. Chu, M. Middendorf, and M. Bernt, “Combinatorics of tandem duplication random loss mutations on circular genomes,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 1, pp. 83–95, 2016.
- [11] H. Kaplan, R. Shamir, and R. E. Tarjan, “A faster and simpler algorithm for sorting signed permutations by reversals,” *SIAM Journal on Computing*, vol. 29, no. 3, pp. 880–892, 2000.
- [12] K. M. Swenson, V. Rajan, Y. Lin, and B. M. Moret, “Sorting signed permutations by inversions in $\mathcal{O}(n \log(n))$ time,” *Journal of Computational Biology*, vol. 17, no. 3, pp. 489–501, 2010.
- [13] E. Tannier, A. Bergeron, and M.-F. Sagot, “Advances on sorting by reversals,” *Discrete Applied Mathematics*, vol. 155, no. 6-7, pp. 881–888, 2007.
- [14] P. Berman and M. Karpinski, “On some tighter inapproximability results,” in *Proc. 26th Int'l Colloq Automata, Languages, and Programming (ICALP '99)*, pp. 200–209, Springer, 1999.
- [15] A. Caprara, “Sorting by reversals is difficult,” in *Proc. 1st Ann. Int'l Conf. Computational Molecular Biology (RECOMB '97)*, pp. 75–83, ACM, 1997.
- [16] L. Bulteau, G. Fertin, and I. Rusu, “Sorting by transpositions is difficult,” *SIAM Journal on Discrete Mathematics*, vol. 26, no. 3, pp. 1148–1180, 2012.
- [17] V. Bafna and P. A. Pevzner, “Genome rearrangements and sorting by reversals,” *SIAM Journal on Computing*, vol. 25, no. 2, pp. 272–289, 1996.
- [18] A. Caprara, G. Lancia, and S. K. Ng, “Fast practical solution of sorting by reversals,” in *Proc. 11th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA '00)*, pp. 12–21, Society for Industrial and Applied Mathematics, 2000.

- [19] J. Kececioglu and D. Sankoff, “Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement,” *Algorithmica*, vol. 13, no. 1-2, p. 180, 1995.
- [20] D. A. Christie, *Genome rearrangement problems*. PhD thesis, University of Glasgow, 1998.
- [21] I. Elias and T. Hartman, “A 1.375-approximation algorithm for sorting by transpositions,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 3, no. 4, 2006.
- [22] M. Bouvel and D. Rossin, “A variant of the tandem duplication-random loss model of genome rearrangement,” *Theoretical Computer Science*, vol. 410, no. 8-10, pp. 847–858, 2009.
- [23] M. Bouvel and E. Pergola, “Posets and permutations in the duplication-loss model: Minimal permutations with d descents,” *Theoretical Computer Science*, vol. 411, no. 26-28, pp. 2487–2501, 2010.
- [24] J.-L. Baril and R. Vernay, “Whole mirror duplication-random loss model and pattern avoiding permutations,” *Information Processing Letters*, vol. 110, no. 11, pp. 474–480, 2010.
- [25] D. V. Lavrov, J. L. Boore, and W. M. Brown, “Complete mtDNA sequences of two millipedes suggest a new model for mitochondrial gene rearrangements: duplication and nonrandom loss,” *Molecular Biology and Evolution*, vol. 19, no. 2, pp. 163–169, 2002.
- [26] Z. Dias and C. C. de Souza, “Polynomial-sized ILP models for rearrangement distance problems,” in *Proc. Brazilian Symp. on Bioinformatics (BSB '07)*, p. 74, 2007.
- [27] D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B. F. Lang, and R. Cedergren, “Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome,” *Proceedings of the National Academy of Sciences*, vol. 89, no. 14, pp. 6575–6579, 1992.
- [28] F. A. M. da Silva, A. R. Oliveira, and Z. Dias, “Machine learning applied to sorting permutations by reversals and transpositions,” IC-PFG-17-03, University of Campinas, 2017.
- [29] M. Blanchette, T. Kunisawa, and D. Sankoff, “Parametric genome rearrangement,” *Gene*, vol. 172, no. 1, pp. GC11–GC17, 1996.

- [30] N. Eriksen, “ $(1+\varepsilon)$ -approximation of sorting by reversals and transpositions,” *Theoretical Computer Science*, vol. 289, no. 1, pp. 517–529, 2002.
- [31] Q.-P. Gu, S. Peng, and H. Sudborough, “A 2-approximation algorithm for genome rearrangements by reversals and transpositions,” *Theoretical Computer Science*, vol. 210, no. 2, pp. 327–339, 1999.
- [32] A. R. Oliveira, G. Fertin, U. Dias, and Z. Dias, “Sorting signed circular permutations by super short operations,” *Algorithms for Molecular Biology*, vol. 13, no. 1, p. 13, 2018.
- [33] M. E. M. Walter, Z. Dias, and J. Meidanis, “Reversal and transposition distance of linear chromosomes,” in *Proc. 5th Int’l Symp. String Processing and Information Retrieval (SPIRE ’98)*, pp. 96–102, 1998.
- [34] M. Bader and E. Ohlebusch, “Sorting by weighted reversals, transpositions, and inverted transpositions,” *Journal of Computational Biology*, vol. 14, no. 5, pp. 615–636, 2007.
- [35] G.-H. Lin and G. Xue, “Signed genome rearrangement by reversals and transpositions: models and approximations,” *Theoretical Computer Science*, vol. 259, no. 1-2, pp. 513–531, 2001.
- [36] T. Hartmann, N. Wieseke, R. Sharan, M. Middendorf, and M. Bernt, “Genome rearrangement with ILP,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 5, pp. 1585–1593, 2018.
- [37] T. Hartmann, M. Bernt, and M. Middendorf, “An exact algorithm for sorting by weighted preserving genome rearrangements,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2017. in press.
- [38] G. Lancia, F. Rinaldi, and P. Serafini, “A unified integer programming model for genome rearrangement problems,” in *Proc. Int’l Conf. Bioinformatics and Biomedical Engineering (iCBBE ’15)*, pp. 491–502, Springer, 2015.
- [39] M. Bernt, D. Merkle, K. Ramsch, G. Fritsch, M. Perseke, D. Bernhard, M. Schlegel, P. F. Stadler, and M. Middendorf, “CREx: inferring genomic rearrangements based on common intervals,” *Bioinformatics*, vol. 23, no. 21, 2007.

- [40] S. Yancopoulos, O. Attie, and R. Friedberg, “Efficient sorting of genomic permutations by translocation, inversion and block interchange,” *Bioinformatics*, vol. 21, no. 16, pp. 3340–3346, 2005.
- [41] A. Bergeron, J. Mixtacki, and J. Stoye, “A unifying view of genome rearrangements,” in *Proc. 6th Int’l Workshop Algorithms in Bioinformatics (WABI ’06)*, pp. 163–173, Springer, 2006.
- [42] F. Jühling, J. Pütz, M. Bernt, A. Donath, M. Middendorf, C. Florentz, and P. F. Stadler, “Improved systematic tRNA gene annotation allows new insights into the evolution of mitochondrial tRNA structures and into the mechanisms of mitochondrial genome rearrangements,” *Nucleic Acids Research*, vol. 40, no. 7, pp. 2833–2845, 2011.
- [43] X. Kong, X. Dong, Y. Zhang, W. Shi, Z. Wang, and Z. Yu, “A novel rearrangement in the mitochondrial genome of tongue sole, *cynoglossus semilaevis*: control region translocation and a tRNA gene inversion,” *Genome*, vol. 52, no. 12, pp. 975–984, 2009.
- [44] W. Shi, X.-L. Dong, Z.-M. Wang, X.-G. Miao, S.-Y. Wang, and X.-Y. Kong, “Complete mitogenome sequences of four flatfishes (Pleuronectiformes) reveal a novel gene arrangement of L-strand coding genes,” *BMC Evolutionary Biology*, vol. 13, no. 1, p. 173, 2013.
- [45] X. Liu, H. Li, Y. Cai, F. Song, W. John-James, and W. Cai, “Conserved gene arrangement in the mitochondrial genomes of barklouse families stenopsocidae and psocidae,” *Frontiers of Agricultural Science and Engineering*, vol. 4, no. 3, pp. 358–365, 2017.

Acknowledgments

TH was funded by a PhD student fellowship of the Leipzig University.