Sorting by Length-Weighted Reversals: Dealing with Signs and Circularity

Firas Swidan[1], Michael A. Bender[2⋆], Dongdong Ge[2], Simai He[2], Haodong Hu[2], and Ron Y. Pinter[1⋆⋆]

[1] Department of Computer Science,
Technion – Israel Institute of Technology, Haifa 32000, Israel.
{firas,pinter}@cs.technion.ac.il
[2] Department of Computer Science,
SUNY Stony Brook, Stony Brook, NY 11794-4400, USA.
{bender,dge,simaihe,huhd}@cs.sunysb.edu

Abstract. We consider the problem of sorting linear and circular permutations and 0/1 sequences by reversals in a length-sensitive cost model. We extend the results on sorting by length-weighted reversals in two directions: we consider the signed case for linear sequences and also the signed and unsigned cases for circular sequences. We give lower and upper bounds as well as guaranteed approximation ratios for these three cases. The main result in this paper is an optimal polynomial-time algorithm for sorting circular 0/1 sequences when the cost function is additive.

1 Introduction

Sorting by reversal (SBR) is of key importance in the study of genome rearrangement. To date, most of the algorithmic work on this problem applies to linear sequences (permutations or strings), and assumes the naïve unit-cost model. This model corresponds to a simple evolutionary model in which inversion mutations of any length are considered equally likely [1]; however, the mechanics of genome-rearrangement events (both inversions and other less ubiquitous transformations) suggest that the probabilities of reversals are dependent on fragment length.

Recently, a length-sensitive model was introduced [2, 3], in which the cost of a reversal is a function of the length of the reversed sequence and the total cost is the sum of the costs of the individual reversals. Several nontrivial lower and upper bounds have been derived for the problem of SBR under this cost model. However, these results pertain only to the unsigned version of the problem, where the direction of the individual elements does not matter (as opposed to the signed version); in the study of genome rearrangement, the direction of the

⋆ Supported in part by NSF Grants EIA-0112849, CCR-0208670, HRL Laboratories, and Sandia National Laboratories.

⋆⋆ Supported in part by the Bar-Nir Bergreen Software Technology Center of Excellence.

elements is often important, since each element represents a whole gene (or a portion thereof) whose direction has significance. Furthermore, many interesting genomes are *circular*; these include bacterial, mitochondrial, and chloroplast genomes, which all play an important role in studies of comparative genomics.

In this paper we extend the recent work for the length-sensitive model on linear sequences in which the cost of reversing a subsequence of length ℓ is $f(\ell) = \ell^\alpha$ (where $\alpha \geq 0$). We consider both permutations as well as 0/1 sequences representing the extreme cases of no repetitions at all among the elements on one hand, and as many repetitions as makes sense on the other hand. The elements are arranged in both linear and circular fashion, and we consider both the signed and unsigned cases.

Note that in general circularity offers more opportunities for lowering the optimal cost to sort a given sequence by reversals, and at the same time circularity presents challenges to providing efficient solutions. A non-unit cost model exacerbates the problems even farther. A sequence that exemplifies this distinction is $10^{n/2-1}1^{n/2-1}0$. One can sort the sequence into $0^{n/2}1^{n/2}$ with 2 reversals. Under a length-sensitive model with $\alpha = 1$ (i.e., the reversal cost is identical to the length of the sequence) the overall cost is $2n - 2$. In the circular case, where we adjoin the two ends of the sequence to form a circular arrangement, 1 reversal operation is enough to sort the sequence; its cost in the length sensitive model is 2. Thus, whereas the ratio between the costs of the two optimal solutions in the unit-cost model is 2, in the length-sensitive model it is $\Theta(n)$. Consequently, the treatment of circular sequences requires more care than the linear case.

Notice that the following relationships between the costs of the four sorting cases hold:

$$\text{unsigned circular} \leq \text{unsigned linear} \leq \text{signed linear} \ . \tag{1}$$

$$\text{unsigned circular} \leq \text{signed circular} \leq \text{signed linear} \ . \tag{2}$$

A summary of our results appears in Tables 1 and 2; note that the lower and upper bounds for the linear, unsigned case were extended verbatim from [3], whereas the approximation ratios vary somewhat among the different cases.

Table 1. Lower and upper bounds for SBR of signed or unsigned and linear or circular 0/1 sequences and permutations.

α Value	**Lower Bounds**	**Upper Bounds**	
		Permutations	**0/1's**
$0 \leq \alpha < 1$	$\Omega(n)$	$O(n \lg n)$	$\Theta(n)$
$\alpha = 1$	$\Omega(n \lg n)$	$O(n \lg^2 n)$	$\Theta(n \lg n)$
$1 < \alpha < 2$	$\Omega(n^\alpha)$	$\Theta(n^\alpha)$	$\Theta(n^\alpha)$
$\alpha \geq 2$	$\Omega(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$

Table 2. Approximation ratios for SBR of signed linear as well as signed and unsigned circular 0/1 sequences and permutations.

α Value	Signed Linear Permutations	0/1's	Unsigned Circular Permutations	0/1's	Signed Circular Permutations	0/1's
$0 \leq \alpha < 1$		$O(1)$		$O(1)$		$O(1)$
$\alpha = 1$	$O(\lg n)$	3	$O(\lg n)$	1	$O(\lg n)$	3
$1 < \alpha < 2$	$O(\lg n)$	$O(1)$				
$\alpha \geq 2$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$

Considerable research has been reported on the algorithmics of SBR (mostly signed) for linear sequences in the unit-cost model, see e.g. [4, 5]. Recently, several papers have addressed the issues of circularity and duplication, most notably Chen and Skiena [6] who deal with fixed-length reversals to sort both linear and circular permutations, Hartman [7] who deals with sorting by transpositions (which are much less common than reversals) on a circle, and Christie and Irving [8] who look at sorting (essentially) 0/1 linear sequences by both reversals and transpositions. None of these results, however, pertain to the length-sensitive model.

The rest of this paper is organized as follows. In Sect. 2 we provide algorithms for sorting 0/1 sequences, in Sect. 3 we deal with sorting circular permutations, and in Sect. 4 we show how to handle signed permutations in our length-sensitive model. Finally, in Sect. 5 we give sorting bounds for permutations and 0/1 sequences when inputs are signed or unsigned and circular or linear.

2 Exact and Approximation Algorithms for Sorting 0/1 Sequences

In this section we give exact and approximation algorithms for sorting linear signed as well as circular singed and unsigned 0/1 sequences. We first introduce approximation algorithms when $0 \leq \alpha < 1$. We then give an exact algorithm to sort unsigned circular 0/1 sequences when $\alpha = 1$. Finally, we introduce a reduction showing that the signed case can be approximated using the unsigned case when $\alpha \geq 1$.

2.1 Approximation Algorithms for $0 \leq \alpha < 1$

We now give $O(1)$ approximation algorithms for sorting linear signed as well as circular signed and unsigned sequences. The results are proved using potential function arguments.

Sorting Unsigned Circular Sequences. Given a circular sequence S, denote the length of the 0 and 1 blocks contained in S by $z_1, \ldots, z_k$ and $w_1, \ldots, w_k$

respectively. Let $Z = \max_{1 \le i \le k}\{z_i\}$ and $W = \max_{1 \le i \le k}\{w_i\}$. We define the potential function $P(S)$ as follows:

$$P(S) = \sum_{i=1}^{k}(z_i^\alpha + w_i^\alpha) - Z^\alpha - W^\alpha \ .$$

Lemma 1. *A reversal ρ of length r acting on a circular sequence S increases the value of the potential function $P(S)$ by at most $4r^\alpha$, that is, $P(S \cdot \rho) - P(S) \le 4r^\alpha$.*

Proof. The proof is by case analysis: since a reversal can cut two blocks at most (the blocks at either edge of the reversal), a reversal increases the value of the term $\sum_{i=1}^{k}(z_i^\alpha + w_i^\alpha)$ by at most $2r^\alpha$. In addition, by similar reasoning, the value of either Z^α or W^α can decrease by at most r^α.

□

Denote $S' = S \cdot \rho$. Notice that $S = S' \cdot \rho^{-1}$. Therefore, the reversal ρ decreases the potential value by the same amount that ρ^{-1} increases it. Thus, by applying Lemma 1 on the inverse reversal, we get a lower bound on the decrease of the potential value.

Corollary 1. *A reversal ρ of length r acting on a circular sequence S decreases the value of the potential function $P(S)$ by at most $4r^\alpha$, that is, $P(S \cdot \rho) - P(S) \ge -4r^\alpha$.*

Because the cost of a reversal of length r is r^α, and the value of $P(\cdot)$ equals 0 for a sorted sequence, we obtain the following lower bound.

Lemma 2. *The function $V(S) = \frac{1}{4}P(S)$ is a lower bound for sorting an unsigned circular sequence S.*

Proof. The proof is by induction on the number of reversals in an optimal solution.

Let m denote the number of reversals in an optimal sorting series. We prove that if a sorting solution uses exactly m reversals, then its cost is at least $V(S)$.

Base case: when $m = 0$, the sequence is already sorted. Induction step: suppose for all $m \le k$ the claim holds. Consider a 0/1 sequence S having an optimal sorting series of length $m = k + 1$. Denote the first reversal by ρ and let r be its length. The reversal ρ changes the sequence S to S', which can be optimally sorted by k reversals. Applying the induction assumption on S', we get that $V(S')$ is a lower bound for sorting S'. By Corollary 1, $P(S') + 4r^\alpha \ge P(S)$. By the definition of $V(\cdot)$ we get: $V(S') + r^\alpha \ge V(S)$. Therefore:

$$\mathrm{opt}(S) = \mathrm{opt}(S') + r^\alpha \ge V(S') + r^\alpha \ge V(S) \ .$$

as needed.

□

Lemma 2 motivates sorting a circular 0/1 sequence by fixing its two maximal blocks Z and W, and linearly sorting the two subsequences between them. The algorithm circularImprovedDC realizes this approach. Given a circular sequence S and two block indices i and j, consider the subsequence of blocks between i and j, counter clockwise, excluding the blocks i and j. If the subsequence starts with a 1, rename the 0's and 1's and define $S(i,j)$ to be the resulting subsequence.Define $S(j,i)$ similarly. In the following we use the algorithm improvedDC, introduced in [3], for sorting linear 0/1 sequences.

Algorithm 1 circularImprovedDC(S)

1: Let $z_{i_0} = Z$ and $w_{j_0} = W$
2: Define $S_1 = S(i_0, j_0)$ and $S_2 = S(j_0, i_0)$
3: $s_1 \leftarrow$ improvedDC(S_1)
4: $s_2 \leftarrow$ improvedDC(S_2)
5: Output $s_1 + s_2$

Theorem 1. *The algorithm circularImprovedDC has an approximation ratio of* $O(1)$.

Sorting Linear and Circular Signed Sequences. Consider a signed 0/1 sequence (linear or circular). Define a block in the sequence to be a contiguous segment of 0's (or 1's) having the same sign. Notice that there are four kinds of blocks in a signed 0/1 sequence.

Represent the 0/1 sequence as a sequence of blocks $b_1, \ldots, b_m$. Consider the potential function $V_1(S) = \frac{1}{2}\sum_{i=1}^{m} b_i^{\alpha}$ for linear sequences S and define $V_2(T)$ as in Lemma 2 for circular sequences T.

Lemma 3. *The potentials* $V_1(S)$ *and* $V_2(T)$ *are lower bounds on the cost of sorting linear and circular signed sequences respectively.*

Given a signed sequence S, let unsign(S) represent the sequence without the signs. The algorithm signedImprovedDC sorts signed linear sequences based on improvedDC.

Algorithm 2 signedImprovedDC(S)

1: $U \leftarrow$ unsign(S)
2: $u \leftarrow$ impovedDC(U)
3: Mimic the reversals used to sort U on S. Denote the resulting sequence by S'
4: Reverse elements of S' with a negative sign. Let s be the cost of this step
5: Output $s + u$

To sort circular signed sequences we modify the algorithm circularImprovedDC in a similar way. We refer to the modified algorithm as signedCircularImprovedDC.

Theorem 2. *The algorithms signedImprovedDC and signedCircularImprovedDC are $O(1)$ approximation algorithms.*

2.2 Optimal Algorithm for $\alpha = 1$

In this section we give a polynomial-time algorithm for sorting circular 0/1 sequences with additive cost functions ($\alpha = 1$). The sorting algorithm is based on dynamic programming. We give enough properties to constrain the set of candidate solutions, so that the optimal solution can be found in polynomial time. This approach was used in [3] to sort linear 0/1 sequences.

We first describe the three properties, "useless", "cutting" and "complex." The first two properties can be proved using the same techniques as in [3]. The main contribution of this part is in proving the third property, which is the most critical in establishing the optimality of the algorithm. Its proof is substantially different from the linear case, reflecting the essential divergence caused by the circularity. This is explained in detail below after we review some preliminaries.

Consider the two extreme blocks in a reversal. If the values of these blocks are identical, that is both equal 0 or 1, we call the reversal *useless*; if one of these two blocks is contained in a bigger block, we call the reversal *cutting*. If the reversal affects more than two blocks, we call the reversal *complex*. We call a reversal that is not complex *simple*. We can show that there are no useless and cutting reversals in a circular optimal solution. As mentioned earlier, the proof follows the same lines as in [3]. Roughly speaking, the same techniques apply because these proofs are "local," involving changes of reversals where the changes affect a single block of 0s or 1s (even though the reversal is longer). This explanation is elaborated below.

In contrast, the proof of the third property for linear sequences introduced in [3] depends heavily on the linearity and does not apply to circular sequences. In the proof by contradiction, one considers the last complex reversal. The changes that are introduced affect the two extreme blocks that the reversal contained. In the case of linear sequences, these blocks are far apart and one can subsequently make changes independently to the reversals involving these blocks. This property is not true for circular sequences. Specifically, reversals that involve one extreme block may also affect the other, because of "wrap-around".

In the following, all the sequences are circular unless mentioned otherwise.

Given a reversal series $\rho_1, \ldots, \rho_m$ acting on a 0/1 sequence $S = s_1, \ldots, s_q$, where $s_i \in \{0, 1\}$, denote the number of reversals in which element s_i participates by $N(s_i)$. Call $N(s_i)$ the *reversal count* of s_i.

When a subsequence $s_i, \ldots, s_j$ of S is never cut by a reversal series $\rho_1, \ldots, \rho_m$, we denote the number of reversals in which the subsequence takes part by $N(s_i, \ldots, s_j)$.

We show that for additive cost functions no optimal reversal series contains useless or cutting reversals, and there exists an optimal reversal series containing no complex reversals. Equation (3), relating the reversal counts to the reversal series cost is useful for the proofs.

$$\sum_{i=1}^{m} |\rho_i| = \sum_{j=1}^{q} N(s_j) \ . \tag{3}$$

The proofs of *useless* and *cutting*, appearing in [3], hold for the circular case.

Lemma 4. *A reversal series containing a useless reversal cannot be optimal.*

Lemma 5. *A reversal series containing a cutting reversal cannot be optimal.*

The following definitions are needed to prove that an optimal reversal series containing no complex reversals exists. Given a reversal ρ affecting a circular $0/1$ sequence S, notice that we can cut S, making it linear, without cutting the reversal. Such a linearization is done in a counter clockwise direction. This linearization is implicitly used throughout the section.

We represent a $0/1$ sequence $1^{w_1}, \ldots, 0^{w_{2\ell}}$ by the lengths of each block, that is we let $w = w_1, \ldots, w_{2\ell}$ denote a $0/1$ sequence. We refer to w as a weighted sequence. A subsequence of w, $sg = w_i, \ldots, w_j$ or for short (i, j), is called a *segment* of w. Let $sg_k = w_{i_k}, \ldots, w_{j_k}$ for $k \in \{1, 2\}$ be two segments of w. We say that sg_1 is a *sub-segment* of sg_2, or *contained* in sg_2, if $i_2 \leq i_1 < j_1 \leq j_2$.

We denote a reversal acting on a segment (i, j) of w by $\rho\,(i, j)$. Let $w = w_1, \ldots, w_{2\ell}$ be a weighted sequence, and let $\rho\,(i, j)$ be a reversal acting on it. If $i \equiv j + 1 (mod 2)$, or for short $i \equiv j + 1$, we say that ρ *unifies* w_i with w_{j+1}, and w_j with w_{i-1}. The reversal ρ is called a *unifying* reversal. In addition, we say that block v_{i-1} of $v = w \cdot \rho = v_1, \ldots, v_{2\ell-2}$ *contains* blocks w_{i-1} and w_j of w. Similarly we say that v_{j-1} of v contains blocks w_{j+1} and w_i of w.

A reversal series $\rho_1, \ldots, \rho_m$ acting on w separates a segment sg of w, if $\rho_1, \ldots, \rho_m$ unifies all the 0 and 1 blocks of sg. If $sg = w$ then $\rho_1, \ldots, \rho_m$ sorts w.

In the sorting process the essence of a reversal is not directly connected to the coordinates (i, j) that define the reversal, but is designated by the blocks that the reversal affects. This essence plays a crucial rule when we try to define *commutative* reversals.

Given a reversal ρ acting on w and a reversal η acting on $w \cdot \rho$, we say that η *commutes* with ρ if two reversals η' and ρ' exist such that $w \cdot \rho \cdot \eta = w \cdot \eta' \cdot \rho'$, where η' and ρ' affect the same blocks of w that η and ρ affected respectively (in the following we drop the primes).

One can verify that the following proposition holds.

Proposition 1. *Let w be a weighted sequence, ρ be a reversal acting on it, and η be a reversal acting on $w \cdot \rho$. The three following conditions are equivalent:*

1. *The reversal η commutes with ρ.*
2. *The blocks that η affects in $w \cdot \rho$ are contiguous in w.*
3. *The segments that ρ and η affect are either contained one in another or disjoint.*

Given a weighted sequence w and a minimum sorting series $\rho_1, \ldots, \rho_m$, define $w^j = w \cdot \rho_1 \cdots \rho_j$. Let ρ_j be the greatest index complex reversal and s the segment that ρ_j affects. Denote by $\bar{s}$ the rest of the sequence. Let ρ_k for $k > j$ be the smallest index reversal such that s is separated in w^k. Consider msg, the maximal segment that contains s and is separated by $\rho_j, \ldots, \rho_k$. If $msg \neq w$, the circularity of w is not used in the separation of msg. Therefore, we can consider msg as a linear 0/1 sequence, and ρ_j as a complex reversal used through the separation of a linear 0/1 sequence. Under these assumptions, [3] proved that msg can be separated optimally without the use of complex reversals. Therefore, without loss of generality, we assume that $msg = w$. That is, s is separated only when w is sorted. The following lemmas characterize the reversals performed after ρ_j.

Lemma 6. *Let w, j, ρ_j, w^j, s, and $\bar{s}$ be as above. Let ρ_k for $k > j$ be a reversal that does not commute with ρ_j. Then ρ_k affects at least one block that contains mixed elements from s and $\bar{s}$. In addition, all the blocks that ρ_k affects become part of a mixed block.*

Lemma 7. *Let w, j, ρ_j, w^j, s, and $\bar{s}$ be as in Lemma 6. Let ρ_k for $k > j$ be the smallest index reversal that commutes with ρ_j. Then ρ_k commutes with all the reversals ρ_q for $k > q > j$.*

Corollary 2. *Let w, j, ρ_j, w^j, s, and $\bar{s}$ be as in Lemma 7. We can rearrange the reversal series so that the reversals ρ_k for $k > j$ do not commute with ρ_j.*

The reversals ρ_k for $k > j$ are simple reversals and by Corollary 2 do not commute with ρ_j.

The following lemmas characterize the sorting process under simple reversals. They are used for proving that the reversal series cannot be optimal if ρ_j remains complex after performing the rearrangement described in Corollary 2, and for calculating the optimal cost in polynomial time.

Given a weighted sequence $w = w_1, \ldots, w_{2\ell}$ and a reversal series $\rho_1, \ldots, \rho_m$ containing no cutting reversals, define c_i to be the number of reversals in which w_i takes part, i.e. $c_i = N(w_i)$. Notice that c_i is well defined, since $\rho_1, \ldots, \rho_m$ contains no cutting reversals.

Lemma 8. *Let $w = w_1, \ldots, w_{2\ell}$ be a weighted sequence and let $\rho_1, \ldots, \rho_m$ be a sorting series having no useless, no cutting, and no complex reversals. Denote $w^k = w \cdot \rho_1 \cdots \rho_k$, and let $w^0 = w$. Then each block of w^k contains a block of w that takes part during $\rho_1, \ldots, \rho_k$ in zero reversals.*

Proof. By induction on k. Base case: the claim is trivial for $k = 0$. Induction step: suppose each block of w^k contains a block of w that takes part during $\rho_1, \ldots, \rho_k$ in zero reversals. We need to prove that each block of w^{k+1} contains a block of w that takes part during $\rho_1, \ldots, \rho_{k+1}$ in zero reversals. Since ρ_{k+1} is simple, we know that it acts on two successive blocks of w^k. Since ρ_{k+1} is not cutting, ρ_{k+1} must be of the form $\rho(i-1, i)$. Such a reversal unifies the block w_i^k with w_{i-2}^k, and w_{i-1}^k with w_{i+1}^k. The other blocks of w^k are not affected by ρ_{k+1}, therefore, they contain by the induction assumption a block of w that takes part during $\rho_1, \ldots, \rho_k$ in zero reversals. The same block takes part during $\rho_1, \ldots, \rho_k$ in zero reversals as well.

Since ρ_{k+1} affects w_i^k unifying it with w_{i-2}^k, and by the induction assumption w_{i-2}^k contains a block v of w that takes part during $\rho_1, \ldots, \rho_k$ in zero reversals, the unification of w_i^k with w_{i-2}^k in w^{k+1} contains v. Since v is not affected by ρ_{k+1}, v takes part during $\rho_1, \ldots, \rho_k$ in zero reversals, and thus fulfills the requirement.

A similar analysis applies to the unification of w_{i-1}^k with w_{i+1}^k in w^{k+1}. $\square$

Lemma 9. *Let $w = w_1, \ldots, w_{2\ell}$ be a weighted sequence and let $\rho_1, \ldots, \rho_m$ be a sorting series having no useless, no cutting, and no complex reversals. There exist indices i and j, where i corresponds to a block of 1's and j corresponds to a block of 0's, such that $c_i = c_j = 0$.*

Lemma 9 is helpful in proving that the outcome of Corollary 2 is a reversal series containing no complex reversals.

Lemma 10. *Let w, j, ρ_j, w^j, s, and $\bar{s}$ be the outcome of Corollary 2, that is ρ_k for $k > j$ do not commute with ρ_j. If ρ_j remains a complex reversal, the reversal series $\rho_1, \ldots, \rho_m$ cannot be optimal.*

Proof. We show that a reversal series having a lower cost exists. Consider the sequence w^j. The reversals ρ_k for $k > j$ are simple. By Lemma 9 there exists indices i' and p', where i' and p' correspond to a block of 1's and 0's respectively, such that $c_{i'} = c_{p'} = 0$. Notice that each of the blocks i' and p' could be contained as a unit in w^{j-1}, or could correspond to two blocks, one in s and one in $\bar{s}$. In the latter case, pick the part of the block that is in $\bar{s}$. In the former pick the blocks themselves. Denote the indices of the picked blocks in w^{j-1} by i and p. We divide the analysis into three cases:

1. The indices i and p are contained in $\bar{s}$. In this case we get $c_p = c_i = 0$, and therefore i and p divide the circular sequence into two linear subsequences. We consider the subsequence between i and p that contains s, denote it v, and the restriction of $\rho_1, \ldots, \rho_m$ to v. This restriction separates v and contains a complex reversal followed by simple reversals that do not commute with it. As mentioned before, [3] proved that such a series cannot be optimal.
2. The indices i and p are contained in s. This imposes that $|i - p| = 1$, or else the reversals acting on the blocks between i and p commute with ρ_j. Suppose that i appears before p in s when scanning the circular sequence in the counter clockwise direction. The other case is handled by renaming the 0's and 1's. Performing the reversals ρ_k for $k > j$ restricted to s and to $\bar{s}$ separates $\bar{s}$, while s gets the form $0^+1^+0^+1^+$. The notation 0^+ corresponds to an arbitrary strictly positive weight of 0's. The whole sequence is of the form $\overbrace{0^+}^{\bar{s}}\ \overbrace{0^+1^+0^+1^+}^{s}\ \overbrace{1^+}^{\bar{s}}$. Notice that the orientation of s and $\bar{s}$ is opposed since ρ_j is not performed on s. To sort the sequence, perform a last reversal

Table 3. Available reversal counts summary.

Segment	s_2			i	s_1		$\bar{s}_2$		
Block	1^+	0^+	1^+	1^+	1^+	0^+	0^+	1^+	0^+
Available reversal count	2	2	1	1	1	1	1	1	0
Explanation: reversals skipped	ρ_j, ρ_m	ρ_j, ρ_m	ρ_j	ρ_j	ρ_j	ρ_j	ρ_m	ρ_m	-

$\overbrace{0^+}^{\bar{s}}\ \ 0^+\ \ \overbrace{1^+0^+}^{\text{last reversal}}\ \ 1^+\ \ \overbrace{1^+}^{\bar{s}}$. The cost of the modified series is not greater than the cost of the original one, since the last reversal's length is not greater than ρ_j's length. However, by Lemma 6, the restriction of ρ_{j+1} to s or to $\bar{s}$ produces a reversal containing a single block. If we omit this reversal from the modified series the cost of the modified series becomes smaller than the original one. A contradiction to the optimality of the original series.

3. The index i is contained in s and p contained in $\bar{s}$ or vice versa. We analyse the former case. The latter is handled by renaming the 0's and 1's. Notice that after performing ρ_j, the indices i and p divide the circle into two independent components. The reversal ρ_m can affect only one of these components. Consider the following division of the sequence w^j (counter clockwise): $\overbrace{0^+}^{p}\ \overbrace{\cdots}^{\bar{s}_1}\ \overbrace{\overbrace{\cdots}^{s_1}\ \overbrace{1^+}^{i}\ \overbrace{\cdots}^{s_2}}^{s}\overbrace{\cdots}^{\bar{s}_2}$, where $\bar{s}_1$ and s_1 are the segments of $\bar{s}$ and s respectively lying between p and i, while $\bar{s}_2$ and s_2 are the segments of $\bar{s}$ and s respectively lying between i and p. Suppose that ρ_m affects $\bar{s}_2$ and s_2. Then the reversals ρ_k for $m > k > j$ separate s_1 and $\bar{s}_1$. Notice that omitting ρ_j does not affect the separation state of these segments. On the other hand, performing ρ_k for $m > k > j$ on $\bar{s}_2$ and s_2 does not separate them. Modify the reversal series as follows: Perform ρ_k for $m > k > j$ on w^{j-1} restricted to each of $\bar{s}_1, s_1, \bar{s}_2$, and s_2 (do not perform ρ_j and ρ_m). The resulting sequence is of the form: $\overbrace{0^+}^{p}\ \overbrace{0^+\,1^+}^{\bar{s}_1}\ \overbrace{1^+\,0^+\,1^+}^{s_2}\ \overbrace{1^+}^{i}\ \overbrace{1^+\,0^+}^{s_1}\ \overbrace{0^+\,1^+\,0^+}^{\bar{s}_2}$. Table 3 summarizes the yet available reversal counts for part of the segments. To sort the remaining blocks perform the following reversals (reversals indicated by brackets $[,]$):

$$\overbrace{1^+\,[0^+\,1^+}^{s_2}\ \overbrace{1^+}^{i}\ \overbrace{1^+]\,0^+}^{s_1}\ \text{and}\ \overbrace{1^+\,[0^+}^{s_2}\ \overbrace{0^+}^{s_1}\ \overbrace{0^+\,1^+]\,0^+}^{\bar{s}_2}\ .$$

The two reversals sort the sequence, and do not violate the available reversal counts. Thus the modified reversal series has a cost not greater than the original one. A contradiction is established similarly to previous cases.

Putting it all together we prove the claim. □

Lemmas 4, 5, and 10 establish Theorem 3.

Theorem 3. *An optimal reversal series contains no useless and no cutting reversals. In addition, an optimal reversal series containing no complex reversals exists.*

Theorem 3 and Lemma 9 implies that finding the optimal sorting cost can be done by taking the minimum over the following set: for all pair of indices (i, j), where $i < j$ and $i - j \equiv 1$, sort the segments $sg_1 = (i, j)$ and $sg_2 = (j, i)$ under the restriction $c_i = c_j = 0$. The sum of the two sorting costs is a candidate for the optimal solution. The calculation of the sorting costs for all pairs (i, j) and (j, i) can be done by running zerOneSort on ww, where w is an arbitrary linearization of a circular 0/1 sequence. The algorithm zerOneSort, introduced in [3], is an exact algorithm for sorting unsigned linear sequences when $\alpha = 1$. The algorithm circularZerOneSort implements this idea.

Algorithm 3 circularZerOneSort(w)

```
Run zerOneSort on ww. Let A be the dynamic programming matrix built by zerOneSort
opt ← ∞
for i = 0 to |w| − 1 in steps of 1 do
  for j = i + 1 to |w| − 1 in steps of 2 do
    tmp ← A(i, j, c_i = c_j = 0) + A(j, i + |w|, c_j = c_i = 0)
    if tmp < opt then
      opt ← tmp
    end if
  end for
end for
Output opt
```

Theorem 4. *The algorithm circularZerOneSort sorts a circular* $0/1$ *sequence optimally with a time complexity in* $O\left(n^3\right)$ *and a space complexity in* $O\left(n^2\right)$

2.3 Approximation Algorithms for $\alpha \geq 1$

We now give a reduction showing that a signed circular or linear sorting algorithm can be derived from an unsigned circular or linear sorting algorithm respectively, while retaining the asymptotic approximation ratio.

Let A be an algorithm for sorting unsigned circular 0/1 sequences and T be a signed circular sequence. We can sort T by running A on unsign(T) and correct the signs of the resulting sequence elements. Lemma 11 shows that this method approximates the optimal sorting cost. A similar result applies to the singed linear case.

Lemma 11. *The sum of* $A(\mathrm{unsign}(T))$ *and the cost of signs correction is less than* $\mathrm{opt}(T) + 2A(\mathrm{unsign}(T))$.

By combining the optimality of circularZerOneSort (Sect. 2.2) and Lemma 11, we obtain an approximation algorithm for signed circular sequences when $\alpha = 1$. A similar result applies to signed linear sequences.

Corollary 3. *The algorithm circularZerOneSort followed by signs correction is a 3-approximation algorithm for sorting signed circular sequences when $\alpha = 1$.*

In [3] its shown that bubble sort is optimal for sorting unsigned 0/1 sequences when $\alpha \geq 2$. By Lemma 11 we get an approximation algorithm for sorting signed sequences when $\alpha \geq 2$.

Corollary 4. *Bubble sort followed by signs correction is a 3-approximation algorithm for sorting signed linear or circular sequences when $\alpha \geq 2$.*

3 Sorting Circular Permutations

In this section we present an approximation algorithm for sorting circular permutations when $\alpha = 1$. The algorithm is based on an analysis that there exists a cut that changes the circular permutation into a linear permutation, such that the sorting cost for the linear permutation is only a constant factor larger than the sorting cost for the circular permutation. The algorithm is therefore to try all possible cuts. The proof follows an accounting argument given in this section, and is heavily dependent on the additivity of the cost function. The main idea of the argument is that we can mimic the circular reversals that we cut with linear ones, such that the linear reversals go the other way around the circle, not crossing the cut. The accounting argument shows that the total cost increases by at most a constant factor.

Let Π be a circular permutation on n elements. We present Π as a sequence of integers. Denote the n linear permutations equivalent to the circular identity permutation by I_j for $1 \leq j \leq n$. In this section, unless mentioned otherwise, permutations are circular.

Associate an *index* with each space between two successive elements of Π. A reversal *affects* an index i, if the reversal affects the two elements neighboring i. Map each index i to the number of reversals r_i affecting the index. We refer to r_i as the *index reversal count.* Let i_0 be an index such that $r_{i_0} = \min_i\{r_i\}$.

In the following we establish a lower bound on the sorting cost using r_{i_0}. Denote the circular optimal sorting cost of a circular permutation Π by $\mathrm{cosc}\,(\Pi)$. The linear optimal sorting cost of a linear permutation $\hat{\Pi}$ is denoted by $\mathrm{losc}(\hat{\Pi})$.

Lemma 12. *Let Π be a circular permutation, $\rho_1, \ldots, \rho_m$ be an optimal reversal sequence, and i_0 be as above. The following holds:*

$$cosc\,(\Pi) \geq n \cdot r_{i_0} \ . \tag{4}$$

Proof. By definition, a reversal affects an index if and only if it affects its two neighbors. Let i be an index with two neighbors π_j and π_{j+1}. We get that $2 \cdot r_i \leq N\,(\pi_j) + N\,(\pi_{j+1})$, where $N\,(\pi_j)$ is the number of reversals in which element π_j takes part. Notice that the set of neighbors of all odd or all even indices does not contain repetition. Thus, we get:

$$\sum_j N(\pi_j) \geq \max\{\sum_{i \text{ odd}} 2 \cdot r_i, \sum_{i \text{ even}} 2 \cdot r_i\} \ .$$

For optimal sorting series, one can verify that $\sum_j N(\pi_j) = \mathrm{cosc}(\Pi)$. Hence, $\mathrm{cosc}(\Pi) \geq \max\{\sum_{i \text{ odd}} 2 \cdot r_i, \sum_{i \text{ even}} 2 \cdot r_i\}$. From the definition of i_0 we get: $\mathrm{cosc}(\Pi) \geq n \cdot r_{i_0}$. □

Lemma 12 enables us to mimic circular reversals by linear reversals, while keeping the sorting cost within a constant factor. Given an index i, define Π_i to be the linear permutation derived by cutting Π at index i. Define Π_{lin} to be Π_{i_0}.

Lemma 13. *Let Π, r_i, and i_0 be as in Lemma 12. Consider the linear permutation Π_{lin} derived by cutting Π at index i_0. Let ρ be a reversal affecting i_0. The reversal ρ can be mimicked by two reversals on Π_{lin} with a total cost less than $2n$.*

Proof. Represent Π and Π_{lin} as follows: $\overbrace{i_0}^{\text{index}} \ \overbrace{\cdots}^{y\text{ elements}} \ \overbrace{\cdots}^{n-y-x\text{ elements}} \ \overbrace{\cdots}^{x\text{ elements}}$, where ρ affects the x and y elements. The reversal ρ can be mimicked on Π_{lin} by the following two reversals (the brackets [,] indicate the reversal):

$$[\ \overbrace{\cdots}^{y\text{ elements}} \quad \overbrace{\cdots}^{n-y-x\text{ elements}} \quad \overbrace{\cdots}^{x\text{ elements}}\ ] \ .$$

$$\overbrace{\cdots}^{x\text{ elements reversed}} \quad [\quad \overbrace{\cdots}^{n-y-x\text{ elements reversed}} \quad] \quad \overbrace{\cdots}^{y\text{ elements reversed}} \qquad .$$

The final result is equivalent to the circular permutation $\Pi \cdot \rho$ and the cost of the two reversals is bounded by $2n$. □

Since there are r_{i_0} reversals to mimic, the total cost of mimicking the circular sorting procedure by a linear one is bounded by $2 \cdot n \cdot r_{i_0}$. This yields a constant factor approximation.

Theorem 5. *The optimal circular sorting cost of Π can be approximated up to a constant factor by the optimal linear distance of Π_{lin} to one of the n linear permutations equivalent to the circular identity permutation.*

Proof. Define i_0 as in Lemma 13. By Lemma 13, we can mimic all reversals in an optimal circular solution by reversals in a linear solution, ending up with a linear permutation I_{j_0} equivalent to the circular identity permutation. The following bound holds for the optimal linear distance between Π_{lin} and I_{j_0}.

$$d(\Pi_{\text{lin}}, I_{j_0}) \leq \mathrm{cosc}(\Pi) + 2 \cdot n \cdot r_{i_0} \ .$$

By Lemma 12 we get: $d(\Pi_{\text{lin}}, I_{j_0}) \leq 3\mathrm{cosc}(\Pi)$. □

Theorem 5 enables approximating the optimal circular sorting cost using algorithms for sorting linear permutations: cut the circular permutation at each index, approximate the distance between the cut and all I_j, and return the minimum result. Sorting linear permutations is done by the algorithm reorderReversalSort, introduced in [3], which is a $O(\log n)$ approximation algorithm. The algorithm circulaReordeReversalSort is based on this idea.

Algorithm 4 *circulaReordeReversalSort*(Π)

$opt \leftarrow \infty$
for all indices i **do**
  for all $j \in \{1, 2, \ldots, |\Pi|\}$ **do**
    $sortingCost \leftarrow$ reorderReversalSort $\left(I_j^{-1} \cdot \Pi_i\right)$
    if $sortingCost < opt$ **then**
      $opt \leftarrow sortingCost$
    end if
  end for
end for
Output opt

Theorem 6. *The algorithm circulaReordeReversalSort has an approximation ratio of* $O(\log n)$.

4 Sorting Signed Permutations

The algorithms for sorting unsigned permutations are modified to sort signed permutations, while retaining their approximation ratios. This is done using the Bafna-Pevzner reduction [9].

Theorem 7. *The algorithm reorderReversalSort can be modified to sort signed permutations, while retaining the algorithm's approximation ratio.*

A similar result applies to the circular case.

5 Basic Sorting Bounds

In this section we give sorting bounds for linear signed as well as circular singed and unsigned permutations and 0/1 sequences The results are extensions of the linear unsigned results introduced in [3]. Equations (1) and (2) make extending the upper and lower bounds straightforward.

Upper Bounds

By (1) and (2), the singed linear case is an upper bound to all other cases. The cost of changing the signs of all elements is in $O(n)$. Thus, the unsigned linear cost summed to $O(n)$ is an upper bound. This fact combined with the upper bounds in [3] implies the results shown in Table 1.

Lower Bounds

A lower bound for the unsigned circular case is by (1) and (2) a lower bound for all other cases. The potential function arguments introduced in [3] can be readily modified to fit the unsigned circular case. Summary of the lower bound results, for both permutations and 0/1, appears in Table 1.

References

1. Nadeau, J.H., Taylor, B.A.: Lengths of chromosomal segments conserved since divergence of man and mouse. Proc. Natl. Acad. Sci. USA **81** (1984) 814–818
2. Pinter, R., Skiena, S.: Sorting with length-weighted reversals. In: Proc. 13th International Conference on Genome Informatics (GIW). (2002) 173–182
3. Bender, M., Ge, D., He, S., Hu, H., Pinter, R., Skiena, S., Swidan, F.: Improved bounds on sorting with length-weighted reversals. In: Proc. 15th ACM-SIAM Symposium on Discrete Algorithms (SODA). (2004) 912–921
4. Pevzner, P.A.: Computational Molecular Biology - an Algorithmic Approach. MIT Press, Cambridge MA (2000)
5. Bergeron, A.: A very elementary presentation of the hannenhalli-pevzner theory. In: Proc. 12th Symp. Combinatorial Pattern Matching (CPM). (2001) 106–117
6. Chen, T., Skiena, S.: Sorting with fixed-length reversals. Discrete Applied Mathematics **71** (1996) 269–295
7. Hartman, T.: A simpler 1.5-approximation algorithm for sorting by transpositions. In: Proc. 14th Symp. Combinatorial Pattern Matching (CPM). (2003) 156–169
8. Christie, D.A., Irving, R.W.: Sorting strings by reversals and by transpositions. SIAM J. on Discrete Math **14** (2001) 193–206
9. Bafna, V., Pevzner, P.: Genome rearrangements and sorting by reversals. SIAM J. Computing **25** (1996) 272–289