On Sorting by Reversals

Jonas Hübotter

July 14, 2020

A reversal is a transformation that inverses the order of some consecutive elements of a permutation. Sorting by reversals is the problem of finding the shortest sequence of reversals transforming a permutation to the identity. The length of such a sequence is said to be the reversal distance of that permutation. In this paper we introduce important concepts used in the literature on this topic. In particular, we introduce the fundamental relationship between the reversal distance and the number of alternating cycles in a cycle decomposition of a suitably-defined bi-colored graph. Then, a 3/2-approximation obtained by Christie is discussed in detail.

1 Introduction

In computational biology a fundamental problem is to determine the evolutionary similarity between genome sequences. By far the most common mutation on genomes is the inversion. A genome can be represented by a permutation. An inversion then simply is the reversal of a substring of a permutation representing a genome [KS95].

We refer by $\langle S_n, \circ \rangle$ to the symmetric group with $S_n = \{(\pi_1 \ \dots \ \pi_n) \mid \{\pi_1, \dots, \pi_n\} = [1, n]\}$. Let $\pi \in S_n$ be a permutation and π_i denote $\pi(i)$. The identity permutation $id = (1 \ \dots \ n)$ is the permutation mapping every input to itself. A *reversal* $\rho(i, j) \in S_n$ inverses the substring from π_i to π_j of a permutation. It is represented by the permutation

$$(1 \ \cdots \ i-1 \ \ j \ \ j-1 \ \cdots \ i+1 \ \ i \ \ j+1 \ \cdots \ n).$$

Composing π with $\rho(i, j)$ yields $(\pi_1 \ \cdots \ \pi_{i-1} \ \pi_j \ \pi_{j-1} \ \dots \ \pi_{i+1} \ \pi_i \ \pi_{j+1} \ \cdots \ \pi_n)$ where the elements $\pi_i, \dots, \pi_j$ have been reversed.

Motivated by the outlined application in computational biology, we define the following problem.

Definition 1.1. Given two permutations $\sigma, \tau \in S_n$, the *reversal distance problem* is the problem of finding the shortest sequence of reversals transforming σ into τ. More formally, we want to find reversals $\rho_1, \dots, \rho_d \in S_n$ such that

$$\sigma \circ \rho_1 \circ \cdots \circ \rho_d = \tau$$

and d is minimal. The number of necessary reversals d is called the *reversal distance* between σ and τ.

We now observe that the very same shortest sequence of reversals transforming σ into τ is also the shortest sequence of reversals transforming $\tau^{-1} \circ \sigma$ into *id*. This simply follows from left-composing τ^{-1} with both sides of the equation.

Definition 1.2. *Sorting by reversals (MIN-SBR)* then is the problem of determining the reversal distance between some permutation $\pi \in S_n$ and *id*. The reversal distance is denoted by $d(\pi)$. A sequence of reversals transforming π into *id*, *sorts* π.

Perhaps the most intuitive algorithm for sorting by reversals is the *ratchet algorithm* proposed by Watterson et al. [Wat+82]. The algorithm sequentially brings all elements from 1 to n into the correct position. More formally, in every step i, if $\pi_i \neq i$, we apply the reversal $\rho(i, \pi_i^{-1})$ (π_i^{-1} refers to the position of element i in the permutation π). Once step $n-1$ is completed, element n must be in position n. Therefore, this algorithm sorts any permutation in at most $n-1$ steps. The approximation of this algorithm, however, can be arbitrarily poor. Consider the permutation $(1\ n\ n-1\ \ldots\ 2)$. The ratchet algorithm sorts this permutation in $n-1$ reversals. But the permutation can also be sorted by the single reversal $\rho(2, n)$ [KS95].

MIN-SBR has been proven NP-hard by Caprara [Cap97] and MAX-SNP hard by Berman et al. excluding the existance of a polynomial time approximation scheme for this problem [BK99]. The problem of sorting signed permutations by reversals, that is permutations where each element may be either positive or negative, was shown to be solvable in polynomial time by Hannenhalli et al. [HP95].

In this paper we give an overview of approximation algorithms for MIN-SBR. We first introduce the notion of breakpoints given by Kececioglu et al. allowing us to formulate a first lower bound for the reversal distance of π and leading to a 2-approximation [KS95] (Section 2). In Section 3 we define a suitable bi-colored graph first introduced by Bafna et al. to find a tighter lower bound [BP96]. In doing so, we make a fundamental observation relating the reversal distance to the number of alternating cycles in a cycle decomposition of such a graph. Lastly, we consider a 3/2-approximation described by Christie [Chr98] (Section 4). A 1.375-approximation was later obtained by Berman et al., but is not considered here [BHK01].

2 Breakpoints

A very fundamental way of approaching the problem of sorting by reversals is to consider the relationship between two consecutive elements of a permutation.

Definition 2.1. Let $i \sim j$ if $|i - j| = 1$. A pair of consecutive elements π_i and π_j forms an *adjacency* if $\pi_i \sim \pi_j$ and a *breakpoint* if $\pi_i \not\sim \pi_j$. We denote by $b(\pi)$ the number of breakpoints in the permutation π. Figure 1 shows an example for the breakpoints of a permutation.

We observe that the number of breakpoints of the identity permutation $b(id)$ is 0. Based on this observation our goal in the following sections is to reduce the number of breakpoints of a given permutation π to 0 as quickly as possible by applying reversals. We also observe that with our current definition of permutations, the identity permutation is in fact not the

$$(7 \mid 5\ 6 \mid 3\ 2 \mid 4 \mid 1)$$

Figure 1: Breakpoints (breakpoints marked by vertical bars)

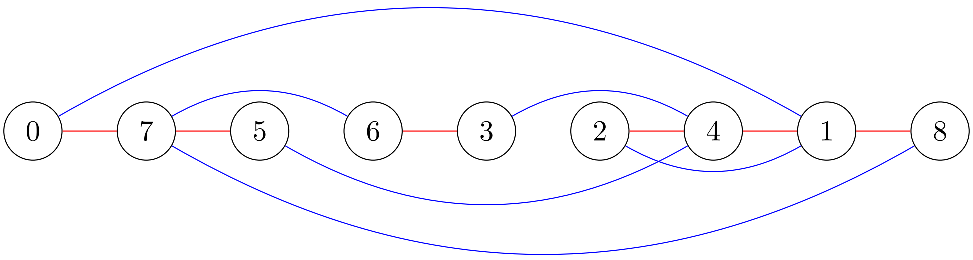

Figure 2: Breakpoint graph

only permutation without any breakpoints. Consider the permutation $(n\ n+1\ \ldots\ 2\ 1)$ (the inversed identity permutation) which also does not have any breakpoints.

In an effort to make the identity permutation the only permutation without any breakpoints, we extend our definition of permutations (and the symmetric group) by expanding permutations with one initial 0 and one trailing $n+1$. In the following we only consider such expanded permutations $\pi = (0\ \pi_1\ \ldots\ \pi_n\ n+1) \in S_n$ where $\pi_0 = 0$ and $\pi_{n+1} = n+1$. It is easy to see that for an expanded permutation π, $b(\pi) = 0$ if and only if $\pi = id$.

We call a reversal a *k-reversal* if it removes k breakpoints. It is immediate that $k \in [-2, 2]$ as a reversal can at most add or eliminate 2 breakpoints.

Corollary 2.1.1. *Using the previous two observations, Kececioglu et al. found a first fundamental lower bound to the reversal distance of a permutation $\pi \in S_n$ [KS95].*

$$d(\pi) \geq \left\lceil \frac{b(\pi)}{2} \right\rceil.$$

Based on this first lower bound, Kececioglu et al. proposed a simple 2-approximation running in $O(n^2)$ time [KS95].

3 Breakpoint graph

We now try to find a tighter lower bound by introducing the breakpoint graph of a permutation that was first defined by Bafna et al.

Definition 3.1. The *breakpoint graph* $G(\pi)$ has as vertices the elements of the permutation π, $[0, n+1]$. We connect two vertices π_i and π_j with a *red* edge if they form a breakpoint in π. We connect two vertices π_i and π_j representing non-consecutive elements in π (that is $|\pi_i^{-1} - \pi_j^{-1}| \neq 1$) with a *blue* edge if $\pi_i \sim \pi_j$.

The red edges in the breakpoint graph represent the elements of π that have to be split apart and the blue edges represent the elements of π that have to be brought together in the process of sorting the permutation π (the missing adjacencies). Figure 2 shows the breakpoint graph of the permutation (0 7 5 6 3 2 4 1 8).

An *alternating cycle decomposition* of $G(\pi)$ is a set of edge-disjoint alternating cycles, such that every edge of $G(\pi)$ is contained in exactly one cycle of the set. We observe that for every breakpoint, a vertex in $G(\pi)$ must have a missing adjacency. Conversely, for every missing adjacency a vertex in $G(\pi)$ must form a breakpoint. Therefore, each vertex in our breakpoint graph has an equal number of incident red and blue edges.

Corollary 3.1.1. *Hence, $G(\pi)$ can be decomposed into edge-disjoint alternating cycles.*

The main result we prove in this section relates the maximum number of alternating cycles of any cycle decomposition of $G(\pi)$ to the reversal distance of π.

First, we introduce some terminology. An alternating cycle in $G(\pi)$ is a *k-cycle* if it has k constituting red and k constituting blue edges. We say a reversal *acts on* two red edges of $G(\pi)$ if those two red edges represent the breakpoints that are split apart by the reversal. Now, an alternating cycle in $G(\pi)$ is *oriented* if there is a 1- or 2-reversal acting on two constituting red edges. An alternating cycle of $G(\pi)$ is *unoriented* if it is not oriented.

To give a more visual means of deducing whether an alternating cycle of $G(\pi)$ is oriented, we assign to each red edge (π_i, π_{i+1}) an orientation from π_i to π_{i+1}, orienting red edges of $G(\pi)$ from the endpoint which appears first in π to the endpoint which appears second. Then, we call an alternating cycle of $G(\pi)$ *directed* with respect to π if it is possible to walk along the whole cycle traversing each red edge in the direction of its orientation. An alternating cycle of $G(\pi)$ is called *undirected* if it is not directed. For example, in Figure 2 the alternating cycle $(\{0,7\},\{6,7\},\{3,6\},\{3,4\},\{1,4\},\{0,1\})$ is directed with respect to (0 7 5 6 3 2 4 1 8), whereas the alternating cycle $(\{5,7\},\{4,5\},\{2,4\},\{1,2\},\{1,8\},\{7,8\})$ is undirected with respect to (0 7 5 6 3 2 4 1 8). Caprara argues that an alternating cycle of $G(\pi)$ is oriented if and only if it is undirected [Cap97].

Lemma 3.2. *An oriented alternating cycle is a 2-cycle if and only if the corresponding reversal acting on both red edges is a 2-reversal.*

Proof. Let $i \sim i'$ and $j \sim j'$. We observe that the 2-cycles shown in Figure 3 are in fact the only types of 2-cycles (with the exception of the symmetric case of an oriented 2-cycle). Given an oriented 2-cycle, we know that the corresponding reversal acting on its two constituting red edges brings j' next to j and i next to i', removing two breakpoints.

On the other hand, given a 2-reversal ρ, we know that ρ has to act on two red edges of $G(\pi)$. Identify with k and l the vertices incident to the first red edge and with m and n the vertices incident to the second red edge. Now, because ρ removes both red edges from the breakpoint graph, we know that either $k \sim m$ and $l \sim n$ or $k \sim n$ and $l \sim m$. Thus, forming a 2-cycle in $G(\pi)$. □

A similar lemma is proven by Christie which we use as a corollary of this lemma in Section 4.1 [Chr98].

(a) Oriented 2-cycle (b) Unoriented 2-cycle

Figure 3: 2-cycles

(a) before reversal (b) after reversal

Figure 4: Case (i)

Let $c(\pi)$ denote the maximum number of alternating cycles in any alternating cycle decomposition of $G(\pi)$. Bafna et al. prove a fundamental theorem relating $b(\pi)$ with $c(\pi)$ [BP96].

Theorem 3.3. *Let $\pi, \rho \in S_n$ and ρ be a reversal. Then*

$$b(\pi) - b(\pi \circ \rho) + c(\pi \circ \rho) - c(\pi) \leq 1.$$

Proof. We consider each case $b(\pi) - b(\pi \circ \rho) \in [-2, 2]$ separately.

1. Let $b(\pi) - b(\pi \circ \rho) = 2$. Lemma 3.2 implies that a 2-reversal ρ removes a 2-cycle from every alternating cycle decomposition of $G(\pi)$. Therefore, $c(\pi \circ \rho) - c(\pi) \leq -1$.

2. Let $b(\pi) - b(\pi \circ \rho) = 1$. A 1-reversal either creates one new adjacency (i) or creates two new adjacencies and simultaneously destroys one adjacency (ii).

   (i) Let $j \sim j'$. Then Figure 4 illustrates that every alternating cycle in $G(\pi \circ \rho)$ containing the edge $\{i, k\}$ induces an alternating cycle in $G(\pi)$ containing the edges $\{j, i\}$, $\{j, j'\}$, and $\{j', k\}$. Therefore, every alternating cycle decomposition of $G(\pi \circ \rho)$ into c cycles induces an alternating cycle decomposition of $G(\pi)$ into c cycles.

   (ii) Let $i \sim j$, $i \sim k$ and $j \sim l$. Then Figure 5 illustrates that every alternating cycle in $G(\pi \circ \rho)$ containing the edge $\{i, j\}$ induces an alternating cycle in $G(\pi)$ containing the edges $\{i, k\}$, $\{k, l\}$, and $\{j, l\}$. Therefore, every alternating cycle decomposition of $G(\pi \circ \rho)$ into c cycles induces an alternating cycle decomposition of $G(\pi)$ into c cycles.

   Therefore, for both subcases, $c(\pi \circ \rho) - c(\pi) \leq 0$ holds.

(a) before reversal (b) after reversal

Figure 5: Case (ii)

Proof for other cases similar. $\square$

Using Theorem 3.3, Bafna et al. give a lower bound of the reversal distance of a permutation π in terms of $b(\pi)$ and $c(\pi)$ [BP96].

Theorem 3.4. *Let $\pi \in S_n$. Then*

$$d(\pi) \geq b(\pi) - c(\pi).$$

Proof. Let $\pi_t = \pi, \pi_0 = id$ and $\rho_1, \ldots, \rho_t$ a shortest sequence of reversals from π_t to π_0. Then

$$\begin{aligned} d(\pi_i) &= d(\pi_{i-1}) + 1 \\ &\overset{(3.3)}{\geq} d(\pi_{i-1}) + b(\pi_i) - b(\pi_{i-1}) + c(\pi_{i-1}) - c(\pi_i) \\ \iff d(\pi_i) - (b(\pi_i) - c(\pi_i)) &\geq d(\pi_{i-1}) - (b(\pi_{i-1}) - c(\pi_{i-1})) \\ &\geq d(\pi_0) - (b(\pi_0) - c(\pi_0)) = 0 \end{aligned}$$

Setting $i = t$, proves the theorem. $\square$

Using this lower bound, Bafna et al. already describe a 7/4-approximation that is not considered here. Christie formulated a lower bound in terms of the number of 2-cycles in a maximum alternating cycle decomposition [Chr98].

Lemma 3.5. *Let $\mathcal{C}$ be a maximum alternating cycle decomposition of $G(\pi)$. Let $c_2(\pi)$ be the minimum number of alternating 2-cycles in any such $\mathcal{C}$ and let $c_{3*}(\pi) = c(\pi) - c_2(\pi)$ be the remaining alternating cycles in $\mathcal{C}$. Then*

$$c_{3*}(\pi) \leq \frac{1}{3}(b(\pi) - 2c_2(\pi)).$$

Proof. $b(\pi)$ is the number of red edges in $G(\pi)$ and $c_2(\pi)$ is the number of red edges in 2-cycles of $G(\pi)$. Therefore, $b(\pi) - 2c_2(\pi)$ is the number of red edges in $G(\pi)$ not occurring in 2-cycles. $3c_{3*}(\pi)$ is a lower bound to the number of red edges in $G(\pi)$ not occurring in 2-cycles as every such cycle has at least three constituting red edges, proving the lemma. $\square$

Theorem 3.6. *Let $\pi \in S_n$. Then*

$$d(\pi) \geq \frac{2}{3}b(\pi) - \frac{1}{3}c_2(\pi).$$

Proof. Follows immediately from Theorem 3.4 and Lemma 3.5. $\square$

4 3/2-approximation

In this section we discuss the 3/2-approximation proposed by Christie in detail [Chr98]. First, we describe how a sorting sequence of reversals can be obtained from an alternating cycle decomposition of $G(\pi)$ (Section 4.1). In Section 4.2 we then describe how to find such a cycle decomposition. This then allows us to prove the approximation bound (Section 4.3). Lastly, we describe the full algorithm and analyze its complexity (Section 4.4).

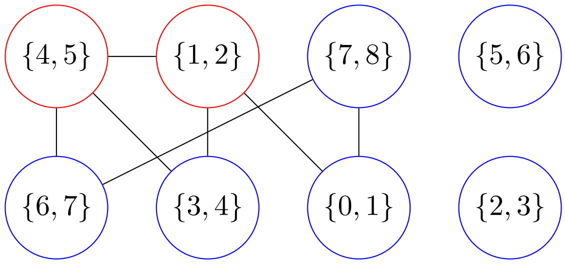

Figure 6: Reversal graph

4.1 Reversal graph

To obtain a sorting sequence of reversals for a permutation π given an alternating cycle decomposition $\mathcal{C}$ of its breakpoint graph $G(\pi)$, Christie defines a new graph.

Definition 4.1. Given an alternating cycle decomposition $\mathcal{C}$ of $G(\pi)$, the *reversal graph* $R(\mathcal{C})$ is constructed as follows.

For each adjacency in π we introduce an isolated blue vertex. For each m-cycle in $\mathcal{C}$ we introduce m vertices, each representing a constituting blue edge. Each vertex u represents the reversal $\rho(u)$ that acts on the two red edges incident to the blue edge. In the case of a 2-cycle, two vertices in the reversal graph represent the same reversal. A vertex is colored red if the represented reversal is a 1- or 2-reversal. Otherwise a vertex is colored blue.

Let u be a vertex in $R(\mathcal{C})$. Define $l_b(u)$ and $r_b(u)$ to be the positions in π of the leftmost and rightmost elements, respectively, that are incident to the blue edge in $G(\pi)$ represented by u. Similarly, $l_r(u)$ and $r_r(u)$ are defined to be the positions in π of the leftmost and rightmost red edges that are adjacent to the blue edge represented by u. The position of a red edge is the position of its leftmost incident vertex. Two vertices u and v are connected with an edge if $l_b(u) < l_b(v) < r_b(u) < r_b(v)$ or $l_b(v) < l_b(u) < r_b(v) < r_b(u)$ or $l_r(u) < l_r(v) < r_r(u) < r_r(v)$ or $l_r(v) < l_r(u) < r_r(v) < r_r(u)$.

Two cycles C and D *interleave* if the reversal graph contains vertices u and v arising from C and D respectively, such that $\{u, v\}$ is an edge in $R(\mathcal{C})$. Blue edges *interleave* in the breakpoint graph if the vertices representing them are adjacent in $R(\mathcal{C})$.

Consider the alternating cycle decomposition

$$\begin{aligned}\mathcal{C} = \{&(\{0,7\}, \{6,7\}, \{3,6\}, \{3,4\}, \{1,4\}, \{0,1\}),\\ &(\{5,7\}, \{4,5\}, \{2,4\}, \{1,2\}, \{1,8\}, \{7,8\})\}\end{aligned}$$

of the breakpoint graph of (0 7 5 6 3 2 4 1 8). The corresponding reversal graph is given in Figure 6.

We observe that the only alternating cycle decomposition of $G(id)$ is $\mathcal{C} = \emptyset$. Therefore, the only reversal graph of the identity permutation $R(\emptyset)$ consists of $n + 1$ isolated blue vertices. In the remainder of this section, our goal is to describe a way of transforming the reversal graph of a given alternating cycle decomposition into the reversal graph only consisting of isolated blue vertices with as few reversals as possible. For that purpose, we first examine the effects of single reversals on the reversal graph.

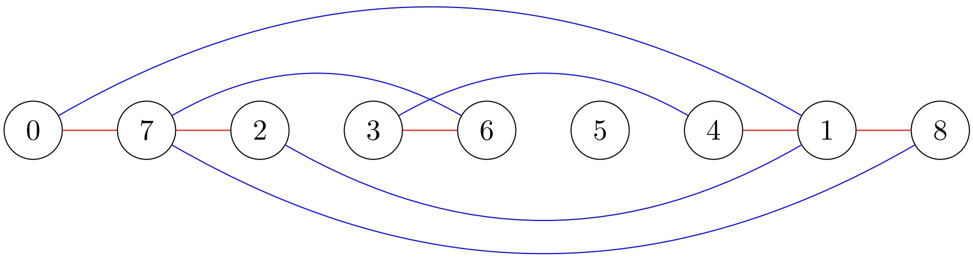

Figure 7: Resulting breakpoint graph

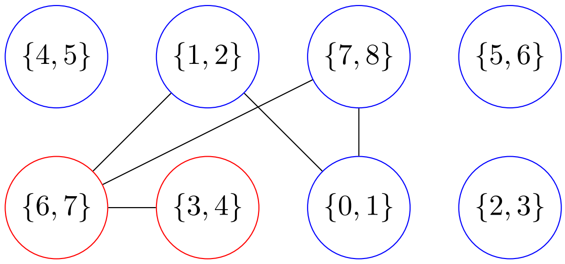

Figure 8: Resulting reversal graph

Let u be a vertex of $R(\mathcal{C})$ representing reversal $\rho(u)$. Denote by $\mathcal{C}_u$ the alternating cycle decomposition of $G(\pi \circ \rho(u))$ that is obtained from $\mathcal{C}$ by removing one 2-cycle or shortening one of its cycles by one red edge and one blue edge.

The following lemma is given without proof.

Lemma 4.2. *$R(\mathcal{C}_u)$ can be derived from $R(\mathcal{C})$ by making the following changes to $R(\mathcal{C})$:*

1. *flip the color of every vertex adjacent to u;*
2. *flip the adjacency of every pair of vertices adjacent to u; and*
3. *if u is a red vertex, turn it into an isolated blue vertex.*

Applying the reversal represented by vertex $\{4, 5\}$ in Figure 6 results in the breakpoint and reversal graphs shown in Figures 7 and 8 respectively.

Corollary 4.2.1. *A reversal $\rho(u)$ affects only vertices that are in the same connected component as u.*

We now consider a couple of lemmas that will later help us in proving the approximation bound.

Lemma 4.3. *$R(\mathcal{C})$ contains no isolated blue vertices arising from alternating cycles in $\mathcal{C}$.*

Proof. Omitted [Chr98]. □

The following lemma is a simple consequence of the above lemma.

Lemma 4.4. *Vertices arising from an unoriented* 2*-cycle of* $\mathcal{C}$ *must be in a connected component of* $R(\mathcal{C})$ *with vertices arising from at least one more alternating cycle of* $\mathcal{C}$*.*

Proof. Let u and v represent the blue edges of an unoriented 2-cycle of $\mathcal{C}$. Then its two constituting blue edges do not interleave. Hence, $R(\mathcal{C})$ does not connect u and v with an edge. If its constituting blue edges do not interleave with blue edges of any other alternating cycle of $\mathcal{C}$, then u and v would be isolated blue vertices. $\lightning$ $\square$

Lemma 4.5. *All vertices arising from the same alternating cycle in* $\mathcal{C}$ *are in the same connected component of* $R(\mathcal{C})$*.*

Proof. Omitted [Chr98]. $\square$

We call a connected component of $R(\mathcal{C})$ *oriented* if it contains a red vertex or if it consists solely of an isolated blue vertex.

Let A be a connected component of $R(\mathcal{C})$. We denote by A_u the subgraph of $R(\mathcal{C}_u)$ that contains all the vertices of A. A sequence of reversals that turns the connected component A of $R(\mathcal{C})$ into an isolated blue vertex, is called an *elemination sequence* of A.

Lemma 4.6. *If a connected component* A *of* $R(\mathcal{C})$ *is oriented and not an isolated blue vertex, it contains a red vertex* u *such that* A_u *is still oriented.*

Proof. Omitted [Chr98]. $\square$

With the above lemma we are now able to find an elimination sequence for the connected components of the reversal graph.

Lemma 4.7. *Let* A *be an oriented connected component of* $R(\mathcal{C})$ *arising from* k *different alternating cycles of* $G(\pi)$*. Then there is an elimination sequence for* A *that contains* k 2*-reversals with all the other reversals being* 1*-reversals.*

Proof. As A is oriented, we can use Lemma 4.6 to repeatedly find a red vertex u in A such that A_u is still oriented and apply $\rho(u)$. By Lemma 4.5, we know that every vertex in $R(\mathcal{C})$ arising from an alternating cycle of $G(\pi)$ is contained within the same connected component. Therefore, every alternating cycle represented by vertices in A eventually reduces to a 2-cycle. Moreover, by Lemma 3.2, $\rho(u)$ is a 2-reversal iff u arises from an oriented 2-cycle. $\square$

Lemma 4.8. *Let* A *be an unoriented connected component of* $R(\mathcal{C})$ *arising from* k *different alternating cycles of* $G(\pi)$*. Then there is an elimination sequence for* A *that contains one* 0*-reversal,* k 2*-reversals, with all the other reversals being* 1*-reversals.*

Proof. Let u be any blue vertex of A, then $\rho(u)$ is a 0-reversal. By Lemma 4.2, A_u is oriented. Now, apply Lemma 4.7 to obtain an elimination sequence for A_u. $\square$

Using the elimination sequences we found for the connected components of $R(\mathcal{C})$ we are finally able to give an upper bound to the number of reversals we need to sort π.

Theorem 4.9. *Let* $\pi \in S_n$ *and* $\mathcal{C}$ *be an alternating cycle decomposition of* $G(\pi)$*. Then*

$$d(\pi) \leq b(\pi) - |\mathcal{C}| + u(\mathcal{C})$$

where $u(\mathcal{C})$ *is the number of unoriented components in* $R(\mathcal{C})$*.*

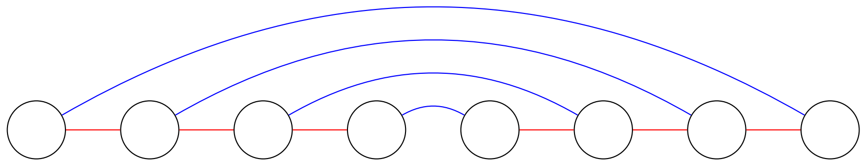

Figure 9: Ladder

4.2 Cycle decomposition

In this section our goal is to find an alternating cycle decomposition of $G(\pi)$. Motivated by the lower bound to the reversal distance in terms of 2-cycles, we try to maximize the number of 2-cycles in our cycle decomposition [Chr98].

To find a large number of 2-cycles in $G(\pi)$, Christie constructs a new graph.

Definition 4.10. Given the permutation π, we construct the *matching graph* $F(\pi)$ with vertices representing red edges in $G(\pi)$ and vertices u, v being adjacent if they form a 2-cycle in $G(\pi)$.

We now observe that 2-cycles of $G(\pi)$ are represented by edges in $F(\pi)$. We then find a maximum cardinality matching M of $F(\pi)$. We call a 2-cycle of $G(\pi)$ *selected* if its corresponding edge of $F(\pi)$ is part of the matching M. Note, that we have not yet found the set of edge-disjoint 2-cycles that we need to construct an alternating cycle decomposition. This is because the matching ensures that all selected 2-cycles are edge-disjoint on their red edges, but does not ensure that they are also edge-disjoint on their blue edges. To find a large set of edge-disjoint 2-cycles, Christie introduces another graph.

Definition 4.11. Given a matching M, we construct the *ladder graph* $L(M)$. $L(M)$ has a vertex for every 2-cycle selected in M. Vertices form connected components when the represented 2-cycles share a blue edge in $G(\pi)$. Such connected components of $L(M)$ are called *ladders*. Figure 9 shows a part of a breakpoint graph with 2-cycles that form a ladder.

A selected 2-cycle is called *independent* if it is not part of a ladder. Otherwise it is called a *ladder 2-cycle*.

As each 2-cycle can share at most two blue edges with other 2-cycles, the connected components of $L(M)$ are simple paths. Therefore, the set of 2-cycles consisting of every other ladder 2-cycle is edge-disjoint on all edges, resulting in the following theorem.

Theorem 4.12. *Given a maximum cardinality matching M of $F(\pi)$ it is possible to find an alternating cycle decomposition $\mathcal{C}$ of $G(\pi)$ that contains at least $\lceil \frac{y}{2} \rceil$ ladder 2-cycles and z independent 2-cycles.*

Proof. Let $\mathcal{C}$ contain all independent 2-cycles from $L(M)$ and every other alternate cycle of each ladder. Obtain the rest of $\mathcal{C}$ by adding any alternating cycle decomposition of the remaining unused edges of $G(\pi)$. □

Another lemma, Christie shows, is given without proof.

Lemma 4.13. *Let C be an unoriented ladder 2-cycle in $\mathcal{C}$. Then the vertices of $R(\mathcal{C})$ that represent C are part of a connected component that contains vertices representing an alternating cycle that is not a selected 2-cycle.*

4.3 Approximation bound

By Theorem 3.6, an algorithm that finds a sorting sequence of reversals of at most length $b(\pi) - \frac{1}{2}c_2(\pi)$ achieves an approximation bound of $\frac{3}{2}$.

Theorem 4.14. *Let $\pi \in S_n$. We are able to find a sorting sequence of at most $b(\pi) - \frac{1}{2}c_2(\pi)$ reversals.*

Proof. Using Theorem 4.12, first find an alternating cycle decomposition $\mathcal{C}$ of $G(\pi)$ with at least $\lceil \frac{y}{2} \rceil$ ladder 2-cycles and z independent 2-cycles.

Let k be the number of 2-cycles in oriented connected components of $R(\mathcal{C})$. Let u be the number of unoriented connected components in $R(\mathcal{C})$ that include l selected 2-cycles and that, by Lemma 4.13, contain vertices representing remaining unselected 2-cycles. Let v be the number of remaining unoriented connected components consisting only of vertices representing m independent selected 2-cycles.

Note that $\mathcal{C}$ has at least $k + l + u + m$ alternating cycles and $R(\mathcal{C})$ has $u + v$ unoriented connected components.

By Theorem 4.9, we can now sort π using at least $k+l+u+m$ 2-reversals and only $u+v$ 0-reversals. Therefore

$$\begin{aligned} d(\pi) &\leq b(\pi) - k - l - u - m + u + v \\ &= b(\pi) - k - l - m + v. \end{aligned}$$

All that is left to show is $k + l + m - v \geq \frac{1}{2}c_2(\pi)$. We know:

1. $l+m$ is the number of selected 2-cycles in unoriented connected components of $R(\mathcal{C})$. k is the total number of 2-cycles in oriented connected components of $R(\mathcal{C})$. As $\lceil \frac{y}{2} \rceil + z$ is the number of selected 2-cycles in $\mathcal{C}$, $k + l + m \geq \lceil \frac{y}{2} \rceil + z$ holds.

2. As each connected components of $R(\mathcal{C})$ counted by v represents at least one unoriented 2-cycle, Lemma 4.4 implies that such a connected component must represent at least two alternating cycles in $\mathcal{C}$. As the connected component only represents independent 2-cycles, we know that it must represent at least two independent 2-cycles. As z is the number of independent 2-cycles, $v \leq \lfloor \frac{z}{2} \rfloor$ holds.

3. We also have that the selected 2-cycles in M are not necessarily edge-disjoint on their blue edges, yielding $|M| = y + z \geq c_2(\pi)$.

And therefore

$$\begin{aligned} k + l + m - v &\geq \left\lceil \frac{y}{2} \right\rceil + z - v && (1) \\ &\geq \left\lceil \frac{y}{2} \right\rceil + z - \left\lfloor \frac{z}{2} \right\rfloor && (2) \\ &= \left\lceil \frac{y}{2} \right\rceil + \left\lceil \frac{z}{2} \right\rceil \\ &\geq \frac{1}{2}c_2(\pi). && (3) \end{aligned}$$

□

Algorithm 1: Approximation algorithm [Chr98]

Input: permutation $\pi \in S_n$
construct the matching graph $F(\pi)$;
find a maximum cardinality matching M for this graph;
construct alternating cycle decomposition $\mathcal{C}$ using this matching;
construct the reversal graph $R(\mathcal{C})$;
find an elimination sequence for $R(\mathcal{C})$;

Figure 10

4.4 Algorithm

The approximation algorithm with a performance ratio of $\frac{3}{2}$, is outlined in Figure 10.

The matching graph can be constructed in $O(n)$ time. The maximum cardinality matching can be obtained from $F(\pi)$ using the algorithm of Micali and Vazirani in $O(|V||E|^{\frac{1}{2}})$ time [MV80] where $|V|$ and $|E|$ are the number of vertices and edges respectively. Therefore, it is possible to find such a matching in $O(n^{\frac{3}{2}})$ time. The alternating cycle decomposition can be obtained from the matching in $O(n)$ time. The reversal graph can be constructed in $O(n^2)$ time. The algorithm to find an elimination sequence of $R(\mathcal{C})$ is outlined in Figure 11 and runs in $O(n^4)$ time. So the overall time-complexity of the algorithm is $O(n^4)$ [Chr98].

Using results from Kaplan et al., the time-complexity of finding an elimination sequence and therefore the time-complexity of the entire algorithm can be improved to $O(n^2)$ [KST97].

Algorithm 2: Finding an elimination sequence [Chr98]

```
Input: reversal graph R(C)
for each unoriented connected component, apply a reversal represented by a blue
  vertex;
while not sorted π do
    u := first red vertex;
    found := false;
    while not found do
        apply ρ(u) to get C_u;
        generate R(C_u);
        count unoriented connected components in R(C_u);
        if there are no unoriented connected components then
            found := true;
        else
            u := next red vertex;
        end
    end
end
```

Figure 11

5 Conclusion

We have seen the fundamental relationship between the reversal distance of a permutation and its number of breakpoints as well as the maximum number of alternating cycles of a cycle decomposition of its breakpoint graph as was first discussed by Bafna et al. Exploiting this relationship, Christie described a 3/2-approximation we discussed in detail. A better but more complex approximation was obtained later by Berman et al.

An interesting property of the described 3/2-approximation is that when constructing the alternating cycle decomposition, any cycle decomposition of the edges that are not part of 2-cycles will suffice [Chr98].

References

[MV80] Silvio Micali and Vijay V. Vazirani. "An $O(|V|^{1/2}|E|)$ algoithm for finding maximum matching in general graphs". In: *21st Annual Symposium on Foundations of Computer Science (sfcs 1980)* (1980), pp. 17–27.

[Wat+82] G.A. Watterson et al. "The chromosome inversion problem". In: *Journal of Theoretical Biology* 99.1 (1982), pp. 1–7. DOI: `https://doi.org/10.1016/0022-5193(82)90384-8`.

[HP95] Sridhar Hannenhalli and Pavel Pevzner. "Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals". In: *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*. STOC '95. 1995, pp. 178–189. DOI: `10.1145/225058.225112`.

[KS95] J Kececioglu and D Sankoff. "Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement". In: *Algorithmica* 13.1 (1995), p. 180. DOI: `10.1007/BF01188586`.

[BP96] Vineet Bafna and Pavel A Pevzner. "Genome Rearrangements and Sorting by Reversals". In: *SIAM J. Comput.* 25.2 (1996), pp. 272–289. DOI: `10.1137/S0097539793250627`.

[Cap97] Alberto Caprara. "Sorting by Reversals is Difficult". In: *Proceedings of the First Annual International Conference on Computational Molecular Biology*. RECOMB '97. 1997, pp. 75–83. DOI: `10.1145/267521.267531`.

[KST97] Haim Kaplan, Ron Shamir, and Robert Tarjan. "Faster and simpler algorithm for sorting signed permutations by reversals". In: vol. 29. 1997, p. 163. DOI: `10.1137/S0097539798334207`.

[Chr98] David A Christie. "A 3/2-Approximation Algorithm for Sorting by Reversals". In: *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '98. 1998, pp. 244–252. ISBN: 0898714109.

[BK99] Piotr Berman and Marek Karpinski. "On Some Tighter Inapproximability Results (Extended Abstract)". In: *Automata, Languages and Programming*. Ed. by Jiří Wiedermann, Peter van Emde Boas, and Mogens Nielsen. 1999, pp. 200–209.

[BHK01] Piotr Berman, Sridhar Hannenhalli, and Marek Karpinski. "1.375-Approximation Algorithm for Sorting by Reversals". In: *Electronic Colloquium on Computational Complexity (ECCC)* 8 (Jan. 2001).