An Approximation Algorithm for Sorting by Reversals and Transpositions

Atif Rahman, Swakkhar Shatabda and Masud Hasan

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)
Dhaka-1000, Bangladesh
atif.bd@gmail.com, swakkhar17@yahoo.com, masudhasan@cse.buet.ac.bd

Abstract. Genome rearrangement algorithms are powerful tools to analyze gene orders in molecular evolution. Analysis of genomes evolving by reversals and transpositions leads to a combinatorial problem of sorting by reversals and transpositions, the problem of finding a shortest sequence of reversals and transpositions that sorts one genome into the other. In this paper we present a $(4 - \frac{2}{k})$-approximation algorithm for sorting by reversals and transpositions for unsigned permutations where k is the approximation ratio of the algorithm used for cycle decomposition. For the best known value of k our approximation ratio becomes $2.5909 + \delta$ for any $\delta > 0$. We also derive a lower bound on reversal and transposition distance of an unsigned permutation.

1 Introduction

The study of evolutionary distance between two organisms using genomic data requires reconstruction of the sequence of evolutionary events that transform one genome into the other. Sequence comparison in computational molecular biology is a powerful tool for deriving evolutional and fundamental relationships among genes. But classical alignment algorithms take into account only local mutations (insertions, deletions and substitutions of nucleotides) and ignore global rearrangements (reversals, transpositions, translocations, fusions, fissions, etc. of long fragments) [1]. While studying genomes of different species, evidence was found that different species have essentially the same set of genes, but their order may differ among species [16, 20]. This suggests that global rearrangement events can be used to trace the evolutionary path among genomes.

In genome rearrangement problems the order of genes in two arbitrary organisms is represented by permutations. The basic task is, given two permutations, to find a shortest sequence of rearrangement operations that transforms one permutation into the other. Assuming that one of the permutations is the identity permutation, the problem is to find the shortest way of sorting a permutation using a given rearrangement operation (or set of operations). Two of the most

M. Kaykobad and Md. Saidur Rahman (Eds.): WALCOM 2007, pp. 97–108, 2007.

studied operations are reversal and transposition. A reversal reverses the order of the elements in a segment. In signed version of the problem each element has a sign and a reversal not only reverses the order of the elements in a segment but also flips their signs. A transposition is a rearrangement operation in which a segment is cut out of the permutation and pasted in a different location.

The problem of sorting by reversals has been studied extensively. For the signed version, Kececioglu and Sankoff [17] conjectured that the problem is NP-hard and gave a 2-approximation algorithm by exploiting the link between reversal distance and the number of breakpoints. Bafna and Pevzner [1] improved the ratio to 1.5 by introducing the breakpoint graph. Finally, Hannenhalli and Pevzner [13] settled the conjecture negative by giving an exact polynomial algorithm.

The unsigned version of the problem was shown to be NP-hard by Caprara [5]. Before that when the complexity was unknown, Kececioglu and Sankoff [17] gave a 2-approximation algorithm for the problem, and Bafna and Pevzner [1] presented a 1.75-approximation algorithm. Later the performance ratio was improved to 1.5 by Christie [7] and to 1.375 by Berman, Hannenhalli and Karpinski [3].

The problem of sorting by transpositions has also been studied by several authors. But unlike sorting by reversal, the complexity of sorting by transpositions is still open. It was first studied by Bafna and Pevzner [2], who devised a 1.5-approximation algorithm. The algorithm was simplified by Christie [8] and further by Hartman [14]. Recently, Elias and Hartman [9] gave a 1.375 approximation algorithm. Eriksson et al. [11] considered the problem in a different way. When the permutation size is n, they gave an algorithm that sorts the permutation by at most $2n/3$ transpositions, but their algorithm does not give any approximation guarantee.

A number of suggestions have been made to consider algorithms for sorting permutations by using more than one rearrangement operations (reversals, transpositions, etc.). Walter, Dias and Meidanis [21] provided a 2-approximation algorithm for signed permutation for sorting by reversals and transpositions. Gu et al. [12] gave a 2-approximation algorithm for sorting signed permutations by transpositions and transreversals (a transreversal combines a transposition and a reversal). Lin and Xue [19] improved this ratio to 1.75 by considering a third operation, called “revrev”, which reverses two contiguous segments. Hartman and Sharan [15] further improved it to 1.5.

Blanchette, Kunisawa, and Sankoff [4] worked on a variation of the problem and developed a computer program Derange II built on a greedy algorithm which attempts to minimize the weighted sum of the number of operations. Eriksen [10] provided a $(1+\epsilon)$-approximation algorithm for sorting signed, circular permutations where reversals are weighted 1 and transpositions and inverted transpositions (transreversals) are weighted 2.

There has been less progress in the problem of sorting permutations by using more than one rearrangement operations for unsigned permutations. Walter, Dias and Meidanis [21] gave a 3-approximation algorithm for sorting by reversals and transpositions for unsigned permutations. Our main result in this paper is to improve this ratio.

Biologists derive gene orders either by sequencing entire genomes or by comparative physical mapping. Physical maps usually do not provide information about directions of genes and so lead to representation of a genome as an unsigned permutation. Most of currently available data on gene orders are based on comparative physical maps [3]. So, an algorithm for sorting unsigned permutation by reversals and transpositions is quite useful.

In this paper we present an algorithm for sorting by reversals and transpositions for unsigned permutations with an approximation ratio $4 - \frac{2}{k}$, where k is the approximation ratio of the cycle decomposition algorithm used. The problem of *cycle decomposition* of a graph G is to decompose the edges of G into maximum number of edge-disjoint cycles and this problem is known to be NP-hard [5]. Using the best known approximation ratio for this problem, which is by Lin and Jiang [18], our algorithm has an approximation ratio $2.5909 + \delta$ for any $\delta > 0$.

In course of our algorithm we also give a lower bound on reversal and transposition distance of an unsigned permutation.

The rest of the paper is organized as follows. In section 2 we give the relevant definitions and derive a lower bound on reversal and transposition distance. In section 3 we present the approximation algorithm and derive the approximation ratio. We conclude by suggesting some future directions in section 4.

2 Preliminaries

Here, several of our definitions and lemmas have similarities with those in [1].

2.1 Sorting by Reversals and Transpositions

Let $\pi = [\pi_0 \pi_1\ \pi_2\ \ldots\ \pi_n \pi_{n+1}]$ be a permutation of $n+2$ distinct elements where $\pi_0 = 0$, $\pi_{n+1} = n+1$, and $1 \leq \pi_i \leq n$ for each $1 \leq i \leq n$ (the middle n elements of π are to be sorted). A *reversal* $\rho = \rho(i,j)$ for some $1 \leq i < j \leq n+1$ applied to π reverses the elements $\pi_i\ \ldots\ \pi_{j-1}$ and thus transforms π into permutation $\pi \cdot \rho = [\pi_0\ \ldots\ \pi_{i-1}\ \pi_{j-1}\ \ldots\ \pi_i\ \pi_j\ \ldots\ \pi_{n+1}]$. A *transposition* $\tau = \tau(i,j,k)$ for some $1 \leq i < j \leq n+1$ and some $1 \leq k \leq n+1$ such that $k \notin [i,j]$ cuts the elements $\pi_i\ \ldots\ \pi_{j-1}$ and pastes between π_{k-1} and π_k and thus transforms π into permutation $\pi \cdot \tau = [\pi_0\ \ldots\ \pi_{i-1}\ \pi_j\ \ldots\ \pi_{k-1}\ \pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1}\ \pi_k\ \ldots\ \pi_{n+1}]$ if $k > j$ or into permutation $\pi \cdot \tau = [\pi_0\ \ldots\ \pi_{k-1}\ \pi_i\ \pi_{i+1}\ \ldots\ \pi_{j-1}\ \pi_k\ \ldots\ \pi_{i-1}\ \pi_j\ \ldots\ \pi_{n+1}]$ if $k < i$.

An *identity permutation* I is a permutation such that $\pi_i = i$ for $0 \leq i \leq n+1$. The *reversal and transposition distance* $d(\pi)$ between π and I is the minimum number of operations such that $\pi \cdot o_1 \cdot o_2 \cdot\ \ldots\ \cdot o_{d(\pi)} = I$, where each operation o_i is a reversal ρ or a transposition τ. The problem of *sorting by reversals and transpositions* is to find a shortest sequence of reversals and transpositions that transforms a permutation π into the identity permeation

I, i.e. finding the distance $d(\pi)$. It is worth mentioning that the motivation of sorting by reversals and transpositions is not to sort but to find a shortest sequence of sorting operations.

2.2 Breakpoints and Cycle Decomposition Graph

Two elements π_i and π_{i+1} of π for all $0 \leq i \leq n$ are called *adjacent* if $|\pi_i - \pi_{i+1}| = 1$. Otherwise there is said to be a *breakpoint* between the two elements. We denote the total number of breakpoints in π by $b(\pi)$.

The *cycle decomposition graph* $G(\pi)$ is an undirected multigraph whose $n+2$ vertices are π_i for $0 \leq i \leq n+1$. $G(\pi)$ has $2(n+1)$ edges and they are of two types: *gray* and *black*. For each $0 \leq i \leq n$, the vertices π_i and π_{i+1} are joined by a black edge. For $0 \leq i, j \leq n+1$, there is a gray edge between π_i and π_j iff $\pi_i = \pi_j + 1$.

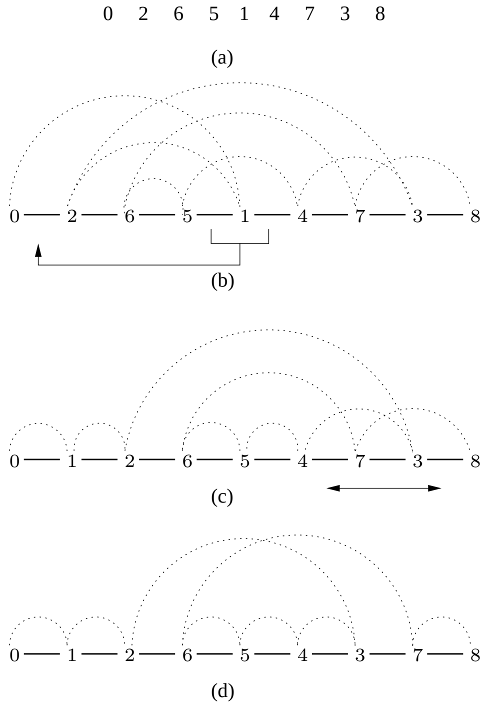


Fig. 1. (a) A permutation π, (b) Cycle decomposition graph $G(\pi)$ of π, (c) A transposition $\tau(4, 5, 1)$ on $G(\pi)$, (d) A reversal $\rho(6, 8)$ on $G(\pi \cdot \tau)$

For convenience of illustration, in this paper the vertices of $G(\pi)$ are drawn horizontally in order $\pi_0, \pi_1, \ldots, \pi_{n+1}$ from left to right. The black edges are usually shown by horizontal lines and the gray ones are shown by dotted arcs. See Fig. 1. We also use broken arcs to show chords having a single gray edge or an odd number of edges of alternating colors starting and ending with gray edges.

An *alternating cycle* of $G(\pi)$ is a cycle of size at least two in which the edges alternate colors. From now on we will use *cycle* to refer to an alternating cycle and *l-cycle* to refer to a cycle having l black edges.

The graph $G(\pi)$ can be completely decomposed into edge-disjoint cycles [7, 6, 18]. However, there may be many different such cycle decompositions. The maximum number of cycles in any cycle decomposition of $G(\pi)$ is denoted by $c(\pi)$. As already mentioned in Section 1, the problem of finding a maximum cycle decomposition is known to be NP-hard [5]. Among several approximation algorithms, Christie [7] gave a 1.5-approximation algorithm, Caprara and Rizzi [6] improved the ratio to $1.4348 + \epsilon$ for any $\epsilon > 0$, and, as the best one so far, Lin and Jiang [18] further improved it to $1.4193 + \epsilon$ for any $\epsilon > 0$.

2.3 A Lower Bound on Reversal and Transposition Distance

Consider a permutation π. The $2(n+1)$ edges of $G(\pi)$ can give at most $(n+1)$ cycles. We have the following crucial lemma whose obvious proof we omit.

Lemma 1. *$G(\pi)$ has $n+1$ cycles iff $\pi = I$.*

For a permutation π and an operation o, denote $\triangle_c(o) = c(\pi \cdot o) - c(\pi)$ as the change in the number of cycles due to operation o. We use *m-transposition* to refer to a transposition τ such that $\triangle_c(\tau) = m$, and *m-reversal* to denote a reversal ρ with $\triangle_c(\rho) = m$.

Lemma 2. $\triangle_c(\tau) \leq 2$.

Proof. A transposition $\tau(i, j, k)$ involves six vertices of $G(\pi)$ (π_{i-1}, π_i, π_{j-1}, π_j, π_{k-1}, π_k). It removes three black edges $((\pi_i, \pi_{i-1}), (\pi_j, \pi_{j-1})$ and $(\pi_k, \pi_{k-1}))$ and adds three new black edges $((\pi_j, \pi_{i-1}), (\pi_i, \pi_{k-1})$ and $(\pi_k, \pi_{j-1}))$, and all other edges are unaffected.

Three removed edges belong to either one, two or three cycles in a cycle decomposition of $G(\pi)$. Again, the added edges belong to either one, two or three cycles in a cycle decomposition of $G(\pi \cdot \tau)$. In the case when the removed edges belong to one cycle and the added edges belong to three cycles the number of cycles increases by two and in all other cases it is less than two. $\mathcal{Q.E.D.}$

We have a similar lemma for reversals.

Lemma 3. $\triangle_c(\rho) \leq 1$.

According to Lemma 1 the sequence of operations that sort a non-identity permutation π must increase the number of cycles from $c(\pi)$ to $n+1$. Since from Lemma 2 and Lemma 3 the maximum increase due to a single operation is two, a lower bound on $d(\pi)$ follows.

Theorem 1. $d(\pi) \geq \frac{n+1-c(\pi)}{2}$.

The lower bound can also be expressed in terms of breakpoints.

Lemma 4. *There exists a maximum cycle decomposition of $G(\pi)$ that contains every 1-cycle in the graph.*

Proof. Let $\mathcal{C}$ denote a maximum cycle decomposition of $G(\pi)$ that does not contain a 1-cycle on vertices π_i and π_{i+1}. The gray and black edges connecting π_i and π_{i+1} must belong to one or two cycles having at least two black edges in $\mathcal{C}$. The gray edge between π_i and π_{i+1} is preceded and succeeded by two black edges in the cycle containing the gray edge in $\mathcal{C}$ and, similarly, the black edge is preceded and succeeded by two gray edges.

We can construct a cycle decomposition $\mathcal{C}'$ from $\mathcal{C}$ by removing the gray and black edge connecting π_i and π_{i+1} to form a 1-cycle and merging the chords remaining after removal of the edges to form another cycle. The number of cycles in $\mathcal{C}'$ is no less than the number of cycles in $\mathcal{C}$ and so $\mathcal{C}'$ is a maximum cycle decomposition of $G(\pi)$. $\mathcal{Q.E.D.}$

Let $c_1(\pi)$ denote the number of 1-cycles and $c_{2^+}(\pi)$ denote the number of cycles having two or more black edges in a maximum cycle decomposition of $G(\pi)$ containing every 1-cycle. An alternative lower bound on reversal and transposition distance is given by the following theorem.

Theorem 2. $d(\pi) \geq \frac{b(\pi)-c_{2^+}(\pi)}{2}$.

Proof. According to Theorem 1, $d(\pi) \geq \frac{n+1-c(\pi)}{2}$. The total number of cycles $c(\pi)$ can be written as the sum of $c_1(\pi)$ and $c_{2^+}(\pi)$. So, $d(\pi) \geq \frac{n+1-c_1(\pi)-c_{2^+}(\pi)}{2}$. But the number of 1-cycles in $G(\pi)$ equals the number of adjacencies in π, and so $n+1-c_1(\pi) = b(\pi)$. Hence, $d(\pi) \geq \frac{b(\pi)-c_{2^+}(\pi)}{2}$. $\mathcal{Q.E.D.}$

3 Approximation Algorithm

The approximation algorithm sorts a permutation π by first constructing the cycle decomposition graph $G(\pi)$. The next step is to partition the edges of $G(\pi)$ into cycles. To do this first all 1-cycles are identified and their edges are removed. Next the remaining edges are partitioned into cycles having two or more black edges using the approximation algorithm of Lin and Jiang [18]. Let $c'(\pi)$ denote the total number of cycles produced in the cycle decomposition step. In each iteration our algorithm attempts to increase $c'(\pi)$ by applying a reversal or a transposition until $G(\pi)$ contains $n+1$ cycles (including the 1-cycles whose edges have been deleted.) If there exists no such operation then the algorithm applies a transposition that will allow a transposition in the next iteration to increase $c'(\pi)$ by two. The algorithm thus sorts a permutation π in at most $n+1-c'(\pi)$ iterations. Following is how the algorithm operates.

We number the black edges of $G(\pi)$ from 1 to $n+1$ by assigning label i to a black edge joining π_{i-1} and π_i. We say that a reversal $\rho(i,j)$ *acts* on edges i and j and a transposition $\tau(i,j,k)$ *acts* on edges i, j and k.

At any time we express an *l-cycle* C as the ordering $(i_1,\ldots,i_l)$ of its black edges along its boundary such that i_1 is the black edge with highest number and is traversed from right to left. We distinguish three different kinds of cycles: *semi-oriented*, *oriented* and *non-oriented*. (The concept of oriented and non-oriented cycles was introduced by Bafna and Pevzner [2] for directed cycle decomposition graphs.) A cycle C is *semi-oriented* if there exist two black edges such that along the boundary of C one is traversed left to right and the other one is traversed right to left (Fig. 2(a)); C is *non-oriented* if its black edges are in decreasing sequence (Fig. 2(b)); otherwise C is an *oriented* cycle (Fig. 2(c)). A gray edge in a cycle C is directed *left* if it is traversed from right to left in C and *right* otherwise. Observe that a non-oriented cycle $C=(i_1,\ldots,i_l)$ has exactly one right edge between black edges i_l and i_1 and an oriented cycle has at least three black edges (for two black edges it becomes non-oriented).

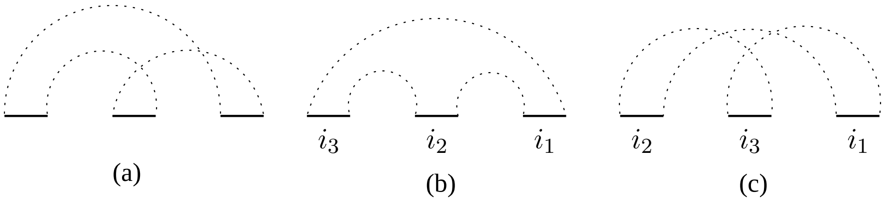


Fig. 2. (a) A semi-oriented, (b) a non-oriented, and (c) an oriented cycle.

Lemma 5. *If C is an oriented cycle, then there exists a 2-transposition acting on C.*

Proof. Let $C=(i_1,\ldots,i_l)$ be an oriented cycle. We find an index $3\le t\le l$ such that $i_t > i_{t-1}$ and apply a transposition $\tau(i_{t-1},i_t,i_1)$ on C creating a 1-cycle (on vertices $\pi_{i_{t-1}-1}$ and π_{i_t}) and some other cycles (Fig. 3). Therefore, τ is a 2-transposition. $\mathcal{Q.E.D.}$

Lemma 6. *If $G(\pi)$ has only non-oriented cycles, then there exists a 0-transposition τ that creates an oriented cycle in $G(\pi\cdot\tau)$.*

Proof. Let $C=(i_1,\ldots,i_l)$ be a non-oriented cycle. Let p be the position of the maximum element of π in between $[i_2-1,i_1]$. Note that p must exist because otherwise there would be a black edge between vertices π_{i_1-1} and π_{i_2} and the gray edge between the vertices would then be part of a 1-cycle. Let s be a position of π_p+1 in π. (We choose the cycle C in such a way that $s>i_1$.) If the gray edge between π_p and π_s is a *right* edge, we consider a transposition $\tau(p+1,s,i_2)$

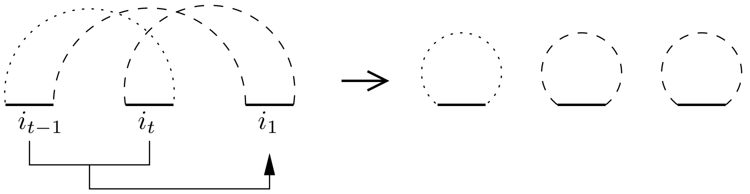


Fig. 3. A 2-transposition acting on an oriented cycle.

(Fig. 4(a)). Otherwise, we consider a transposition $\tau(p, i_1, s+1)$ (Fig. 4(b)). In both cases the edges removed by τ belong to two different cycles and the edges added belong to a 1-cycle and another cycle C' in $G(\pi \cdot \tau)$. Therefore, τ is a 0-transposition. Again, in both cases C' contains two right edges, and therefore, C' is an oriented cycle.

$\mathcal{Q.E.D.}$

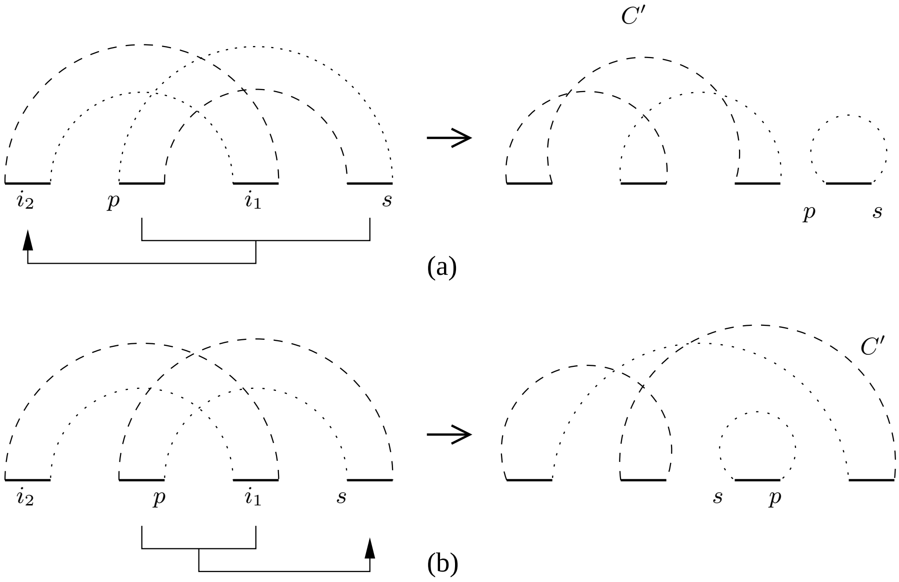
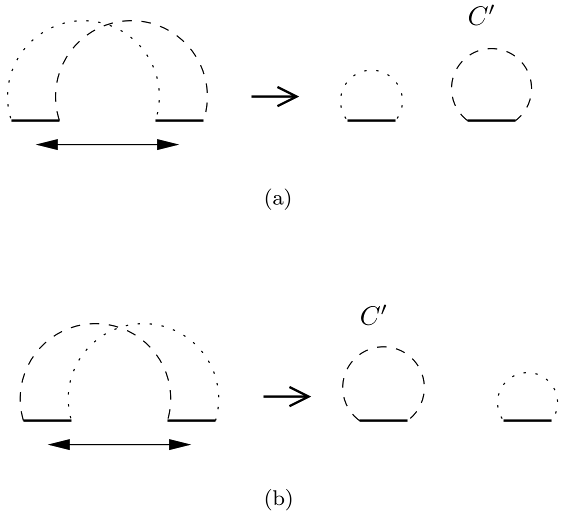


Fig. 4. A 0-transposition on a non-oriented cycle.

Lemma 7. *If C is a semi-oriented cycle, then there exists a 1-reversal acting on C.*

Proof. Consider a semi-oriented cycle $C = (i_1, \ldots, i_l)$. Let t $(1 \leq t \leq l)$ be an index such that the edges i_{t-1} (if $t = 1$ then $t - 1 = l$) and i_t are traversed in opposite directions. We consider a reversal ρ on edges i_{t-1} and i_t (Fig. 5). The removed edges belong to same cycle and the edges added belong to a 1-cycle and another cycle C' in $G(\pi \cdot \rho)$. Therefore, ρ is a 1-reversal. $\mathcal{Q}.\mathcal{E}.\mathcal{D}.$



(a)

(b)

Fig. 5. A 1-reversal acting on a semi-oriented cycle.

Lemmas 5, 6 and 7 imply the following corollary and the Algorithm RTSort summarizes our approximation algorithm. It is easy to verify that RTSort runs in polynomial time.

Corollary 1. *Any permutation π can be sorted in $n + 1 - c'(\pi)$ operations.*

3.1 Approximation Ratio

In section 2 we established the following lower bound,

$$d(\pi) \geq \frac{n + 1 - c(\pi)}{2}.$$

From Corollary 1, our algorithm sorts a permutation in at most $n+1-c'(\pi)$ steps. So we get an approximation ratio,

$$r = \frac{2(n + 1 - c'(\pi))}{n + 1 - c(\pi)},$$

where the value of $c'(\pi)$ depends on the algorithm used for cycle decomposition.

Remember that $G(\pi)$ has $b(\pi)$ black edges that are not elements of 1-cycles. If a maximum cycle decomposition of $G(\pi)$ contains $c_{2^+}(\pi)$ cycles with two or

Algorithm 1 RTSort(π)

```
Construct undirected cycle decomposition graph $G(\pi)$ of $\pi$
Identify all 1-cycles and remove their edges
Decompose $G(\pi)$ into edge disjoint cycles (by [18])
while $G(\pi)$ does not have $n+1$ cycles do
  if $G(\pi)$ has an oriented cycle then
    Apply a 2-transposition (Lemma 5)
  else if $G(\pi)$ has a semi-oriented cycle then
    Apply a 1-reversal (Lemma 7)
  else
    Apply a 0–transposition (Lemma 6)
  end if
  $G(\pi) \leftarrow G(\pi \cdot o)$
end while
```

more black edges, then an algorithm with approximation ratio k on $G(\pi)$ will decompose $b(\pi)$ black edges into at least $\frac{c_{2^+}(\pi)}{k}$ l-cycles with $l \geq 2$. So,

$$\begin{aligned} r &\leq \frac{2(n+1-c_1(\pi)-c_{2^+}(\pi)/k)}{n+1-c_1(\pi)-c_{2^+}(\pi)} \\ &= \frac{2b(\pi)-2c_{2^+}(\pi)/k}{b(\pi)-c_{2^+}(\pi)} \\ &= \frac{2b(\pi)-2c_{2^+}(\pi)+(2-2/k)c_{2^+}(\pi)}{b(\pi)-c_{2^+}(\pi)} \\ &= 2+\frac{(2-2/k)c_{2^+}(\pi)}{b(\pi)-c_{2^+}(\pi)}. \end{aligned}$$

The second term on the right hand side decreases with increase of $b(\pi)$. But each of the $c_{2^+}(\pi)$ cycles contains at least two black edges. So, $b(\pi) \geq 2c_{2^+}(\pi)$. Therefore,

$$\begin{aligned} r &\leq 2+\frac{(2-2/k)c_{2^+}(\pi)}{2c_{2^+}(\pi)-c_{2^+}(\pi)} \\ &= 2+(2-\frac{2}{k}) \\ &= 4-(\frac{2}{k}). \end{aligned}$$

Putting $k = 1.4193+\epsilon$, where $\epsilon > 0$, from [18] in the above equation, we get

$$\begin{aligned} r &\leq 4-\frac{2}{1.4193+\epsilon} \\ &= 4-\frac{2+2\epsilon/1.4193-2\epsilon/1.4193}{1.4193+\epsilon} \\ &= 4-\frac{2(1+\epsilon/1.4193)}{1.4193(1+\epsilon/1.4193)}+\frac{2\epsilon}{1.4193(1.4193+\epsilon)} \\ &= 4-\frac{2}{1.4193}+\delta \\ &= 2.5909+\delta. \end{aligned}$$

The algorithm thus has an approximation ratio $2.5909 + \delta$ for any $\delta > 0$.

Theorem 3. *The algorithm RTSort is a* $(4 - \frac{2}{k})$*-approximation algorithm for sorting by reversal and transposition for unsigned permutation, where k is the approximation ratio for the cycle decomposition of a graph. With the best known value of* $k = 1.4193 + \epsilon$*, where* $\epsilon > 0$ *[18], our algorithm gives the approximation ratio of* $2.5909 + \delta$*, for any* $\delta > 0$*.*

4 Conclusion

In this paper we presented a $(4-\frac{2}{k})$-approximation algorithm for sorting unsigned permutations by reversals and transpositions where k is the approximation ratio of cycle decomposition algorithm used. As cycle decomposition approaches optimality the approximation ratio of our algorithm approaches two. In future the performance ratio may be improved and other rearrangement operations such as translocation, fission, fusion may be considered. The ultimate goal would be to combine merits of both sequence alignment and genome rearrangement in a single algorithm. In a variation of the problem each rearrangement operation could be weighted according to the likelihood of the operation in molecular evolution. The goal would be to find a sequence of sorting operations such that the total cost is minimized.

References

1. V. Bafna and P. Pevzner. Genome rearrangements and sorting by reversals. *Proc. of 34th IEEE FOCS*, pages 148–157, 1993. Also in SIAM Journal on Computing 25:272-289,1996.
2. V. Bafna and P.A. Pevzner. Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240, May 1998.
3. P. Berman, S. Hannenhalli, and M. Karpinski. 1.375-approximation algorithm for sorting by reversals. *Proc. of 10th European Symposium on Algorithms (ESA'02)*, 2461:200–210, 2002.
4. M. Blanchette, T. Kunisawa, and D. Sankoff. Parametric genome rearrangement. *Gene*, 172:GC11–17, 1996.
5. A. Caprara. Sorting by reversals is difficult. *Proc. of 1st ACM RECOMB*, pages 75–83, 1997.
6. A. Caprara and R. Rizzi. Improved approximation for breakpoint graph decomposition and sorting by reversals. *Journal of Combinatorial Optimization*, 6(2):157–182, 2002.
7. D.A. Christie. A 3/2 approximation algorithm for sorting by reversals. *Proc. of 9th ACM-SIAM SODA*, pages 244–252, 1998.
8. D.A. Christie. *Genome Rearrangement Problems.* PhD thesis, University of Glasgow, 1999.
9. I. Elias and T. Hartman. A 1.375-approximation algorithm for sorting by transposition. *Proc. of the 5th International Workshop on Algorithms in Bioinformatics (WABI'05)*, 3692:204–214, October 2005.

10. N. Eriksen. (1+ϵ)-approximation of sorting by reversals and transpositions. *Theoretical Computer Science*, 289(1):517–529, 2002.
11. H. Eriksson, K. Eriksson, J. Karlander, L. Svensson, and J. Wastlund. Sorting a bridge hand. *Discrete Mathematics*, 241(1-3):289–300, 2001.
12. Q.P. Gu, S. Peng, and H. Sudborough. A 2-approximation algorithm for genome rearrangements by reversals and transpositions. *Theoretical Computer Science*, 210(2):327–339, 1999.
13. S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip. *Proc. of 27th ACM STOC*, pages 178–189, 1995.
14. T. Hartman. A simpler 1.5-approximation algorithm for sorting by transpositions. *Combinatorial Pattern Matching (CPM '03)*, 2676:156–169, 2003.
15. T. Hartman and R. Sharan. A 1.5-approximation algorithm for sorting by transpositions and transreversals. *Proc. 4th Workshop on Algorithms in Bioinformatics (WABI'04)*, 2004.
16. S.B. Hoot and J.D. Palmer. Structural rearrangements, including parallel inversions, within the chloroplast genome of anemone and related genera. *Journal of Molecular Evolution*, 38:274–281, 1994.
17. J. Kececioglu and D. Sankoff. Exact and approximation algorithms for the inversion distance between two permutations. *Combinatorial Pattern Matching, Proc. 4th Annual Symposium (CPM'93)*, 684:87–105, 1993. Extended version has appeared in Algorithmica, 13:180-210, 1995.
18. G. Lin and T. Jiang. A further improved approximation algorithm for breakpoint graph decomposition. *Journal of Combinatorial Optimization*, 8(2):183–194, 2004.
19. G.H. Lin and G. Xue. Signed genome rearrangements by reversals and transpositions: Models and approximations. *Proc. COCOON '99*, 1627:71–80, 1999.
20. J.D. Palmer and L.A. Herbon. Tricircular mitochondrial genomes of brassica and raphanus: reversal of repeat configurations by inversion. *Nucleic Acids Approach*, 14:9755–9764, 1986.
21. M.E. Walter, Z. Dias, and J. Meidanis. Reversal and transposition distance of linear chromosomes. *String Processing and Information Retrieval: A South American Symposium (SPIRE 98)*, pages 96–102, 1998.