Chapter 67
Sorting Permutations by Transpositions *

Vineet Bafna† Pavel Pevzner‡

Abstract

Sequence comparison in computational molecular biology is a powerful tool for deriving evolutionary and functional relationships between genes. However, classical alignment algorithms handle only local mutations (i.e. insertions, deletions and substitutions of nucleotides) and ignore global rearrangements (i.e. inversions and transpositions of long fragments). As a result, the applications of sequence alignment to analyze highly rearranged genomes (i.e. herpes viruses or plant mitochondrial DNA) are rather limited. The paper addresses the problem of *genome* comparison versus classical *gene* comparison and presents algorithms to analyze rearrangements in genomes evolving by *transpositions*. In the simplest form the problem corresponds to *sorting by transpositions*, i.e. sorting of an array using transpositions of arbitrary fragments. We derive lower bounds on *transposition distance* between permutations and present approximation algorithms for sorting by transpositions. The algorithms also imply a non-trivial upper bound on the *transposition diameter* of the symmetric group. Finally, we formulate two *biological* problems in genome rearrangements and describe the first *algorithmic* steps towards their solution.

1 Introduction

Studies of molecular evolution of herpes viruses raised many more questions than they answered. Genomes of herpes viruses evolve so rapidly that the extremes of present-day phenotypes may appear quite unrelated. As a result, the similarity between many genes in herpes viruses is so low that it is frequently indistinguishable from the background noise (Karlin et al, 1994). In particular, there is little or no cross-hybridization between DNAs of Epstein-Barr virus EBV and Herpes simplex virus HSV-1 and till recently there was no unambiguous evidence that these herpes viruses actually had a common evolutionary origin (McGeoch, 1992). As a result the classical methods of *sequence comparison* are not very useful for such highly diverged genomes and the ventures into the quagmire of molecular phylogeny of herpes viruses may lead to contradictions since different genes give rise to different evolutionary trees (Griffin and Boursnell, 1990). However, recently a new approach to analyze highly diverged genomes was proposed which is based on comparison of *gene orders* versus traditional comparison of *DNA sequences* (Sankoff et al.,1992). Since it is often found that the order of genes is much more conserved than the DNA sequence (Franklin, 1985) this approach seems to be a method of choice for many "hard-to-analyze" genomes.

Analysis of genomes of EBV and HSV-1 reveals that evolution of these viruses involved a number of *inversions* and *transpositions* of large fragments. The analysis of such rearrangements at the *genome* level might be more conclusive than the analysis at the *gene* level traditionally used in molecular evolution. However, there are almost no computer science results allowing a biologist to analyze genome rearrangements.

Genomes evolve by inversions and transpositions as well as more simple operations of deletion, insertion and duplication of fragments. Inversions seem to be a very common rearrangement, in fact some genomes are believed to evolve almost solely by inversions (Palmer and Herbon, 1988). A combinatorial problem of *sorting by reversals* (corresponding to genome rearrangements by inversions) studied intensively in the last two years and currently there are two software programs which proved to be useful to analyze rearrangements in animal (Sankoff et al., 1992) and plant (Bafna and Pevzner,1994) organelle DNA. In 1992 Kececioglu and Sankoff suggested the first perfomance guarantee algorithm for sorting by reversal (see Kececioglu and Sankoff, 1993). Later Bafna and Pevzner, 1993 devised 1.75 perfomance guarantee algorithm for sorting by reversals and proved Gollan's conjecture on the reversal diameter of the symmetric group (see also, Kececioglu and Sankoff, 1994). An interesting problem related to sorting by reversals is the problem of *sorting by prefix reversals* also known as *pancake flipping problem* (Gates and Papadimitriou, 1979, Cohen and Blum, 1993, Heydari and Sudburough, 1993).

In a study of herpes viruses, Hannenhalli et al.,

*Supported by NSF Young Investigator Award, NSF grant CCR-9308567, NIH grant 1R01 HG00987 and DOE grant DE-FG02-94ER61919 to P.A.P.

†bafna@dimacs.rutgers.edu, DIMACS Center, Rutgers Univ., Piscataway, NJ 08854. Work done while the author was at Dept. of CSE, Penn State.

‡pevzner@cse.psu.edu, Department of CSE, Penn State Univ., University Park, PA 16802

1994, faced the problem of analyzing an entire spectrum of genome rearrangements, in particular, transpositions. As a first approximation, transpositions in genome rearrangements can be modelled in a straightforward but limited manner by *sorting by transpositions*, described below:

We assume that the order of genes in a genome is represented by a permutation $\pi = \pi_1\pi_2 \dots \pi_n$. For a permutation π, a *transposition* $\rho(i,j,k)$ (defined for all $1 \leq i < j \leq n+1$ and all $1 \leq k \leq n+1$ such that $k \notin [i,j]$) "inserts" an interval $[i, j-1]$ of π between π_{k-1} and π_k (Fig. 1), i.e. $\rho(i,j,k)$ corresponds to a permutation

$$\begin{pmatrix} 1 \dots i-1 & \boxed{\text{i} \dots \dots \dots \text{j-1}} & \boxed{\text{j} \dots \text{k-1}} & k \dots n \\ 1 \dots i-1 & \boxed{\text{j} \dots \text{k-1}} & \boxed{\text{i} \dots \dots \dots \text{j-1}} & k \dots n \end{pmatrix}$$

Clearly $\pi \cdot \rho(i,j,k)$ has the effect of moving genes $\pi_i, \pi_{i+1}, \dots \pi_{j-1}$ to a new location in a genome. Also, note that for $i < j < k$, $\rho(i,j,k)$ has the effect of exchanging blocks $\pi_i \dots \pi_{j-1}$ and $\pi_j \dots \pi_{k-1}$, and $\rho(i,j,k) = \rho(j,k,i)$.

Given permutations π and σ, the *transposition distance problem* is to find a series of transpositions $\rho_1, \rho_2, \dots, \rho_t$ such that $\pi \cdot \rho_1 \cdot \rho_2 \cdots \rho_t = \sigma$ and t is minimum. We call t the *transposition distance* between π and σ. Note that transposition distance between π and σ equals the transposition distance between $\sigma^{-1}\pi$ and the *identity* permutation ι. *Sorting π by transpositions* is the problem of finding transposition distance $d(\pi)$, between π and ι. Note that the "biological" definition of transpositions used in this paper is different from the usual "algebraic" definition.

Transpositions generate the *symmetric group* S_n and we seek a shortest product of *generators* $\rho_1 \cdot \rho_2 \cdots \rho_t$ that equals $\pi \epsilon S_n$. Even and Goldreich 1981 show that given a set of generators of a permutation group, determining the shortest product of generators that equals π is NP-hard. In our problem, the generator set is fixed and the complexity status of sorting by transpositions is unknown. The only known polynomially-solvable variant of sorting by transpositions is sorting by transpositions $\rho(i, i+1, i+2)$ where the operation is an exchange of adjacent elements. For this problem polynomial algorithms exist for both linear and circular permutations (Jerrum, 1985). Aigner and West, 1987 found a simple algorithm for sorting by transpositions $\rho(1,2,i)$ when the operation is reinsertion of the first element,

Sorting by transpositions is somehow harder combinatorial problem than the previously studied sorting by reversals; in particular the *transposition diameter* of the symmetric group is still unknown. To devise a performance guarantee algorithm for sorting by transpositions we establish lower bounds for transposition distance based on the notion of the *cycle* graph of a permutation. In section 2 we show that the number of alternating cycles in this edge-colored graph is a bottleneck for sorting by transposition. In section 3 we derive upper bounds for sorting by transposition based on the analysis of *crossing* cycles in the cycle graph. More involved analysis in section 4 provides even better upper bounds in the case the cycle graph contains *long* cycles. However, this construction breaks for *short* cycles. Somewhat surprisingly the analysis of *parity* of cycles in the cycle graph provides a compromise and leads to 1.75 performance guarantee algorithm (section 5). Finally, in section 6 we devise 1.5 performance guarantee algorithm for sorting by transpositions by exploiting both the structure and parity of crossing cycles in the cycle graph. As an application, we derive a non-trivial upper bound on the transposition diameter of the symmetric group. Algorithms for sorting by reversals and transpositions present the first steps towards the solutions of two biological problems described in the last section.

2 Lower Bounds for Sorting by Transpositions

Augment a permutation $\pi = \pi_1\pi_2 \dots \pi_n$ by adding $\pi_0 = 0$, and $\pi_{n+1} = n+1$. For all $0 \leq i \leq n$, the pair (π_i, π_{i+1}) is a *breakpoint* if $\pi_{i+1} \neq \pi_i + 1$. Observe that the identity permutation is the only permutation with 0 breakpoints, and therefore, sorting a permutation corresponds to decreasing the number of breakpoints. Also, it is easy to see that a transposition can decrease the number of breakpoints by at most 3, implying a trivial lower bound of $d(\pi) \geq \frac{\#\text{breakpoints}(\pi)}{3}$. However, not all permutations allow transpositions that reduce the number of breakpoints by 3 so the bound is not tight. We introduce the notion of a cycle graph of a permutation and use it to obtain improved lower bounds.

A directed edge-colored *cycle graph* of π, denoted by $G(\pi)$, has a vertex set $\{0, 1, 2, \dots, n+1\}$ and edge set defined as follows. For all $1 \leq i \leq n+1$, *gray* edges are directed from $i-1$ to i, and *black* edges from π_i to π_{i-1} (In Fig. 1, black edges are shown by thick lines, and gray edges are shown by thin lines).

An *alternating cycle* of $G(\pi)$ is a directed cycle in which the edges alternate colors. Observe that for each vertex in $G(\pi)$, every incoming edge is uniquely paired with an outgoing edge of different color. This implies that there is a unique decomposition of the edge set of $G(\pi)$ into alternating cycles. In what follows, we will use *cycle* to refer to an alternating cycle, and *k-cycle* to refer to an alternating cycle of length $2k$. Also, we call a k-cycle *long* if $k > 2$, and *short* otherwise.

There are a total of $2(n+1)$ edges and at most $(n+1)$ cycles in $G(\pi)$ (the identity permutation is the only permutation with $n+1$ cycles). For a permutation π denote the number of cycles in $G(\pi)$ as $c(\pi)$. Then the

sequence of transpositions that sort π must increase the number of cycles from $c(\pi)$ to $n+1$. For a permutation π, and a transposition ρ, denote $\Delta c(\rho) = c(\pi\rho) - c(\pi)$, as the change in number of cycles due to transposition ρ.

LEMMA 2.1. $\Delta c(\rho) \in \{2, 0, -2\}$

Proof. A transposition $\rho(i,j,k)$ involves 6 vertices of graph $G(\pi)$ $(\pi_{i-1}, \pi_i, \pi_{j-1}, \pi_j, \pi_{k-1}, \pi_k)$ and leads to removing 3 black edges $((\pi_i, \pi_{i-1}), (\pi_j, \pi_{j-1})$ and $(\pi_k, \pi_{k-1}))$ and adding 3 new black edges $((\pi_j, \pi_{i-1}), (\pi_i, \pi_{k-1})$ and $(\pi_k, \pi_{j-1}))$ (removed and added edges form a cycle shown by dotted lines in Fig. 1).

Three removed edges belong to either three or two or one cycle in the cycle decomposition of $G(\pi)$. In the case the removed edges belong to three cycles $c(\pi\rho) = c(\pi) - 3 + 1$ since these 3 cycles correspond to one cycle in $G(\pi\rho)$ (Fig. 2a). In the case the removed edges belong to two cycles $c(\pi\rho) = c(\pi) - 2 + 2$ since these two cycles correspond to two cycles in $G(\pi\rho)$ (Fig. 2b). In the case the removed edges belong to a single cycle C there are two subcases (Fig. 2c and Fig. 2d). In the subcase shown in Fig. 2c $c(\pi\rho) = c(\pi) - 1 + 1$ since C corresponds to 1 cycle in $G(\pi\rho)$. In the subcase shown in Fig. 2d $c(\pi\rho) = c(\pi) - 1 + 3$ since C corresponds to three cycles in $G(\pi\rho)$. □

Lemma 2.1 immediately gives a lower bound on $d(\pi)$.

THEOREM 2.1. $d(\pi) \geq \frac{n+1-c(\pi)}{2}$

A cycle in $G(\pi)$ is *odd* if it has an odd number of black edges, and *even* otherwise. To establish a better lower bound we analyze odd and even cycles separately. Define $c_{odd}(\pi)$ $(c_{even}(\pi))$ as the number of odd (even) cycles in π. For a permutation π, and a transposition ρ, denote $\Delta c_{odd}(\rho) = c_{odd}(\pi\rho) - c_{odd}(\pi)$, as the change in number of odd cycles due to transposition ρ. The following lemma implies that the parity of $c_{odd}(\pi)$ (and $c_{even}(\pi)$) is the same for all $\pi \in S_n$.

LEMMA 2.2. $\Delta c_{odd}(\rho) \in \{2, 0, -2\}$

As the identity permutation has $n+1$ odd cycles, lemma 2.2 implies a better bound

THEOREM 2.2. $d(\pi) \geq \frac{n+1-c_{odd}(\pi)}{2}$

Define $d(n) = \max_{\pi \in S_n} d(\pi)$ to be the *transposition diameter* of the symmetric group of order n. Observing that for $\pi = n\ n-1\ \ldots\ 2\ 1$, $c_{odd}(\pi)$ equals 1 if n is even and equals 0 or 2 if n is odd, the transposition diameter of the symmetric group S_n is at least $\lfloor \frac{n}{2} \rfloor$. One can verify that $d(n\ n-1 \ldots\ 1) \leq \lfloor \frac{n}{2} \rfloor + 1$ for all n and $d(n) = d(n\ n-1 \ldots\ 1) = \lfloor \frac{n}{2} \rfloor + 1$ for $3 \leq n \leq 10$.

3 Upper Bounds for Sorting by Transpositions

For $x \in \{2, 0, -2\}$, define an *x-move* on π as a transposition ρ, such that $\Delta c(\rho) = x$. In order to sort faster, we would like to use as many 2-moves as possible. In this section, we study the structure of cycles which allow 2-moves to devise an approximation algorithm for sorting by transpositions.

We number the black edges of the cycle graph $G(\pi)$ from 1 to $n+1$ by assigning label i to a black edge from π_i to π_{i-1}. We say that transposition $\rho(i,j,k)$ *acts* on edges i, j and k. We also say that a transposition $\rho(i,j,k)$ *acts* on a cycle C if all 3 black edges i, j and k belong to C. The proof of lemma 2.1 implies the following simple observations.

LEMMA 3.1. *If a transposition ρ acts on a cycle and creates more than one new cycle in $G(\pi\rho)$ then ρ is a 2-move.*

LEMMA 3.2. *If a transposition ρ acts on edges belonging to exactly two different cycles then ρ is a 0-move.*

There exist two different kinds of cycles - *non-oriented* for which no 2-moves are possible (for example, a cycle in $G(n\ n-1 \ldots\ 1)$) and *oriented* for which a 2-move is possible (Fig. 2d). Below we give a formal definition of oriented and non-oriented cycles.

Consider a k-cycle C visiting (in order) the black edges $i_1, \ldots i_k$. A cycle C can be written in k possible ways depending on the choice of the first black edge. Below we assume that the initial black edge i_1 of cycle C starts at its "rightmost" vertex in π, i.e. $i_1 = \max_{1 \leq t \leq k} i_t$.

For all $k > 1$, a cycle $C = (i_1 \ldots i_k)$ is *non-oriented* if $i_1 \ldots i_k$ is a decreasing sequence, otherwise C is an *oriented* cycle. We will also use a characterization of non-oriented cycles in the terms of *edge directions*. A gray edge joining $\pi_t = i - 1$ with $\pi_s = i$ in $G(\pi)$ is directed *left* if $t > s$ and is directed *right* otherwise. Clearly, a cycle $C = (i_1 \ldots i_k)$ is non-oriented iff $k > 1$ and C has exactly one right edge (a gray edge between black edges i_k and i_1).

LEMMA 3.3. *If C is an oriented cycle then there exists a 2-move acting on C. If C is a non-oriented cycle there exist no 2-moves acting on C.*

Proof. Let $C = (i_1 \ldots i_k)$ be an oriented cycle and let $3 \leq t \leq k$ be an index such that $i_t > i_{t-1}$. Consider a transposition $\rho(i_{t-1}, i_t, i_1)$ acting on C. This transposition creates a 1-cycle (on vertices $\pi_{i_{t-1}-1}$ and π_{i_t}) and some other cycles. Therefore, by lemma 3.1, ρ is a 2-move. □

Lemmas 3.2 and 3.3 imply

THEOREM 3.1. *For an arbitrary (unsorted) permutation π there exists either a 2-move or a 0-move followed by a 2-move.*

Proof. If $G(\pi)$ has an oriented cycle then, by lemma 3.3, a 2-move is possible. Otherwise consider a non-oriented cycle $C = (i_1, \ldots, i_k)$ and let r be a position of the maximal element of π in the interval

$[i_2, i_1 - 1]$. Let s be a position of $\pi_r + 1$ in π. Clearly $s \notin [i_2, i_1]$. W.l.o.g, assume that $s > i_1$, and consider a transposition $\rho(r+1, s, i_2)$ (Fig. 3). The transposition ρ acts on edges of two different cycles, therefore by lemma 3.2 ρ is a 0-move. Since ρ changes the direction of the left edge (π_{i_1-1}, π_{i_2}), and does not change direction of the right edge (π_{i_k-1}, π_{i_1}), the cycle C' containing these edges in $G(\pi\rho)$ has at least two right edges. Therefore C' is an oriented cycle allowing a 2-move (lemma 3.3). □

Theorem 3.1 provides an increase of $c(\pi)$ by at least 2 in two consecutive moves and implies the following upper bound for sorting by transpositions.

THEOREM 3.2. *Any permutation π can be sorted in $n + 1 - c(\pi)$ transpositions.*

Theorems 2.1 and 3.2 imply an approximation algorithm for sorting by transpositions with performance guarantee 2. In the following sections we give a better upper bound by disallowing -2-moves, and forcing at least two consecutive 2-moves between any two 0-moves. In our approximation algorithm, we will use only 0- and 2-moves, although we do not have a proof that an optimal sequence of transpositions exists which does not use -2-moves.

4 Crossing Cycles

Theorem 3.1 shows that the number of 2-moves can be made greater than or equal to the number of 0-moves. In order to improve the performance ratio for sorting by transposition, we need to further increase the number of 2-moves. Theorem 4.1 provides the first step towards such an improvement. But, first we need to prove a series of technical lemmas.

Consider a triple of black edges x, y, z belonging to the same cycle C in $G(\pi)$. C induces a cyclic order on x, y, z and among 3 possible representations of this order we choose the one starting from the rightmost black edge $\max\{x, y, z\}$ as the canonical representation for a triple (x, y, z). A triple (in a canonical order) is called *non-oriented* if $x > y > z$ and *oriented* otherwise. For example, a triple (k, j, i) in Fig. 2c is non-oriented while triple (k, i, j) in Fig. 2d is oriented. All triples of a non-oriented cycle are non-oriented. On the other hand every oriented cycle has at least one oriented triple.

Ordered sequences of integers $\{v_1 < \ldots < v_k\}$ and $\{w_1 < \ldots < w_k\}$ are *interleaving* if either $v_1 < w_1 < v_2 < w_2 < \ldots < v_k < w_k$ or $w_1 < v_1 < w_2 < v_2 \ldots < w_k < v_k$. Sets of integers V and W are interleaving if orderings of V and W are interleaving. A transposition $\rho(i, j, k)$ is a *shuffling* transposition with respect to a triple (x, y, z) if the sets $\{i, j, k\}$ and $\{x, y, z\}$ interleave.

LEMMA 4.1. *Let (x, y, z) be a triple in a cycle C, and $i, j, k \notin C$ be black edges in $G(\pi)$. Then $\rho(i, j, k)$ changes the orientation of triple (x, y, z) (i.e., it transforms oriented triple into non-oriented and vice-versa) iff ρ is a shuffling transposition for (x, y, z).*

LEMMA 4.2. *If C is non-oriented, then for all triples $(x, y, z) \in C$, transposition $\rho(z, y, x) = \rho(y, x, z)$ transforms C into a non-oriented cycle in $G(\pi\rho)$.*

We will also need the following lemma specifying some 2-moves acting on oriented cycles.

LEMMA 4.3. *If (x, y, z) is an oriented triple then $\rho(y, z, x) = \rho(z, x, y)$ is a 2-move.*

Cycles C and C' are *crossing* if there exists an oriented triple in C and a non-oriented triple in C' that are interleaving (Fig. 4a). Cycles C and C' are *non-interfering* if there exist oriented triples in C and C' that are not interleaving (Fig. 4b).

LEMMA 4.4. *If permutation π has crossing or non-interfering cycles then there exist two consecutive 2-moves in π.*

Proof. If cycles C and C' in $G(\pi)$ are crossing, there exist an oriented triple $(x, z, y) \in C$ and a non-oriented triple $(x', y', z') \in C'$ which are interleaving (Fig. 4a). By lemma 4.3 a transposition $\rho(z, y, x)$ defines a 2-move on C. On the other hand, since (x, y, z) and (x', y', z') are interleaving, $\rho(z, y, x)$ is a shuffling transposition with respect to (x', y', z'). Thus, by lemma 4.1 ρ transforms C' into an oriented cycle in $G(\pi\rho)$ and by lemma 3.3, provides a second 2-move.

Alternatively, if C and C' are non-interfering then there exist oriented triples $(x, z, y) \in C$ and $(x', z', y') \in C'$ which are non-interleaving (Fig. 4b). By lemma 4.3 a transposition $\rho(z, y, x)$ defines a 2-move on C. Furthermore, (x', z', y') remains an oriented triple (lemma 4.1) of C' in $G(\pi\rho)$, which provides a second 2-move. □

We say that a transposition *acts* on two cycles C and C' in $G(\pi)$ if it acts on black edges of both C and C'. To prove theorem 4.1 below we will need the following observation about transpositions acting on two cycles.

LEMMA 4.5. *Let C be a cycle containing black edges x and y and D be a cycle containing black edges x' and y'. Let ρ be a transposition acting on three of four black edges x, y, x', y'.*

- *If $\{x, y\}$ does not interleave with $\{x', y'\}$ then ρ creates a cycle with a non-oriented triple.*
- *If $\{x, y\}$ interleaves with $\{x', y'\}$ then ρ creates a cycle with an oriented triple.*

We say that cycle $C = (i_1, \ldots, i_k)$ *spans* cycle $D = (j_1, \ldots, j_l)$, if $i_k < j_l < j_1 < i_1$. The following lemma illustrates an important property of non-oriented cycles.

LEMMA 4.6. *For every non-oriented cycle $C = (\ldots a \ldots b \ldots)$, with arbitrary edges a, b, there exists a*

cycle $D(\ldots c \ldots d \ldots)$ such that (a,b) and (c,d) interleave.

THEOREM 4.1. *If there exists a long cycle in $G(\pi)$, then either a 2-move or a 0-move followed by two consecutive 2-moves are possible in π.*

Proof. If $G(\pi)$ has an oriented cycle then, by lemma 3.3 a 2-move is possible. Also, if there exist non-oriented long cycles C and D with interleaving triples $(r,s,t) \in C$ and $(x,y,z) \in D$, then a 0-move ρ acting on edges z,y,x is a shuffling transposition for C. By lemma 4.1, ρ transforms C into an oriented cycle C'. By lemma 4.2 ρ transforms D into a non-oriented cycle D'. It is easy to see that C' and D' are crossing, therefore, by lemma 4.4 there exist two consecutive 2-moves in $G(\pi\rho)$.

Therefore, assume that no two cycles have interleaving triples. Pick a non-oriented long cycle $C = (i_1, \ldots, i_k)$, such that C is not spanned by any long cycle. Find a cycle $D = (x \ldots c \ldots d \ldots y)$ such that the pairs (c,d) and (i_1, i_k) interleave (lemma 4.6). Note that if $y < i_k$, then $x < i_1$ otherwise D would span C. On the other hand, if $y > i_k$, then $x > i_1$ otherwise (c,d) and (i_1, i_k) would not interleave. Therefore, either $y < i_k < x < i_1$ or $i_k < y < i_1 < x$. W.l.o.g, we assume the latter. Let s be the rightmost edge in C to the left of y, i.e. $s = \max_{i \in C, i<y} i$. Two cases arise:

$s > i_k$: Find cycle $E = (v \ldots c \ldots d \ldots u)$, such that the pairs (c,d) and (i_k, s) interleave (lemma 4.6). If $u < i_k$ then $v < s$ because otherwise E either spans C ($v > i_1$) or has an interleaving triple with $(i_k, s, i_1) \in C$ ($s < v < i_1$). If $u > i_k$ (Fig. 5a), then four cases arise depending on v lying in one of the intervals $[s,y], [y,i_1], [i_1,x]$ or $[x, n+1]$ (Fig. 5b-e). The transpositions $\rho(x,y,v)$ in Fig. 5a and $\rho(x,y,u)$ in Fig. 5b-e are shuffling w.r.t the triple (i_1, s, i_k) of C, and by lemma 4.1, transform C into an oriented cycle C' in $G(\pi\rho)$. ρ also transforms D and E into D' and a 1-cycle in $G(\pi\rho)$. From lemma 4.5, D' is oriented in Fig. 5a. In the remaining cases, D' is oriented when $v \in [y, x-1]$ and non-oriented otherwise (lemma 4.5). Observe that in the first case C' and D' are crossing (Fig. 5b,e), otherwise they are non-interfering (Fig. 5c,d). In either case, two 2 moves are possible in $G(\pi\rho)$(lemma 4.4).

$s = i_k$: Let t be the leftmost black edge in C to the right of y, i.e. $t = \min_{i \in C, i>y} i$. As C is a long cycle, $t < i_1$. Find $E = (v, \ldots, c, \ldots, d, \ldots, u)$ such that the pairs (c,d) and (t, i_1) interleave (lemma 4.6). Cycle E is different from cycle D because otherwise E and C would have interleaving triples. If $v > i_1$ then $u > t$ because otherwise E either spans C ($u < i_1$) or has an interleaving triple with $(i_1, t, i_k) \in C$ ($i_1 < u < t$). This case is similar to the cases shown in Fig. 5d,e. If $v < i_1$, then three cases arise depending on which of the intervals $[0, i_k], [i_k, y]$ or $[y,t]$ contains u. The first of these cases is shown in Fig. 5f, while the other two are symmetric to cases in Fig. 5c and 5e respectively. In Fig. 5f, the transposition $\rho(x,y,u)$ transforms C into a non-oriented cycle C' (lemma 4.1), and cycles D, E into an oriented cycle D' and a 1-cycle in $G(\pi\rho)$ (lemma 4.5). Further, C' and D' are crossing, and therefore two 2-moves are possible in $G(\pi\rho)$. □

5 Mixing Odd and Even Cycles

Theorem 4.1 guarantees creating at least four cycles in three moves thus providing $\Delta c(\rho) = \frac{4}{3}$ on average which is better than $\Delta c(\rho) = 1$ given by theorem 3.1. However, it can be applied only when $G(\pi)$ has long cycles. In case $G(\pi)$ only has short cycles, the best we can guarantee is a 0-move followed by a 2-move creating four 1-cycles from two 2-cycles (theorem 3.1). Theorems 3.1 and 4.1 motivate the algorithm *Tsort*.

Algorithm *Tsort*(π)

1. While $G(\pi)$ has a long cycle, perform either a 2-move or a 0, 2, 2-move. (theorem 4.1).
2. If $G(\pi)$ has only short cycles, perform a 0-move followed by a 2-move (theorem 3.1).

Does *Tsort* achieve a performance ratio of better than 2? Unfortunately in the case that $G(\pi)$ has only short cycles, the 0-move followed by a 2-move provides only $\Delta c(\rho) = \frac{4-2}{2} = 1$ on average. However, for these two moves $\Delta c_{odd}(\rho) = \frac{4-0}{2} = 2$ thus achieving a maximal rate of creating of odd cycles from the perspective of the theorem 2.2. On the other hand, theorem 4.1 does not guarantee yet that $\Delta c_{odd}(\rho) = 2$ for every 2-move. Therefore, if we use either the number of cycles or the number of odd cycles as our objective function, we cannot guarantee a performance ratio better than 2. Somewhat surprisingly, we show that a *mixed* objective function which gives different weights to odd and even cycles leads to an improved performance guarantee.

THEOREM 5.1. *Tsort provides a performance guarantee of* 1.75 *for sorting by transpositions.*

Proof. For arbitrary $x \geq 1$, define the objective function $f(\pi) = x c_{odd}(\pi) + c_{even}(\pi)$, where $c_{odd}(\pi)$ and $c_{even}(\pi)$ are the number of odd and even cycles in $G(\pi)$, respectively. Clearly, for this range of x, $f(\pi)$ is uniquely maximized by the identity permutation, and sorting a permutation corresponds to maximizing f. Observe that the maximum gain any transposition ρ can achieve is $\Delta f(\rho) = f(\pi\rho) - f(\pi) = 2x$. We now evaluate the maximum Δf guaranteed by theorems 3.1

and 4.1.

In the case that $G(\pi)$ only has short cycles theorem 3.1 guarantees that in two moves four 1-cycles are created from two 2-cycles, implying a gain of $4x-2$ over two moves, or an average gain of $2x-1$ in one transposition. In any 2-move two new cycles are created and, in the worst case (if both are even) we can still guarantee a gain of 2. By construction, a 0-move in theorem 4.1 either creates a 1-cycle or does not change the number of black edges in any cycle. Therefore $\Delta f \geq 0$ for any 0-move. Moreover, theorem 4.1 guarantees that any such 0-move is followed by two 2-moves implying an average gain of $\frac{4}{3}$. It follows that $\Delta f \geq \min\{\frac{4}{3}, 2x-1\}$ on the average. Comparing the best possible gain of $2x$ against the gain provided by *Tsort*, we get a performance guarantee of

$$\frac{2x}{\min\{\frac{4}{3}, 2x-1\}}$$

The best performance is achieved for $x = \frac{7}{6}$, resulting in the approximation ratio 1.75. □

6 1.5-Approximation for Sorting By Transposition

In order to improve performance still further, we need to strengthen theorem 4.1. Note that theorem 4.1 only guarantees an increase in the number of cycles, whereas theorem 2.2 indicates that we need to increase the number of odd cycles. By choosing appropriate 2-moves, we shall ensure that the number of odd cycles increases by at least two in every 2-move.

We call a transposition ρ *valid* if $\Delta c(\rho) = \Delta c_{odd}(\rho)$.

LEMMA 6.1. *If there exists an oriented cycle in $G(\pi)$, then either a valid 2-move or a valid 0-move followed by two consecutive valid 2-moves is possible in π.*

Proof. For a cycle C containing edges i and j define $d(i,j)$ as the number of black edges between vertices π_i and π_j in C (in particular $d(i,j) = 1$ for consecutive black edges i and j). Suppose there is no valid 2-move in π. For an arbitrary oriented cycle C in $G(\pi)$ consider the set S of oriented triples of C such that the distance between the first and the second element of the triple is odd:

$$S = \{(x,y,z) : x,y,z \in C \text{ and } d(x,y) \text{ is odd } \}$$

The observation that every oriented cycle C has an oriented triple (x,y,z) such that x and y are the consecutive black edges in C implies that S is non-empty. Let (x,y,z) be a triple in S with maximal x.

A transposition ρ acting on edges y,z and x transforms C into 3 cycles C_1, C_2 and C_3 consisting of $d(x,y), d(y,z)$ and $d(z,x)$ black edges. As $(x,y,z) \in S$, cycle C_1 is odd. If either C_2 or C_3 is odd then $\Delta c_{odd}(\rho) = 2$ and ρ is a valid 2-move contradicting to the assumption that there is no valid 2-moves in π. Therefore both $d(y,z)$ and $d(z,x)$ are even. As both $d(y,z)$ and $d(z,x)$ are even the fragments of C from y to z and from z to x contain at least two edges. Let a be a black edge preceding z in C and b be a black edge following z in C (Fig. 6a).

If $y < a < x$ then transposition ρ acting on edges y, a and x creates cycles of length $d(y,z)-1$ and $d(x,y)$. Both these numbers are odd and, therefore ρ is a valid 2-move thus contradicting to the assumption. Therefore $a \notin [y,x]$. Symmetric arguments demonstrate that $b \notin [y,x]$.

If $a > x$ then (a,z,x) is an oriented triple with odd $d(a,z) = 1$ thus contradicting to the choice of (x,y,z). Therefore $a < y$. If $b > x$ then (b,a,z) is an oriented triple with odd $d(b,a) = d(b,x) + d(x,y) + d(y,a) = (d(z,x)-1) + d(x,y) + (d(y,z)-1)$ thus contradicting to the choice of (x,y,z). Therefore $a, b < y$.

The situations described by conditions $b < a$ and $a < b$ are presented in the Fig. 6b,c. If $b < a$, then $\rho(b,a,z)$ is a valid 2-move (Fig. 6b). If $a < b$, then there exist 2-moves but no valid 2-moves in π. However, there exists a valid 0-move followed by two consecutive valid 2-moves (Fig. 6c). □

Fig. 6c presents an example of an oriented cycle which does not allow valid 2-moves. This cycle has a complicated "self-interleaving" structure and in the following, we try to avoid creating such cycles. In order to achieve this goal, we define *strongly oriented* cycles, which have the simplest "self-interleaving" structure among all oriented cycles. This idea leads to theorem 6.1.

Let $C = (i_1, \ldots, i_k)$ be a cycle in $G(\pi)$ and let $C^* = (i_1 = j_1 > \ldots > j_k)$ be a sequence of black edges of C in decreasing order. Sequences C and C^* coincide for a non-oriented cycle and are different otherwise. Define *strongly-oriented* cycles as oriented cycles for which C^* can be transformed into C by a single transposition, i.e. C can be partitioned into strips $C_1 = (i_1, \ldots, i_a)$, $C_2 = (i_{a+1}, \ldots, i_b)$, $C_3 = (i_{b+1}, \ldots, i_c)$ and $C_4 = (i_{c+1}, \ldots, i_k)$ such that $C = C_1C_2C_3C_4$ and $C^* = C_1C_3C_2C_4$ (C_4 might be empty). For example, Fig. 6b gives an example of a strongly oriented cycle, as $C = xyabz$ is transformed into $C^* = xzyab$ by a single transposition. Clearly, every strongly oriented cycle has exactly two right edges. On the other hand, not every oriented cycle with two right edges is strongly oriented (Fig. 6c).

LEMMA 6.2. *A strongly oriented cycle allows a valid 2-move.*

Next, we present two lemmas which show how strongly oriented cycles arise from non-oriented cycles.

LEMMA 6.3. *If ρ is a shuffling transposition on*

a non-oriented cycle C, then ρ transforms C into a strongly oriented cycle in $G(\pi\rho)$.

LEMMA 6.4. *Let $D(x,\ldots,y)$ and $E(x',\ldots,y')$ be two non-oriented cycles in $G(\pi)$ with no interleaving triples and let ρ be a transposition acting on three of four black edges x, y, x', y'. Then ρ creates strongly oriented cycle iff D and E have interleaving pairs of edges.*

Every strongly oriented cycle has exactly two right edges, one of which is of the form (r, i_1). Label the other as (s, t). For strongly oriented cycles of the first kind (Fig. 7a) define

$$r' = \max_{i \in left} i \text{ and } t' = \min_{i \in right} i$$

and consider three intervals $I_1(C) = [r', r], I_2(C) = [t, t']$ and $I_3 = [0, s] \cup [i_1, n+1]$.
For strongly oriented cycles of the second kind (Fig. 7c) define

$$s' = \max_{i \in left} i \ , \ t' = \min_{i \in right} i \ , \ a = \max_{i \in mid'} i \text{ and } a' = \min_{i \in mid''} i$$

and consider intervals $I_1(C) = [s', s], I_2(C) = [t, t']$ and $I_3(C) = [a, a']$.

A strongly oriented cycle C and a non-oriented cycle $C' = (i_1, \ldots, i_k)$ are *strongly crossing* if there exists a black edge x in C' such that each of the sets $I_1(C), I_2(C)$ and $I_3(C)$ contains exactly one element of the triple (i_1, x, i_k). Note that 2-moves for C described in the proof of lemma 6.2 form shuffling transpositions w.r.t (i_1, x, i_k). This observation and lemma 6.2 imply

LEMMA 6.5. *If $G(\pi)$ has strongly crossing cycles then there exist two consecutive valid 2-moves in $G(\pi)$.*

Finally, we use theorem 4.1 and lemmas 6.1– 6.5 to prove theorem 6.1 (see[3] for proofs).

THEOREM 6.1. *If there exists a long cycle in $G(\pi)$, then either a valid 2-move or a valid 0-move followed by two consecutive valid 2-moves are possible in π.*

Algorithm *TransSort(π)*

1. While $G(\pi)$ has a long cycle, perform a valid 2-move or a valid $0, 2, 2$-move (theorem 6.1).
2. If $G(\pi)$ has only short cycles, perform a valid 0-move followed by a valid 2-move (theorem 3.1).

Theorem 6.1 has a constructive proof which implies an $O(n^2)$ algorithm *TransSort* for sorting by transpositions. Also, theorems 2.2, 3.1 and 6.1 imply

COROLLARY 6.1. *Algorithm TransSort sorts π in no more than $\frac{3}{4} \cdot (n + 1 - c_{odd}(\pi))$ transpositions, thereby ensuring a performance guarantee of* 1.5

COROLLARY 6.2. *The transposition diameter of the symmetric group S_n is at most $\frac{3}{4}n$.*

7 Open problems

Recent advances in large-scale comparative genetic mapping offer exciting prospects for understanding mammalian genome evolution. The large number of conserved segments in the maps of man and mouse suggest that multiple chromosomal rearrangements have occured since the divergence of lineages leading to humans and mice. In their pioneering paper, Nadeau and Taylor, 1984 estimated that just 178 ± 39 rearrangements have occured since this divergence. This estimate survived a ten-fold increase in the amount of the comparative man/mouse mapping information; the new estimate based on the latest data (Copeland et al., 1993) almost did not change compared to Nadeau and Taylor, 1984 . However, the arguments used by Nadeau and Taylor, 1984 are non-constructive and do not provide any solution to an open biological problem of reconstructing evolutionary scenario explaining man and mouse genome rearrangements.

Chromosomal rearrangements include not only inversions and transpositions but *translocations* as well. A combinatorial analysis of all such rearrangements to derive a scenario of mammalian evolution is far beyond the possibilities of current algorithms. However, some limited applications of algorithms for inversions, transpositions and translocations to study chromosome evolutions are already possible. In particular, extreme conservation of genes on X chromosome provides an opportunity to study evolutionary history of X chromosome independently of the rest of the genomes thus reducing computational complexity of the problem. According to Ohno's law (Ohno, 1967) gene content of X chromosome is assumed to have remained the same throughout mammalian development in the last 125 million years. However, the order of genes on X chromosome has been disrupted several times. The conservative gene content of X chromosome implies that the only translocations which affected the gene order in X chromosome were translocations between two copies of X chromosome and thus might be ignored for our purposes. A recently found violation of the Ohno law by *Csfgmra* gene does not affect this conclusion since this gene is located at the very end of the human X chromosome. Davisson, 1987 and Lyon,1988 suggested two conflicting scenarios of rearrangements in X chromosome under the assumption that X chromosome was not involved in translocations. Based on the analysis of the latest data on comparative man/mouse mapping Bafna and Pevzner, 1995 found the most parsimonious scenario for evolutionary history of X chromosome and corrected the previously suggested scenarios.

Another open problem on genome rearrangements is related to viral evolution. As was mentioned in the introduction, herpes viruses, present a particularly

hard case for classical sequence comparison. On the other hand, they present particularly suitable test case for study of genome rearrangements since complete sequences of seven diverse herpesviruses are known. Herpes virus genomes contain from 70 to about 200 genes. Detailed comparison of amino acid sequences of viral proteins resulted in an "alphabet" of about 30 conserved genes which were rearranged in different herpes viruses (Hannenhalli et al., 1994). Derived lower bounds for the pairwise genome rearrangements of viral genomes allowed us to construct the most parsimonious scenarios for herpes virus evolution. Moreover, there are only three alternative equally parsimonious scenarios of genome rearrangements in herpes viruses corresponding to three different gene arrangements in an ancestor herpesvirus (Hannenhalli et al., 1994). Although, based on the currently available data, it is impossible to delineate the true scenario among these three, ongoing efforts to map and sequence different herpes virus genomes provide a warrant that a true evolutionary scenario will be found in the future.

References

[1] M. Aigner and D. B. West. Sorting by insertion of leading element. *Journal of Combinatorial Theory*, 45:306–309, 1987.

[2] V. Bafna and P. Pevzner. Genome rearrangements and sorting by reversals. In *34th IEEE Symp. on Foundations of Computer Science*, pages 148–157, 1993 (to appear in SIAM J. of Comp.).

[3] V. Bafna and P. Pevzner. Sorting by Transpositions. Technical Report CSE-94-031, The Pennsylvania State University, 1994.

[4] V. Bafna and P. Pevzner. Sorting by reversals: Genome rearrangements in plant organelles and evolutionary history of X chromosome. (To appear in Mol. Biol. and Evol., **12**, 1995).

[5] D. Cohen and M. Blum. Improved bounds for sorting pancakes under a conjecture. 1993 (manuscript).

[6] N. G. Copeland, N. A. Jenkins, D. J. Gilbert, J. T. Eppig, L. J. Maltals, J. C. Miller, W. F. Dietrich, A. Weaver, S. E. Lincoln, R. G. Steen, L. D. Steen, J. H. Nadeau, and E. S. Lander. A genetic linkage map of the mouse: Current applications and future prospects. *Science*, 262:57–65, 1993.

[7] M. Davisson. X-linked genetic homologies between mouse and man. *Genomics*, 1:213–227, 1987.

[8] S. Even and O. Goldreich. The minimum-length generator sequence problem is NP-hard. *Journal of Algorithms*, 2:311–313, 1981.

[9] N. Franklin. Conservation of genome form but not sequence in the transcription antitermination determinants of bacteriophages $\lambda, \phi 21$ and $P22$. *Journal of Molecular Evolution*, 181:75–84, 1985.

[10] W. H. Gates and C. H. Papadimitriou. Bounds for sorting by prefix reversals. *Discrete Mathematics*, 27:47–57, 1979.

[11] S. Hannenhalli, C. Chappey, E. Koonin, and P. Pevzner. Algorithms for genome rearrangements: Herpesvirus evolution as a test case. In *Proc. of 3rd Intl. Conference on Bioinformatics and Complex Genome Analysis*, 1994 (to appear).

[12] M. Heydari and I. H. Sudborough. On sorting by prefix reversals and the diameter of pancake networks. 1993 (manuscript).

[13] M. Jerrum. The complexity of finding minimum-length generator sequences. *Theoretical Computer Science*, 36:265–289, 1985.

[14] S. Karlin, E. S. Mocarski, and G. A. Schachtel. Molecular evolution of herpesviruses: genomic and protein sequence comparisons. *Journal of Virology*, 68:1886–1902, 1994.

[15] J. Kececioglu and D. Sankoff. Exact and approximation algorithms for the inversion distance between two permutations. In *Proc. of 4th Ann. Symp. on Combinatorial Pattern Matching*, LNCS 684, pages 87–105. Springer Verlag, 1993.

[16] J. Kececioglu and D. Sankoff. Efficient bounds for oriented chromosome inversion distance. In *Proc. of 5th Ann. Symp. on Combinatorial Pattern Matching*, LNCS 807, pages 307–325. Springer Verlag, 1994.

[17] M. F. Lyon. X-Chromosome Inactivation and the Location and Expression of X-linked Genes. *Am. J. Hum. Genet.*, 42:008–016, 1988.

[18] D. J. McGeoch. Molecular evolution of large DNA viruses of eukaryotes. *Seminars in Virology*, 3:399–408, 1992.

[19] J. H. Nadeau and B. A. Taylor. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proc. Natl. Acad. Sci. USA*, 81:814–818, 1984.

[20] S. Ohno. *Sex chromosomes and sex-linked genes.* Springer, Heidelberg, 1967.

[21] J. D. Palmer and L. A. Herbon. Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. *Journal of Molecular Evolution*, 27:87–97, 1988.

[22] D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B. F. Lang, and R. Cedergren. Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome. *Proc. Natl. Acad. Sci. USA*, 89:6575–6579, 1992.

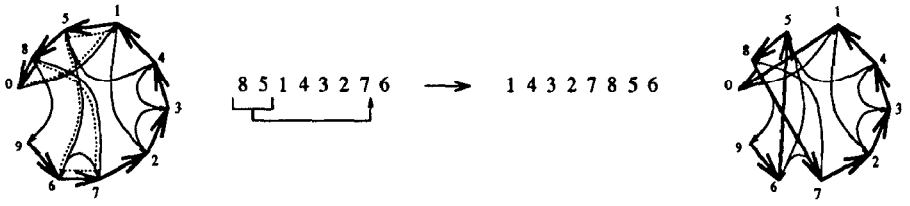

Figure 1: Transposition $\rho(1,3,8)$ on π transforms cycle graph $G(\pi)$ into $G(\pi\rho)$

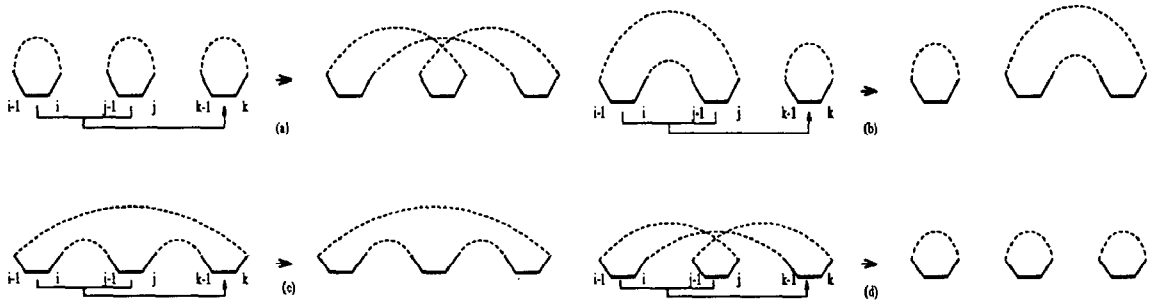

Figure 2: Transposition $\rho(i,j,k)$ changes the number of cycles and transforms black edges $(\pi_i, \pi_{i-1}), (\pi_j, \pi_{j-1})$ and (π_k, π_{k-1}) into $(\pi_j, \pi_{i-1}), (\pi_i, \pi_{k-1})$ and (π_k, π_{j-1}).

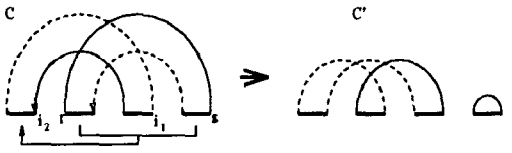

Figure 3: A 0-move creating an oriented cycle

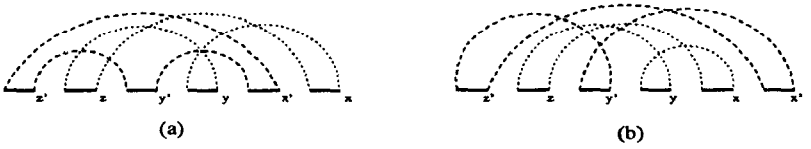

Figure 4: Crossing and non-interfering cycles

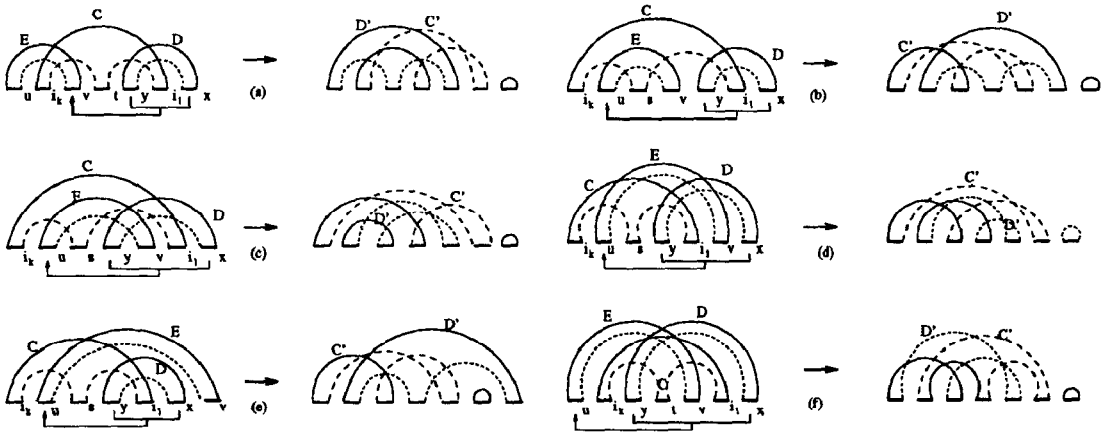

Figure 5: 0-move leading to two 2-moves

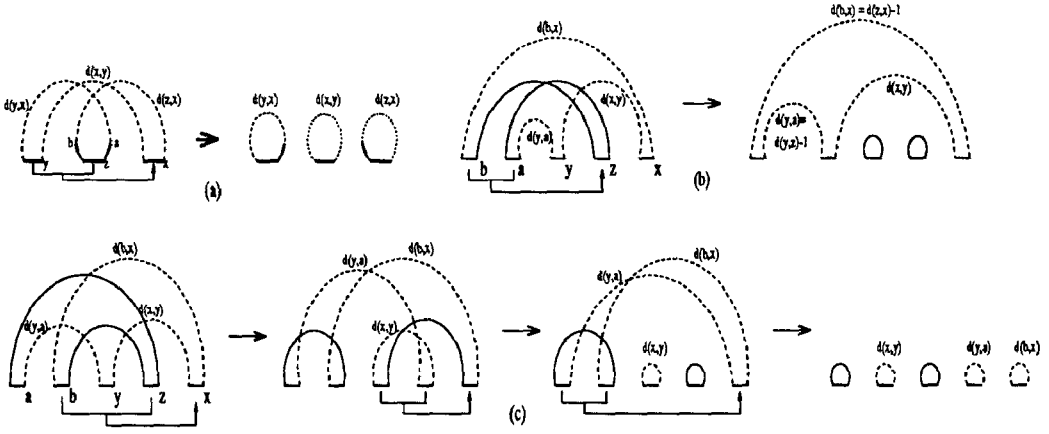

Figure 6: Valid 2-moves and 0, 2, 2-moves on an oriented cycle

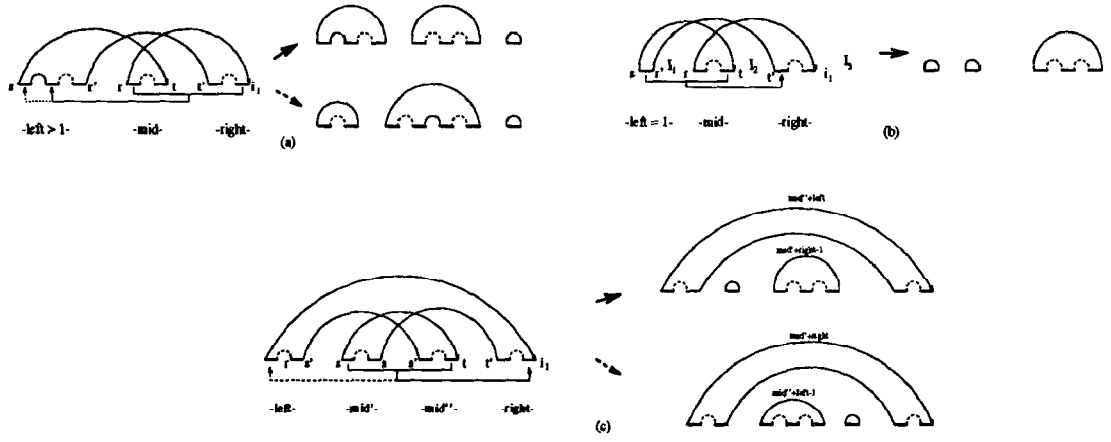

Figure 7: Strongly oriented cycles: (a,b) First kind. (c) Second kind