Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals

SRIDHAR HANNENHALLI

Bioinformatics, SmithKline Beecham Pharmaceuticals, King of Prussia, Pennsylvania

AND

PAVEL A. PEVZNER

University of Southern California, Los Angeles, California

Abstract. Genomes frequently evolve by reversals $\rho(i, j)$ that transform a gene order $\pi_1 \cdots \pi_i \pi_{i+1} \cdots \pi_{j-1} \pi_j \cdots \pi_n$ into $\pi_1 \cdots \pi_i \pi_{j-1} \cdots \pi_{i+1} \pi_j \cdots \pi_n$. Reversal distance between permutations π and σ is the minimum number of reversals to transform π into σ. Analysis of genome rearrangements in molecular biology started in the late 1930's, when Dobzhansky and Sturtevant published a milestone paper presenting a rearrangement scenario with 17 inversions between the species of *Drosophila*. Analysis of genomes evolving by inversions leads to a combinatorial problem of *sorting by reversals* studied in detail recently. We study sorting of *signed* permutations by reversals, a problem that adequately models rearrangements in small genomes like chloroplast or mitochondrial DNA. The previously suggested approximation algorithms for sorting signed permutations by reversals compute the *reversal distance* between permutations with an astonishing accuracy for both simulated and biological data. We prove a duality theorem explaining this intriguing performance and show that there exists a "hidden" parameter that allows one to compute the reversal distance between signed permutations in polynomial time.

Categories and Subject Descriptors: F.1.3 [**Computation by Abstract Devices]:** Modes of Computation; G.2.1 [**Discrete Mathematics]:** Combinatorics; J.3 [**Life and Medical Sciences]:** *biology and genetics*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Computational biology, genetics

---

A preliminary version of this paper appeared in *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing* (Las Vegas, Nev., May 29–June 1). ACM, New York, 1995, pp. 178–189.

This work is supported by National Science Foundation (NSF) Young Investigator Award, NSF grant CCR 93-08567, NIH grant 1R01 HG00987, and DOE grant DE-FG02-94ER61919.

Authors' addresses: S. Hannenhalli, Bioinformatics, SmithKline Beecham Pharmaceuticals, King of Prussia, PA 19406-0939, e-mail: hannes00@mh.us.sbphrd.com, and P. A. Pevzner, Departments of Mathematics and Computer Science, University of Southern California, DRB-155, Los Angeles, CA 90089-1113, e-mail: ppevzner@hto.usc.edu.

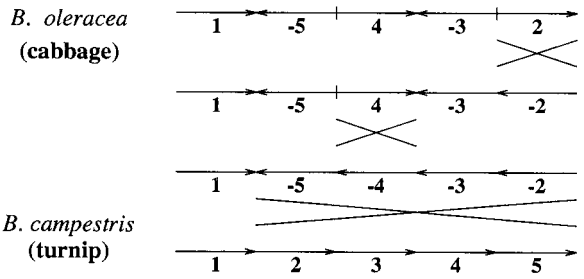

FIG. 1. "Transformation" of cabbage into turnip. Mitochondrial DNA of cabbage and turnip are composed of five conserved blocks of genes that are shuffled in cabbage as compared to turnip. Every conserved block has a direction that is shown by + or − sign.

1. *Introduction*

1.1. MOTIVATION AND BIOLOGICAL BACKGROUND. In the late 1980's, Jeffrey Palmer and colleagues discovered a remarkable and novel pattern of evolutionary change in plant organelles. They compared the mitochondrial genomes of *Brassica oleracea* (cabbage) and *Brassica campestris* (turnip), which are very closely related (many genes are 99%–99.9% identical). To their surprise, these molecules, which are almost identical in *gene* sequence, differ dramatically in gene *order* (Figure 1). This discovery and many other studies in the last decade convincingly proved that genome rearrangements is a common mode of molecular evolution in mitochondrial, chloroplast, viral and bacterial DNA (see Bafna and Pevzner, [1995]).

Every study of genome rearrangements involves solving a combinatorial "puzzle" to find a shortest series of *reversals* to transform one genome into another. (Three such reversals "transforming" cabbage into turnip are shown in Figure 1.) In cases of genomes consisting of small number of "conserved blocks," Palmer and co-authors were able to find the most parsimonious scenarios for rearrangements. However, for genomes consisting of more than 10 blocks, exhaustive search over all potential solutions is far beyond the possibilities of "pen-and-pencil" methods. As a result, Palmer and Herbon [1988] and Makaroff and Palmer [1988] overlooked the most parsimonious scenarios of rearrangements in more complicated cases like turnip vs. black mustard or turnip vs. radish (see Bafna and Pevzner [1995] for optimal solutions).

In the problem we consider, the genes are numbered $1, \ldots, n$ and the order of genes in two organisms is represented by permutations $\pi = (\pi_1 \pi_2 \cdots \pi_n)$ and $\sigma = (\sigma_1 \sigma_2 \cdots \sigma_n)$. A *reversal* $\rho(i, j)$ is the permutation

$$\begin{pmatrix} 1\ 2 \cdots i-1\ \mathbf{i\ i+1} \cdots \mathbf{j-1\ j}\ j+1 \cdots n \\ 1\ 2 \cdots i-1\ \mathbf{j\ j-1} \cdots \mathbf{i+1\ i}\ j+1 \cdots n \end{pmatrix}.$$

Clearly $\pi \cdot \rho(i, j)$ has the effect of reversing the order of genes $\pi_i \pi_{i+1} \cdots \pi_j$. In the case of *signed* permutations with + or − signs associated with every element of π, $\pi \cdot \rho(i, j)$ reverses *both* the order and signs of the elements $\pi_i \pi_{i+1} \cdots \pi_j$ (see below).

Given permutations π and σ, the *reversal distance problem* is to find a series of reversals $\rho_1, \rho_2, \ldots, \rho_t$ such that $\pi \cdot \rho_1 \cdot \rho_2 \cdots \rho_t = \sigma$ and t is minimum. We call t the *reversal distance* between π and σ. Note that the reversal distance

between π and σ equals the reversal distance between $\sigma^{-1}\pi$ and the *identity* permutation $(1\ 2\ \cdots\ n)$. *Sorting π by reversals* is the problem of finding the reversal distance, $d(\pi)$, between π and the identity permutation (Figure 2(a)).

1.2. Previous Results. Analysis of genome rearrangements provides a multitude of challenges for computer scientists; see Pevzner and Waterman [1995] for a review of open combinatorial problems motivated by genome rearrangements. A computational approach based on comparison of gene orders versus traditional comparison of DNA sequences was pioneered by Sankoff (see Sankoff et al. [1990; 1992] and Sankoff [1992]). Kececioglu and Sankoff [1995] first formulated the reversal distance problem and derived the lower and upper bounds for reversal distance. This approach led to the first approximation algorithm for sorting by reversals, which generated the exact solutions in a number of difficult instances. The problem was further studied by Bafna and Pevzner [1996], who introduced the notion of *breakpoint* graph of a permutation and revealed important links between the *maximum cycle decomposition* of this graph and reversal distance.[1]

1.3. Breakpoint Graph and Cycle Decomposition. What makes it hard to sort a permutation? In the very first computational studies of genome rearrangements, Watterson et al. [1982], and Nadeau and Taylor [1984] introduced the notion of *breakpoint* and noticed some correlations between the reversal distance and the number of breakpoints. (In fact, Sturtevant and Dobzhansky [1936] implicitly discussed these correlations 60 years ago!) Below, we define the notion of breakpoint.

Let $i \sim j$, if $|i - j| = 1$. Extend a permutation $\pi = \pi_1\pi_2 \cdots \pi_n$ by adding $\pi_0 = 0$ and $\pi_{n+1} = n + 1$. We call a pair of elements (π_i, π_{i+1}), $0 \le i \le n$, of π an *adjacency* if $\pi_i \sim \pi_{i+1}$, and a *breakpoint* if $\pi_i \not\sim \pi_{i+1}$. Since the identity permutation has no breakpoints, sorting by reversals corresponds to eliminating breakpoints. An observation that every reversal can eliminate *at most* 2 breakpoints immediately implies that $d(\pi) \ge (b(\pi)/2)$, where $b\pi)$ is the number of breakpoints in π. However, the estimate of reversal distance in terms of breakpoints is very inaccurate. Bafna and Pevzner [1996] showed that there exists another parameter (size of a maximum cycle decomposition of the breakpoint graph) that estimates reversal distance with much greater accuracy.

The *breakpoint graph* of a permutation π is an edge-colored graph $G(\pi)$ with $n + 2$ vertices $\{\pi_0, \pi_1, \ldots, \pi_n, \pi_{n+1}\} = \{0, 1, \ldots, n, n + 1\}$. We join vertices π_i and π_j by a *black* edge if (π_i, π_j) is a breakpoint in π (i.e., $\pi_i \not\sim \pi_j$ and $i \sim j$) and by a *gray* edge if (i, j) is a breakpoint in π^{-1} (i.e., $\pi_i \sim \pi_j$ and $i \not\sim j$). See Figure 2(b).

A *cycle* in an edge-colored graph G is called *alternating* if the colors of every two consecutive edges of this cycle are distinct. In the following, by *cycles*, we mean alternating cycles. The *length* of a cycle C, denoted by $l(C)$, is the number

[1] See also Kececioglu and Sankoff [1994], Kececioglu and Gusfield [1994], Kececioglu and Ravi [1995], Hannenhalli [1995], Hannenhalli and Pevzner [1995, 1996], Berman and Hannenhalli [1996], Caprara [1997], Tarjan et al. [1997], and Bafna and Pevzner [1998] for recent progress on the computational aspects of genome rearrangements, as well as Gates and Papadimitriou [1979], Even and Goldreich [1981], Jerrum [1985], Aigner and West [1987], Cohen and Blum [1993], and Heydari and Sudborough [1993] for studies of related combinatorial problems.

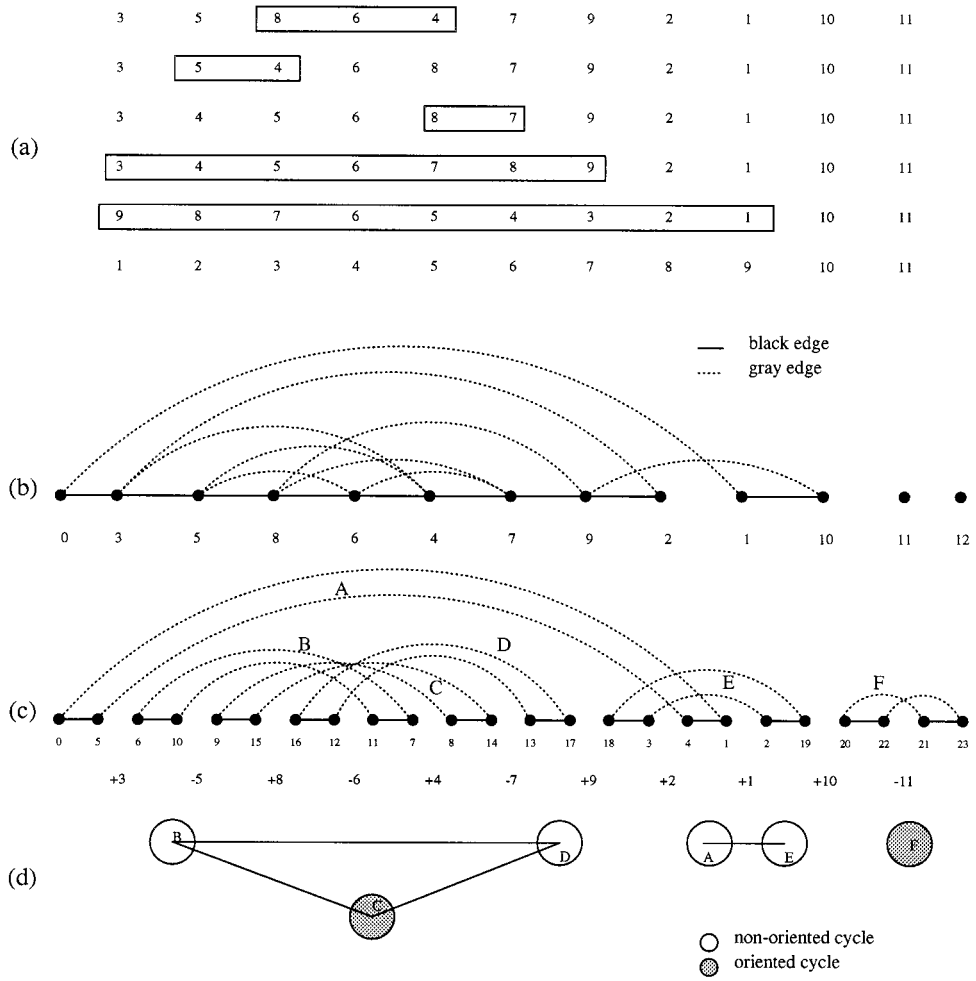

FIG. 2. (a) Optimal sorting of a permutation (3 5 8 6 4 7 9 2 1 10 11) by 5 reversals (b) Breakpoint graph of this permutation: black edges connect adjacent vertices that are not consecutive, gray edges connect consecutive vertices that are not adjacent. (c) Transformation of a signed permutation into an unsigned permutation π and the breakpoint graph $G(\pi)$; (d) Interleaving graph H_π with two oriented and one unoriented component.

of black (or equivalently, gray) edges in it. A cycle C is *short* if $l(C) = 2$ and *long* if $l(C) > 2$. A permutation π is *simple* if its breakpoint graph has no long cycles.

Consider a *cycle decomposition* of $G(\pi)$ into a *maximum* number $c(\pi)$ of edge-disjoint alternating cycles. For the permutation π in Figure 2(b) $c(\pi) = 4$ since $G(\pi)$ can be decomposed into three short cycles (8, 9, 7, 6, 8), (8, 5, 4, 7, 8), and (3, 5, 6, 4, 3) and one long cycle (0, 1, 10, 9, 2, 3, 0). Bafna and Pevzner [1996] showed that every reversal changes the parameter $b(\pi) - c(\pi)$ by at most 1 and therefore the maximum cycle decomposition provides a better bound for the reversal distance:

$$d(\pi) \geq b(\pi) - c(\pi). \tag{1}$$

However, finding a maximum cycle decomposition is a difficult problem (Kececioglu and Sankoff [1995] gave a linear programming bound for the size of the maximal cycle decomposition). Fortunately, in the biologically relevant case of *signed permutations*, this problem is trivial. Genes are *directed* fragments of DNA and a sequence of n genes in a genome is represented by a *signed* permutation on $\{1, \ldots, n\}$ with + or − sign associated with every element of π. For example, a gene order for *B. oleracea* presented in Figure 1 is modeled by a signed permutation $(+1\ -5\ +4\ -3\ +2)$. In the signed case, every reversal of a fragment changes *both* the order and the signs of the elements within that fragment (Figure 1). We are interested in the minimum number of reversals $d(\pi)$ required to transform a signed permutation π into the identity signed permutation $(+1\ +2\ \cdots\ +n)$.

1.4. New Results. Bafna and Pevzner [1996] noted that the concept of breakpoint graph extends naturally to signed permutations and devised an approximation algorithm for sorting signed permutations by reversals with performance ratio 1.5. For signed permutations, the bound (1) approximates the reversal distance extremely well for both simulated [Kececioglu and Sankoff 1994] and biological data [Bafna and Pevzner, 1995; Hannenhalli et al., 1995]. Kececioglu and Sankoff [1994] observed that an average difference between the bound (1) and the exact distance is less than 1 for random permutations. This intriguing performance raises a question whether the bound (1) overlooked a third parameter (in addition to the number of breakpoints and the size of a maximum cycle decompositions) that would allow closing the gap between $d(\pi)$ and $b(\pi) - c(\pi)$. Below, we answer this question by revealing the third "hidden" parameter (number of *hurdles* in π) making it harder to sort a permutation. We show that

$$b(\pi) - c(\pi) + h(\pi) \leq d(\pi) \leq b(\pi) - c(\pi) + h(\pi) + 1, \tag{2}$$

where $h(\pi)$ is the number of hurdles in π. Based on this result, we devise a polynomial algorithm for sorting signed permutations by reversals. This is the first polynomial algorithm for a realistic model of genome rearrangements.

The paper is organized as follows: In Section 2, we extend the definition of breakpoint graph for signed permutations and introduce the notions of oriented and unoriented cycles. In Section 3, we introduce the notion of a hurdle and prove the bound

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

Previous studies revealed that a complicated *interleaving* structure of *long* cycles in the breakpoint graph poses major difficulties in analyzing genome rearrangements. To get around this problem we develop a new technique called *equivalent transformations* of permutations (Section 4). The technique allows one to mimic sorting permutations with long cycles by sorting simple permutations. In Section 5, we prove important structural theorems for simple permutations and make the first step towards proving the bound

$$d(\pi) \leq b(\pi) - c(\pi) + h(\pi) + 1.$$

In Section 6, we associate a partial order with every permutation and show how this partial order is affected by reversals. The properties of this partial order allow us to introduce *safe* reversals that are the key operations for *clearing the hurdles* in our algorithm. In Section 7, we further develop a characterization of "hard-to-sort" permutations (called *fortresses*) that can not be sorted in $b(\pi) - c(\pi) + h(\pi)$ steps and prove the duality theorem.

$$d(\pi) = \begin{cases} b(\pi) - c(\pi) + h(\pi) + 1, & \text{if} \quad \pi \text{ is a fortress} \\ b(\pi) - c(\pi) + h(\pi), & \text{otherwise.} \end{cases}$$

Finally, in Section 8, we present a polynomial algorithm for sorting by reversals based on equivalent transformations, duality theorem and clearing the hurdles. The applications of these results are given in Hannenhalli and Pevzner [1996] where the duality theorem was used to settle two conjectures by Kececioglu and Sankoff (*reversals do not cut long strips* and *reversals do not increase the number of breakpoints*), while the algorithm for sorting signed permutations was used to analyze evolution of extensively rearranged plant and animal organelles.

2. *Breakpoint Graph of Signed Permutation*

Define a transformation from a signed permutation π of order n to an unsigned permutation π' of order $2n$ as follows: To model the directions of elements in π, replace the positive elements $+x$ by $2x - 1$, $2x$ and negative elements $-x$ by $2x$, $2x - 1$ (Figure 2(c)). We call the unsigned permutation π' the *image* of the signed permutation π. Observe that in the breakpoint graph of the image of a signed permutation, every vertex has degree at most 2.

Therefore, the cycle decomposition is unique, thus making the case of signed permutations easier to handle. We observe that the identity signed permutation of order n maps to the identity (unsigned) permutation of order $2n$, and the effect of a reversal on π can be mimicked by a reversal on π' thus implying $d(\pi) \geq d(\pi')$. In the following, by sorting of the image $\pi' = (\pi'_1\pi'_2 \cdots \pi'_{2n})$ of a signed permutation $\pi = (\pi_1\pi_2 \cdots \pi_n)$, we mean a sorting of π' by reversals $\rho(2i + 1, 2j)$, which "cut" only after even positions of π'. The effect of a reversal $\rho(2i + 1, 2j)$ on π' can be mimicked by a reversal $\rho(i + 1, j)$ on π, thus implying that $d(\pi) = d(\pi')$ if the cuts between π'_{2i-1} and π'_{2i} are forbidden (see Bafna and Pevzner [1996] for details). In the rest of the paper, all unsigned permutations we consider are images of some signed permutations. For convenience, we extend the term *signed permutation* for unsigned permutations $\pi = (\pi_1\pi_2 \cdots \pi_{2n})$ such that π_{2i-1} and π_{2i} are consecutive numbers for $1 \leq i \leq n$.

Given an arbitrary reversal ρ, denote $\Delta b \equiv \Delta b(\pi, \rho) = b(\pi\rho) - b(\pi)$ (increase in breakpoints), and $\Delta c \equiv \Delta c(\pi, \rho) = c(\pi\rho) - c(\pi)$ (increase in the size of the cycle decomposition). Bafna and Pevzner [1996] proved that for every permutation π and reversal ρ, $\Delta(b - c) \equiv \Delta b(\pi, \rho) - \Delta c(\pi, \rho) \geq -1$ (i.e., every reversal reduces the parameter $b(\pi) - c(\pi)$ by at most 1). We call a reversal *proper* if $\Delta(b - c) = -1$.

If (π_{i-1}, π_i) and (π_j, π_{j+1}) are breakpoints (black edges in $G(\pi)$), we say that reversal $\rho(i, j)$ *acts* on black edges (π_{i-1}, π_i) and (π_j, π_{j+1}). $\rho(i, j)$ is a *reversal* (*acting*) *on a cycle* C of $G(\pi)$ if the breakpoints (π_{i-1}, π_i) and $(\pi_j,$

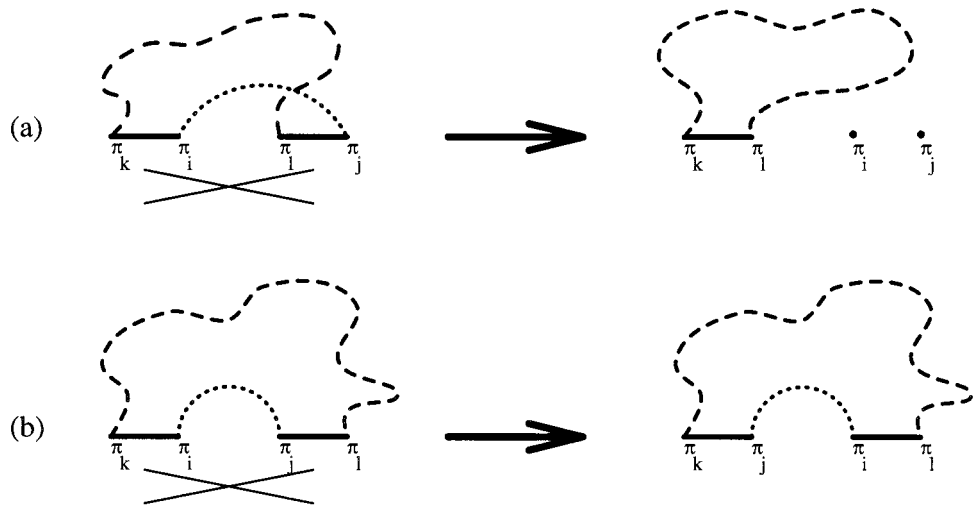

FIG. 3. (a) A proper reversals on an oriented gray edge. (b) A nonproper reversal on an unoriented gray edge.

π_{j+1}) belong to C. A gray edge g is *oriented* if a reversal acting on two black edges incident to g is proper and *unoriented*, otherwise. For example, a gray edge (8, 9) in Figure 2(c) is oriented (since a reversal acting on black edges (8, 14) and (9, 15) destroys two breakpoints and one cycle) while a gray edges (4, 5) is unoriented. To provide an intuition for the notion of an oriented edge, we state the following lemma:

LEMMA 1. *Let* (π_i, π_j) *be a gray edge incident to black edges* (π_k, π_i) *and* (π_j, π_l). *Then* (π_i, π_j) *is oriented iff* $i - k = j - l$.

PROOF. Notice that $k = i \pm 1$ and $l = j \pm 1$. If $i - k = j - l$, then either $k = i - 1, l = j - 1$ or $k = i + 1, l = j + 1$ (Figure 3(a)). Clearly $\Delta(b - c) = -1$, hence, the reversal acting on (π_i, π_j) is proper. If $i - k \neq j - l$, then either $k = i - 1, l = j + 1$ or $k = i + 1, l = j - 1$ (Figure 3(b)). In this case, $\Delta(b) = 0$ and $\Delta(c) = 0$; hence, the reversal acting on (π_i, π_j) is not proper. □

A cycle in $G(\pi)$ is *oriented* if it has an oriented gray edge and unoriented, otherwise. Cycles C and F in Figure 2(c) are oriented while cycles A, B, D, and E are unoriented. Clearly, there is no proper reversal acting on an unoriented cycle. It is easy to see that a permutation has a proper reversal iff it has an oriented cycle.

3. *Interleaving Graph and Hurdles*

Gray edges (π_i, π_j) and (π_k, π_t) in $G(\pi)$ are *interleaving* if the intervals $[i, j]$ and $[k, t]$ overlap but neither of them contains the other. For example, edges (4, 5) and (18, 19) in Figure 2(c) are interleaving while edges (4, 5) and (22, 23) or (4, 5) and (16, 17) are noninterleaving. Cycles C_1 and C_2 are *interleaving* if there exist interleaving gray edges $g_1 \in C_1$ and $g_2 \in C_2$.

Let $\mathcal{C}_\pi$ be the set of cycles in the breakpoint graph of a permutation π. Define

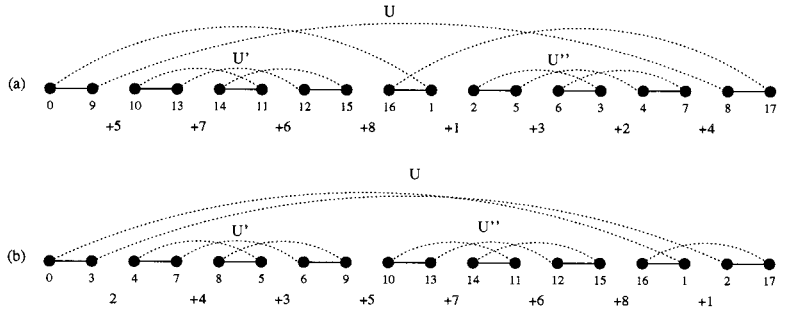

FIG. 4. (a) Unoriented component U separates U' and U'' by virtue of the edge (0, 1); (b) Hurdle U does not separates U' and U''.

an *interleaving* graph $H_\pi(\mathcal{C}_\pi, \mathcal{I}_\pi)$ of π with the edge set

$$\mathcal{I}_\pi = \{(C_1, C_2) : C_1 \text{ and } C_2 \text{ are interleaving cycles in } G(\pi)\}.$$

Figure 2(d) shows the interleaving graph H_π consisting of three connected components. The vertex set of H_π is partitioned into *oriented* and *unoriented* vertices (cycles in $\mathcal{C}_\pi$). A connected component of H_π is *oriented* if it has at least one oriented vertex and *unoriented* otherwise. For a connected component U, define leftmost and rightmost positions of U as

$$U_{\min} = \min_{C \in U} \min_{\pi_i \in C} i \qquad \text{and} \qquad U_{\max} = \max_{C \in U} \max_{\pi_i \in C} i.$$

Let $Extent(U)$ be the interval $[U_{\min}, U_{\max}]$. For example, a component U containing cycles B, C and D in Figure 2(c) has leftmost vertex $\pi_2 = 6$ and rightmost vertex $\pi_{13} = 17$; therefore, $Extent(U) = [2, 13]$.

We say that a component U *separates* components U', U'' in π if there exists a gray edge (π_i, π_j) in U such that $Extent(U') \subset [i, j]$, but $Extent(U'') \cap [i, j] = \emptyset$. For example, the component U in Figure 4(a) separates the components U' and U''.

Let $\prec$ be a partial order on a set P. An element $x \in P$ is called a *minimal* element in $\prec$ if there is no element $y \in P$ with $y \prec x$. An element $x \in P$ is the *greatest* in $\prec$ if $y \prec x$ for every $y \in P$.

Consider the set of unoriented components $\mathcal{U}_\pi$ in H_π and define the *containment* partial order on this set, that is, $U \prec W$ iff $Extent(U) \subset Extent(W)$ for U, $W \in \mathcal{U}_\pi$. A *hurdle* is defined as follows: An unoriented component $U \in \mathcal{U}_\pi$ that is a minimal element in $\prec$ is a hurdle, called *minimal hurdle*. The unoriented component $U \in \mathcal{U}_\pi$ that is the greatest element in $\prec$ is a hurdle, called the *greatest hurdle*, if U does not separate any two minimal hurdles. Obviously, there can be at most one greatest hurdle. We wish to emphasize that the notions of minimal and greatest hurdles become equivalent if we circularize the linear permutation. Let $h(\pi)$ be the total number of hurdles in π. Permutation π in Figure 2(c) has one unoriented component and $h(\pi) = 1$. Permutation in Figure 4(b) has two minimal and one greatest hurdles ($h(\pi) = 3$). Permutation in Figure 4(a) has 2 minimal and no greatest hurdles ($h(\pi) = 2$) since the greatest unoriented component U in Figure 4(a) separates U' and U''.

The following theorem further improves the bound (1):

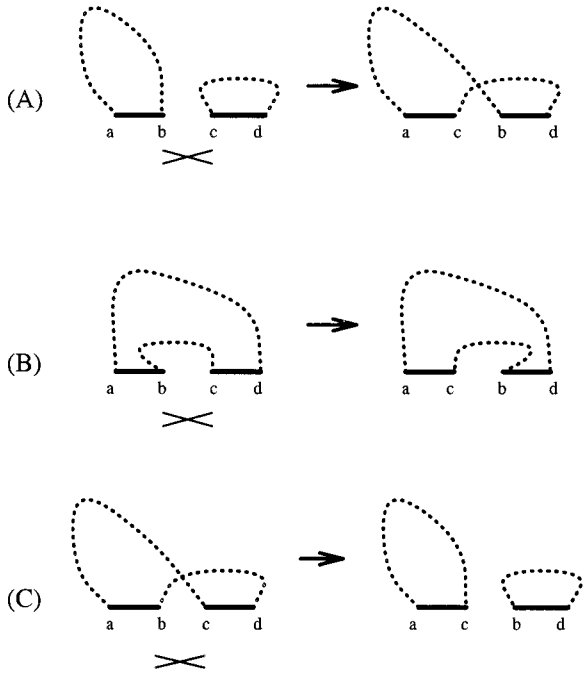

FIG. 5. (A) For reversals acting on two cycles, $\Delta(b - c) = 1$. (B) For reversals acting on an unoriented cycle, $\Delta(b - c) = 0$. (C) For reversals acting on an oriented cycle, $\Delta(b - c) = -1$.

THEOREM 1. *For arbitrary (signed) permutation* π, $d(\pi) \geq b(\pi) - c(\pi) + h(\pi)$.

PROOF. Given an arbitrary reversal ρ, denote $\Delta h \equiv \Delta h(\pi, \rho) = h(\pi\rho) - h(\pi)$. Clearly, every reversal ρ acts on black edges of at most two hurdles and therefore ρ "destroys" (i.e., transforms an unoriented component into an oriented component) at most two minimal hurdles. Note that, if ρ destroys two minimal hurdles in $\mathcal{U}_\pi$, then ρ can not destroy the greatest hurdle in $\mathcal{U}_\pi$ (see the definition of the greatest hurdle). Therefore, $\Delta h \geq -2$ for every reversal ρ.

Bafna and Pevzner [1993] proved that $\Delta(b - c) \in \{-1, 0, 1\}$ (Figure 5). If $\Delta(b - c) = -1$, then ρ acts on an oriented cycle and hence it does not destroy any hurdles in π. Therefore, $\Delta h \geq 0$ and $\Delta(b - c + h) \equiv \Delta b - \Delta c + \Delta h \geq -1$. If $\Delta(b - c) = 0$, then ρ acts on a cycle and therefore it affects at most one hurdle. It implies $\Delta h \geq -1$ and $\Delta(b - c + h) \geq -1$. If $\Delta(b - c) = 1$, then $\Delta(b - c + h) \geq -1$ since $\Delta h \geq -2$ for every reversal ρ.

Therefore, for an arbitrary reversal ρ, $\Delta(b - c + h) \geq -1$ thus implying $d(\pi) \geq b(\pi) - c(\pi) + h(\pi)$. □

In the following, we show that the lower bound $d(\pi) \geq b(\pi) - c(\pi) + h(\pi)$ is very tight. As the first step towards the upper bound $d(\pi) \leq b(\pi) - c(\pi) + h(\pi) + 1$, we develop the technique called *equivalent transformations* of permutations.

4. *Equivalent Transformations of Permutations*

Previous studies revealed that complicated interleaving structure of long cycles in the breakpoint graphs poses serious difficulties in analyzing sorting by reversals

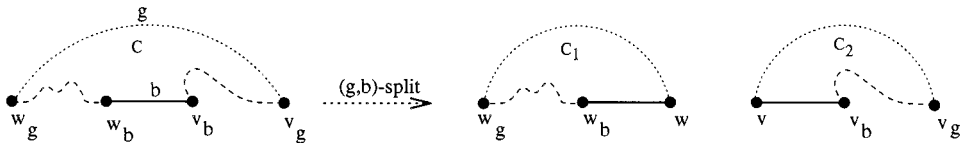

FIG. 6. Example of a (g, b)-split.

[Bafna and Pevzner 1996] and by transpositions [Bafna and Pevzner 1995]. To get around this problem, we introduce equivalent transformations of permutations based on the following idea. If a permutation $\pi \equiv \pi(0)$ has a long cycle, transform it into a new permutation $\pi(1)$ by "breaking" this long cycle into two smaller cycles. Continue with $\pi(1)$ in the same manner and form a sequence of permutations $\pi \equiv \pi(0), \pi(1), \ldots, \pi(k) \equiv \sigma$ ending with a simple permutation (i.e., one having no long cycles). In this section, we show that these transformations can be arranged in such a way that every sorting of σ mimics a sorting of π with the same number of reversals. In the following sections, we show how to optimally sort simple permutations. Optimal sorting of the *simple* permutation σ mimics optimal sorting of the *arbitrary* permutation π leading to a polynomial algorithm for sorting by reversals.

Let $b = (v_b, w_b)$ be a black edge and $g = (w_g, v_g)$ be a gray edge belonging to a cycle $C = \ldots, v_b, w_b, \ldots, w_g, v_g, \ldots$ in the breakpoint graph $G(\pi)$ of a permutation π. A (g, b)-*split* of $G(\pi)$ is a new graph $\hat{G}(\pi)$ obtained from $G(\pi)$ by

—removing edges g and b,

—adding two new vertices v and w,

—adding two new black edges (v_b, v) and (w, w_b),

—adding two new gray edges (w_g, w) and (v, v_g).

Figure 6 shows a (g, b)-split transforming a cycle C in $G(\pi)$ into cycles C_1 and C_2 in $\hat{G}(\pi)$. If $G(\pi)$ is a breakpoint graph of a signed permutation π, then every (g, b)-split of $G(\pi)$ corresponds to the breakpoint graph of a signed *generalized* permutation $\hat{\pi}$ such that $\hat{G}(\pi) = G(\hat{\pi})$. Below, we define generalized permutations and describe the *padding* procedure to find a generalized permutation $\hat{\pi}$ corresponding to a (g, b)-split of G.

A generalized permutation $\pi = (\pi_1 \pi_2 \cdots \pi_n)$ is a permutation of arbitrary distinct *reals* (versus permutations of *integers* $\{1, 2, \ldots, n\}$ we considered before). In this section, by *permutations*, we mean generalized permutations, and by *identity generalized permutation* we mean a generalized permutation $\pi = (\pi_1 \pi_2 \cdots \pi_n)$ with $\pi_i < \pi_{i+1}$ for $1 \le i \le n - 1$. Extend a permutation $\pi = (\pi_1 \pi_2 \cdots \pi_n)$ by adding $\pi_0 = \min_{1 \le i \le n} \pi_i - 1$ and $\pi_{n+1} = \max_{1 \le i \le n} \pi_i + 1$. Elements π_j and π_k of π are *consecutive* if there is no element π_l such that $\pi_j < \pi_l < \pi_k$ for $1 \le l \le n$. Elements π_i and π_{i+1} of π are *adjacent* for $0 \le i \le n$. The *breakpoint graph* of a (generalized) permutation $\pi = (\pi_1 \pi_2 \cdots \pi_n)$ is defined as the graph on vertices $\{\pi_0, \pi_1, \ldots, \pi_n, \pi_{n+1}\}$ with black edges between adjacent elements that are not consecutive and gray edges between consecutive elements that are not adjacent. Obviously the definition of the breakpoint graph for generalized permutation is consistent with the notion of the breakpoint graph described earlier.

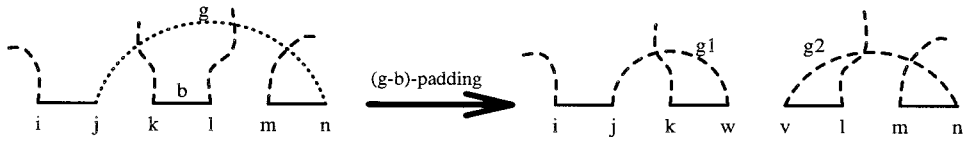

FIG. 7. A (g, b)-padding deletes an oriented edge g and adds an oriented edge g_1 and unoriented edge g_2.

Let $b = (\pi_{i+1}, \pi_i)$ be a black edge and $g = (\pi_j, \pi_k)$ be a gray edge belonging to a cycle $C = \ldots, \pi_{i+1}, \pi_i, \ldots, \pi_j, \pi_k, \ldots$ in the breakpoint graph $G(\pi)$. Define $\Delta = \pi_k - \pi_j$ and let $v = \pi_j + (\Delta/3)$, $w = \pi_k - (\Delta/3)$. A (g, b)-*padding* of $\pi = (\pi_1 \pi_2 \cdots \pi_n)$ is a permutation on $n + 2$ elements obtained from π by inserting v and w after the ith element of π $(0 \leq i \leq n)$:

$$\hat{\pi} = (\pi_1 \pi_2 \cdots \pi_i v w \pi_{i+1} \cdots \pi_n).$$

Note that v and w are both consecutive and adjacent in $\hat{\pi}$ thus implying that if π is (the image of) a signed permutation then $\hat{\pi}$ is also (the image of) a signed permutation. The (g, b)-split in Figure 6 corresponds to (g, b)-padding for $g = (w_g, v_g)$ and $b = (v_b, w_b)$. The following lemma establishes the correspondence between (g, b)-paddings and (g, b)-splits.[2]

LEMMA 2. $\hat{G}(\pi) = G(\hat{\pi})$.

If g and b are nonincident edges of a *long* cycle C in $G(\pi)$, then the (g, b)-padding breaks C into two *smaller* cycles in $G(\hat{\pi})$. Therefore, paddings may be used to transform an arbitrary permutation π into a simple permutation. Note that $b(\hat{\pi}) = b(\pi) + 1$ and $c(\hat{\pi}) = c(\pi) + 1$. Below, we prove that, for every permutation with a long cycle, there exists a padding on nonincident edges of this cycle such that $h(\hat{\pi}) = h(\pi)$, thus indicating that padding provides a way to eliminate long cycles in a permutation without changing the parameter $b(\pi) - c(\pi) + h(\pi)$. First, we need a series of technical lemmas.

LEMMA 3. *Let a (g, b)-padding on a cycle C in $G(\pi)$ delete the gray edge g and add two new gray edges g_1 and g_2. If g is oriented, then either g_1 or g_2 is oriented in $G(\hat{\pi})$. If C is unoriented, then both g_1 and g_2 are unoriented in $G(\hat{\pi})$.*

The case when g is oriented is illustrated in Figure 7.

LEMMA 4. *Let a (g, b)-padding break a cycle C in $G(\pi)$ into cycles C_1 and C_2 in $G(\hat{\pi})$. Then C is oriented iff either C_1 or C_2 is oriented.*

PROOF. Note that a (g, b)-padding preserves orientation of gray edges in $G(\hat{\pi})$, which are "inherited" from $G(\pi)$ (Lemma 1). If C is oriented, then it has an oriented gray edge. If this edge is different from g, then it remains oriented in a (g, b)-padding of π and therefore a cycle (C_1 or C_2) containing this edge is oriented. If $g = (w_g, v_g)$ is the only oriented gray edge in C, then (g, b)-padding

[2] Of course, a (g, b)-padding of a permutation $\pi = (\pi_1 \pi_2 \cdots \pi_n)$ on $\{1, 2, \ldots, n\}$ can be modeled as a permutation $\hat{\pi} = (\hat{\pi}_1 \hat{\pi}_2 \cdots \hat{\pi}_i v w \hat{\pi}_{i+1} \cdots \hat{\pi}_n)$ on $\{1, 2, \ldots, n + 2\}$ where $v = \pi_j + 1$, $w = \pi_k + 1$, $\hat{\pi}_i = \pi_i + 2$ if $\pi_i > \min\{\pi_j, \pi_k\}$ and $\hat{\pi}_i = \pi_i$ otherwise. The generalized permutations were introduced to make the following "mimicking" procedure more intuitive.

adds two new gray edges $((w_g, w)$ and $(v, v_g))$ to $G(\hat{\pi})$, one of which is oriented (Lemma 3). Therefore, a cycle (C_1 or C_2) containing this edge is oriented.

If C is an unoriented cycle, then all edges of C_1 and C_2 "inherited" from C remain unoriented. Lemma 3 implies that new edges $((w_g, w)$ and $(v, v_g))$ in C_1 and C_2 are also unoriented. □

The following lemma shows that paddings preserve interleaving of gray edges:

LEMMA 5. *Let g' and g'' be two gray edges of $G(\pi)$ different from g. Then g' and g'' are interleaving in π iff g' and g'' are interleaving in a (g, b)-padding of π.*

This lemma immediately implies the following:

LEMMA 6. *Let a (g, b)-padding breaks a cycle C in $G(\pi)$ into cycles C_1 and C_2 in $G(\hat{\pi})$. Then every cycle D interleaving with C in $G(\pi)$ interleaves with either C_1 or C_2 in $G(\hat{\pi})$.*

PROOF. Let $d \in D$ and $c \in C$ be interleaving gray edges in $G(\pi)$. If c is different from g, then Lemma 5 implies that d and c are interleaving in $G(\hat{\pi})$ and therefore D interleaves with either C_1 or C_2. If $c = g$, then it is easy to see that one of the new gray edges in $G(\hat{\pi})$ interleaves with d and therefore D interleaves with either C_1 or C_2 in $G(\hat{\pi})$. □

LEMMA 7. *For every gray edge g, there exists a gray edge f interleaving with g in $G(\pi)$.*

PROOF. Let $g = (\pi_i, \pi_j)$. Gray edges and adjacencies in the breakpoint graph form a path from vertex 0 to $n + 1$ that visits *all* vertices in $G(\pi)$. In particular, this path visits the interval $i + 1, \ldots, j - 1$ between the endpoints of g and thus contains a gray edge f connecting a vertex from this interval with the "outside" of this interval (i.e., the set $\{0, 1, \ldots, i - 1, j + 1, \ldots, n + 1\}$). □

LEMMA 8. *Let C be a cycle in $G(\pi)$ and $g \notin C$ be a gray edge in $G(\pi)$. Then g interleaves with an even number of gray edges in C.*

PROOF. Let $g = (\pi_i, \pi_j)$. The gray edges of cycle C enter and leave the interval $i + 1, \ldots, j - 1$ the same number of times (to leave a room, one first needs to enter it). □

A (g, b)-padding ϕ transforming π into $\hat{\pi}$ (i.e., $\hat{\pi} = \pi \cdot \phi$) is *safe* if it acts on nonincident edges of a long cycle and $h(\pi) = h(\hat{\pi})$. Clearly, every safe padding breaks a long cycle into two smaller cycles.

THEOREM 2. *If C is a long cycle in $G(\pi)$, then there exists a safe (g, b)-padding acting on C.*

PROOF. If C has a pair of interleaving gray edges $g_1, g_2 \in C$, then removing these edges transforms C into two paths. Since C is a long cycle at least one of these paths contains a gray-edge g. Pick a black-edge b from the other path and consider the (g, b)-padding transforming π into $\hat{\pi}$ (clearly, g and b are nonincident edges). This (g, b)-padding breaks C into cycles C_1 and C_2 in $G(\hat{\pi})$ with g_1 and g_2 belonging to different cycles C_1 and C_2. By Lemma 5, g_1 and g_2 are interleaving, thus implying that C_1 and C_2 are interleaving. Also this (g, b)-padding does not "break" the component K in H_π containing the cycle C since by

Lemma 6 all cycles from K belong to the component of H_π containing C_1 and C_2. Moreover, according to Lemma 4 the orientation of this component in H_π and $H_{\hat{\pi}}$ is the same. Therefore, the chosen (g, b)-padding preserves the set of hurdles and $h(\pi) = h(\hat{\pi})$.

If all gray edges of C are mutually noninterleaving, then C is an unoriented cycle (refer to Figure 3). Lemmas 7 and 8 imply that there exists a gray edge $e \in C'$ interleaving with at least two gray edges $g_1, g_2 \in C$. Removing g_1 and g_2 transforms C into two paths and since C is a long cycle at least one of these paths contains a gray edge g. Pick a black-edge b from the other path and consider the (g, b)-padding of π. This padding breaks C into cycles C_1 and C_2 in $G(\hat{\pi})$ with g_1 and g_2 belonging to different cycles C_1 and C_2. By Lemma 5, both C_1 and C_2 interleave with C' in $\hat{\pi}$. Therefore, this (g, b)-padding does not break the component K in H_π containing C and C'. Moreover, according to Lemma 4, both C_1 and C_2 are unoriented thus implying that the orientation of this component in H_π and $H_{\hat{\pi}}$ is the same. Therefore, the chosen (g, b)-padding preserves the set of hurdles and hence, $h(\pi) = h(\hat{\pi})$. □

A permutation π is *equivalent* to a permutation σ ($\pi \rightsquigarrow \sigma$) if there exists a series of permutations $\pi \equiv \pi(0), \pi(1), \ldots, \pi(k) \equiv \sigma$ such that $\pi(i + 1) = \pi(i) \cdot \phi(i)$ for a safe (g, b)-padding $\phi(i)$ acting on π_i $(0 \leq i \leq k - 1)$.

THEOREM 3. *For every permutation, there exists an equivalent simple permutation.*

PROOF. Define the *complexity* of a permutation π as $\Sigma_{C \in \mathcal{C}_\pi} (l(C) - 2)$ where $\mathcal{C}_\pi$ is the set of cycles in $G(\pi)$ and $l(C)$ is the length of a cycle C. The complexity of a simple permutation is 0. Note that every padding on nonincident edges of a long cycle C breaks C into cycles C_1 and C_2 with $l(C) = l(C_1) + l(C_2) - 1$. Therefore,

$$(l(C) - 2) = (l(C_1) - 2) + (l(C_2) - 2) + 1,$$

implying that a padding on nonincident edges of a cycle reduces the complexity of permutations. This observation and Theorem 2 imply that every permutation with long cycles can be transformed into a permutation without long cycles by a series of paddings preserving $b(\pi) - c(\pi) + h(\pi)$. □

Let $\hat{\pi}$ be a (g, b)-padding of π and ρ be a reversal acting on two black edges of $\hat{\pi}$. Then ρ can be mimicked on π by ignoring the padded elements. We need a generalization of this observation.

A sequence of permutations $\pi \equiv \pi(0), \pi(1), \ldots, \pi(k) \equiv \sigma$ is called a *generalized sorting* of π if σ is the identity (generalized) permutation and $\pi(i + 1)$ is obtained from $\pi(i)$ *either* by a reversal or by a padding. Note that reversals and paddings in generalized sorting of π may interleave. Interleaving of reversals and paddings in generalized sorting is necessary to mimic sorting of the (genuine) permutation since a reversal may merge two short cycles into a long one.

LEMMA 9. *Every generalized sorting of π mimics a (genuine) sorting of π with the same number of reversals.*

PROOF. Ignore padded elements. □

In the following, we show how to find a generalized sorting of a permutation π by a series of paddings and reversals containing $d(\pi)$ reversals. Lemma 9 implies that this generalized sorting of π mimics an optimal (genuine) sorting of π.

5. *Safe Reversals in Oriented Components*

Recall that for an arbitrary reversal, $\Delta(b - c + h) \geq -1$ (see proof of Theorem 1). A reversal ρ is *safe* if $\Delta(b - c + h) = -1$. In the following, we prove the existence of a safe reversal acting on a cycle in an oriented component by analyzing actions of reversals on simple permutations. In this section, by cycles, we mean *short* cycles and by permutations we mean simple permutations.

Denote the set of all cycles interleaving with a cycle C in $G(\pi)$ as $V(C)$ (i.e., $V(C)$ is the set of vertices adjacent to C in H_π). Define the sets of edges in the subgraph of H_π induced by $V(C)$

$$E(C) = \{(C_1, C_2) : C_1, C_2 \in V(C) \text{ and } C_1 \text{ interleaves with } C_2 \text{ in } \pi\}$$

and its complement

$$\bar{E}(C) = \{(C_1, C_2) : C_1, C_2 \in V(C) \text{ and } C_1 \text{ does not interleave with } C_2 \text{ in } \pi\}.$$

A reversal ρ acting on an oriented (short) cycle C "destroys" C (i.e., removes the edges of C from $G(\pi)$) and transforms every other cycle in $G(\pi)$ into a corresponding cycle on the same vertices in $G(\pi\rho)$. As a result ρ transforms the interleaving graph $H_\pi(\mathcal{C}_\pi, \mathcal{I}_\pi)$ of π into the interleaving graph $H_{\pi\rho}(\mathcal{C}_\pi \backslash C, \mathcal{I}_{\pi\rho})$ of $\pi\rho$. This transformation results in complementing the subgraph induced by $V(C)$ as described by the following lemma (Figure 8). We denote $\bar{\mathcal{I}}_\pi = \mathcal{I}_\pi \backslash \{(C, D) : D \in V(C)\}$.

LEMMA 10. *Let ρ be a reversal acting on an oriented (short) cycle C. Then*

—$\mathcal{I}_{\pi\rho} = (\bar{\mathcal{I}}_\pi \backslash E(C)) \cup \bar{E}(C)$, *that is ρ removes edges $E(C)$ and adds edges $\bar{E}(C)$ to transform H_π into $H_{\pi\rho}$.*

—*ρ changes the orientation of a cycle $D \in \mathcal{C}_\pi$ iff $D \in V(C)$.*

Lemma 10 immediately implies the following:

LEMMA 11. *Let ρ be a reversal acting on a cycle C and A, B be nonadjacent vertices in $H_{\pi\rho}$. Then (A, B) is an edge in H_π iff $A, B \in V(C)$.*

Let K be an oriented component of H_π and let $\mathcal{R}(K)$ be a set of reversals acting on oriented cycles from K. Assume that a reversal $\rho \in \mathcal{R}(K)$ "breaks" K into a number of connected components $K_1(\rho), K_2(\rho), \ldots$ in $H_{\pi\rho}$ and the first m of these components are unoriented. If $m > 0$, then ρ may be unsafe since some of the components $K_1(\rho), \ldots, K_m(\rho)$ may form new hurdles in $\pi\rho$ thus increasing $h(\pi\rho)$ as compared to $h(\pi)$. In the following, we show that there is a flexibility in choosing a reversal from the set $\mathcal{R}(K)$ allowing one to substitute a safe reversal σ for an unsafe reversal ρ.

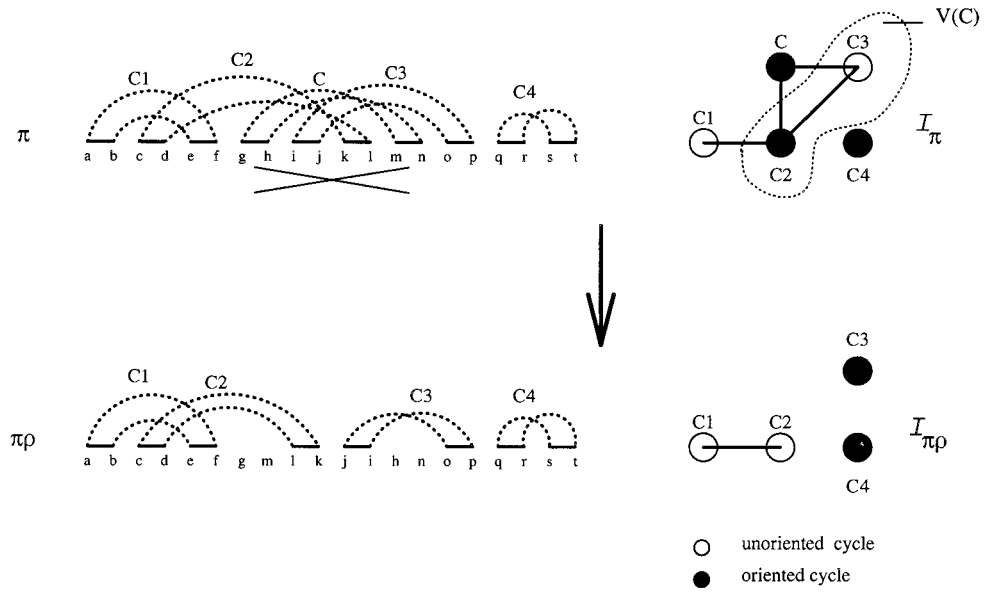

FIG. 8. Reversal on a cycle C (i) deletes vertex C from the interleaving graph; (ii) changes the orientation of vertices in $V(C)$; (iii) complements the subgraph induced by $V(C)$.

LEMMA 12. *Let ρ and σ be the reversals acting on two interleaving oriented cycles C and C' in $G(\pi)$, respectively. If C' belongs to an unoriented component $K_1(\rho)$ in $H_{\pi\rho}$ then*

—*every two vertices outside $K_1(\rho)$ which are adjacent in $H_{\pi\rho}$ are also adjacent in $H_{\pi\sigma}$.*

—*orientation of vertices outside $K_1(\rho)$ does not change in $H_{\pi\sigma}$ as compared to $H_{\pi\rho}$.*

PROOF. Let D, E be two vertices outside $K_1(\rho)$ connected by an edge in $H_{\pi\rho}$. If one of these vertices, say D, does not belong to $V(C)$ in H_π, then Lemma 11 implies (i) (C', D) is not an edge in H_π and (ii) (D, E) is an edge in H_π. Therefore, by Lemma 10, reversal σ preserves the edge (D, E) in $H_{\pi\sigma}$. If both vertices D and E belong to $V(C)$, then Lemma 10 implies that (D, E) is not an edge in H_π. Since vertex C' and vertices D, E are in different components of $H_{\pi\rho}$, Lemma 11 implies that (C', D) and (C', E) are edges in H_π. Therefore, by Lemma 10, (D, E) is an edge in $H_{\pi\sigma}$. In both cases, σ preserves the edge (D, E) in $H_{\pi\sigma}$ and the first part of the lemma holds.

Lemma 11 implies that for every vertex D outside $K_1(\rho)$, $D \in$ V(C) *iff* $D \in V(C')$. This observation and Lemma 10 imply that the orientation of vertices outside $K_1(\rho)$ does not change in $H_{\pi\sigma}$ as compared to $H_{\pi\rho}$. □

LEMMA 13. *Every unoriented component in the interleaving graph* (*of a simple permutation*) *contains at least* 2 *vertices*.

PROOF. By Lemma 7, every gray edge in $G(\pi)$ has an interleaving gray edge. Therefore every unoriented (short) cycle in $G(\pi)$ has an interleaving cycle. □

THEOREM 4. *For every oriented component K in H_π there exists a (safe) reversal $\rho \in \mathcal{R}(K)$ such that all components $K_1(\rho), K_2(\rho), \ldots$ are oriented in $H_{\pi\rho}$.*

PROOF. Assume that a reversal $\rho \in \mathcal{R}(K)$ "breaks" K into a number of connected components $K_1(\rho), K_2(\rho), \ldots$ in $H_{\pi\rho}$ and the first m of these components are unoriented. Denote the overall number of vertices in these unoriented components as $index\ (\rho) = \sum_{i=1}^{m} |K_i(\rho)|$ where $|K_i(\rho)|$ is the number of vertices in $K_i(\rho)$. Let ρ be a reversal such that

$$index(\rho) = \min_{\sigma \in \mathcal{R}(K)} index(\sigma).$$

This reversal acts on a cycle C and breaks K into a number of components. If all these components are oriented (i.e., $index(\rho) = 0$), the theorem holds. Otherwise, $index(\rho) > 0$ and let $K_1(\rho), \ldots, K_m(\rho)$ $(m \geq 1)$ be unoriented components in $H_{\pi\rho}$. Below, we find another reversal $\sigma \in \mathcal{R}(K)$ with $index(\sigma) < index(\rho)$, a contradiction.

Let V_1 be the set of vertices of the component $K_1(\rho)$ in $H_{\pi\rho}$. Note that $K_1(\rho)$ contains at least one vertex from $V(C)$ and consider the (nonempty) set $V = V_1 \cap V(C)$ of vertices from component $K_1(\rho)$ adjacent to C in H_π. Since $K_1(\rho)$ is an unoriented component in $\pi\rho$ all cycles from V are oriented in π and all cycles from $V_1 \backslash V$ are unoriented in π (Lemma 10). Let C' be an (oriented) cycle in V and let σ be the reversal acting on C' in $G(\pi)$. Lemma 12 implies that for $i \geq 2$ all edges of the component $K_i(\rho)$ in $H_{\pi\rho}$ are preserved in $H_{\pi\sigma}$ and the orientation of vertices in $K_i(\rho)$ does not change in $H_{\pi\sigma}$ as compared to $H_{\pi\rho}$. Therefore all unoriented components $K_{m+1}(\rho), K_{m+2}(\rho), \ldots$ of $\pi\rho$ "survive" in $\pi\sigma$ and

$$index(\sigma) \leq index(\rho).$$

Below, we prove that there exists a reversal σ acting on a cycle from V such that $index(\sigma) < index(\rho)$, a contradiction.

If $V_1 \neq V$, then there exists an edge between an (oriented) cycle $C' \in V$ and an (unoriented) cycle $C'' \in V_1 \backslash V$ in $\mathcal{J}_\pi$. Lemma 10 implies that a reversal σ acting on C' in π orients the cycle C'' in $G(\sigma\pi)$. This observation and Lemma 12 imply that σ reduces $index(\sigma)$ by at least 1 as compared to $index(\rho)$, a contradiction (refer to Figure 9(a)).

If $V_1 = V$ (all cycles of K_1 interleave with C), then there exist at least two vertices in $V(C)$ (Lemma 13). Moreover, there exist (oriented) cycles $C', C'' \in V_1$ such that (C', C'') are not interleaving in π (otherwise, Lemma 10 would imply that $K_1(\rho)$ is a graph with no edges, a contradiction to connectivity of $K_1(\rho)$). Define σ as a reversal acting on C'. Lemma 10 implies that σ preserves the orientation of C'' thus reducing $index(\sigma)$ by at least 1 as compared to $index(\rho)$, a contradiction (refer to Figure 9(b)).

The above discussion implies that there exists a reversal $\rho \in \mathcal{R}(K)$ such that $index(\rho) = 0$, that is, ρ does not create new unoriented components. Therefore, $\Delta b(\pi, \rho) = -2$, $\Delta c(\pi, \rho) = -1$ and $\Delta h(\pi, \rho) = 0$ implying that ρ is safe. □

6. *Clearing the Hurdles*

If π has an oriented component, then Theorem 4 implies that there exists a safe reversal in π. In this section, we search for a safe reversal in the absence of any

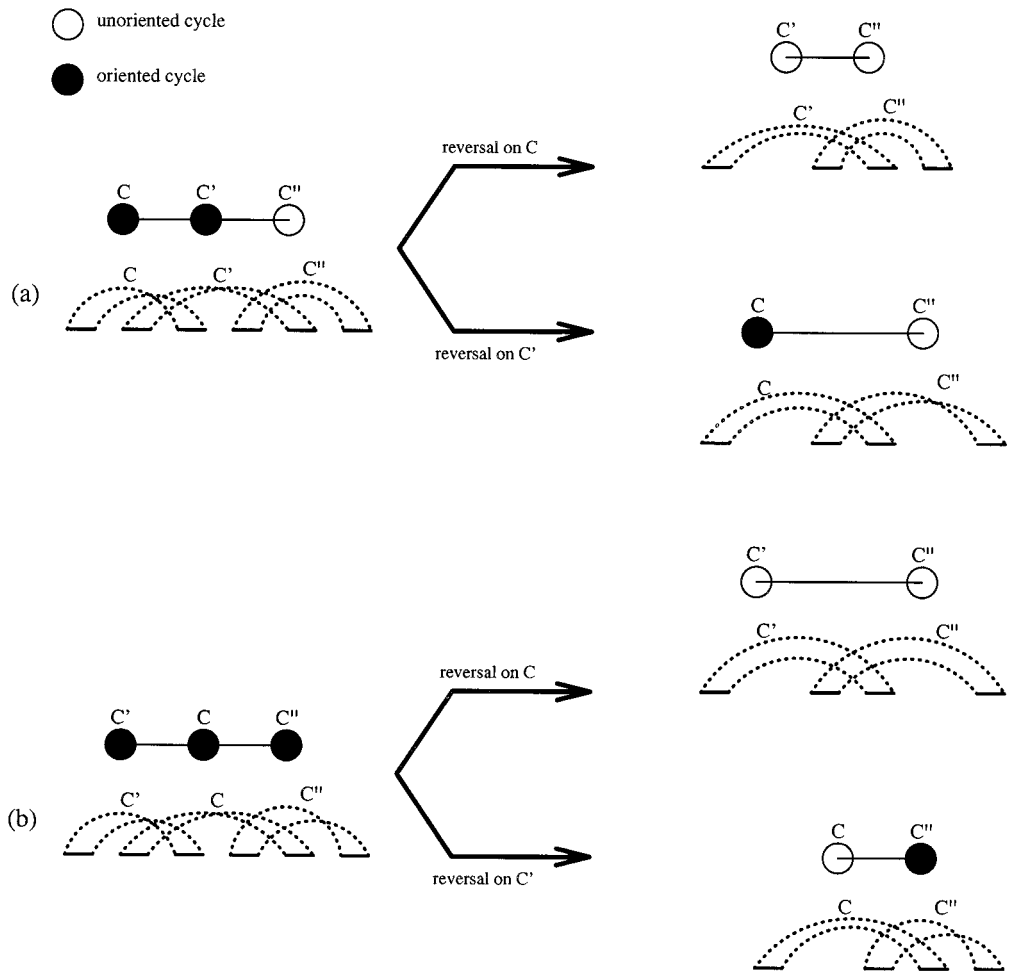

FIG. 9. Proof of Theorem 4. A reversal on a cycle C' has a smaller *index* than a reversal on a cycle C.

oriented component. Let $\prec$ be a partial order on a set P. We say x is *covered* by y in P if $x \prec y$ and there is no element $z \in P$ for which $x \prec z \prec y$. The *cover graph* Ω of $\prec$ is an (undirected) graph with vertex set P and edge set $\{(x, y) : x, y \in P$ and x is covered by $y\}$.

Let $\mathcal{U}_\pi$ be the set of unoriented components in H_π and let $Extent(U) = [U_{min}, U_{max}]$ be the interval between the leftmost and rightmost positions in an unoriented component $U \in \mathcal{U}_\pi$ (see Section 3). Define $\bar{U}_{min} = \min_{U \in \mathcal{U}_\pi} U_{min}$, $\bar{U}_{max} = \max_{U \in \mathcal{U}_\pi} U_{max}$ and let $[\bar{U}_{min}, \bar{U}_{max}]$ be the interval between the leftmost and rightmost positions among all the unoriented components of π. Let $\bar{U}$ be an (*artificial*) component associated with the interval $[\bar{U}_{min}, \bar{U}_{max}]$.

Define $\bar{\mathcal{U}}_\pi$ as the set of $|\mathcal{U}_\pi| + 1$ elements consisting of $|\mathcal{U}_\pi|$ elements $\{U : U \in \mathcal{U}_\pi\}$ combined with an additional element $\bar{U}$. Let $\prec\equiv\prec_\pi$ be the containment partial order on $\bar{\mathcal{U}}_\pi$ defined by the rule $U \prec W$ iff $Extent(U) \subset Extent(W)$ for $U, W \in \bar{\mathcal{U}}_\pi$. If there exists the *greatest* unoriented component U in π (i.e., $Extent(U) = [\bar{U}_{min}, \bar{U}_{max}]$), we assume that there exist two elements ("real"

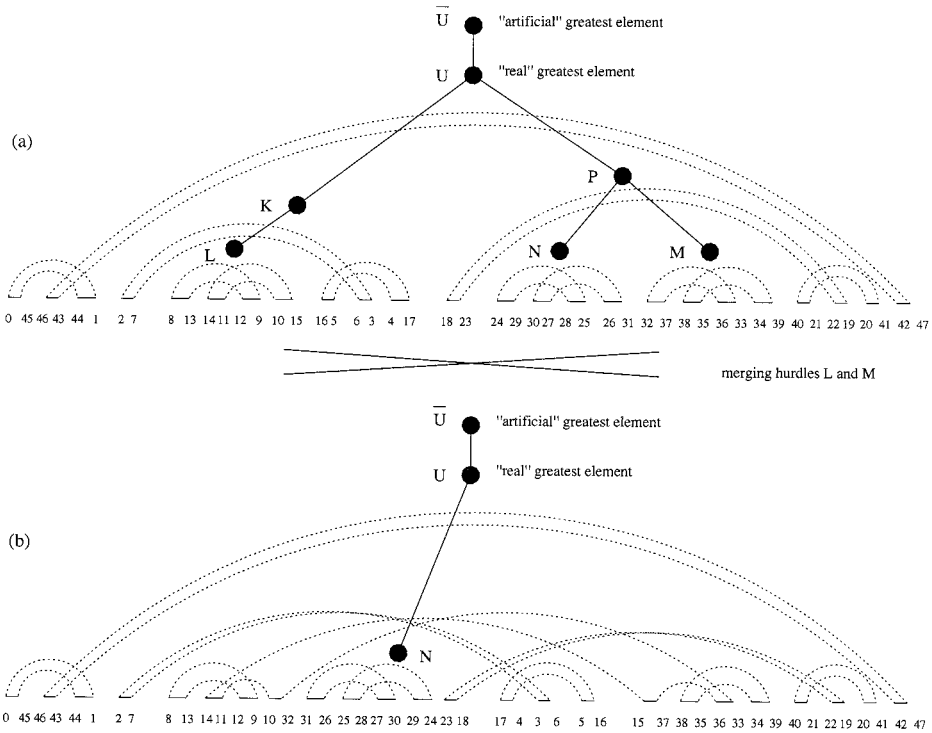

FIG. 10. (a) A cover graph Ω_π of a permutation π with "real" unoriented components K, L, M, N, P, U and an "artificial" component $\bar{U}$; (b) A reversal ρ merging hurdles L and M in π transforms unoriented components L, K, P and M into an oriented component which "disappears from $\Omega_{\pi\rho}$. This reversal transforms unoriented cycles (32, 33, 36, 37, 32) and (10, 11, 14, 15, 10) in π into an oriented cycle (15, 14, 11, 10, 32, 33, 36, 37, 15) in $\pi\rho$. $LCA(L, M) = \overline{LCA}(L, M) = U$ and $PATH(A, F) = \{L, K, U, P, M\}$.

component U and "artificial" component $\bar{U}$), corresponding to the greatest interval and that $U <_\pi \bar{U}$. Let Ω_π be the *tree* representing the cover graph of the partial order $<_\pi$ on $\overline{\text{hk}}_\pi$ (Figure 10(a)). Every vertex in Ω_π but $\bar{U}$ is associated with an unoriented component in $\mathcal{U}_\pi$. In the case, π has the greatest hurdle we assume that the leaf $\bar{U}$ is associated with this greatest hurdle (i.e., in this case, there are *two vertices* corresponding to the greatest hurdle, leaf $\bar{U}$ and its neighbor, the greatest hurdle $U \in \mathcal{U}_\pi$). Every leaf in Ω_π, corresponding to a minimal element in $<_\pi$, is a hurdle. In the case $\bar{U}$ is a leaf in Ω_π, it is not necessarily a hurdle (e.g., $\bar{U}$ is a leaf in Ω_π but not a hurdle for a permutation π shown in Figure 4(a)). Therefore, the number of leaves in Ω_π coincides with the number of hurdles $h(\pi)$, except for the cases when[3]

—there exists only one unoriented component in π (in this case, Ω_π consists of two copies of this component and has two leaves while $h(\pi) = 1$)

—there exists the greatest element in $\mathcal{U}_\pi$, which is not a hurdle, that is, this element separates other hurdles (in this case the number of leaves equals $h(\pi) + 1$).

[3] Although an addition of an "artificial" component $\bar{U}$ might seem unnecessary, we will find below that such an addition greatly facilitates the analysis of technical details.

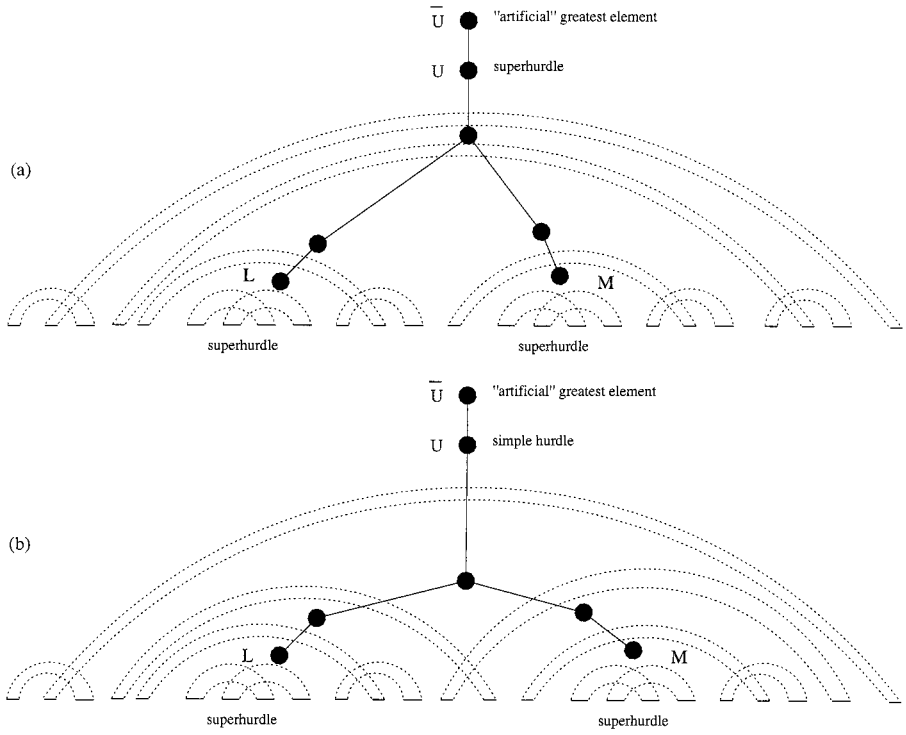

FIG. 11. Permutation in (a) is a 3-fortress while permutation in (b) with the same cover graph is not a fortress (hurdle U is not a superhurdle since deleting U leaves a greatest component that *separates* hurdles L and M).

LEMMA 14. (HURDLE CUTTING). *Every reversal ρ on a cycle in a hurdle K cuts off the leaf K from the cover graph of π, that is, $\Omega_{\pi\rho} = \Omega_\pi \backslash K$.*

PROOF. If ρ acts on an unoriented cycle of a component K in π, then K remains "unbroken" in $\pi\rho$. Also Lemma 7 implies that every reversal on an (unoriented) cycle of an (unoriented) component K orients at least one cycle in K. Therefore, ρ transforms K into an oriented component in $\pi\rho$ and deletes the leaf K from the cover graph. □

A hurdle $K \in \mathcal{U}_\pi$ *protects* a nonhurdle $U \in \mathcal{U}_\pi$ if deleting K from $\mathcal{U}_\pi$ transforms U from a nonhurdle into a hurdle (i.e., U is a hurdle in $\mathcal{U}_\pi \backslash K$). A hurdle in π is a *superhurdle* if it protects a nonhurdle $U \in \mathcal{U}_\pi$ and a *simple hurdle*, otherwise. Components M, N, and U in Figure 10(a) are simple hurdles while component L is a superhurdle (deleting L transforms a nonhurdle K into a hurdle). In Figure 11(a), all three hurdles are superhurdles while in Figure 11(b) there are two superhurdles and one simple hurdle (note that the cover graphs in Figure 11(a) and Figure 11(b) are the same!). The following lemma immediately follows from the definition of a simple hurdle.

LEMMA 15. *A reversal acting on a cycle of a simple hurdle is safe.*

PROOF. Lemma 14 implies that for every reversal ρ acting on a cycle of a simple hurdle, $b(\pi) = b(\pi\rho)$, $c(\pi) = c(\pi\rho)$ and $h(\pi\rho) = h(\pi) - 1$ implying that ρ is safe. □

Unfortunately, a reversal acting on a cycle of a superhurdle is unsafe since it transforms a nonhurdle into a hurdle implying $\Delta(b - c + h) = 0$. Below, we define a new operation (hurdles merging) allowing one to search for safe reversals even in the absence of simple hurdles.

If L and M are two hurdles in π, define $PATH(L, M)$ as the set of (unoriented) components on the (unique) path from the leaf L to the leaf M in the cover graph Ω_π. If *both* L and M are *minimal* elements in $\prec$, define $LCA(L, M)$ as an (unoriented) component that is the *least common ancestor* of L and M and define $\overline{LCA}(L, M)$ as the *least common ancestor* of L and M *that does not separate* L and M. Obviously, $\overline{LCA}(L, M)$ is either $LCA(L, M)$ or its parent. If L corresponds to the *greatest* hurdle U, there are two elements U and $\bar{U}$ in $\mathcal{U}_\pi$ corresponding to the same (greatest) interval $[U_{min}, U_{max}] = [\bar{U}_{min}, \bar{U}_{max}]$. In this case, define $LCA(L, M) = \overline{LCA}(L, M) = U$. Let $G(V, E)$ be a graph, $w \in V$ and $W \subset V$. A *contraction of* W *into* w *in* G is defined as a new graph with vertex set $V \backslash (W \backslash w)$ and edge set $\{(p(x), p(y)) : (x, y) \in E\}$, where $p(v) = w$ if $v \in W$ and $p(v) = v$, otherwise. Note that, if $w \in W$, then a contraction reduces the number of vertices in G by $|W| - 1$, while, if $w \notin W$, the number of vertices is reduced by $|W|$.

Let L and M be two hurdles in π and Ω_π be the cover graph of π. We define $\Omega_\pi(L, M)$ as the graph obtained from Ω_π by the contraction of $PATH(L, M)$ into $\overline{LCA}(L, M)$ (loops in $\Omega_\pi(L, M)$ are ignored). Note that in the case $LCA(L, M) = \overline{LCA}(L, M)$, $\Omega_\pi(L, M)$ corresponds to deleting the elements of the set $PATH(L, M) \backslash LCA(L, M)$ from the partial order $\prec_\pi$ while in the case $LCA(L, M) \neq \overline{LCA}(L, M)$, $\Omega_\pi(L, M)$ corresponds to deleting the entire set $PATH(L, M)$ from $\prec_\pi$.

LEMMA 16. (HURDLES MERGING). *Let* π *be a permutation with cover graph* Ω_π *and let* ρ *be a reversal acting on black edges of* (*different*) *hurdles* L *and* M *in* π. *Then*, ρ *acts on* Ω_π *as the contraction of* $PATH(L, M)$ *into* $\overline{LCA}(L, M)$, *that is*, $\Omega_{\pi\rho} = \Omega_\pi(L, M)$.

PROOF. The reversal ρ acts on black edges of the cycles $C_1 \in L$ and $C_2 \in M$ in $G(\pi)$ and transforms C_1 and C_2 into an oriented cycle C in $G(\pi\rho)$ (Figure 10). It is easy to verify that every cycle interleaving with C_1 or C_2 in $G(\pi)$ interleaves with C in $G(\pi\rho)$. It implies that ρ transforms hurdles L and M in π into parts of an oriented component in $\pi\rho$ and, therefore, L and M "disappear" from $\Omega_{\pi\rho}$.

Moreover, every (unoriented) component from $PATH(L, M) \backslash \overline{LCA}(L, M)$ has at least one cycle interleaving with C in $G(\pi\rho)$. It implies that every such component in π becomes a part of an oriented component in $\pi\rho$ and therefore "disappears" from $\Omega_{\pi\rho}$. Every component from $\mathcal{U}_\pi \backslash PATH(L, M)$ remains unoriented in $\pi\rho$. Component $LCA(L, M)$ remains unoriented iff $LCA(L, M) = \overline{LCA}(L, M)$. Every component that is covered by a vertex from $PATH(L, M)$ in $\prec_\pi$ will be covered by $\overline{LCA}(L, M)$ in $\prec_{\pi\rho}$. □

We write $U < W$ for hurdles U and W if the rightmost position of U is smaller than the rightmost position of W, that is, $U_{max} < W_{max}$. Order the hurdles of π in the increasing order of their rightmost positions

$$U(1) < \cdots < U(l) \equiv L < \cdots < U(m) \equiv M < \cdots < U(h(\pi))$$

and define the sets of hurdles

$$BETWEEN(L, M) = \{U(i) : l < i < m\} \qquad \text{and} \qquad OUTSIDE(L, M) = \{U(i) : i \notin [l, m]\}.$$

Notice that any hurdle from $BETWEEN(L, M)$ must lie sandwiched between L and M.

LEMMA 17. *Let ρ be a reversal merging hurdles L and M in π. If both sets of hurdles $BETWEEN(L, M)$ and $OUTSIDE(L, M)$ are nonempty, then ρ is safe.*

PROOF. Let $U' \in BETWEEN(L, M)$ and $U'' \in OUTSIDE(L, M)$. Lemma 16 implies that the reversal ρ deletes the hurdles L and M from Ω_π. There is also a "danger" that ρ adds a new hurdle K in $\pi\rho$ by transforming K from a nonhurdle in π into a hurdle in $\pi\rho$. If it is the case, K does not separate L and M in π (otherwise, by Lemma 16, K would be deleted from $\pi\rho$). Without loss of generality, we assume that $L < U' < M$.

If K is a *minimal* hurdle in $\pi\rho$, then either $L \prec_\pi K$ or $M \prec_\pi K$ (otherwise, K would be a hurdle in π). Since K does not separate L and M in π, it implies that $L \prec_\pi K$ *and* $M \prec_\pi K$. Since U' is sandwiched between L and M, it implies that $U' \prec_\pi K$. Thus, $U' \prec_{\pi\rho} K$, a contradiction to minimality of K in $\pi\rho$.

If K is the *greatest* hurdle in $\pi\rho$, then either $L, M \not\prec_\pi K$ or $L, M \prec_\pi K$ (if, otherwise, $L \not\prec_\pi K$ and $M \prec_\pi K$, then, according to Lemma 16, K would be deleted from $\pi\rho$). If $L, M \not\prec_\pi K$, then $L < U' \prec_\pi K < M$, that is, K is sandwiched between L and M. Therefore, U'' lies outside K in π and $U'' \not\prec_{\pi\rho} K$, a contradiction. If $L, M \prec_\pi K$, then, since K is a nonhurdle in π, K separates L, M from another hurdle N. Therefore, K separates U' from N. Since both N and U' "survive" in $\pi\rho$, it implies that K separates N and U' in $\pi\rho$, a contradiction.

Therefore, ρ deletes the hurdles L and M from Ω_π and does not add a new hurdle in $\pi\rho$, thus implying that $\Delta h = -2$. Since $b(\pi\rho) = b(\pi)$ and $c(\pi\rho) = c(\pi) - 1$, $\Delta(b - c + h) = -1$ and the reversal ρ is safe. □

LEMMA 18. *If $h(\pi) > 3$, then there exists a safe reversal merging two hurdles in π.*

PROOF. Order $h(\pi)$ hurdles of π in the increasing order of their rightmost positions and let L and M be the first and $\lfloor 1 + (h(\pi)/2) \rfloor$-th hurdles in this order. Since $h(\pi) > 3$, both sets $BETWEEN(L, M)$ and $OUTSIDE(L, M)$ are nonempty and by Lemma 17, the reversal ρ merging L and M is safe. □

LEMMA 19. *If $h(\pi) = 2$, then there exists a safe reversal merging two hurdles in π. If $h(\pi) = 1$, then there exists a safe reversal cutting the only hurdle in π.*

PROOF. If $h(\pi) = 2$, then Ω_π is either a path graph or contains the greatest component separating two hurdles in π. In both cases, merging the hurdles in π is a safe reversal (Lemma 16). If $h(\pi) = 1$, then Lemma 14 provides a safe reversal cutting the only hurdle in π. □

The previous lemmas show that hurdles merging provides a way to find safe reversals even in the absence of simple hurdles. On the negative note, hurdles merging does not provide a way to transform a superhurdle into a simple hurdle.

LEMMA 20. *Let ρ be a reversal in π merging two hurdles L and M. Then every superhurdle in π (different from L and M) remains a superhurdle in $\pi\rho$.*

PROOF. Let U be a superhurdle in π (different from L and M) protecting a nonhurdle U'. Clearly if U' is a minimal hurdle in $\mathcal{U}_\pi \backslash U$, then U remains a superhurdle in $\pi\rho$. If U' is the greatest hurdle in $\mathcal{U}_\pi \backslash U$, then U' does not separate any hurdles in $\mathcal{U}_\pi \backslash U$. Therefore, U' does not belong to $PATH(L, M)$ and hence "survives" in $\pi\rho$ (Lemma 16). It implies that U' remains protected by U in $\pi\rho$. □

7. *Fortresses*

Lemmas 18 and 19 imply that unless Ω_π is a homeomorph of the 3-*star* (a graph with three edges incident on the same vertex) there exists a safe reversal in π. On the other hand, if at least one hurdle in π is simple, then Lemma 15 implies that there exists a safe reversal in π. Therefore, the only case in which a safe reversal might not exist is when Ω_π is a homeomorph of 3-star with three superhurdles, called a 3-*fortress* (Figure 11(a)).

LEMMA 21. *If ρ is a reversal destroying a 3-fortress π (i.e. $\pi\rho$ is not a 3-fortress), then ρ is unsafe.*

PROOF. Every reversal on a permutation π can reduce $h(\pi)$ by at most 2 and the *only* operation that can reduce the number of hurdles by 2 is merging of hurdles. On the other hand, Lemma 16 implies that merging of hurdles in a 3-fortress can reduce $h(\pi)$ by at most 1. Therefore, $\Delta h \geq -1$. Note that, for every reversal that does not act on edges of the *same* cycle, $\Delta(b - c) = 1$ ($\Delta b = 0$, $\Delta c = -1$ if ρ acts on breakpoints of different cycles, $\Delta b = 1$, $\Delta c = 0$ if ρ acts on a breakpoint and an adjacency and $\Delta b = 2$, $\Delta c = 1$ if ρ acts on two adjacencies) and therefore every reversal that does not act on edges of the same cycle in a 3-fortress is unsafe.

If ρ acts on a cycle in an unoriented component of a 3-fortress, then it does not reduce the number of hurdles. Since $\Delta(b - c) = 0$ for a reversal on an unoriented cycle, ρ is unsafe.

If ρ acts on a cycle in an oriented component of a 3-fortress, then it does not destroy any unoriented components in π and, does not reduce the number of hurdles. If ρ increases the number of hurdles, then $\Delta h \geq 1$ and $\Delta(b - c) \geq -1$ imply that ρ is unsafe. If the number of hurdles in $\pi\rho$ remains the same, then every superhurdle in π remains a superhurdle in $\pi\rho$, thus implying that $\pi\rho$ is a 3-fortress, a contradiction. □

LEMMA 22. *If π is a 3-fortress, then $d(\pi) = b(\pi) - c(\pi) + h(\pi) + 1$.*

PROOF. Lemma 21 implies that every sorting of 3-fortress contains at least one unsafe reversal. Therefore, $d(\pi) \geq b(\pi) - c(\pi) + h(\pi) + 1$.

If π has oriented cycles, all oriented components in π can be destroyed by safe paddings (Theorem 2) and safe reversals in oriented components (Theorem 4) without affecting unoriented components.

If π is a 3-fortress without oriented cycles, then an (unsafe) reversal ρ merging arbitrary hurdles in π leads to a permutation $\pi\rho$ with two hurdles (Lemma 16). Once again, oriented cycles appearing in $\pi\rho$ after such merging can be destroyed

by safe paddings and safe reversals in oriented components (Theorems 2 and 4) leading to a permutation σ with $h(\sigma) = 2$. Theorems 2 and 4 and Lemma 19 imply that σ can be sorted by safe paddings and safe reversals. Hence, there exists a generalized sorting of π such that all paddings and all reversals but one in this sorting are safe. Therefore, this generalized sorting contains $b(\pi) - c(\pi) + h(\pi) + 1$ reversals. Lemma 9 implies that the generalized sorting of π mimics an optimal (genuine) sorting of π by $d(\pi) = b(\pi) - c(\pi) + h(\pi) + 1$ reversals. □

In the following, we try to avoid creating 3-fortresses in the course of sorting by reversals. If we are successful in this task, the permutation π can be sorted in $b(\pi) - c(\pi) + h(\pi)$ reversals. Otherwise, we show how to sort π in $b(\pi) - c(\pi) + h(\pi) + 1$ reversals and prove that such permutations can not be sorted with fewer number of reversals. Permutation π is called a *fortress* if it has an odd number of hurdles and all these hurdles are superhurdles.

LEMMA 23. *If ρ is a reversal destroying a fortress π with $h(\pi)$-superhurdles (i.e., $\pi\rho$ is not a fortress with $h(\pi)$ superhurdles), then either ρ is unsafe or $\pi\rho$ is a fortress with $h(\pi) - 2$ superhurdles.*

PROOF. Every reversal acting on a permutation can reduce the number of hurdles by at most 2 and the *only* operation that can reduce the number of hurdles by 2 is a merging of hurdles. Arguments similar to the proof of Lemma 21 demonstrate that, if ρ does not merge hurdles, then ρ is unsafe. If a safe reversal ρ does merge (super)hurdles L and M in π, then Lemma 16 implies that every such reversal reduces the number of hurdles by 2, and in the case $h(\pi) > 3$, does not create new hurdles. Also, Lemma 20 implies that every superhurdle in π but L and M remains a superhurdle in $\pi\rho$, thus implying that $\pi\rho$ is a fortress with $h(\pi) - 2$ superhurdles. □

LEMMA 24. *If π is a fortress, then $d(\pi) \geq b(\pi) - c(\pi) + h(\pi) + 1$.*

PROOF. Lemma 23 implies that every sorting of π either contains an unsafe reversal or gradually decreases the number of superhurdles in π by transforming a fortress with h (super)hurdles into a fortress with $h - 2$ (super)hurdles. Therefore, if sorting of π uses only safe reversals, then it will eventually lead to a 3-fortress. Therefore, by Lemma 21, every sorting of a fortress contains at least one unsafe reversal and hence $d(\pi) \geq b(\pi) - c(\pi) + h(\pi) + 1$. □

Finally, we formulate the duality theorem for sorting signed permutations by reversals.

THEOREM 5. *For every permutation π,*

$$d(\pi) = \begin{cases} b(\pi) - c(\pi) + h(\pi) + 1, & \text{if } \pi \text{ is a fortress} \\ b(\pi) - c(\pi) + h(\pi), & \text{otherwise.} \end{cases}$$

PROOF. If π has an even number of hurdles then safe paddings (Theorems 2), safe reversals in oriented components (Theorem 4) and safe hurdles mergings (Lemmas 18 and 19) lead to a generalized sorting of π by $b(\pi) - c(\pi) + h(\pi)$ reversals.

If π has an odd number of hurdles at least one of which is simple, then there exists a safe reversal cutting this simple hurdle (Lemma 15). This safe reversal leads to a permutation with an even number of hurdles. Therefore, similar to the previous case, there exists a generalized sorting of π using only safe paddings and $b(\pi) - c(\pi) + h(\pi)$ safe reversals.

Therefore, if π is not a fortress, there exists a generalized sorting of π by $b(\pi) - c(\pi) + h(\pi)$. Lemma 9 implies that this generalized sorting mimics optimal (genuine) sorting of π.

If π is a fortress, there exists a sequence of safe paddings (Theorem 2), safe reversals in oriented components (Theorem 4), and safe hurdles merging (Lemma 18) leading to a 3-fortress that can be sorted by a series of reversals having exactly one unsafe reversal. Therefore, there exists a generalized sorting of π using $b(\pi) - c(\pi) + h(\pi) + 1$ reversals. Lemma 24 implies that this generalized sorting mimics optimal (genuine) sorting of π with $d(\pi) = b(\pi) - c(\pi) + h(\pi) + 1$ reversals. □

8. *Polynomial Algorithm*

Lemmas 9, 18, 15, and 19, and Theorems 2, 4, and 5 motivate the algorithm *Reversal_Sort*, which optimally sorts signed permutations.

Algorithm *Reversal_Sort*(π)
1. **while** π is not sorted
2. **if** π has a long cycle
3. select a safe (g, b)-padding ρ of π (Theorem 2)
4. **else if** π has an oriented component
5. select a safe reversal ρ in this component (Theorem 4)
6. **else if** π has an even number of hurdles
7. select a safe reversal ρ merging two hurdles in π (Lemmas 18 and 19)
8. **else if** π has at least one simple hurdle
9. select a safe reversal ρ cutting this hurdle in π (Lemmas 15 and 19)
10. **else if** π is a fortress with more than three superhurdles
11. select a safe reversal ρ merging two (super)hurdles in π (Lemma 18)
12. **else** /* π is a 3-fortress */
13. select an (un)safe reversal ρ merging two arbitrary (super)hurdles in π
14. $\pi \leftarrow \pi \cdot \rho$
15. **endwhile**
16. mimic (genuine) sorting of π using the computed generalized sorting of π (Lemma 9)

THEOREM 6. *Reversal_Sort*(π) *optimally sorts a permutations* $\pi = (\pi_1 \pi_2 \cdots \pi_n)$ *in* $O(n^4)$ *time*.

PROOF. Theorem 5 implies that *Reversal_Sort* provides generalized sorting of π by a series of reversals and paddings containing $d(\pi)$ reversals. Lemma 9 implies that this generalized sorting mimics an optimal (genuine) sorting of π by $d(\pi)$ reversals.

We sketch an $O(n^4)$ implementation of *Reversal_Sort*(π) (the description of data structures is omitted). Note that every iteration of **while** loop in *Reversal_Sort* reduces the amount *complexity*(π) + $3d(\pi)$ by at least 1 thus implying that the number of iterations of *Reversal_Sort* is bounded by $4n$. The most "expensive" iteration is a search for a safe reversal in an oriented component. Since for simple permutations it can be implemented in $O(n^3)$ time, the overall running time of *Reversal_Sort* is $O(n^4)$. □

A more careful analysis of *Reversal_Sort* (omitted here) leads to further reduction of running time. Below, we describe a simpler version of *Reversal_Sort*, which does not use paddings and runs in $O(n^5)$ time.

Define

$$f(\pi) = \begin{cases} 1, & \text{if} \quad \pi \text{ is a fortress} \\ 0, & \text{otherwise.} \end{cases}$$

A reversal ρ is *valid*, if $\Delta(b - c + h + f) = -1$. Proofs of Theorem 1 and Lemma 24 imply that there is a reversal with $\Delta(b - c + h + f) \geq -1$. This observation and Theorem 5 imply the following:

THEOREM 7. *For every permutation π there exists a valid reversal in π. Every sequence of valid reversals sorting π is optimal.*

Theorem 7 motivates the following simple version of *Reversal_Sort*, which is very fast in practice:

Algorithm *Reversal_Sort_Simple*(π)
1. **while** π is not sorted
2. select a valid reversal ρ in π (Theorem 7)
3. $\pi \leftarrow \pi \cdot \rho$
4. **endwhile**

Step 1 can be executed at most n times for a permutation of size n. There are $n * (n - 1)$ reversals to search among. Computing $(b - c + h + f)$ for any permutation can be done in $O(n^2)$ time. Hence, the worst-case time complexity of the above algorithm is $O(n^5)$.

Reversal_Sort_Simple was implemented and tested on biological data. The results of these tests are described in Hannenhalli and Pevzner [1996], in particular, we found optimal evolutionary scenarios for extensively rearranged genomes, which were considered as too hard to analyze in the previous studies. Experiments with *Reversal_Sort_Simple* on simulated data explained the mystery of astonishing performance of previously suggested approximation algorithms for sorting signed permutations by reversals. A simple explanation for this performance is that the bound (1) is extremely tight since $h(\pi)$ is small for "random" permutations and zero for most of the biological data.

ACKNOWLEDGMENTS. We are indebted to Vineet Bafna, Piotr Berman, Webb Miller, and Anatoly Rubinov for many helpful discussions and suggestions. We are also grateful to Eric Boudreau for sending us unpublished experimental data on gene orders of *Chlamydomonas gelatinosa* and *Chlamydomonas reinhardtii* for testing *Reversal_Sort_Simple*. Both referees provided many valuable comments.

REFERENCES

AIGNER, M., AND WEST, D. B. 1987. Sorting by insertion of leading element. *J. Combin. Theory 45*, 306–309.

BAFNA, V., AND PEVZNER, P. 1995. Sorting by reversals: Genome rearrangements in plant organelles and evolutionary history of X chromosome. *Mol. Biol. Evol. 12*, 239–246.

BAFNA, V., AND PEVZNER, P. 1996. Genome rearrangements and sorting by reversals. *SIAM J. Comput. 25*, 272–289.

BAFNA, V., AND PEVZNER, P. 1998. Sorting by transpositions. *SIAM J. Disc. Math. 11*, 224–240.

BERMAN, P., AND HANNENHALLI, S. 1996. Fast sorting by reversals. In *Combinatorial Pattern Matching*, *Proceedings of the 6th Annual Symposium* (*CPM'96*). Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, pp. 168–185.

CAPRARA, A. 1997. Sorting by reversals is difficult. In *Proceedings of the 1st Annual International Conference on Computational Molecular Biology* (*RECOMB97*). pp. 75–83.

COHEN, D., AND BLUM, M. 1993. Improved bounds for sorting pancakes under a conjecture, Manuscript.

EVEN, S., AND GOLDREICH, O. 1981. The minimum-length generator sequence problem is NP-hard. *J. Algorithms 2*, 311–313.

GATES, W. H., AND PAPADIMITRIOU, C. H. 1979. Bounds for sorting by prefix reversals. *Disc. Math. 27*, 47–57.

HANNENHALLI, S. 1995. Polynomial algorithm for computing translocation distance between genomes. In *Combinatorial Pattern Matching, Proceedings of the 6th Annual Symposium* (*CPM'95*). Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, pp. 162–176.

HANNENHALLI, S., CHAPPEY, C., KOONIN, E., AND PEVZNER, P. 1995. Genome sequence comparison and scenarios for gene rearrangements: A test case. *Genomics, 30*, 299–311.

HANNENHALLI, S., AND PEVZNER, P. 1995. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, Calif., pp. 581–592.

HANNENHALLI, S., AND PEVZNER, P. 1996. To cut . . . or not to cut (applications of comparative physical maps in molecular evolution). In *Proceedings of the 7th Annual ACM–SIAM Symposium on Discrete Algorithms* (Atlanta, Ga., Jan. 28–30). ACM, New York, pp. 304–313.

HEYDARI, M., AND SUDBOROUGH, I. H. 1993. On sorting by prefix reversals and the diameter of pancake networks, Manuscript.

JERRUM, M. 1985. The complexity of finding minimum-length generator sequences. *Theoret. Comput. Sci. 36*, 265–289.

KECECIOGLU, J., AND GUSFIELD, D. 1994. Reconstructing a history of recombinations from a set of sequences. In *Proceedings of the 5th Annual ACM–SIAM Symposium on Discrete Algorithms*, ACM, New York, pp. 471–480.

KECECIOGLU, J., AND RAVI, R. 1995. Of mice and men: Algorithms for evolutionary distances between genomes with translocation. In *Proceedings of the 6th Annual ACM–SIAM Symposium on Discrete Algorithms* (San Francisco, Calif., Jan. 22–24). ACM, New York, pp. 604–613.

KECECIOGLU, J., AND SANKOFF, D. 1995. Exact and approximation algorithms for the inversion distance between two permutations. *Algorithmica 13*, 180–210.

KECECIOGLU, J., AND SANKOFF, D. 1994. Efficient bounds for oriented chromosome inversion distance. In *Combinatorial Pattern Matching*, *Proceedings of the 5th Annual Symposium* (*CPM'94*). Lecture Notes in Computer Science, vol. 807. Springer-Verlag, Berlin, Germany, pp. 307–325.

MAKAROFF, C. A., AND PALMER, J. D. 1988. Mitochondrial DNA rearrangements and transcriptional alterations in the male sterile cytoplasm of Ogura radish. *Mol. Cell. Biol. 8*, 1474–1480.

NADEAU, J. H., AND TAYLOR, B. A. 1984. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proc. Natl. Acad. Sci. USA 81*, 814–818.

PALMER, J. D., AND HERBON, L. A. 1988. Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. *J. Mol. Evolut. 27*, 87–97.

PEVZNER, P. A., AND WATERMAN, M. S. 1995. Open combinatorial problems in computational molecular biology. In *Proceedings of the 3rd Israel Symposium on Theory of Computing and Systems*. IEEE Computer Society Press, Los Alamitos, Calif., pp. 158–163.

SANKOFF, D. 1992. Edit distance for genome comparison based on non-local operations. In *Combinatorial Pattern Matching*, *Proceedings of the 3rd Annual Symposium* (*CPM'92*). Lecture Notes in Computer Science, vol. 644. Springer-Verlag, Berlin, Germany, pp. 121–135.

SANKOFF, D., CEDERGREN, R., AND ABEL, Y. 1990. Genomic divergence through gene rearrangement. In *Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences*, Chap. 26. Academic Press, Orlando, Fla., pp. 428–438.

SANKOFF, D., LEDUC, G., ANTOINE, N., PAQUIN, B., LANG, B. F., AND CEDERGREN, R. 1992. Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome. *Proc. Natl. Acad. Sci. USA 89*, 6575–6579.

STURTEVANT, A. H., AND DOBZHANSKY, T. 1936. Inversions in the third chromosome of wild races of *drosophila pseudoobscura*, and their use in the study of the history of the species. *Proc. Nat. Acad. Sci. 22*, 448–450.

TARJAN, R., KAPLAN, H., AND SHAMIR, R. 1997. Faster and simpler algorithm for sorting by reversals. In *Proceedings of the 8th Annual ACM–SIAM Symposium on Discrete Algorithms*. ACM, New York, pp. 614–623.

WATTERSON, G. A., EWENS, W. J., HALL, T. E., AND MORGAN, A. 1982. The chromosome inversion problem. *J. Theoret. Biol. 99*, 1–7.

RECEIVED FEBRUARY 1995; REVISED JULY 1998; ACCEPTED SEPTEMBER 1998