Transforming Cabbage into Turnip

(polynomial algorithm for sorting signed permutations by reversals)*

Sridhar Hannenhalli Pavel Pevzner†

March 2, 1995

Abstract

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 for the species of *Drosophila*. Analysis of genomes evolving by inversions leads to a combinatorial problem of *sorting by reversals* studied in detail recently. Kececioglu and Sankoff conjectured that sorting by reversals is NP-hard, but despite many attempts their conjecture remains open. We study sorting of *signed* permutations by reversals, a problem which adequately models rearrangements in small genomes like chloroplast or mitochondrial DNA. The previously suggested performance guarantee algorithms for sorting signed permutations by reversals approximate 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 which allows one to efficiently compute the reversal distance between signed permutations.

1 Introduction

In the late 1980's Jeffrey Palmer and colleagues discovered a remarkable and novel pattern of evolutionary change in plant organelles. They mapped 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* (Fig. 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, 1995a).

Every study of genome rearrangements in mitochondrial DNA 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 Fig. 1.) For genomes consisting of small number of "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, 1995a for correct solutions).

Analysis of genome rearrangements provides a multitude of challenges; 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, 1993 suggested the first performance guarantee algorithm to analyze genome rearrangements and conjectured that sorting by reversals in NP-hard. The problem was further studied by Bafna and Pevzner, 1993 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. See also Kececioglu and Sankoff, 1994, Kececioglu and Gusfield, 1994, Kececioglu and Ravi, 1995, Hannenhalli, 1994 and Bafna and Pevzner, 1995b 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.

In the problem we consider, the order of genes

*supported by NSF Young Investigator Award, NSF grant CCR-9308567, NIH grant 1R01 HG00987 and DOE grant DE-FG02-94ER61919

†Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802; e-mail: hannenha,pevzner@cse.psu.edu. . . . .

is represented by a permutation. A *reversal* $\rho(i,j)$ transforms a permutation

$$\pi = (\pi_1 \dots \pi_{i-1}\pi_i\pi_{i+1} \dots \pi_{j-1}\pi_j\pi_{j+1} \dots \pi_n)$$

into a permutation

$$\pi \cdot \rho(i,j) = (\pi_1 \dots \pi_{i-1}\pi_j\pi_{j-1} \dots \pi_{i+1}\pi_i\pi_{j+1} \dots \pi_n)$$

and has the effect of reversing the order of elements $\pi_i\pi_{i+1} \dots \pi_{j-1}\pi_j$. Given permutations π and σ, the *reversal distance problem* is to find a series of reversals $\rho_1, \rho_2, \dots, \rho_t$ such that $\pi \cdot \rho_1 \cdot \rho_2 \cdots \rho_t = \sigma$ and t is minimum (Fig. 2a). We call t the *reversal distance* between π and σ. *Sorting π by reversals* is the problem of finding the reversal distance, $d(\pi)$, between π and the *identity permutation* $(12 \dots n)$.

What makes it hard to sort a permutation? In the very first computational studies of 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!) Breakpoints are defined as follows. Let $i \sim j$ if $|i-j| = 1$. Extend a permutation $\pi = \pi_1\pi_2 \dots \pi_n$ by adding $\pi_0 = 0$ and $\pi_{n+1} = n+1$. We call a pair of elements $(\pi_i,\ \pi_{i+1})$, $0 \leq i \leq n$, of π an *adjacency* if $\pi_i \sim \pi_{i+1}$, and a *breakpoint* if $\pi_i \not\sim \pi_{i+1}$. As the identity permutation has no breakpoints, sorting by reversals corresponds to eliminating breakpoints. An observation that every reversal can eliminate *at most* 2 breakpoints implies that $d(\pi) \geq \frac{b(\pi)}{2}$, 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, 1993 found another parameter which estimates reversal distance with much greater accuracy.

The *breakpoint graph* of a permutation $\pi = (\pi_1\pi_2 \dots \pi_n)$ is an edge-colored graph $G(\pi)$ with $n+2$ vertices $\{\pi_0, \pi_1, \dots, \pi_n, \pi_{n+1}\}$. We join vertices π_i and π_j by a *black* edge if $(\pi_i,\ \pi_j)$ 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 Fig. 2b.) 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 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 it has no long cycles.

Consider a *cycle decomposition* of $G(\pi)$ into a *maximum* number $c(\pi)$ of edge-disjoint alternating cycles (Fig. 2b). Bafna and Pevzner, 1993 showed that the maximum cycle decomposition provides a better bound for the reversal distance:

$$d(\pi) \geq b(\pi) - c(\pi) \qquad (1.1)$$

However, finding a maximum cycle decomposition is a difficult problem. Fortunately in biologically more relevant case of *signed permutations* this problem is trivial. Genes are *directed* fragments of DNA and sequence of n genes in a genome is represented by a *signed* permutation on $\{1, \dots n\}$ with $+$ or $-$ sign associated with every element of π. For example, a gene order for *B. oleracea* presented in Fig. 1 is modeled by a signed permutation $(+1 - 5 + 4 - 3 + 2)$. In the signed case, every reversal $\rho(i,j)$ changes *both* the order and the signs of the elements within the fragment $\pi_i\pi_{i+1} \dots \pi_{j-1}\pi_j$ (Fig. 1). We are interested in the minimum number of reversals $d(\pi)$ required to transform a signed permutation π into the *identity signed permutation* $(+1 + 2 \dots + n)$. Bafna and Pevzner, 1993 noted that the concept of breakpoint graph extends naturally to signed permutations and devised an approximation algorithm for sorting signed permutations with performance ratio 1.5. For signed permutations the bound (1.1) approximates the reversal distance extremely well for both simulated (Kececioglu and Sankoff, 1994) and biological data (Bafna and Pevzner, 1995a, Hannenhalli et al., 1994). This intriguing performance raises a question whether the bound (1.1) overlooked a third parameter (in addition to the number of breakpoints and the size of a maximum cycle decomposition) 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). We show that $b(\pi) - c(\pi) + h(\pi) \leq d(\pi) \leq b(\pi) - c(\pi) + h(\pi) + 1$, 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 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 3). The technique allows one to mimic sorting permutations with long cycles by sorting simple permutations. In section 4 we study combinatorics of simple permutations and make the first step towards proving the bound $d(\pi) \leq b(\pi) - c(\pi) + h(\pi) + 1$. In section 5 we associate a partial order with every permutation. The properties of this partial order allow us to introduce *safe* reversals (*hurdle cutting* and *hurdles merging*) which are the key operations for *clearing the hurdles* in our algorithm. In section 6 we further develop a characterization of "hard-to-sort" permutations (called *fortresses*) which can not be sorted in $b(\pi) - c(\pi) + h(\pi)$ steps and prove the duality theorem

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

Finally in section 7 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, 1994 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 Interleaving graph and hurdles

Define a transformation from a signed permutation π of order n to an unsigned permutation π' of order $2n$ as follows. To model the signs of elements in π replace the positive elements $+x$ by $2x-1, 2x$ and negative elements $-x$ by $2x, 2x-1$ (Fig. 2c) 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 signed permutations easier to analyze. 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 $\pi' = (\pi'_1 \pi'_2 \dots \pi'_{2n})$ of a signed permutation $\pi = (\pi_1 \pi_2 \dots \pi_n)$ we mean sorting of π' by reversals $\rho(2i+1, 2j)$ which "cut" *only after even positions* in 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 after odd positions (i.e. between π'_{2i-1} and π'_{2i}) are forbidden (see Bafna and Pevzner, 1993). 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 $\pi = (\pi_1 \pi_2 \dots \pi_{2n})$ such that π_{2i-1} and π_{2i} are consecutive numbers for $1 \leq i \leq n$.

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 edges (π_{i-1}, π_i) and (π_j, π_{j+1}). $\rho(i,j)$ is a *reversal (acting) on a cycle*, if the breakpoints (π_{i-1}, π_i) and (π_j, π_{j+1}) belong to the *same* cycle in $G(\pi)$. Given a 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, 1993 proved that every reversal reduces $b(\pi) - c(\pi)$ by at most 1, i.e. $\Delta(b-c) \equiv \Delta b(\pi,\rho) - \Delta c(\pi,\rho) \geq -1$. We call a reversal *proper* if $\Delta(b-c) = -1$. A gray edge g is *oriented* if a reversal acting on two black edges incident to g is proper and *unoriented* otherwise. A cycle in $G(\pi)$ is *oriented* if it has an oriented gray edge and unoriented otherwise. It can be verified that there is no proper reversal acting on an unoriented cycle.

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. Cycles C_1 and C_2 in $G(\pi)$ are *interleaving* if there exist interleaving gray edges $g_1 \in C_1$ and $g_2 \in C_2$. See Fig. 2c for examples.

Let $\mathcal{C}_\pi$ be the set of cycles in the breakpoint graph of a permutation π. Define 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 and C_2 are interleaving cycles in $G(\pi)\}$ (Fig. 2d). 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 as $U_{min} = \min_{C \in U} \min_{\pi_i \in C} i$ and and $U_{max} = \max_{C \in U} \max_{\pi_i \in C} i$. 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 $[U'_{min}, U'_{max}] \subset [i,j]$, but $[U''_{min}, U''_{max}] \not\subset [i,j]$. For example, the component U in Fig. 3a 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, i.e. $U \prec W$ iff $[U_{min}, U_{max}] \subset [W_{min}, W_{max}]$ for $U, W \in \mathcal{U}_\pi$. A *hurdle* is defined as an unoriented component U which is either a minimal hurdle or the greatest hurdle where a *minimal hurdle* is a minimal element in $\prec$ and the *greatest hurdle* satisfies the following two conditions (i) U is the greatest element in $\prec$ *and* (ii) U does not separate any two hurdles. Let $h(\pi)$ be the *overall* number of hurdles in π (see Fig. 3). The following theorem further improves the bound (1.1)

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

Proof: Given a reversal ρ, denote $\Delta h \equiv \Delta h(\pi, \rho) = h(\pi\rho) - h(\pi)$. Since ρ acts on black edges of at most two hurdles, it may "destroy" 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 condition (ii) in the definition of the greatest hurdle). Therefore $\Delta h \geq -2$.

Bafna and Pevzner, 1993 proved that $\Delta(b-c) \in \{-1, 0, 1\}$. If $\Delta(b-c) = -1$, then ρ acts on an oriented cycle (Bafna and Pevzner, 1993) and hence it does not affect any unoriented components 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 affects at most one hurdle (see condition (ii)). 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 $d(\pi) \geq b(\pi) - c(\pi) + h(\pi)$. □

3 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 (Bafna and Pevzner, 1993) and by transpositions (Bafna and Pevzner, 1995b). 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 (i.e. having no long cycles) permutation. 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 (i) removing edges g and b, (ii) adding two new vertices v and w, (iii) adding two new black edges (v_b, v) and (w, w_b), (iv) adding two new gray edges (w_g, w) and (v, v_g) (Fig. 5). 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 a permutation $\hat{\pi}$ corresponding to a (g,b)-split of $G(\pi)$.

A generalized permutation $\pi = (\pi_1\pi_2\ldots\pi_n)$ 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 $\pi = (\pi_1\pi_2\ldots\pi_n)$ with $\pi_i < \pi_{i+1}$ for $1 \leq i \leq n-1$. Extend π by adding $\pi_0 = \min_i \pi_i - 1$ and $\pi_{n+1} = \max_i \pi_i + 1$. 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 $1 \leq l \leq n$. Elements π_i and π_{i+1} of π are *adjacent* for $0 \leq i \leq n$. The *breakpoint graph* of a (generalized) permutation $\pi = (\pi_1\pi_2\ldots\pi_n)$ 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.

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 $v = \pi_j + \frac{\Delta}{3}$, $w = \pi_k - \frac{\Delta}{3}$. A (g,b)-*padding* of $\pi = (\pi_1\pi_2\ldots\pi_n)$ is a permutation $\hat{\pi} = (\pi_1\pi_2\ldots\pi_i v w \pi_{i+1}\ldots\pi_n)$ on $n+2$ elements obtained from π by inserting v and w after the i-th element of π. 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. [1] The following lemma establishes the correspondence between (g,b) paddings and (g,b)-splits.

Lemma 1. $\hat{G}(\pi) = G(\hat{\pi})$.

A (g,b)-padding ϕ transforming π into $\hat{\pi} = \pi \cdot \phi$ is *safe* if it acts on non-incident edges of a long cycle and $h(\pi) = h(\hat{\pi})$. Clearly, every safe padding breaks a long cycle into two smaller cycles. Therefore safe paddings may be used to transform an arbitrary permutation π into a simple permutation without changing the parameter $b(\pi)-c(\pi)+h(\pi)$ (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 safe padding. First we need a series of lemmas.

Lemma 2. *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.*

Lemma 3. *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 $G(\hat{\pi})$.*

Lemma 4. *For every gray edge g there exists a gray edge f interleaving with g in $G(\pi)$. If C is a cycle in $G(\pi)$ and $g \notin C$ then g interleaves with an even number of gray edges in C.*

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 another path and consider the (g,b)-padding transforming π into $\hat{\pi}$ (clearly g and b are non-incident 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. Clearly, 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 3 all cycles from K belong to the component of $H_{\hat{\pi}}$ containing C_1 and C_2. Moreover, according to lemma 2 the orientation of this component in H_π and

[1] Of course, a (g,b)-padding of a permutation $\pi = (\pi_1\pi_2\ldots\pi_n)$ on $\{1,2,\ldots,n\}$ can be modeled as a permutation $\hat{\pi} = (\hat{\pi}_1\hat{\pi}_2\ldots\hat{\pi}_i v w \hat{\pi}_{i+1}\ldots\hat{\pi}_n)$ 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. We prefer generalized permutations to make the following "mimicking" more intuitive.

$H_{\hat{\pi}}$ is the same. Therefore the chosen (g, b)-padding preserves the set of hurdles and hence, $h(\pi) = h(\hat{\pi})$.

If all gray edges of C are mutually non-interleaving then C is an unoriented cycle. Lemmas 4 implies 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 another 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. Clearly, 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 2 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 permutations π is *equivalent* to a permutation σ 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 $\phi(i)$ on $\pi(i)$ $(0 \leq i \leq k-1)$. Define the $complexity(\pi) = \sum_{C \in \mathcal{C}_\pi}(l(C) - 2)$ 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 non-incident edges of a long cycle C breaks C into cycles C_1 and C_2 with $(l(C)-2) = (l(C_1)-2)+(l(C_2)-2)+1$. Therefore safe paddings reduce the complexity of permutations. This observation and theorem 2 imply that every permutation with long cycles can be transformed into a simple permutation by a series of safe 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.

Lemma 5. *Every generalized sorting of π mimics a (genuine) sorting of π with the same number of reversals.*

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 5 implies that this generalized sorting mimics an *optimal* (genuine) sorting of π.

4 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 in a simple permutation with an oriented cycle. In this section, by cycles we mean *short* cycles and by permutations we mean *simple* permutations.

Denote the set of 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). Let $E(C) = \{(C_1, C_2) : C_1, C_2 \in V(C)$ and C_1 interleaves with C_2 in $\pi\}$ be the set of edges in the subgraph of H_π induced by $V(C)$. Denote the complement of $E(C)$ as $\overline{E}(C) = \{(C_1, C_2) : C_1, C_2 \in V(C)$ and C_1 does not interleave with $C_2\}$.

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 $H_{\pi\rho}(\mathcal{C}_\pi \setminus C, \mathcal{I}_{\pi\rho})$ of $\pi\rho$. This transformation results in complementing the subgraph induced by $V(C)$ as described by the following lemma. We denote $\overline{\mathcal{I}}_\pi = \mathcal{I}_\pi \setminus \{(C, D) : D \in V(C)\}$.

Lemma 6. *Let ρ be a reversal acting on an oriented (short) cycle C. Then (i) $\mathcal{I}_{\pi\rho} = (\overline{\mathcal{I}}_\pi \setminus E(C)) \cup \overline{E}(C)$, i.e. ρ removes edges $E(C)$ and adds edges $\overline{E}(C)$ to transform H_π into $H_{\pi\rho}$; (ii) ρ changes the orientation of a cycle $D \in \mathcal{C}_\pi$ iff $D \in V(C)$.*

Lemma 7. *Let ρ be a reversal acting on a cycle C and A, B be non-adjacent 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.

Lemma 8. *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 (i) every two vertices outside $K_1(\rho)$ which are adjacent in $H_{\pi\rho}$ are also adjacent in $H_{\pi\sigma}$; (ii) 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 7 implies (i) (C', D) is not an edge in H_π and (ii) (D, E) is an edge in H_π. Therefore, by lemma 6 reversal σ preserves the edge (D, E) in $H_{\pi\sigma}$. If both vertices D and E belong to $V(C)$ then lemma 6 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 7 implies that (C', D) and (C', E) are edges in H_π. Therefore, by lemma 6, (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 7 implies that for every vertex D outside $K_1(\rho)$, $D \in V(C)$ iff $D \in V(C')$. This observation and lemma 6 imply that the orientation of vertices outside $K_1(\rho)$ in $H_{\pi\sigma}$ and $H_{\pi\rho}$ is the same. □

Theorem 3. *For every oriented component K in H_π there exists a (safe) reversal $\rho \in \mathcal{R}(K)$ such that all components $K_1(\rho), K_2(\rho), \dots$ are oriented in $H_{\pi\rho}$.*

Proof Assume that a reversal $\rho \in \mathcal{R}(K)$ "breaks" K into a number of connected components $K_1(\rho), K_2(\rho), \dots$ in $H_{\pi\rho}$ and the first m of them 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 $K_1(\rho), \dots, K_m(\rho)$ $(m \geq 1)$ 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 $K_1(\rho)$. Note that $K_1(\rho)$ contains at least one vertex from $V(C)$ and consider the (non-empty) 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 $V_1 \setminus V$ are unoriented in π (lemma 6). Let C' be an (oriented) cycle in V and let σ be the reversal acting on C' in $G(\pi)$. Lemma 8 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 $K_{m+1}(\rho), K_{m+2}(\rho), \dots$ 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 $V_1 \neq V(C)$ then there exists an edge between an (oriented) cycle $C' \in V$ and an (unoriented) cycle $C'' \in V_1 \setminus V$ in $G(\pi)$. Lemma 6 implies that a reversal σ acting on C' in π orients the cycle C'' in $G(\pi)$. This observation and lemma 8 imply that σ reduces $index(\sigma)$ by at least 1 as compared to $index(\rho)$, a contradiction. If $V_1 = V(C)$ then all cycles of K_1 interleave with C. Since every unoriented short cycle in $G(\pi)$ has an interleaving cycle (4), there exist at least two vertices in $V(C)$. Moreover, there exist (oriented) cycles $C', C'' \in V_1$ such that (C', C'') are not interleaving in π (otherwise, lemma 6 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 6 implies that σ preserves the orientation of C'' thus reducing $index(\sigma)$ by at least 1 as compared to $index(\rho)$, a contradiction.

The found reversal 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. □

5 Clearing the hurdles

If π has an oriented component then theorem 3 implies that there exists a safe reversal in π. In this section we search for a safe reversal in the absence of any 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 $\{(x, y) : x, y \in P \text{ and } x \text{ is covered by } y\}$. Let $\mathcal{U}_\pi$ be the set of unoriented components in H_π and let $[U_{min}, U_{max}]$ be the interval between the leftmost and rightmost positions in an unoriented component $U \in \mathcal{U}_\pi$ (see section 2). Define $\overline{U}_{min} = \min_{U\in\mathcal{U}_\pi} U_{min}$, $\overline{U}_{max} = \max_{U\in\mathcal{U}_\pi} U_{max}$ and let $\overline{U}$ be an (*artificial*) component associated with the interval $[\overline{U}_{min}, \overline{U}_{max}]$.

Define $\overline{\mathcal{U}}_\pi$ as the set $\mathcal{U}_\pi \cup \overline{U}$ consisting of $|\mathcal{U}_\pi| + 1$ elements. Let $\prec \equiv \prec_\pi$ be the *containment* partial order on $\overline{\mathcal{U}}_\pi$ defined by the rule $U \prec W$ iff $[U_{min}, U_{max}] \subset [W_{min}, W_{max}]$ for $U, W \in \overline{\mathcal{U}}_\pi$. If there exists the *greatest* unoriented component U in π (i.e. $[U_{min}, U_{max}] = [\overline{U}_{min}, \overline{U}_{max}]$) we assume that there exist two elements ("real" component U and "artificial" component $\overline{U}$) corresponding to the greatest interval and that $U \prec_\pi \overline{U}$. Let Ω_π be the *tree* representing the cover graph of the partial order $\prec_\pi$ on $\overline{\mathcal{U}}_\pi$ (Fig. 4a). Every vertex in Ω_π but $\overline{U}$ is associated with an unoriented component in $\mathcal{U}_\pi$. In the case π has the greatest hurdle we assume that the leaf $\overline{U}$ is associated with this greatest hurdle (i.e. in this case there are *two vertices* in Ω_π corresponding to the greatest hurdle, leaf $\overline{U}$ and its neighbor, the greatest hurdle $U \in \mathcal{U}_\pi$). In the case $\overline{U}$ is a leaf in Ω_π it is not necessarily a hurdle (for example, $\overline{U}$ is a leaf in Ω_π but not a hurdle for a permutation π shown in Fig. 3a). Therefore, the number of leaves in Ω_π coincides with the number of hurdles $h(\pi)$ except for the cases when (i) there exists only one unoriented component in π (in this case Ω_π has two leaves while $h(\pi) = 1$) and (ii) there exists the greatest element in $\mathcal{U}_\pi$ which is not a hurdle, i.e. this element separates other hurdles (in this case the number of leaves equals $h(\pi)+1$). [2] Lemma 4 implies

Lemma 9. *(Hurdle cutting) Every reversal ρ on a cycle in a hurdle K cuts off the leaf K from the cover*

[2] Although an addition of an "artificial" component $\overline{U}$ might look a bit strange and unnecessary we will find below that such an addition greatly facilitates the analysis of technical details.

graph of π, i.e. $\Omega_{\pi\rho} = \Omega_\pi \setminus K$.

A hurdle $K \in \mathcal{U}_\pi$ *protects* a non-hurdle $U \in \mathcal{U}_\pi$ if deleting K from $\mathcal{U}_\pi$ transforms U from a non-hurdle into a hurdle (i.e. U is a hurdle in $\mathcal{U}_\pi \setminus K$). A hurdle in π is a *superhurdle* if it protects a non-hurdle $U \in \mathcal{U}_\pi$, and a *simple hurdle* otherwise (see Fig. 4 and 6 for examples). Clearly

Lemma 10. *A reversal acting on a cycle of a simple hurdle is safe.*

Unfortunately, a reversal acting on cycle of a superhurdle is unsafe since it transforms a non-hurdle 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 which is the *least common ancestor* of L and M and define $\overline{LCA}(L, M)$ as the *least common ancestor* of L and M *which does not separate* L and M. If L corresponds to the *greatest* hurdle U there are two elements U and $\overline{U}$ in $\mathcal{U}_\pi$ corresponding to the same (greatest) interval $[U_{min}, U_{max}] = [\overline{U}_{min}, \overline{U}_{max}]$. 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 $V \setminus (W \setminus w)$ 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 a contraction reduces the number of vertices in G by $|W| - 1$ if $w \in W$ and by $|W|$ if $w \notin W$. Let L and M be two hurdles in 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 $\Omega_\pi(L, M)$ corresponds to deleting the elements of the set $PATH(L, M) \setminus LCA(L, M)$ from the partial order $\prec_\pi$.

Lemma 11. *(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 $\overline{LCA}(L, M)$, i.e. $\Omega_{\pi\rho} = \Omega_\pi(L, M)$.*

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 $G(\pi\rho)$ (Fig. 4). 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 component from $PATH(L, M) \setminus \overline{LCA}(L, M)$ 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 $\mathcal{U}_\pi \setminus PATH(L, M)$ remains unoriented in $\pi\rho$. Component $LCA(L, M)$ remains unoriented iff $LCA(L, M) = \overline{LCA}(L, M)$. Every component which 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 then the rightmost position of W, i.e. $U_{max} < W_{max}$. Order the hurdles of π in the increasing order of their rightmost positions $U(1) < \ldots < U(l) \equiv L < \ldots < U(m) \equiv M < \ldots < U(h(\pi))$ and define the sets of hurdles $BETWEEN(L, M) = \{U(i) : l < i < m\}$ and $OUTSIDE(L, M) = \{U(i) : i \notin [l, m]\}$.

Lemma 12. *Let ρ be a reversal merging hurdles L and M in π. If both sets $BETWEEN(L, M)$ and $OUTSIDE(L, M)$ are non-empty then ρ is safe.*

Proof Let $U' \in BETWEEN(L, M)$ and $U'' \in OUTSIDE(L, M)$. Lemma 11 implies that ρ deletes 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 non-hurdle 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 11 K would be deleted from $\pi\rho$). W.l.o.g. 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 11, K would be deleted from $\pi\rho$). If $L, M \not\prec_\pi K$ then $L < U' \prec_\pi K < M$, i.e. 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 non-hurdle 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 $h(\pi\rho) = h(\pi) - 2$. Since $b(\pi\rho) = b(\pi)$ and $c(\pi\rho) = c(\pi) - 1$, the reversal ρ is safe. □

Lemma 13. *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 $(1 + \frac{h(\pi)}{2})$-th hurdles in this order. Since $h(\pi) > 3$, both sets $BETWEEN(L, M)$ and $OUTSIDE(L, M)$ are non-empty and by lemma 12 the reversal ρ merging L and M is safe. □

Lemma 9 and 11 imply

Lemma 14. *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.*

Lemmas 13 shows that hurdles merging guarantees a safe reversals even in the absence of simple hurdles. On the negative note, hurdles merging never transforms a superhurdle into a simple hurdle.

Lemma 15. *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 $\pi\rho$.*

6 Fortresses

Lemmas 13 and 14 imply that unless Ω_π is a *homeomorph* of the *3-star* (a graph with 3 edges incident on the same vertex) there exists a safe reversal in π. On the other hand, if π has a simple hurdle then lemma 10 implies that there exists a safe reversal in π. Therefore, unless Ω_π is a homeomorph of 3-star with 3 superhurdles, called a *3-fortress*, there exists a safe reversal in π (Fig. 6a).

Lemma 16. *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 which can reduce $h(\pi)$ by 2 is hurdles merging. On the other hand, lemma 11 implies that hurdles merging in a 3-fortress can reduce $h(\pi)$ by at most 1. Therefore, $\Delta h \geq -1$. Note that for every reversal ρ which does not act on edges of the *same* cycle, $\Delta(b-c) = 1$ and therefore every such reversal 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 $h(\pi)$. If $h(\pi\rho) > h(\pi)$ then $\Delta(b-c+h) \geq 0$ and ρ is unsafe. If $h(\pi\rho) = h(\pi)$ then the number of superhurdles in $\pi\rho$ and π is the same thus implying that $\pi\rho$ is a 3-fortress, a contradiction. □

Lemma 17. *If π is a 3-fortress then $d(\pi) = b(\pi) - c(\pi) + h(\pi) + 1$.*

Proof Lemma 16 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 3) without affecting unoriented components. Therefore we can assume that π is a 3-fortress without oriented 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 11). 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 3) leading to a permutation σ with $h(\sigma) = 2$. Theorems 2, 3 and lemma 14 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 5 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 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 18. *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 Arguments similar to the proof of lemma 16 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 11 implies that every such reversal reduces the number of hurdles by 2, and in the case $h(\pi) > 3$, does not create new hurdles. Thus lemma 15 implies that $\pi\rho$ is a fortress with $h(\pi) - 2$ superhurdles. □

Lemma 19. *If π is a fortress then $d(\pi) \geq b(\pi) - c(\pi) + h(\pi) + 1$.*

Proof Lemma 18 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 16 every sorting of a fortress contains at least one unsafe reversal and hence $d(\pi) \geq b(\pi) - c(\pi) + h(\pi) + 1$. □

Theorem 4. *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 3) and safe hurdles mergings (lemmas 13 and 14) 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 hurdle cutting (lemma 10) leading 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. Thus, if π is not a fortress, there exists a generalized sorting of π using only safe paddings and $b(\pi) - c(\pi) + h(\pi)$ reversals. Theorem 1 and lemma 5 imply that this generalized sorting mimics optimal (genuine) sorting of π. If π is a fortress there exists a sequence of safe paddings (theorems 2), safe reversals in oriented components (theorem 3) and safe hurdles mergings (lemma 13) leading to a 3-fortress. This 3-fortress can be sorted by a series of reversals having at most one unsafe reversal. Therefore there exists a generalized sorting of π using $b(\pi)-c(\pi)+h(\pi)+1$ reversals. Lemmas 5 and 19 imply that this generalized sorting mimics optimal (genuine) sorting of π by $b(\pi) - c(\pi) + h(\pi) + 1$ reversals. □

7 Polynomial algorithm

Theorem 4 motivates the algorithm *Reversal_Sort* which optimally sorts a signed permutation of order n in $O(n^4)$ time. We sketch an $O(n^4)$ implementation of *Reversal_Sort*. Note that every iteration of **while** loop reduces the function $complexity(\pi) + 3d(\pi)$ 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. For a simple permutation, such search can be implemented in $O(n^3)$ time, thus implying that the overall running time of *Reversal_Sort* is $O(n^4)$.

```
Algorithm Reversal_Sort(π)
  while π is not sorted
      if π has a long cycle
          select a safe (g, b)-padding ρ of π (theorem 2)
      else if π has an oriented component
          select a safe reversal ρ in this component
          (theorem 3)
      else if π has an even number of hurdles
          select a safe reversal ρ merging two hurdles in π
          (lemma 13 and 14)
      else if π has at least one simple hurdle
          select a safe reversal ρ cutting this hurdle in π
          (lemmas 10 and 14)
      else if π is a fortress with more than 3 superhurdles
          select a safe reversal ρ merging two
          (super)hurdles in π (lemma 13)
      else   /* π is a 3-fortress */
          select an (un)safe reversal ρ merging two
          arbitrary (super)hurdles in π
      π ← π · ρ
  endwhile
  mimic (genuine) sorting of π using the computed
  generalized sorting of π (lemma 5)
```

We omit an amortized analysis of *Reversal_Sort* leading to further reduction of running time. Instead, we describe a simpler version of *Reversal_Sort* which does not use paddings and runs in $O(n^5)$ time. Define $f(\pi) = \begin{cases} 1, & \text{if } \pi \text{ is a fortress} \\ 0, & \text{otherwise} \end{cases}$. A reversal ρ is *valid*, if $\Delta(b - c + h + f) = -1$. Proofs of theorem 1 and lemma 19 imply that $\Delta(b - c + h + f) \geq -1$. This inequality and theorem 4 imply that (i) for every permutation π there exists a valid reversal in π and (ii) every sequence of valid reversals sorting π is optimal. These observations motivate *Reversal_Sort_Simple* which is very fast in practice.

```
Algorithm Reversal_Sort_Simple(π)
    while π is not sorted
        select a valid reversal ρ in π
        π ← π · ρ
    endwhile
```

Reversal_Sort_Simple was implemented and tested on biological data. The results of these tests are described in Hannenhalli and Pevzner, 1994, 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 (2) is extremely tight since $h(\pi)$ is small for "random" permutations and zero for most of the biological data.

8 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 and Monique Turmel for sending us unpublished experimental data on gene orders of *Chlamydomonas gelatinosa* and *Chlamydomonas reinhardtii* for testing *Reversal_Sort_Simple*.

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. Computing).

[3] V. Bafna and P. Pevzner. Sorting by reversals: Genome rearrangements in plant organelles and evolutionary history of X chromosome. *Mol. Biol. and Evol.*, 12:239–246, 1995.

[4] V. Bafna and P. Pevzner. Sorting by transpositions. In *Proc. of 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 614–623, 1995b.

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

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

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

[8] S. Hannenhalli. Polynomial algorithm for computing translocation distance between genomes. In *Proc. of 6th Ann. Symp. on Combinatorial Pattern Matching*, 1995. (to appear).

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

[10] S. Hannenhalli and P. Pevzner. Reversals do not cut long strips. Technical Report: CSE-94-074, Department of Computer Science and Engineering, The Pennsylvania State University, 1994.

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

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

[13] J. Kececioglu and D. Gusfield. Reconstructing a history of recombinations from a set of sequences. In *5th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 471–480, 1994.

[14] J. Kececioglu and R. Ravi. Of mice and men: Evolutionary distances between genomes under translocation. In *Proc. of 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 604–613, 1995.

[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*, Lecture Notes in Computer Science 684, pages 87–105. Springer Verlag, 1993. (Extended version has appeared in Algorithmica, 13: 180-210, 1995.).

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

[17] C. A. Makaroff and J. D. Palmer. Mitochondrial DNA rearrangements and transcriptional alterations in the male sterile cytoplasm of Ogura radish. *Molecular Cellular Biology*, 8:1474–1480, 1988.

[18] 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.

[19] 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.

[20] P.A. Pevzner and M.S. Waterman. Open combinatorial problems in computational molecular biology. In *3rd Israel Symposium on Theory of Computing and Systems*, pages 158–163. IEEE Computer Society Press, 1995.

[21] D. Sankoff. Edit distance for genome comparison based on non-local operations. In *Proc. of 3rd Ann. Symp. on Combinatorial Pattern Matching*, Lecture Notes in Computer Science 644, pages 121–135. Springer Verlag, 1992.

[22] D. Sankoff, R. Cedergren, and Y. Abel. Genomic divergence through gene rearrangement. In *Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences*, chapter 26, pages 428–438. Academic Press, 1990.

[23] 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.

[24] A. H. Sturtevant and T.Dobzhansky. 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, 1936.

[25] G. A. Watterson, W. J. Ewens, T. E. Hall, and A. Morgan. The chromosome inversion problem. *Journal of Theoretical Biology*, 99:1–7, 1982.

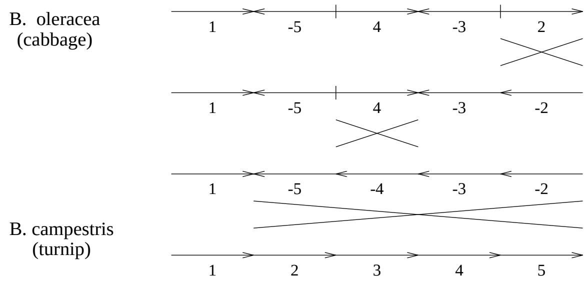

Figure 1: "Transformation" of cabbage into turnip

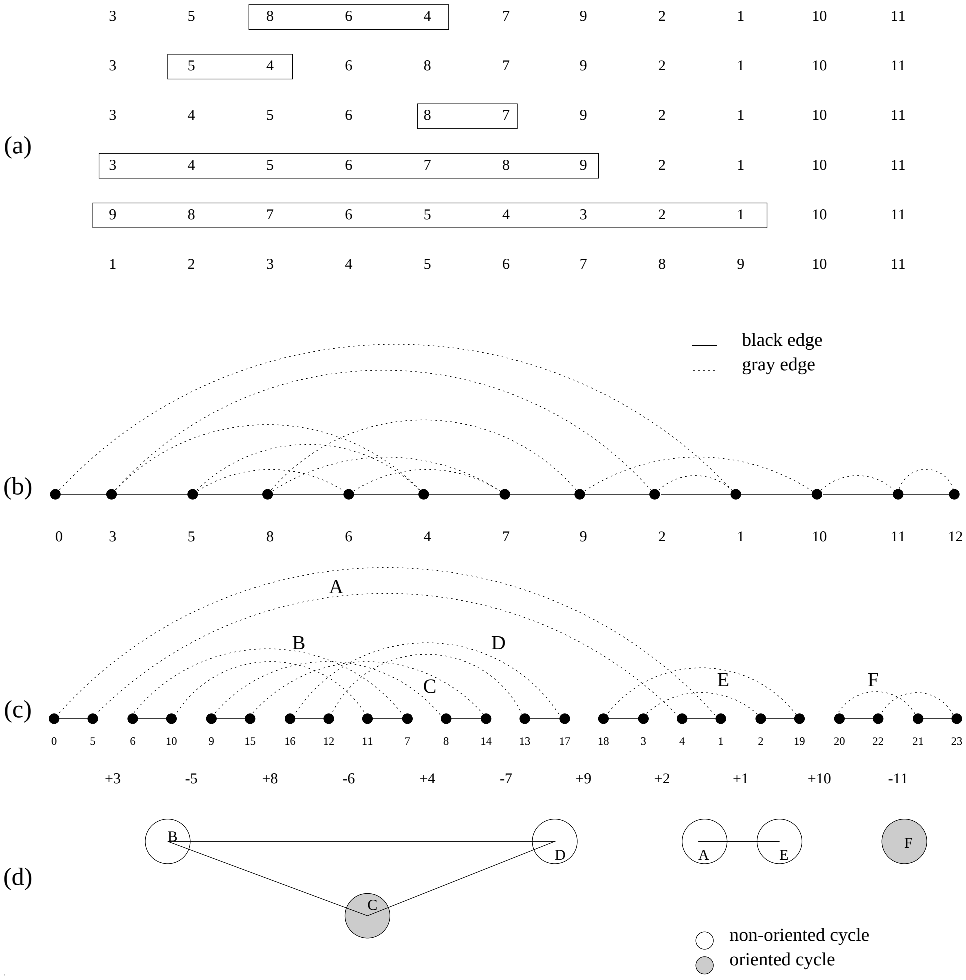

Figure 2: (a) Optimal sorting of a permutation $\sigma = (3\ 5\ 8\ 6\ 4\ 7\ 9\ 2\ 1\ 10\ 11)$ by 5 reversals (b) Breakpoint graph $G(\sigma)$ can be decomposed into three short cycles ($(8,9,7,6,8)$, $(8,5,4,7,8)$ and $(3,5,6,4,3)$) and one long cycle $(0,1,10,9,2,3,0)$ thus implying that $c(\sigma) = 4$. (c) Transformation of a signed permutation into an unsigned permutation π and the breakpoint graph $G(\pi)$. Gray edges $(8,9)$ and $(22,23)$ are oriented while gray edges $(4,5)$ and $(18,19)$ are unoriented (note that a gray edge (π_i, π_j) incident to black edges (π_k, π_i) and (π_j, π_l) is oriented iff $i - k = j - l$). Cycles C and F are oriented while cycles A, B, D and E are unoriented. Gray edges $(6,7)$ and $(12,13)$ are interleaving while gray edges $(6,7)$ and $(4,5)$ are non-interleaving. (d) Interleaving graph H_π with two oriented and one unoriented component.

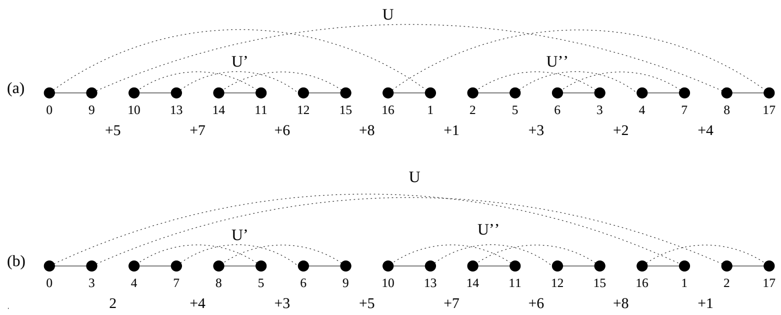

Figure 3: Permutation in (a) has 2 minimal and no greatest hurdle ($h(\pi) = 2$) since the greatest unoriented component U separates hurdles U' and U''. Permutation in (b) has 2 minimal (U' and U'') and 1 greatest hurdle U ($h(\pi) = 3$).

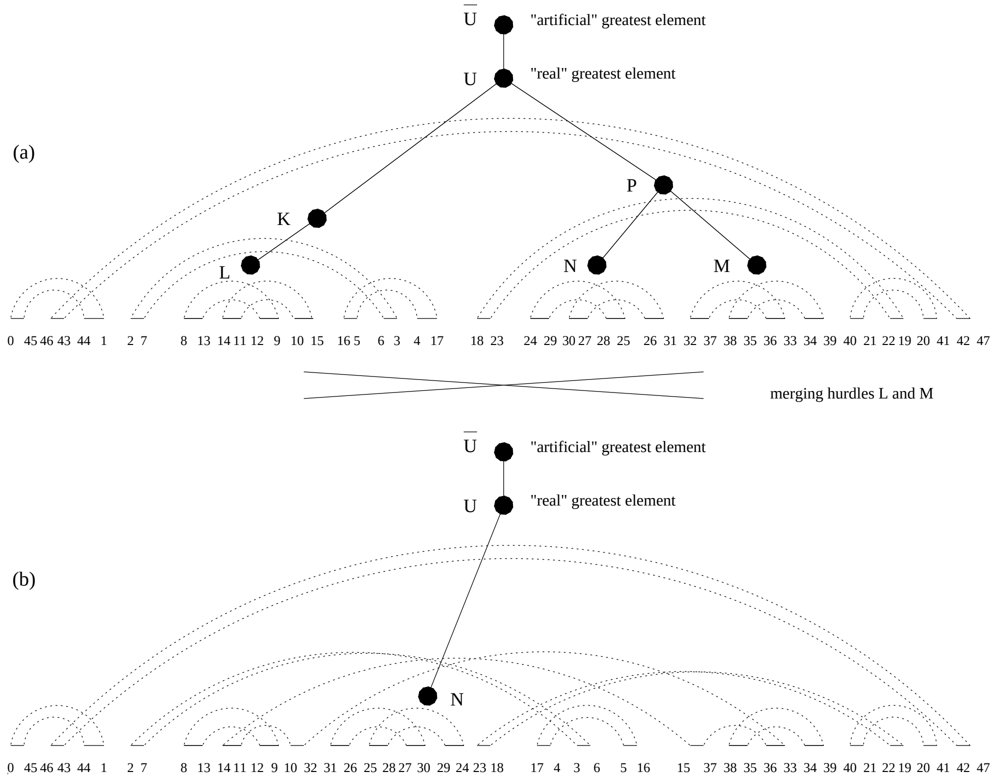

Figure 4: (a) A cover graph Ω_π of a permutation π with "real" unoriented components K, L, M, N, P, U and an "artificial" component $\overline{U}$; (b) A reversal ρ merging hurdles L and M in π transforms unoriented components L, K, P and M into an oriented component which "disappears from $\Omega_{\pi\rho}$. 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 $PATH(A, F) = \{L, K, U, P, M\}$. Components M, N and U are simple hurdles while component L is a superhurdle (deleting L transforms a non-hurdle K into a hurdle).

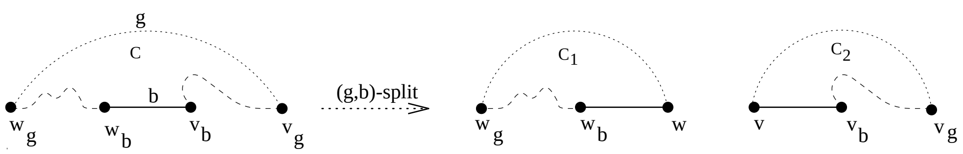

Figure 5: A (g, b)-split transforming a cycle C in $G(\pi)$ into cycles C_1 and C_2 in $\hat{G}(\pi)$.

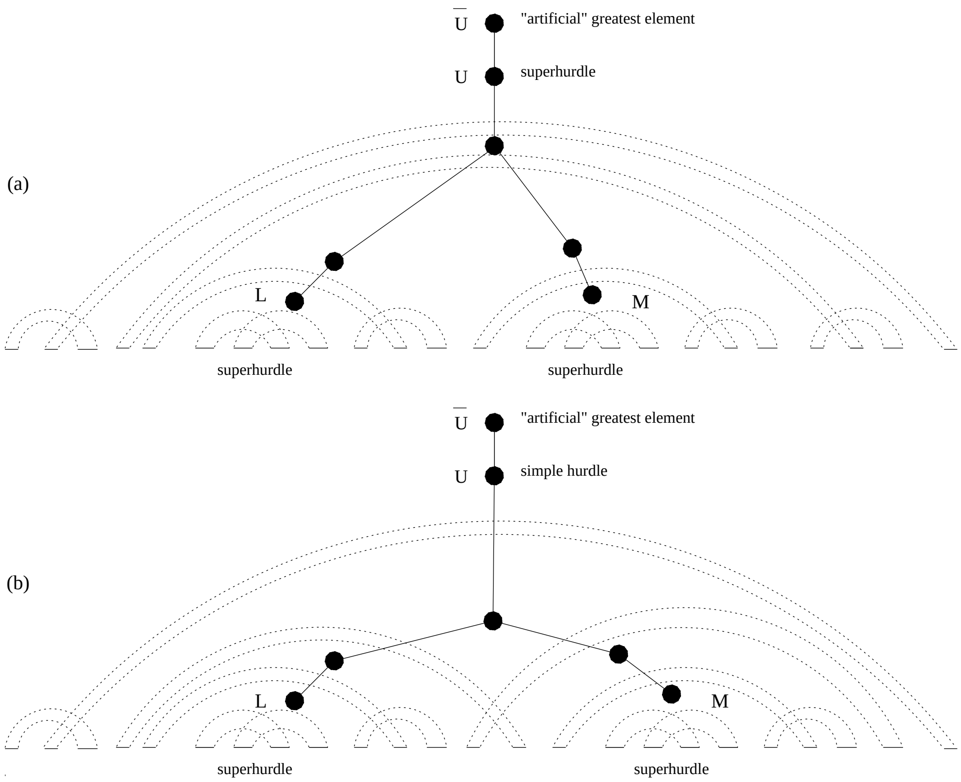

Figure 6: Permutation in (a) is a 3-fortress while permutation in (b) with the same cover graph is not a fortress.