Lou XW, Zhu DM. Sorting unsigned permutations by weighted reversals, transpositions, and transreversals. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 25(4): 853–863 July 2010. DOI 10.1007/s11390-010-1066-7

Sorting Unsigned Permutations by Weighted Reversals, Transpositions, and Transreversals

Xiao-Wen Lou (娄晓文) and Da-Ming Zhu* (朱大铭), *Senior Member, CCF*

School of Computer Science and Technology, Shandong University, Jinan 250101, China

E-mail: xwlou@mail.sdu.edu.cn; dmzhu@sdu.edu.cn

Received January 7, 2009; revised March 1, 2010.

Abstract Reversals, transpositions and transreversals are common events in genome rearrangement. The genome rearrangement sorting problem is to transform one genome into another using the minimum number of given rearrangement operations. An integer permutation is used to represent a genome in many cases. It can be divided into disjoint strips with each strip denoting a block of consecutive integers. A singleton is a strip of one integer. And the genome rearrangement problem turns into the problem of sorting a permutation into the identity permutation equivalently. Hannenhalli and Pevzner designed a polynomial time algorithm for the unsigned reversal sorting problem on those permutations with $O(\log n)$ singletons. In this paper, first we describe one case in which Hannenhalli and Pevzner's algorithm may fail and propose a corrected approach. In addition, we propose a $(1+\varepsilon)$-approximation algorithm for sorting unsigned permutations with $O(\log n)$ singletons by reversals of weight 1 and transpositions/transreversals of weight 2.

Keywords approximation algorithm, genome rearrangement, sorting, reversal, transposition

1 Introduction

As one of the most promising way to understand evolution between species, the genome rearrangement sorting problem is to find a shortest sequence of evolutionary events (such as reversals, transpositions, transreversals) that transform one genome into another. The parsimony approach has been widely used in many challenging algorithmic problems[1]. Genomes are represented as signed or unsigned permutations, where each element stands for a gene. When comparing two permutations, one of them can be viewed as the identity permutation $\iota = [1, 2, \ldots, n]$ or the signed version $\iota' = [+1, +2, \ldots, +n]$ simply by substitution.

The problem of sorting unsigned permutation by reversals is proved to be NP-hard by Caprara[2], and several approximation algorithms have been suggested[3-5]. For the problem of sorting signed permutation by reversals, a polynomial algorithm was first given by Hannenhalli and Pevzner[6], and the running time has been improved subsequently[7-10]. Hannenhalli and Pevzner discovered that singleton is the major obstacle for sorting unsigned permutations by reversals. They designed a polynomial algorithm for unsigned permutations with $O(\log n)$ singletons[11]. However, there is one case in which Hannenhalli and Pevzner's algorithm may fail. We describe the case and give a corrected approach.

For the problem of sorting by transpositions, several 1.5-approximation algorithms are proposed[4,12-14]. Elias and Hartman improved the performance ratio to 1.375 recently[15].

Bafna and Pevzner suggested the sorting problem that considers reversals and transpositions simultaneously as an approach for understanding the genome rearrangements related to the evolution of mammalian and viral[12]. For signed permutations, Walter *et al.* gave a 2-approximation algorithm for sorting by reversals and transpositions[16]. Gu *et al.* gave a 2-approximation algorithm for sorting by transpositions and transreversals[17]. Lin and Xue introduced the operation *revrev*, and gave a 1.75-approximation algorithm for sorting by reversals, transpositions, transreversals and revrevs[18]. Hartman and Sharan gave a 1.5-approximation algorithm for sorting circular permutations by transpositions and transreversals[19]. For unsigned permutations, Walter *et al.* gave a 3-approximation algorithm which allows reversal and

Regular Paper

Supported by the National Natural Science Foundation of China under Grant No. 60970003, the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20090131110009, the Key Science-Technology Project of Shandong Province of China under Grant No. 2006GG2201005.

*Corresponding Author

transposition operations[16]. Rahman *et al.* improved the performance ratio to $2.8386 + \delta$[20]. Lou and Zhu gave a 2.25-approximation algorithm for sorting circular permutations by reversals, transpositions and transreversals[21].

Eriksen observed that an algorithm looking for the minimal number of such operations will produce a solution heavily biased towards transposition[22]. Instead, he studied the weighted reversal and transposition/transreversal sorting problem, which assigns 1 to reversal and 2 to transposition or transreversal. The task is to sort a permutation by reversals or transpositions or transreversals with the minimum value of $\mathrm{rev}(s) + 2\mathrm{trp}(s)$, where $\mathrm{rev}(s)$ and $\mathrm{trp}(s)$ are the number of reversals and transpositions/transreversals in a certain scenario s respectively. He designed a $(1+\varepsilon)$-approximation algorithm for this weighted sorting problem on signed permutations. Later, Bader *et al.* gave a 1.5-approximation algorithm for any weight proportion between $1:1$ and $1:2$ on signed permutations[23].

In this paper, we focus on unsigned permutations, and the weight proportion is $1:2$ as in [22]. We present a $(1+\varepsilon)$-approximation algorithm for sorting unsigned permutations by weighted reversals and transpositions/transreversals, where the permutation is restricted to containing $O(\log n)$ singletons.

2 Preliminaries

A genome can be represented as a permutation of integers, where each integer stands for a gene. Let $\pi = [g_1, \ldots, g_n]$ be a permutation of $\{1, \ldots, n\}$. If each element of π has no signs, call it an *unsigned permutation*. Otherwise, if each element of π has a sign of "+" or "−" representing the gene's direction, call it a *signed permutation*. Let $\pi = [g_1, \ldots, g_n]$ be an unsigned permutation and $\pi' = [g'_1, \ldots, g'_n]$ be a signed permutation. π' is called a *spin* of π if either $g'_i = +g_i$ or $g'_i = -g_i$ for $1 \leqslant i \leqslant n$. A *segment* of π is a sequence of consecutive elements in π. For example, segment $[g_i, \ldots, g_j]$ $(i \leqslant j)$ of π contains all the elements from g_i to g_j. In unichromosomal genomes, three kinds of rearrangement operations are commonly considered: reversal, transposition and transreversal. A *reversal* $r(i,j)$ $(i<j)$ on π reverses the order of elements in the segment $[g_{i+1}, \ldots, g_j]$, and accordingly the permutation becomes $\sigma = [g_1, \ldots, g_i, g_j, \ldots, g_{i+1}, g_{j+1}, \ldots, g_n]$. A *transposition* $t(i,j,k)$ $(i<j<k)$ exchanges two consecutive segments $[g_{i+1}, \ldots, g_j]$ and $[g_{j+1}, \ldots, g_k]$ of π, and the permutation becomes $\sigma = [g_1, \ldots, g_i, g_{j+1}, \ldots, g_k, g_{i+1}, \ldots, g_j, g_{k+1}, \ldots, g_n]$. A *transreversal* $t_{\mathrm{r}}(i,j,k)$ $(i<j<k)$ reverses the segment $[g_{i+1}, \ldots, g_j]$ while exchanging the two consecutive segments $[g_{i+1}, \ldots, g_j]$ and $[g_{j+1}, \ldots, g_k]$, hence the permutation becomes $\sigma = [g_1, \ldots, g_i, g_{j+1}, \ldots, g_k, g_j, \ldots, g_{i+1}, g_{k+1}, \ldots, g_n]$. For a signed permutation π', reversing a segment of π' changes the direction of genes in the segment. Thus each element in the reversed segment also has its sign flipped.

The *reversal sorting* problem asks to transform π into the identity permutation ι by the minimum number of reversals, and this number is called the *reversal distance* of π, denoted by $d_{\mathrm{r}}(\pi)$. A *reversal scenario* is a sequence of reversal operations that transforms π into ι. The *weighted sorting* problem asks to transform π into ι by the minimum weighted sum of reversals, transpositions, and transreversals, in which one reversal counts for once and one transposition or transreversal counts for twice. A *weighted scenario* is a sequence of rearrangement operations (reversal or transposition or transreversal) that transforms π into ι. Formally, the *weighted distance* of π is defined as: $d_{\mathrm{r+t}}(\pi) = \min_{\phi\in\Phi}\{\mathrm{rev}(\phi) + 2\mathrm{trp}(\phi)\}$, where Φ is the set of weighted scenarios sorting π, $\mathrm{rev}(\phi)$ and $\mathrm{trp}(\phi)$ is the number of reversals and transpositions/transreversals in a certain weighted scenario ϕ respectively. For an arbitrary unsigned permutation π of order n, let Π be the set of all 2^n spins of π. If there is a spin $\pi' \in \Pi$ satisfying $d_{\mathrm{r}}(\pi') = d_{\mathrm{r}}(\pi)$, call π' an *optimal r-spin*; if there is a spin $\pi' \in \Pi$ satisfying $d_{\mathrm{r+t}}(\pi') = d_{\mathrm{r+t}}(\pi)$, call π' an *optimal rt-spin*. The process of finding an optimal r-spin or rt-spin of an unsigned permutation π can be viewed as the process of finding an optimal assignment of "+" or "−" to each element of π.

Let $i \sim j$ if $|i-j| = 1$. For an unsigned permutation π, a pair of consecutive elements (g_i, g_{i+1}) forms an *adjacency* if $g_i \sim g_{i+1}$; otherwise a *breakpoint*. A segment $[g_i, \ldots, g_j]$ of π is called a *strip* if each (g_k, g_{k+1}) is an adjacency for $i \leqslant k < j$, but both (g_{i-1}, g_i) and (g_j, g_{j+1}) are breakpoints. A strip of one element is a *singleton*; a strip of k elements is a *k-strip* $(k \geqslant 2)$; if $k \geqslant 3$, call it a *long strip*. If π has no singletons, call it a *singleton-free permutation*. For the problem of sorting by transpositions, a strip can be reduced to one element (i.e., singleton) since it has no signs[4]. However, when reversals are taken into account, doing this will lose the orientation information. The basic idea of the algorithm is to figure out the optimal assignment for long strips and 2-strips, thus a singleton-free permutation becomes a signed permutation. Based on this, we can allow a few singletons ($O(\log n)$ to be precise) by enumerating their signs. But with more singletons, the computation cannot be completed in polynomial time.

A strip $s = [g_i, \ldots, g_j]$ of π is *increasing* (*decreasing*) if $g_i < g_j$ $(g_i > g_j)$. Let π' be a spin of π, an increasing (decreasing) strip s of π is *canonical* in π' if all the elements of s are positive (negative) in π'. An increasing (decreasing) strip s of π is *anti-canonical* in π' if all the elements of s are negative (positive) in π'. An anti-canonical 2-strip is called an *anti-strip*. If every strip of π is canonical in π', then call π' a *canonical spin* of π. Two spins π'_1 and π'_2 are *twins* (with respect to segment $s = [g_i, \ldots, g_j]$) if they differ only in the signs of elements in s. In the following, we use *flip*(s) to denote flipping the sign of every element in s.

For a signed permutation π', there exists a polynomial algorithm computing the reversal distance. Here are some details. First, transform a signed permutation $\pi' = [g'_1, \ldots, g'_n]$ of order n to a permutation $\pi'' = [0, g_1, g_2, \ldots, g_{2n-1}, g_{2n}, 2n+1]$ of order $2(n+1)$ as follows. Replace the positive element $+x$ by $(2x-1, 2x)$, and the negative element $-x$ by $(2x, 2x-1)$; then add 0 at the beginning and $2n+1$ at the end. Call π'' the *extended permutation* of π'. The reversal operation on π'' never touches 0 and $2n+1$, and never breaks the pair $(2x-1, 2x)$ or $(2x, 2x-1)$. Therefore, it has the same effect of operating on π'. The *breakpoint graph* $G(\pi'')$ of π'' is defined as follows: set a vertex of $G(\pi'')$ for each element of π''; draw a gray edge between vertices $2x$ and $2x+1$ if their positions are not consecutive in π''; draw a *black edge* between vertices g_{2i} and g_{2i+1} if they form a breakpoint. Note that every vertex has degree 2 in $G(\pi'')$, so it can be uniquely decomposed into *alternating cycles*, i.e., the colors of every two consecutive edges of the cycle are distinct. An alternating cycle with l black edges has *length* l, denoted as an *l-cycle*. Note that $l \geqslant 2$. A gray edge is *oriented* if the two black edges incident to it have the opposite directions when we travel along the path; otherwise, *unoriented*. A cycle is *oriented* if it contains at least one oriented gray edge; otherwise, an *unoriented cycle*. A gray edge in a breakpoint graph $G(\pi'')$ is *inside* a strip s if it is incident to an element of s[11]. Two gray edges $e_1 = (g_{i_1}, g_{i_2})$ and $e_2 = (g_{j_1}, g_{j_2})$ are *crossing* if $i_1 < j_1 < i_2 < j_2$ or $j_1 < i_1 < j_2 < i_2$. Cycles C_1 and C_2 are *crossing* if there exist crossing gray edges $e_1 \in C_1$ and $e_2 \in C_2$. If we take each alternating cycle of $G(\pi'')$ as a vertex, and draw an edge between two vertices if the cycles they represent are crossing, we get another graph G_π. Those alternating cycles of $G(\pi'')$ corresponding to all the vertices of one connected component of G_π together form a *component* of $G(\pi'')$. For every component of $G(\pi'')$, if it contains at least one oriented cycle, call it an *oriented component*; otherwise, an *unoriented component*. Imagining we bend a permutation counterclockwise into a circle and make $(0, 2n+1)$ an adjacency. If there is an interval on the circle which contains one and only one unoriented component, then this component is called a *hurdle*. If an unoriented component is not a hurdle, call it a *non-hurdle*. Note that a non-hurdle always stretches over an interval that contains a hurdle. A *hurdle* is called a *super hurdle* if removing it will turn a non-hurdle into a hurdle[6,22]; otherwise, a *simple hurdle*. If $G(\pi'')$ has an odd number of hurdles and all these hurdles are super hurdles, call the permutation π' a *fortress*. Let $b(\pi')$, $c(\pi')$, $h(\pi')$ denote the numbers of breakpoints, cycles and hurdles in $G(\pi'')$, respectively. Let $f(\pi') = 1$ if π' is a fortress; otherwise, $f(\pi') = 0$. It is proved in [6] that:

$$d_{\mathrm{r}}(\pi') = b(\pi') - c(\pi') + h(\pi') + f(\pi'). \tag{1}$$

Fig.1 gives an example of a breakpoint graph $G(\pi'')$. It consists of six unoriented components $K_1, \ldots, K_6$, where K_3, K_4 and K_6 are simple hurdles, K_1 is a super hurdle, K_2 and K_5 are non-hurdles.

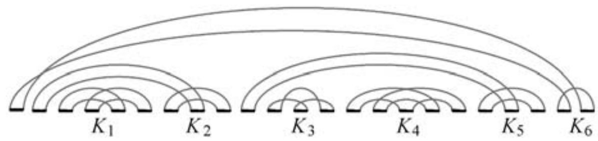

Fig.1. A breakpoint graph with six unoriented components.

For a signed permutation π', there exists a $(1+\varepsilon)$-approximation algorithm for sorting π' by weighted reversals, transpositions, and transreversals. First construct the breakpoint graph $G(\pi'')$ from π' as aforementioned. For a component K of $G(\pi'')$, we use $b(K)$ and $c(K)$ to denote the numbers of breakpoints and cycles in K respectively. If K is oriented, we can use $b(K) - c(K)$ reversals to transform K into $b(K)$ disjoint adjacencies, which is just an optimal weighted scenario sorting K. However, for an unoriented component, we cannot always have a method to sort it optimally by the weighted operations. Thus if $d_{\mathrm{r+t}}(K) > b(K) - c(K)$, call K a *strongly unoriented component* (*SUC*). If an unoriented component is not an SUC, call it a *non-SUC*. Again imagine bending a permutation into a circle, if there is an interval on the circle which contains one and only one SUC, then this SUC is called a *strong hurdle*. A strong hurdle is a *super-strong-hurdle* if removing it will make another SUC become a strong hurdle. If a strong hurdle is not a super-strong-hurdle, call it a *simple-strong-hurdle*. If $G(\pi'')$ has an odd number of strong hurdles and all these strong hurdles are super-strong-hurdles, the permutation π' is called a *strong fortress*. Let $b(\pi')$, $c(\pi')$ and $h_{\mathrm{t}}(\pi')$ denote the numbers of breakpoints, cycles and strong hurdles in $G(\pi'')$, respectively. Let $f_{\mathrm{t}}(\pi') = 1$ if π' is a strong fortress; otherwise $f_{\mathrm{t}}(\pi') = 0$. It is proved in [22] that:

$$d_{\mathrm{r+t}}(\pi') = b(\pi') - c(\pi') + h_{\mathrm{t}}(\pi') + f_{\mathrm{t}}(\pi'). \tag{2}$$

Since $h_{\mathrm{t}}(\pi')$ and $f_{\mathrm{t}}(\pi')$ cannot be computed exactly, only an approximation algorithm is available[22]. As an example, in Fig.1, K_1, K_2, K_4 and K_5 are all SUC's, K_3 and K_6 are non-SUC's, since each of them can be removed by one transposition, thus $d_{\rm r+t}(K_3) = 2 = b(K_3) - c(K_3)$. And both K_1 and K_4 are super-strong-hurdles.

3 A Note on Sorting Unsigned Permutations by Reversals

Using (1), Hannenhalli and Pevzner proved the following lemmas:

Lemma 1[11]. *For any unsigned permutation π, there exists an optimal r-spin π' of π such that all the long strips of π are canonical in π'; every 2-strip of π is either canonical or anti-canonical in π'.*

Lemma 2[11]. *For any unsigned permutation π, there exists an optimal r-spin π' of π such that (I) an unoriented component in π' does not contain any anti-strip, and (II) an oriented component in π' contains at most one anti-strip.*

Suppose π is a singleton-free unsigned permutation and π' is the canonical spin of π. In Hannenhalli and Pevzner's algorithm[11], for every unoriented component K of $G(\pi'')$, if K contains 2-strips, select any one of them and transform it into an anti-strip, all the other 2-strips remain canonical. Such a spin of π is called a *super spin*. They proved that every super spin of a singleton-free permutation is an optimal r-spin. In fact, such a super spin is not always optimal. Here is an example: suppose $\pi = [5, 6, 3, 4, 1, 2]$, the canonical spin of π is $\pi'_1 = [+5, +6, +3, +4, +1, +2]$. Note that $G(\pi''_1)$ consists of only one unoriented component. According to their algorithm, suppose we select a canonical strip [+5,+6] to be turned into anti-strip and get a super spin $\pi'_2 = [-5, -6, +3, +4, +1, +2]$. From (1), $d_{\rm r}(\pi'_1) = 4 - 2 + 1 = 3$, $d_{\rm r}(\pi'_2) = 5 - 1 + 0 = 4 > d_{\rm r}(\pi'_1)$, thus π'_2 cannot be an optimal r-spin. Moreover, if we select the canonical 2-strip $[+3, +4]$ or $[+1, +2]$ to be turned into an anti-strip, the resulted super spin cannot be optimal either. The breakpoint graphs of $G(\pi''_1)$ and $G(\pi''_2)$ are shown in Fig.2.

As the example shows, the reason that HP algorithm fails is that they assumed that the edges incident to a canonical 2-strip belonged to the same cycle. Thus, when a canonical 2-strip s belongs to two cycles of an unoriented component, turning s into an anti-strip will increase the reversal distance. To correct this, let $s = [g_i, g_{i+1}]$ $(g_i < g_{i+1})$ be a 2-strip of an unsigned permutation π, its canonical form in π's canonical spin π' is $s = [+g_i, +g_{i+1}]$, which is transformed to $[2g_i - 1, 2g_i, 2g_{i+1} - 1, 2g_{i+1}]$ in π''s extended permutation π''. If $2g_i - 1$ and $2g_{i+1}$ belong to the same cycle of $G(\pi'')$, call s a *satisfied 2-strip*. Using this definition, we correct the definition of super spin as follows:

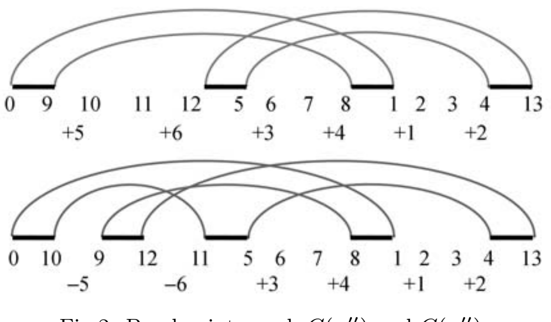

Fig.2. Breakpoint graph $G(\pi''_1)$ and $G(\pi''_2)$.

Definition 1 (Super-r-Spin). *Suppose π' is a canonical spin of π, for every unoriented component K of $G(\pi'')$, if K contains satisfied 2-strips, arbitrarily select only one of them and transform it into an anti-strip; otherwise, all the strips of K remain canonical. We call such a spin a super-r-spin.*

Theorem 1. *For a singleton-free unsigned permutation π, every super-r-spin π' of π is an optimal r-spin of π.*

Note that the only difference between a super spin and a super-r-spin is that only the satisfied 2-strips are turned into anti-strips. Detailed proof for Theorem 1 is given in the Appendix, it uses some lemmas proved in the following section, and the technique for proof is similar to Theorem 2. For an unsigned permutation π with k singletons, there are 2^k possible canonical spins. Constructing the super-r-spin of a canonical spin takes $O(n)$ time, and computing the reversal distance of a super-r-spin also takes $O(n)$ time[24]. Thus it takes $O(2^k n)$ time to get the super-r-spin π^* of π with the minimum reversal distance $d_{\rm r}(\pi^*)$. Computing the optimal reversal scenario of π^* and taking it as the optimal reversal scenario for π will complete the algorithm. Since it takes $O(n \log n)$ time to get an optimal reversal scenario of a signed permutation[10], the time complexity of the algorithm is $O(2^k n + n \log n)$. If the number of singletons, k, is $O(\log n)$, we can complete the computation in polynomial time.

4 Approximation Algorithm for Sorting Unsigned Permutations by Weighted Operations

In this section, we consider sorting an unsigned permutation π by weighted reversals, transpositions and transreversals. The weighted distance is defined as $d_{\rm r+t}(\pi) = \min_{\phi \in \Phi} \{{\rm rev}(\phi) + 2{\rm trp}(\phi)\}$. Eriksen studied the signed version of this problem, and gave a $(1 + \varepsilon)$-approximation algorithm[22].

For a signed permutation π', a component K of $G(\pi'')$ is either an oriented component or an unoriented component, and an unoriented component is either an SUC or a non-SUC. If K is an oriented component, we can eliminate it using $b(K) - c(K)$ reversals. If K is a non-SUC, we can eliminate it using $((b(K) - c(K))/2$ transpositions, which contributes $b(K) - c(K)$ to the weighted distance, while it costs more than $b(K) - c(K)$ to eliminate a non-SUC by reversals. If K is an SUC, we need more than $((b(K) - c(K))/2$ transpositions to eliminate it, while just $b(K) - c(K) + 1$ reversals will do the job. However, it is difficult to distinguish between an SUC and a non-SUC, only an approximation algorithm is available. The reason we do not mention transreversals is that in both signed case and unsigned case, each transreversal can be replaced by two reversals without affecting the objective function[22]. Therefore, we only consider reversals and transpositions.

4.1 Spinning Long Strips and 2-Strips

Lemma 3. *For any unsigned permutation* π, $d_{\mathrm{r+t}}(\pi) = \min_{\pi' \in \Pi} d_{\mathrm{r+t}}(\pi')$.

Proof. For every spin π' of π, any weighted scenario for π' can be used to sort π simply by ignoring the signs, so $d_{\mathrm{r+t}}(\pi') \geqslant d_{\mathrm{r+t}}(\pi)$. Let $\phi = \{\rho_1, \rho_2, \ldots, \rho_x\}$ represent an optimal weighted scenario for π, where $\mathrm{rev}(\phi) + \mathrm{trp}(\phi) = x$ and $\mathrm{rev}(\phi) + 2\mathrm{trp}(\phi) = d_{\mathrm{r+t}}(\pi)$. Consider the signed permutation $\pi' = \iota' \cdot \rho_x \cdot \cdots \cdot \rho_2 \cdot \rho_1$, where $\iota' = [+1, +2, \ldots, +n]$ is the signed identity permutation. Since $\pi' \in \Pi$ and $\pi' \cdot \rho_1 \cdot \rho_2 \cdot \cdots \cdot \rho_x = \iota'$, $d_{\mathrm{r+t}}(\pi') \leqslant d_{\mathrm{r+t}}(\pi)$. Therefore, $d_{\mathrm{r+t}}(\pi) = \min_{\pi' \in \Pi} d_{\mathrm{r+t}}(\pi')$. □

Lemma 4. *Let* π'_1 *be a spin of* π, *if* π'_1 *has a canonical* 2*-strip* s, *then flip*(s) *will increase* $b(\pi'_1)$ *by* 1; *if* π'_1 *has a canonical* 3*-strip* s, *then flip*(s) *will increase* $b(\pi'_1)$ *by* 2.

Proof. Without loss of generality, let $s = [+1, +2]$ be a canonical 2-strip of π'_1 (see Fig.3(a)). After *flip*(s), we get π'_2 (see Fig.3(b)). Obviously *flip*(s) brings in a new pair of black edge and gray edge: (m, n). Similarly, if s is a canonical 3-strip, as shown in Figs. 3(c) and 3(d), *flip*(s) brings in two new pairs of black edge and gray edge: (m_1, n_1) and (m_2, n_2). □

Lemma 5. *Let* π'_1 *be a spin of* π *with canonical* 2*-strip* s, *the two black edges incident to* s *be* b_1 *and* b_2, *and let* $\pi'_2 = \pi'_1 \cdot$ *flip*(s) *be a twin of* π'_1 *with anti-strip* s. *If* b_1 *and* b_2 *belong to two cycles of* $G(\pi''_1)$, *then* $c(\pi'_2) = c(\pi'_1) - 1$; *if* s *is a satisfied* 2*-strip, then* $c(\pi'_2) = c(\pi'_1) + 1$ *if* b_1 *has the same direction with* b_2, *or* $c(\pi'_2) = c(\pi'_1)$ *if* b_1 *has the opposite direction with* b_2.

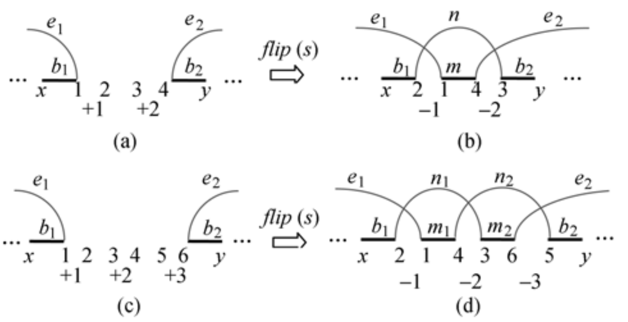

Fig.3. *flip*(s) on a canonical 2-strip and a canonical 3-strip.

Proof. Similar to Lemma 4, let $s = [+1, +2]$, $b_1 = (x, 1)$ and $b_2 = (4, y)$ ($e_1 = (x', 1)$ and $e_2 = (4, y')$) denote the two black edges (gray edges) incident to 1 and 4 in $G(\pi''_1)$ respectively (see Fig.3(a)).

If b_1 and b_2 belong to two cycles of $G(\pi''_1)$, i.e., vertex x' joins x through more than one edges and y' joins y through more than one edges. After *flip*(s), although both b_1 and b_2 have one end point altered, x' still joins x through more than one edges and y' joins y through more than one edges. On the other hand, e_1 and e_2 are joined by m, b_1 and b_2 are joined by n; so the two cycles are merged into one cycle. Thus we have $c(\pi'_2) = c(\pi'_1) - 1$.

If s is a satisfied 2-strip, i.e., b_1 and b_2 belong to the same cycle of $G(\pi''_1)$, the change of $c(\pi'_1)$ after *flip*(s) depends on the relative direction of b_1 and b_2. For convenience, we add an arrow to each edge, the head and tail of the arrow are called the *head* and *tail* of the edge.

Case 1. b_1 and b_2 have the same direction, without loss of generality, let they direct right. The only possible case is that e_1's head joins e_2's tail through one or more edges and b_2' head joins b_1' tail through one or more edges (see Fig.4(a)). After *filp*(s), n joins b_1's head with b_2's tail, m joins e_2's head with e_1's tail (see Fig.4(b)). For convenience, we stretch the cycles into a circle while retaining the edges' joining relationship (see Fig.4(c)). Since all the cycles in $G(\pi''_2)$ must be alternating cycles, the configuration of the stretched circles in $G(\pi''_2)$ must be the one shown in Fig.4(d). It is clear that $c(\pi'_2) = c(\pi'_1)$+1.

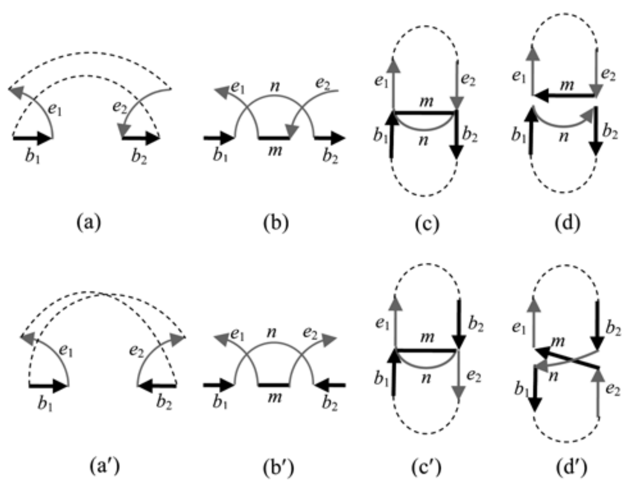

Fig.4. Stretched circle for 2-strip case.

Case 2. b_1 and b_2 have the opposite direction. The only possible case is that e_1's head joins b_2's tail directly or through more than one edges and e_2's head joins b_1's tail directly or through more than one edges (see Fig.4(a$'$)). Similar to Case 1, Figs. 4(b$'$), 4(c$'$), 4(d$'$) show that $c(\pi_2') = c(\pi_1')$. □

Lemma 6. *Let π_1' be a spin of π with canonical 3-strip $s = [g_i, g_{i+1}, g_{i+2}]$, and let $\pi_2' = \pi_1' \cdot flip(s)$ be a twin of π_1' with anti-canonical s. Then $d_{\mathrm{r+t}}(\pi_1') \leqslant d_{\mathrm{r+t}}(\pi_2')$.*

Proof. Without loss of generality, let $s = [g_i, g_{i+1}, g_{i+2}] = [1, 2, 3]$. Spin π_1' with canonical s is shown in Fig.3(c). The twin π_2' with anti-canonical $s = [-1, -2, -3]$ is shown in Fig.3(d). From Lemma 4, $b(\pi_2') = b(\pi_1') + 2$. Let $b_1 = (x, 1)$ and $b_2 = (6, y)$ ($e_1 = (x', 1)$ and $e_2 = (6, y')$) denote the two black edges (gray edges) incident to 1 and 6 in $G(\pi_1'')$ respectively. The transformation from π_1' to π_2' brings in two black edges m_1, m_2 and two gray edges n_1, n_2. Note that n_1 joins m_2, m_1 join n_2. The change of $c(\pi_1')$ after $flip(s)$ is very similar to Lemma 5, the only difference is that the newly created edges are two pairs.

If b_1 and b_2 belong to two cycles of $G(\pi_1'')$, i.e., vertex x' joins x through more than one edges and y' joins y through more than one edges. After $flip(s)$, x' still joins x through more than one edges and y' joins y through more than one edges. On the other hand, e_1 and b_2 are joined by m_1, n_2; b_1 and e_2 are joined by n_1, m_2; so the two cycles are merged into one cycle. Thus we have $c(\pi_2') = c(\pi_1') - 1$. To get $d_{\mathrm{r+t}}(\pi_2') < d_{\mathrm{r+t}}(\pi_1')$, from (2), $(h_{\mathrm{t}} + f_{\mathrm{t}})\pi_2' \leqslant (h_{\mathrm{t}} + f_{\mathrm{t}})\pi_1' - 4$[①] must hold. However, the transformation from π_1' to π_2' affects at most two SUC's, which makes it impossible.

If b_1 and b_2 belong to the same cycle of $G(\pi_1'')$, the change of $c(\pi_1')$ after $flip(s)$ still depends on the relative direction of b_1 and b_2.

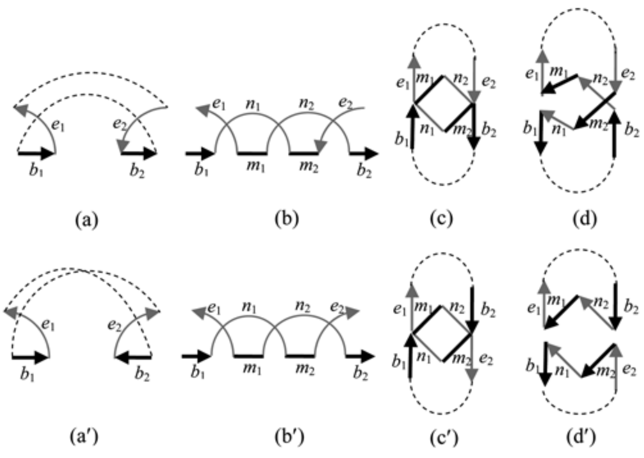

Fig.5. Stretched circle for 3-strip case.

Case 1. b_1 and b_2 have the same direction, from Figs. 5(a)~5(d), $c(\pi_2') = c(\pi_1')$. To get $d_{\mathrm{r+t}}(\pi_2') < d_{\mathrm{r+t}}(\pi_1')$, from (2), $(h_{\mathrm{t}} + f_{\mathrm{t}})\pi_2' \leqslant (h_{\mathrm{t}} + f_{\mathrm{t}})\pi_1' - 3$ must hold. However, since b_1 and b_2 belong to the same cycle of $G(\pi_1'')$, the transformation from π_1' to π_2' affects at most one SUC, which makes it impossible.

Case 2. b_1 and b_2 have the opposite direction, from Figs. 5(a$'$)~5(d$'$), $c(\pi_2') = c(\pi_1') + 1$. To get $d_{\mathrm{r+t}}(\pi_2') < d_{\mathrm{r+t}}(\pi_1')$, from (2), $(h_{\mathrm{t}} + f_{\mathrm{t}})\pi_2' \leqslant (h_{\mathrm{t}} + f_{\mathrm{t}})\pi_1' - 2$ must hold. However, since b_1 and b_2 have the opposite direction, they must belong to an oriented component of $G(\pi_1'')$, also makes it impossible. □

Lemma 7. *Let π_1' be a spin of π with canonical 3-strip $s = [g_i, g_{i+1}, g_{i+2}]$, and let π_2' be a twin of π_1' with respect to s. Then $d_{\mathrm{r+t}}(\pi_1') \leqslant d_{\mathrm{r+t}}(\pi_2')$.*

Proof. Without loss of generality, let $s = [g_i, g_{i+1}, g_{i+2}] = [1, 2, 3]$. Spin π_1' with canonical s is shown in Fig.6(a), π_1''s twins with respect to s are shown in Figs. 6(b)~6(f). We will discuss them respectively.

The twin π_b' in Fig.6(b) has $s = [+1, -2, +3]$, $b(\pi_b') = b(\pi_1') + 2$, $c(\pi_b') = c(\pi_1') + 1$. Since the transformation from π_1' to π_b' only adds one oriented cycle (component), the number of strong hurdles is not changed, hence $(h_{\mathrm{t}} + f_{\mathrm{t}})(\pi_b') = (h_{\mathrm{t}} + f_{\mathrm{t}})(\pi_1')$. From (2), $d_{\mathrm{r+t}}(\pi_b') > d_{\mathrm{r+t}}(\pi_1')$.

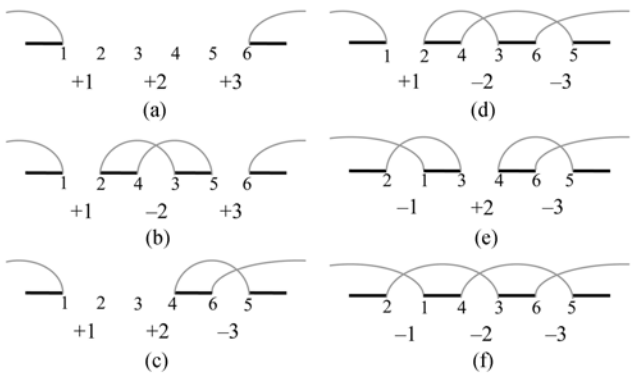

Fig.6. All the twins of 3-strip [1, 2, 3].

The twin π_c' in Fig.6(c) has $s = [+1, +2, -3]$, $b(\pi_c') = b(\pi_1') + 1$, $c(\pi_c') = c(\pi_1')$. Since the transformation from π' to π_c' affects at most one SUC, $h_{\mathrm{t}}(\pi_c') \geqslant h_{\mathrm{t}}(\pi_1') - 1$. If $h_{\mathrm{t}}(\pi_c') = h_{\mathrm{t}}(\pi_1') - 1$, this SUC must be a simple hurdle, which implies $f_{\mathrm{t}}(\pi_c') = f_{\mathrm{t}}(\pi_1')$. If $f_{\mathrm{t}}(\pi_c') = f_{\mathrm{t}}(\pi_1') - 1$, this SUC must be a super-strong-hurdle, which implies $h_{\mathrm{t}}(\pi_c') = h_{\mathrm{t}}(\pi_1')$. So $(h_{\mathrm{t}} + f_{\mathrm{t}})(\pi_c') \geqslant (h_{\mathrm{t}} + f_{\mathrm{t}})(\pi_1') - 1$, hence $d_{\mathrm{r+t}}(\pi_c') \geqslant d_{\mathrm{r+t}}(\pi_1')$.

The twin π_d' in Fig.6(d) has $s = [+1, -2, -3]$, $b(\pi_d') = b(\pi_1') + 2$, $c(\pi_d') = c(\pi_1')$. To get $d_{\mathrm{r+t}}(\pi_d') < d_{\mathrm{r+t}}(\pi_1')$, $(h_{\mathrm{t}} + f_{\mathrm{t}})(\pi_d') \leqslant (h_{\mathrm{t}} + f_{\mathrm{t}})(\pi_1') - 3$ must hold.

①$(h_{\mathrm{t}} + f_{\mathrm{t}})(\pi_1')$ is a simplified notation of $h_{\mathrm{t}}(\pi_1') + f_{\mathrm{t}}(\pi_1')$.

However, the transformation from π'_1 to π'_d affects at most one SUC, which makes it impossible.

The twin π'_e in Fig.6(e) has $s = [-1, +2, -3]$, $b(\pi'_e) = b(\pi') + 2$, $c(\pi'_e) = c(\pi'_1)$. Since the transformation from π'_1 to π'_e affects at most two SUC's, to get $d_{\mathrm{r+t}}(\pi'_e) < d_{\mathrm{r+t}}(\pi'_1)$, $h_{\mathrm{t}}(\pi'_e) = h_{\mathrm{t}}(\pi'_1) - 2$ and $f_{\mathrm{t}}(\pi'_e) = f_{\mathrm{t}}(\pi'_1) - 1$ must hold. However, in a strong fortress, two super-strong-hurdles cannot be adjacent by a strip. That is when $h_{\mathrm{t}}(\pi'_e) = h_{\mathrm{t}}(\pi'_1) - 2$, the two affected SUC's must both be simple-strong-hurdles, thus $f_{\mathrm{t}}(\pi'_e) = f_{\mathrm{t}}(\pi'_1)$. Therefore, $d_{\mathrm{r+t}}(\pi'_e) \geqslant d_{\mathrm{r+t}}(\pi'_1)$.

The twin π'_f in Fig.6(f) has $s = [-1, -2, -3]$, from Lemma 6, $d_{\mathrm{r+t}}(\pi'_f) \geqslant d_{\mathrm{r+t}}(\pi'_1)$. □

Lemma 8. *For any unsigned permutation π, there exists an optimal rt-spin π' of π such that all the long strips of π are canonical in π'.*

Proof. Immediately from Lemma 7. □

Lemma 9. *For any unsigned permutation π, there exists an optimal rt-spin π' of π such that every 2-strip of π is either canonical or anti-canonical in π'.*

Proof. For a spin π' of π, define $index(\pi')$ as the number of 2-strips in π' that are neither canonical nor anti-canonical in π'. Let π'_1 be a spin with the minimum *index* value among all the optimal rt-spins. The following will prove that $index(\pi'_1) = 0$.

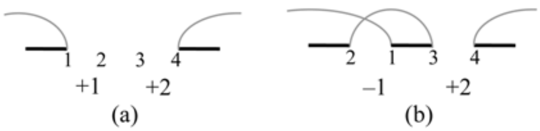

Fig.7. A canonical 2-strip and a twin of it.

Suppose for contradiction that $index(\pi'_1) > 0$, without loss of generality, we can assume that $s = [-1, +2]$ in π'_1, s is neither canonical nor anti-canonical (see Fig.7(b)). Let π'_2 be a twin of π'_1 with canonical 2-strip $s = [+1, +2]$ (see Fig.7(a)). Note that $b(\pi'_1) = b(\pi'_2) + 1$, $c(\pi'_1) = c(\pi'_2)$. Since the transformation from π'_2 to π'_1 affects at most one strong hurdle of $G(\pi''_2)$, if $h_{\mathrm{t}}(\pi'_1) = h_{\mathrm{t}}(\pi'_2) - 1$, this strong hurdle must be a simple-strong-hurdle, which makes $f_{\mathrm{t}}(\pi'_1) = f_{\mathrm{t}}(\pi'_2) - 1$ impossible; if $f_{\mathrm{t}}(\pi'_1) = f_{\mathrm{t}}(\pi'_2) - 1$, this strong hurdle must be a super-strong-hurdle, which leads to $h_{\mathrm{t}}(\pi'_1) = h_{\mathrm{t}}(\pi'_2)$. Therefore, $(h_{\mathrm{t}} + f_{\mathrm{t}})(\pi'_1) \geqslant (h_{\mathrm{t}} + f_{\mathrm{t}})(\pi'_2) - 1$. According to (2), $d_{\mathrm{r+t}}(\pi'_1) \geqslant d_{\mathrm{r+t}}(\pi'_2)$. It implies that π'_2 is an optimal rt-spin with $index(\pi'_2) < index(\pi'_1)$, a contradiction. □

From Lemma 8 and Lemma 9, there exists an optimal rt-spin such that all the long strips are canonical and all the 2-strips are either canonical or anti-canonical. To search for the optimal rt-spin of a singleton-free unsigned permutation, first construct its canonical spin δ, then the remaining task is to decide which canonical 2-strips of δ have to be turned into anti-strips.

Lemma 10. *If s is a canonical 2-strip of an unoriented component K, flip(s) will turn K into an oriented component K'.*

Proof. Let $s = [2g_i - 1, 2g_i, 2g_{i+1} - 1, 2g_{i+1}]$ in the extended form. Since K is an unoriented component, the two gray edges n_1 and n_2 incident to s are both unoriented gray edges. According to the two forms of an unoriented gray edge, there are four configurations of s with its two incident gray edges (Figs. 8(a)∼8(d)). The corresponding configuration after *flip*(s) is shown in Figs. 8(a′)∼8(d′) respectively. Note that after *flip*(s), the two end points $2g_i - 1$ and $2g_{i+1}$ of s move close to each other and are joined by the newly created black edge. It is clear that in each case, both n_1 and n_2 become oriented gray edges, and they are the only oriented gray edges of K'. On the other hand, in K', the newly created gray edge crosses with both n_1 and n_2, thus K' remains one component. □

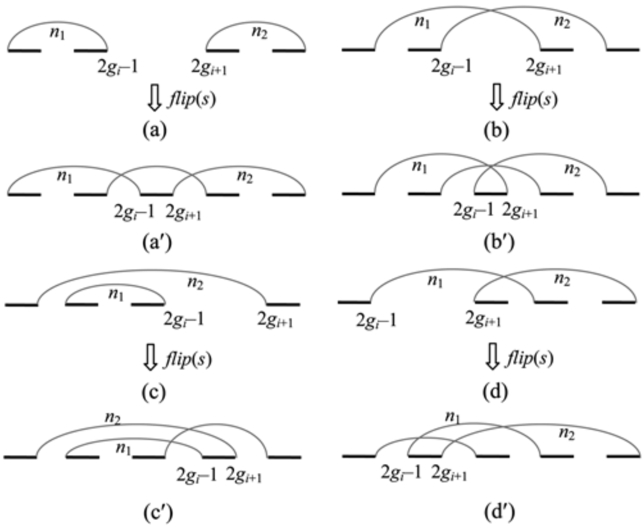

Fig.8. Change of the gray edges inside s on *flip*(s).

Lemma 11. *For any unsigned permutation π, there exists an optimal rt-spin π' of π such that* (*I*) *an unoriented component in π' does not contain any anti-strip, and* (*II*) *an oriented component in π' contains at most one anti-strip.*

Proof. Let π'_1 be an optimal rt-spin with the minimum number of anti-strips among all the optimal rt-spins of π. Suppose for contradiction that π'_1 has an unoriented component K and K contains an anti-strip s. Then let π'_2 be a twin of π'_1 with canonical 2-strip s. From Lemma 4, $b(\pi'_1) = b(\pi'_2) + 1$. From Lemma 5, $c(\pi'_1) = c(\pi'_2) + 1$ or $c(\pi'_1) = c(\pi'_2)$ or $c(\pi'_1) = c(\pi'_2) - 1$. Since K is an unoriented component of π'_1 and an SUC must be an unoriented component[22], from Lemma 10, K becomes an oriented component in π_2', thus $(h_t + f_t)(\pi_1') \geqslant (h_t + f_t)(\pi_2')$. From (2), $d_{r+t}(\pi_1') \geqslant d_{r+t}(\pi_2')$, it implies that π_2' is an optimal rt-spin with fewer anti-strips than π_1', a contradiction.

If π_1' has an oriented component K which contains at least two anti-strips s_1 and s_2, consider the twin π_3' of π_1' with canonical s_1. From Lemma 4, $b(\pi_1') = b(\pi_3')+1$. From Lemma 5, $c(\pi_1') = c(\pi_3') - 1$ or $c(\pi_1') = c(\pi_2')$ or $c(\pi_1') = c(\pi_3')+1$. So we have: $(b-c)(\pi_1') \geqslant (b-c)(\pi_3')$. Since π_1' is an optimal rt-spin, $d_{r+t}(\pi_1') \leqslant d_{r+t}(\pi_3')$, i.e., $(b-c)(\pi_1') + (h_t + f_t)(\pi_1') \leqslant (b-c)(\pi_3') + (h_t + f_t)(\pi_3')$. Therefore, $(h_t + f_t)(\pi_1') \leqslant (h_t + f_t)(\pi_3')$. If $(h_t + f_t)(\pi_1') = (h_t + f_t)(\pi_3')$, then $d_{r+t}(\pi_1') \geqslant d_{r+t}(\pi_3')$, which implies that π_3' is an optimal rt-spin with fewer anti-strips than π_1', a contradiction. If $(h_t + f_t)(\pi_1') < (h_t + f_t)(\pi_3')$, it implies that K becomes an SUC in π_3'. Since an SUC must be an unoriented component, from Lemma 10, the two oriented gray edges inside s_1 are the only oriented gray edges in K. Similarly, transforming s_2 from an anti-strip into a canonical strip can also turn K into an SUC, which implies that the two oriented gray edges inside s_2 are the only oriented gray edges in K. It means that the oriented gray edges inside s_1 and s_2 are the same, a contradiction. □

4.2 Algorithm for Singleton-Free Permutations

Lemma 12. *Suppose π_1' is an optimal rt-spin of π containing the minimum number of anti-strips among all the optimal rt-spins satisfying the conditions* (I) *and* (II) *of Lemma* 11. *Let s be an anti-strip of π_1', then π_1''s twin π_2' with canonical s must have s in one cycle of an unoriented component of $G(\pi_2'')$ and $(h_t + f_t)(\pi_1') = (h_t + f_t)(\pi_2') - 1$ holds.*

Proof. From the hypothesis, $\pi_1' = \pi_2' \cdot flip(s)$ and π_2' also satisfies the conditions (I) and (II) of Lemma 11. From Lemma 4, $b(\pi_1') = b(\pi_2') + 1$. Note that a 2-strip s may belong to two cycles of $G(\pi_2'')$, or belong to the same cycle of $G(\pi_2'')$. The following will discuss them respectively.

Case 1. If s belongs to two cycles of $G(\pi_2'')$. From Lemma 5, $c(\pi_1') = c(\pi_2') - 1$. Since *flip*$(s)$ can affect at most two strong hurdles of $G(\pi_2'')$, and when *flip*(s) removes two strong hurdles, they must both be simple-strong-hurdles. From the definition of strong fortress, we know $(h_t + f_t)(\pi_1') \geqslant (h_t + f_t)(\pi_2') - 2$. From (2), $d_{r+t}(\pi_2') \leqslant d_{r+t}(\pi_1')$. It implies that π_2' is an optimal rt-spin of π satisfying the conditions (I) and (II) of Lemma 11, and that π_2' contains fewer anti-strips than π_1', a contradiction.

Case 2. If s belongs to the same cycle of $G(\pi_2'')$. From Lemma 5, the change of cycle number due to *flip*(s) on π_2' depends on the relative direction of black edges b_1 and b_2 which are incident to s.

Case 2.1. If b_1 and b_2 have the opposite direction, then $c(\pi_1') = c(\pi_2')$. Since s is in an oriented component of $G(\pi_2'')$, $(h_t + f_t)(\pi_1') \geqslant (h_t + f_t)(\pi_2')$. From (2), $d_{r+t}(\pi_1') > d_{r+t}(\pi_2')$, a contradiction.

Case 2.2. If b_1 and b_2 have the same direction, then $c(\pi_1') = c(\pi_2') + 1$. If b_1 and b_2 are in an oriented component of $G(\pi_2'')$, then $(h_t + f_t)(\pi_1') \geqslant (h_t + f_t)(\pi_2')$. From (2), $d_{r+t}(\pi_1') \geqslant d_{r+t}(\pi_2')$. It implies that π_2' is an optimal rt-spin of π satisfying the conditions of Lemma 11 with fewer anti-strips than π_1', a contradiction. So b_1 and b_2 must belong to one cycle of an unoriented component K of $G(\pi_2'')$, i.e., s is a satisfied 2-strip of K. In this case, *flip*(s) will never increase $(h_t + f_t)(\pi_2')$, and can at most decrease $(h_t + f_t)(\pi_2')$ by 1. If $(h_t+f_t)(\pi_1') = (h_t+f_t)(\pi_2')$, then $d_{r+t}(\pi_1') = d_{r+t}(\pi_2')$, a contradiction. Therefore $(h_t+f_t)(\pi_1') = (h_t+f_t)(\pi_2')-1$ and $d_{r+t}(\pi_1') = d_{r+t}(\pi_2') - 1$.

From the above analysis, s must be a satisfied 2-strip of an unoriented component of $G(\pi_2'')$ and $(h_t + f_t)(\pi_1') = (h_t + f_t)(\pi_2') - 1$ holds. □

Theorem 2. *For a singleton-free unsigned permutation π, every super-r-spin π' of π is an optimal rt-spin of π.*

Proof. Let π_1' be an optimal rt-spin of π containing the minimum number of anti-strips among all the optimal rt-spins satisfying the conditions (I) and (II) of Lemma 11. Suppose that π_1' contains u anti-strips $s_1', s_2', \ldots, s_u'$. Applying *flip*(s) to all of them will turn them into canonical 2-strips $s_1, s_2, \ldots, s_u$, thus turning π_1' into a canonical spin, denoted by δ. From Lemmas 4, 5, 12, $d_{r+t}(\delta) = d_{r+t}(\pi_1') + u$, each s_i $(1 \leqslant i \leqslant u)$ is a satisfied 2-strip of δ, at most one in every component of δ. Let $K_1, K_2, \ldots, K_u$ be the u unoriented components of δ containing $s_1, s_2, \ldots, s_u$. Let π_3' be a spin of π obtained from δ by arbitrarily choosing a satisfied 2-strip of K_i (for $1 \leqslant i \leqslant u$), and transforming it into an anti-strip. From Lemma 4, $b(\pi_3') = b(\delta)+u = b(\pi_1')$, from Lemma 5, $c(\pi_3') = c(\delta) + u = c(\pi_1')$. Moreover, $(h_t + f_t)(\pi_1') = (h_t + f_t)(\pi_3')$. This is because for an unoriented component K_i, if it contains more than one satisfied 2-strips, turning any one of them into an anti-strip has the same effect of turning K_i into an oriented component, thus has the same effect on (h_t+f_t). Therefore, $d_{r+t}(\pi_3') = d_{r+t}(\pi_1')$.

If π_3' does not have additional unoriented components containing satisfied 2-strips, π_3' is a super-r-spin of π. Otherwise, suppose π_3' has unoriented components $K_{u+1}, \ldots, K_x$, each of them contains satisfied 2-strips. Then for every K_j $(u + 1 \leqslant j \leqslant x)$, arbitrarily select a satisfied 2-strip $s_j \in K_j$, *flip*(s_j) will transform K_j into an oriented component without increasing the value of $(b - c)(\pi_3')$ and $(h_t + f_t)(\pi_3')$, which means $d_{r+t}(\pi_3' \cdot flip(s_j)) \leqslant d_{r+t}(\pi_3')$. Let $\pi_4' =$

$\pi_3' \cdot \mathit{flip}(s_{u+1}) \cdot \ldots \cdot \mathit{flip}(s_x)$. Then π_4' is a super-r-spin of π with $d_{\mathrm{r+t}}(\pi_4') \leqslant d_{\mathrm{r+t}}(\pi_3') = d_{\mathrm{r+t}}(\pi_1')$, which implies that π_4' is an optimal rt-spin. This completes the proof of the theorem. □

According to Theorem 2, if π is a singleton-free permutation, by running Eriksen's algorithm[22] on a super-r-spin of π, we immediately get a $(1+\varepsilon)$-approximation solution for sorting π by weighted reversals and transpositions/transreversals.

4.3 Algorithm for Permutations with $O(\log n)$ Singletons

The following algorithm *Weighted_Sorting* shows how to sort an unsigned permutation with $O(\log n)$ singletons within polynomial time.

Algorithm. *Weighted_ Sorting*(π)

```
A_{r+t}(π) ← n;
for every canonical spin π′ of π {
    construct G(π″);
    for every unoriented component K_i of G(π″) {
      if K_i has satisfied 2-strips {
        arbitrarily select one of them, denoted as s;
        π′ ← π′ · flip(s);
      } //endif
    } //endfor
    compute A_{r+t}(π′) by running the (1 + ε)-approximation algorithm on π′[22];
    if A_{r+t}(π′) < A_{r+t}(π) {
      A_{r+t}(π) ← A_{r+t}(π′);
      π* ← π′;
    } //endif
} //endfor
sort π*;
take the weighted scenario for π* as the weighted scenario for π with approximate distance A_{r+t}(π*).
```

Theorem 3. *For any unsigned permutation π with $O(\log n)$ singletons, algorithm Weighted_Sorting(π) is a $(1+\varepsilon)$-approximation algorithm for sorting π by weighted reversals and transpositions/transreversals.*

Proof. Algorithm *Weighted_Sorting*(π) first enumerates all the singleton's signs and gets the set of all the canonical spins of π. Let $P = \{\pi_1', \pi_2', \ldots, \pi_m'\}$ be the set of canonical spins by the enumeration. Let δ_i be the super-r-spin of π_i', $A_{\mathrm{r+t}}(\delta_i)$ be the distance of δ_i computed by running Eriksen's algorithm on δ_i. From Lemma 3 and Theorem 2, the optimal weighted distance of π is $d_{\mathrm{r+t}}(\pi) = \min\{d_{\mathrm{r+t}}(\delta_1), d_{\mathrm{r+t}}(\delta_2), \ldots, d_{\mathrm{r+t}}(\delta_m)\}$. Let $d_{\mathrm{r+t}}(\delta_y) = d_{\mathrm{r+t}}(\pi)$ $(1 \leqslant y \leqslant m)$. The approximate solution on π got from *Weighted_Sorting*(π) is $A_{\mathrm{r+t}}(\pi) = \min\{A_{\mathrm{r+t}}(\delta_1), A_{\mathrm{r+t}}(\delta_2), \ldots, A_{\mathrm{r+t}}(\delta_m)\} = A_{\mathrm{r+t}}(\delta_x)$ $(1 \leqslant x \leqslant m)$. Since $A_{\mathrm{r+t}}(\delta_i)/d_{\mathrm{r+t}}(\delta_i) \leqslant 1+\varepsilon$ for $1 \leqslant i \leqslant m$, we have:

$$\frac{A_{\mathrm{r+t}}(\pi)}{d_{\mathrm{r+t}}(\pi)} = \frac{\min\{A_{\mathrm{r+t}}(\delta_1), A_{\mathrm{r+t}}(\delta_2), \ldots, A_{\mathrm{r+t}}(\delta_m)\}}{\min\{d_{\mathrm{r+t}}(\delta_1), d_{\mathrm{r+t}}(\delta_2), \ldots, d_{\mathrm{r+t}}(\delta_m)\}} = \frac{A_{\mathrm{r+t}}(\delta_x)}{d_{\mathrm{r+t}}(\delta_y)} \leqslant \frac{A_{\mathrm{r+t}}(\delta_y)}{d_{\mathrm{r+t}}(\delta_y)} \leqslant 1+\varepsilon. \quad (3)$$

When there are $O(\log n)$ singletons in π, the size of set P is $O(n^k)$, where k is a fixed constant. Therefore, algorithm *Weighted_Sorting*(π) is a polynomial time algorithm on such permutations and guarantee the $(1+\varepsilon)$-approximation ratio. □

5 Conclusion

This paper rectifies the polynomial time algorithm for the reversal sorting problem on unsigned permutations with limited singletons and gives a $(1+\varepsilon)$-approximation algorithm for the weighted sorting problem on the same permutation set. We discover that for a singleton-free unsigned permutation, its super-r-spin is an optimal rt-spin as well as an optimal r-spin. Although we cannot compute the exact weighted distance of a super-r-spin in polynomial time due to the existence of a class of structures called SUC, we know for sure that such a spin is an optimal rt-spin. As future work, it is meaningful to design a polynomial time approximation algorithm for sorting all the unsigned permutations by reversals and transpositions under weight proportion 1:2.

References

[1] Jiang T. Some algorithmic challenges in genome-wide ortholog assignment. *Journal of Computer Science and Technology*, 2010, 25(1): 42-52.

[2] Caprara A. Sorting by reversals is difficult. In *Proc. the 1st International Conference on Research in Computational Molecular Biology*, Santa Fe, USA, Jan. 20-23, 1997, pp.75-83.

[3] Bafna V, Pevzner P. Genome rearrangements and sorting by reversals. *SIAM J. Computing*, 1996, 25(2): 272-289.

[4] Christie D. Genome rearrangement problems [Ph.D. Dissertation]. University of Glasgow, 1998.

[5] Berman P, Hannenhalli S, Karpinski M. 1.375-approximation algorithm for sorting by reversals. In *Proc. the 10th Annual European Symposium on Algorithms*, Rome, Italy, Sept. 16-21, 2002, pp.200-210.

[6] Hannenhalli S, Pevzner P. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). *Journal of ACM*, 1999, 46(1): 1-27.

[7] Kaplan H, Shamir R, Tarjan R. A faster and simpler algorithm for sorting signed permutations by reversals. *SIAM J. Computing*, 1999, 29(3): 880-892.

[8] Tannier E, Sagot M-F. Sorting by reversals in subquadratic time. In *Proc. the 15th Annual Symposium on Combinatorial Pattern Matching*, İstanbul, Turkey, Jul. 5-7, 2004, pp.1-13.

[9] Bergeron A. A very elementary presentation of the Hannenhalli-Pevzner theory. *Discrete Applied Mathematics*,

2005, 146(2): 134-145.

[10] Swenson K, Rajan V, Lin Y, Moret B. Sorting signed permutations by inversions in $O(n \log n)$ time. In *Proc. the 13th International Conference on Research in Computational Molecular Biology*, Tucson, USA, May 18-21, 2009, pp.386-399.

[11] Hannenhalli S, Pevzner P. To cut ... or not to cut (applications of comparative physical maps in molecular evolution). In *Proc. the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, Atlanta, USA, Jan. 28-30, 1996, pp.304-313.

[12] Bafna V, Pevzner P. Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 1998, 11(2): 224-240.

[13] Hartman T, Shamir R. A simpler and faster 1.5-approximation algorithm for sorting by transpositions. *Information and Computation*, 2006, 204(2): 275-290.

[14] Feng J, Zhu D. Faster algorithm for sorting by transpositions and sorting by block-interchanges. *ACM Transactions on Algorithms*, 2007, 3(3): Article No.25.

[15] Elias I, Hartman T. A 1.375-approximation algorithm for sorting by transpositions. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2006, 3(4): 369-379.

[16] Walter M, Dias Z, Meidanis J. Reversal and transposition distance of linear chromosomes. In *Proc. String Processing and Information Retrieval*, Santa Cruz de La Sierra, Bolivia, Sept. 9-11, 1998, pp.96-102.

[17] Gu Q, Peng S, Sudborough H. A 2-approximation algorithm for genome rearrangements by reversals and transpositions. *Theoretical Computer Science*, 1999, 210(2): 327-339.

[18] Lin G, Xue G. Signed genome rearrangement by reversals and transpositions: Models and approximations. *Theoretical Computer Science*, 2001, 259(1/2): 513-531.

[19] Hartman T, Sharan R. A 1.5-approximation algorithm for sorting by transpositions and transreversals. *Journal of Computer and System Sciences*, 2005, 70(3): 300-320.

[20] Rahman A, Shatabda S, Hasan M. An approximation algorithm for sorting by reversals and transpositions. *Journal of Discrete Algorithms*, 2008, 6(3): 449-457.

[21] Lou X, Zhu D. A 2.25-approximation algorithm for cut-and-paste sorting of unsigned circular permutations. In *Proc. the 14th Annual International Computing and Combinatorics Conference*, Dalian, China, Jun. 27-29, 2008, pp.331-341.

[22] Eriksen N. $(1+\varepsilon)$-approximation of sorting by reversals and transpositions. *Theoretical Computer Science*, 2002, 289(1): 517-529.

[23] Bader M, Ohlebusch E. Sorting by weighted reversals, transpositions, and inverted transpositions. *Journal of Computational Biology*, 2007, 14(5): 615-636.

[24] Bader D, Moret B, Yan M. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *Journal of Computational Biology*, 2001, 8(5): 483-491.

Xiao-Wen Lou received her B.S. degree in 2004 from School of Computer Science and Technology, Shandong University, where she is currently a Ph.D. candidate. Her research interests include algorithm design and analysis and computational biology.

Da-Ming Zhu received his B.S. degree from University of Science and Technology of China in 1987 and his Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences in 1999, both in computer science. He is currently a professor and Ph.D. supervisor in School of Computer Science and Technology, Shandong University, and senior member of China Computer Federation. His main research interests include algorithm design and analysis, computational biology and neural network.

Appendix

Proof of Theorem 1. Let π_1' be an optimal r-spin of π containing the minimum number of anti-strips among all the optimal r-spins satisfying the conditions (I) and (II) of Lemma 2. Let s be an anti-strip of π_1', and let π_2' be the twin of π_1' with canonical 2-strip s, i.e., $\pi_1' = \pi_2' \cdot \mathit{flip}(s)$. Then π_2' also satisfies the conditions (I) and (II) of Lemma 2. From Lemma 4, $b(\pi_1') = b(\pi_2')+1$. Note that a 2-strip s may belong to two cycles of $G(\pi_2'')$, or belong to the same cycle of $G(\pi_2'')$. The following will discuss them respectively.

Case 1. If s belongs to two cycles of $G(\pi_2'')$. From Lemma 5, $c(\pi_1') = c(\pi_2') - 1$. Since *flip*$(s)$ can affect at most two hurdles of π_2', and when *flip*(s) removes two hurdles, they must both be simple hurdles. From the definition of fortress, we know $(h+f)(\pi_1') \geqslant (h+f)(\pi_2')-2$. From (1), $d_{\mathrm{r}}(\pi_2') \leqslant d_{\mathrm{r}}(\pi_1')$. This implies that π_2' is an optimal r-spin of π satisfying the conditions (I) and (II) of Lemma 2, and that π_2' contains fewer anti-strips than π_1', a contradiction.

Case 2. If s belongs to the same cycle of $G(\pi_2'')$. From Lemma 5, the change of cycle number due to *flip*(s) on π_2' depends on the relative direction of black edges b_1 and b_2 which are incident to s.

Case 2.1. If b_1 and b_2 have the opposite direction, then $c(\pi_1') = c(\pi_2')$. Since s is in an oriented component of $G(\pi_2'')$, $(h+f)(\pi_1') \geqslant (h+f)(\pi_2')$. From (1), $d_{\mathrm{r}}(\pi_1') > d_{\mathrm{r}}(\pi_2')$, a contradiction.

Case 2.2. If b_1 and b_2 have the same direction, then $c(\pi_1') = c(\pi_2') + 1$. If b_1 and b_2 are in an oriented component of $G(\pi_2'')$, then $(h+f)(\pi_1') \geqslant (h+f)(\pi_2')$. From (1), $d_{\mathrm{r}}(\pi_1') \geqslant d_{\mathrm{r}}(\pi_2')$. It implies that π_2' is an optimal r-spin of π satisfying the conditions of Lemma 2 with fewer anti-strips than π_1', a contradiction. So b_1 and b_2 must belong to one cycle of an unoriented component K of $G(\pi_2'')$, i.e., s is a satisfied 2-strip of K. In this case, *flip*(s) will never increase $(h+f)(\pi_2')$, and can at most decrease $(h+f)(\pi_2')$ by 1. If $(h+f)(\pi_1') = (h+f)(\pi_2')$, then $d_{\mathrm{r}}(\pi_1') = d_{\mathrm{r}}(\pi_2')$, a contradiction. Therefore $(h+f)(\pi_1') = (h+f)(\pi_2') - 1$ and $d_{\mathrm{r}}(\pi_1') = d_{\mathrm{r}}(\pi_2') - 1$.

The above analysis shows that if π_1' is an optimal r-spin of π containing the minimum number of anti-strips among all the optimal r-spins satisfying the conditions (I) and (II) of Lemma 2, then its twin π_2' with canonical s must have s in one cycle of an unoriented component of $G(\pi_2'')$ and $(h+f)(\pi_1') = (h+f)(\pi_2') - 1$ holds.

Suppose that π_1' contains u anti-strips $s_1', s_2', \ldots, s_u'$. Applying *flip*(s) to all of them will turn them into canonical 2-strips $s_1, s_2, \ldots, s_u$, thus turning π_1' into a canonical spin, denoted by δ. Note that $d_{\mathrm{r}}(\delta) = d_{\mathrm{r}}(\pi_1') + u$ must hold, and each s_i $(1 \leqslant i \leqslant u)$ must be a satisfied 2-strip of δ, at most one in every component of δ. Let $K_1, K_2, \ldots, K_u$ be the u unoriented components of δ containing satisfied 2-strips $s_1, s_2, \ldots, s_u$. Let π_3' be a spin of π obtained from δ by arbitrarily choosing a satisfied 2-strip of K_i (for $1 \leqslant i \leqslant u$), and transforming it into an anti-strip. From Lemma 4, $b(\pi_3') = b(\delta) + u = b(\pi_1')$, from Lemma 5, $c(\pi_3') = c(\delta) + u = c(\pi_1')$. Moreover, $(h+f)(\pi_1') = (h+f)(\pi_3')$. This is because for an unoriented component K_i, if it contains more than one satisfied 2-strips, turning any one of them into an anti-strip has the same effect of turning K_i into an oriented component, thus has the same effect on $(h+f)$. Therefore, $d_{\mathrm{r}}(\pi_3') = d_{\mathrm{r}}(\pi_1')$.

If π_3' does not have additional unoriented components containing satisfied 2-strips, π_3' is a super-r-spin of π. Otherwise, suppose π_3' has unoriented components $K_{u+1}, \ldots, K_x$, each of which contains satisfied 2-strips. For every K_j $(u+1 \leqslant j \leqslant x)$, arbitrarily select a satisfied 2-strip $s_j \in K_j$, *flip*(s_j) will transform K_j into an oriented component without increasing the value of $(b-c)(\pi_3')$ and $(h+f)(\pi_3')$, thus $d_{\mathrm{r}}(\pi_3' \cdot \mathit{flip}(s_j)) \leqslant d_{\mathrm{r}}(\pi_3')$. Let $\pi_4' = \pi_3' \cdot \mathit{flip}(s_{u+1}) \cdot \ldots \cdot \mathit{flip}(s_x)$, then π_4' is a super-r-spin of π with $d_{\mathrm{r}}(\pi_4') \leqslant d_{\mathrm{r}}(\pi_3') = d_{\mathrm{r}}(\pi_1')$. Thus π_4' is an optimal r-spin. This completes the proof. □