Volume 73 Number 7 November 2007 ISSN 0022-0000

ELSEVIER

JOURNAL OF COMPUTER AND SYSTEM SCIENCES

Special Issue: Bioinformatics III

Available online at www.sciencedirect.com

ScienceDirect

Journal of Computer and System Sciences 73 (2007) 1045–1059

JOURNAL OF COMPUTER AND SYSTEM SCIENCES

www.elsevier.com/locate/jcss

A 1.75-approximation algorithm for unsigned translocation distance

Yun Cui [a], Lusheng Wang [b,*], Daming Zhu [a]

[a] *School of Computer Science and Technology, Shandong University, PR China*
[b] *Department of Computer Science, City University of Hong Kong, Hong Kong*

Received 20 August 2005; received in revised form 11 April 2006

Available online 14 March 2007

Abstract

The translocation operation is one of the popular operations for genome rearrangement. In this paper, we present a 1.75-approximation algorithm for computing unsigned translocation distance which improves upon the best known 2-approximation algorithm [J. Kececioglu, R. Ravi, Of mice and men: Algorithms for evolutionary distances between genomes with translocation, in: 6th ACM–SIAM Symposium on Discrete Algorithms, 1995, pp. 604–613].

Keywords: Unsigned translocation distance; Approximation algorithm

1. Introduction

Genome rearrangement is an important area in computational biology [6–10]. There are several basic operations, e.g., *reversal*, *translocation*, and *transposition*. Here we study the translocation operations. A *chromosome* $X = x_1, x_2, \dots, x_p$ is a sequence of genes, where each gene x_i is represented by an integer. A gene x_i has a direction. When the direction of every gene is known, we use a signed integer to indicate the direction. When the directions of genes are unknown, we use unsigned integers to represent the genes. Throughout this paper, each x_i in a *signed chromosome* is a signed integer, and each x_i in an *unsigned chromosome* is an unsigned integer. A *signed genome* is a set of signed chromosomes and an *unsigned* genome is a set of unsigned chromosomes.

For two unsigned chromosomes $X = x_1, x_2, \dots, x_m$ and $Y = y_1, y_2, \dots, y_n$ in a genome, a *translocation* swaps the segments in the chromosomes and generates two new chromosomes. A prefix–prefix translocation $\rho_{pp}(X, Y, i, j)$ generates two new chromosomes: $x_1, \dots, x_{i-1}, y_j, \dots, y_n$ and $y_1, \dots, y_{j-1}, x_i, \dots, x_m$. A prefix–suffix translocation $\rho_{ps}(X, Y, i, j)$ generates two new chromosomes: $x_1, \dots, x_{i-1}, y_{j-1}, \dots, y_1$ and $x_m, \dots, x_i, y_j, \dots, y_n$.

For two signed chromosomes $X = x_1, x_2, \dots, x_m$ and $Y = y_1, y_2, \dots, y_n$ in a genome, a prefix–prefix translocation $\rho_{pp}(X, Y, i, j)$ generates two new chromosomes: $x_1, \dots, x_{i-1}, y_j, \dots, y_n$ and $y_1, \dots, y_{j-1}, x_i, \dots, x_m$. A prefix–suffix translocation $\rho_{ps}(X, Y, i, j)$ generates two new chromosomes: $x_1, \dots, x_{i-1}, -y_{j-1}, \dots, -y_1$ and $-x_m, \dots, -x_i, y_j, \dots, y_n$.

[*] Corresponding author.
E-mail addresses: yuncuiyc@hotmail.com (Y. Cui), cswangl@cityu.edu.hk (L. Wang), dmzhu@sdu.edu.cn (D. Zhu).

0022-0000/$ – see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcss.2007.03.009

The *translocation distance* between two (signed/unsigned) genomes is the minimum number of translocations used to transform one genome into the other.

Hannenhalli designed the first $O(n^3)$ algorithm [2] for computing translocation distance for signed genomes. The time complexity was improved to $O(n^2)$ in [3]. In [5], an error originated in [2] was fixed. The problem of computing translocation distance for unsigned genomes was recently proved to be NP-hard [4]. Kececioglu and Ravi gave a ratio-2 approximation algorithm for the translocation distance for unsigned genomes [1].

In this paper, we present a ratio-1.75 approximation algorithm for computing the translocation distance of unsigned genomes which improves upon the best known 2-approximation algorithm [1]. Our algorithm uses the maximum match method to find a cycle decomposition that contains enough number of 2-cycles (cycle containing exactly two black edges). By doing this, we give each unsigned gene a sign and the problem becomes the computation of translocation distance for signed genomes. Thus, we can use the algorithm in [3,5] for signed genomes to finally get an approximation solution.

2. Signed and unsigned translocation

The basic idea of our approximation algorithm for unsigned genomes is to carefully assign a sign to each gene in the genomes and use the algorithm for signed genomes to compute the translocation distance. The approximation ratio purely depends on the quality of the sign assignment of each gene.

First, let us introduce the computation method for signed genomes.

2.1. Signed translocation

Given signed genomes $\boldsymbol{A}$ and $\boldsymbol{B}$, the *breakpoint graph* $G_s(\boldsymbol{A}, \boldsymbol{B})$ can be obtained as follows: for every chromosome $X = x_1, x_2, \ldots, x_n$ of $\boldsymbol{A}$, replace each x_i with an ordered pair $(l(x_i), r(x_i))$ of vertices. If x_i is positive, $(l(x_i), r(x_i)) = (x_i^t, x_i^h)$; if x_i is negative, $(l(x_i), r(x_i)) = (x_i^h, x_i^t)$. The vertices $r(x_i)$ and $l(x_{i+1})$ are *neighbors* in $\boldsymbol{A}$. The neighbors in $\boldsymbol{B}$ are defined analogously. For two vertices u and v, if they are neighbors in $\boldsymbol{A}$, then we use a black edge to connect them; if they are neighbors in $\boldsymbol{B}$, then we use a grey edge to connect them.

Example 1. Let the two genomes be $A = \{(1, 2, 3), (4, -6, -5, 7)\}$ and $B = \{(1, 2, 3), (4, 5, 6, 7)\}$. Both A and B contain two chromosomes. The breakpoint graph is shown in Fig. 1(a).

Every vertex in $G_s(\boldsymbol{A}, \boldsymbol{B})$ is incident with at most one black and one grey edge. Therefore, $G_s(\boldsymbol{A}, \boldsymbol{B})$ can be uniquely decomposed into *cycles*. A cycle containing exactly i black (grey) edges is called an i*-cycle*. A cycle is *long* if it is not a 1-cycle.

Let $X = x_1, x_2, \ldots, x_p$ be a chromosome in $\boldsymbol{A}$. A *subpermutation* (*SP*) is an interval $x_i, x_{i+1}, \ldots, x_{i+l}$ in X containing at least three genes such that there is another interval of the same length $y_j, y_{j+1}, \ldots, y_{j+l}$ in a chromosome Y of $\boldsymbol{B}$ satisfying $\{|x_i|, |x_{i+1}|, \ldots, |x_{i+l}|\} = \{|y_j|, |y_{j+1}|, \ldots, |y_{j+l}|\}$, $x_i = y_j$, $x_{i+l} = y_{j+l}$ and $x_i, x_{i+1}, \ldots, x_{i+l-1}, x_{i+l} \neq y_j, y_{j+1}, \ldots, y_{j+l-1}, y_{j+l}$. Here x_i and x_{i+l} are the two *ending* genes of the *SP*. A *minimal subpermutation* (*minSP*) is a *SP* not containing any other *SP*. By the definition of *SP*, we have

Lemma 1. *Let* $I = r(x_i), l(x_{i+1}), r(x_{i+1}), \ldots, l(x_{j-1}), r(x_{j-1}), l(x_j)$ *denote a SP in* $G_s(\boldsymbol{A}, \boldsymbol{B})$*, then the grey edge* $(r(x_i), l(x_j))$ *is not in* $G_s(\boldsymbol{A}, \boldsymbol{B})$*. Moreover, the two (ending) genes* x_i *and* x_j *cannot be neighbors in* $\boldsymbol{B}$.

The translocation distance for signed genomes is closely related to the number of cycles and the number of *minSP*'s. If all *minSP*'s in $G_s(\boldsymbol{A}, \boldsymbol{B})$ are in a *SP*, say, I, and the total number of *minSP*'s is even, then I is an *even-isolation*. Clearly there is at most one even-isolation in $G_s(\boldsymbol{A}, \boldsymbol{B})$.

Let n be the number of genes in the two genomes and N the number of chromosomes in the genomes. c denotes the total number of cycles in the breakpoint graph and s denotes the number of *minSP*'s. f is the *remaining index* which is defined as follows: (1) $f = 1$ if s is odd; (2) $f = 2$ if there is an even-isolation; (3) $f = 0$ otherwise. Lemma 1 gives the formula to compute the translocation distance $d_s(\boldsymbol{A}, \boldsymbol{B})$ for the two signed genomes $\boldsymbol{A}$ and $\boldsymbol{B}$.

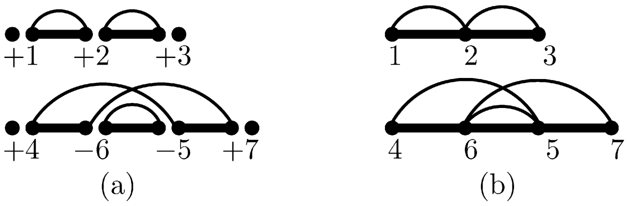

Fig. 1. (a) The breakpoint graph for signed genome. (b) The breakpoint graph for unsigned genome.

Lemma 2. *(See [2].)*

$$d_s(\boldsymbol{A}, \boldsymbol{B}) = n - N - c + s + f. \tag{1}$$

2.2. Unsigned translocation

Consider unsigned genomes $\boldsymbol{A}$ and $\boldsymbol{B}$. For every chromosome $X = x_1, x_2, \ldots, x_n$ of $\boldsymbol{A}$, x_i and x_{i+1} are *neighbors* in $\boldsymbol{A}$. The neighbors in $\boldsymbol{B}$ are defined analogously. To define the *breakpoint graph* $G(\boldsymbol{A}, \boldsymbol{B})$, we use a vertex to represent a gene. Two vertices are connected with a black edge if they are neighbors in $\boldsymbol{A}$ and two vertices are connected with a grey edge if they are neighbors in $\boldsymbol{B}$.

Example 2. Let the two genomes be $A = \{(1, 2, 3), (4, 6, 5, 7)\}$ and $B = \{(1, 2, 3), (4, 5, 6, 7)\}$. Both A and B contain two chromosomes. The breakpoint graph is shown in Fig. 1(b).

Note that every vertex is incident either with one black and one grey edge, or with two black and two grey edges. Therefore, the cycle decompositions for $G(\boldsymbol{A}, \boldsymbol{B})$ are not unique. Once we have a cycle decomposition for the breakpoint graph of two unsigned genomes, we actually assign a sign to each gene in the genomes. Thus, one way to compute the translocation distance for two unsigned genomes is to (1) try all possible ways to get cycle decomposition (thus we can get a sign for each gene), and (2) compute the translocation distance for signed genomes and select the minimum value among all possible cycle decompositions.

3. The approximation algorithm

If we can give a good approximation of the cycle decomposition of the unsigned case, we can get a good approximation solution for the unsigned translocation distance. Our main idea of the approximation algorithm is to give a cycle decomposition of $G(\boldsymbol{A}, \boldsymbol{B})$ that contains the maximum number of 1-cycles and a sufficient number of 2-cycles.

3.1. Why the ratio could be better than 2?

Now, we give an intuitive explanation that if we keep the maximum number of 1-cycles and maximum number of 2-cycles in assigning signs to genes, then the best performance ratio we can expect is 1.5.

Suppose that we ignore the effect of s and f in formula (1). That is, we assume that $s = 0$ and $f = 0$ in the optimal cycle decomposition. Then $d_s(\boldsymbol{A}, \boldsymbol{B}) = n - N - c$. Let c_i^* be the number of i-cycles in the optimal cycle decomposition. Then

$$d_s(\boldsymbol{A}, \boldsymbol{B}) = n - N - c = n - N - c_1^* - c_2^* - \sum_{i \geqslant 3} c_i^*. \tag{2}$$

$n - N$ is the number of black edges in the breakpoint graph. We further assume that $c_1^* = 0$, $c_2^* = 0$ and all black edges are in 3-cycles in the optimal cycle decomposition. In this case, $d_s(\boldsymbol{A}, \boldsymbol{B}) = n - N - \frac{n-N}{3} = \frac{2}{3}(n - N)$. If in the approximation solution, we do not care about i-cycles for $i \geqslant 3$, the distance for the approximation solution could be $n - N$. Thus, the ratio becomes $\frac{3}{2}$. In our approximation algorithm, we cannot get the maximum number of 2-cycles, but we get a large number of 2-cycles. Besides, we have to design sophisticated ways to deal with the other two parameters s and f in the analysis.

3.2. The cycle decomposition algorithm

Given unsigned genomes $\boldsymbol{A}$ and $\boldsymbol{B}$, a cycle decomposition of $G(\boldsymbol{A}, \boldsymbol{B})$ can be computed in the following three steps.

Step 1. Decomposition of 1-cycles.

If two vertices are joined by a black edge and a grey edge in $G(\boldsymbol{A}, \boldsymbol{B})$, then assign proper signs to the two vertices to obtain the 1-cycle containing the black edge and the grey edge. Thus, if two genes are neighbors in both genomes, the corresponding 1-cycle is kept in the cycle decomposition.

Step 2. Decomposition of 2-cycles.

From $G(\boldsymbol{A}, \boldsymbol{B})$, we define a new graph, called *match graph*, $F_{\boldsymbol{AB}}$ as follows: (1) For every black edge in $G(\boldsymbol{A}, \boldsymbol{B})$ with at least one end not assigned a sign in Step 1, we create a vertex of $F_{\boldsymbol{AB}}$. (2) For every two vertices of $F_{\boldsymbol{AB}}$ (representing two black edges in $G(\boldsymbol{A}, \boldsymbol{B})$), we create an edge connecting them in $F_{\boldsymbol{AB}}$ if the two black edges in $G(\boldsymbol{A}, \boldsymbol{B})$ can form a 2-cycle. $F_{\boldsymbol{AB}}$ can be constructed in $O(n^2)$ time where n is the number of genes.

Let M denote a maximum match of $F_{\boldsymbol{AB}}$. $|M|$ is the size of the match. A maximum match of any graph can be found in $O(|V||E|^{\frac{1}{2}})$ time, where $|V|$ is the number of vertices and $|E|$ is the number of edges [11]. Since $F_{\boldsymbol{AB}}$ contains at most n vertices and $O(n)$ edges, M can be found in $O(n^{\frac{3}{2}})$ time. Every edge in M represents a 2-cycle of $G(\boldsymbol{A}, \boldsymbol{B})$. By the construction, two 2-cycles in M cannot share any black edge of $G(\boldsymbol{A}, \boldsymbol{B})$. However, they may share a grey edge in $G(\boldsymbol{A}, \boldsymbol{B})$. In that case, the two 2-cycles cannot be kept in the cycle decomposition simultaneously. A 2-cycle in M is *isolated* if it does not share any grey edge with any other 2-cycles in M. Otherwise, the 2-cycle is *related*. Since a 2-cycle has two grey edges, it is related to at most two 2-cycles.

A *related component* U consists of related cycles $C_1, C_2, \ldots, C_k$, where C_i is related to C_{i-1} ($2 \leqslant i \leqslant k$), and every 2-cycle in U is not related to any 2-cycle not in U. A related component involves at most two chromosomes, and can be one of the four types shown in Fig. 2. In our cycle decomposition, we keep all the isolated 2-cycles and alternatively select 2-cycles from every related component. Assume that a maximum match M of $F_{\boldsymbol{AB}}$ contains z isolated 2-cycles. In our cycle decomposition approach, we can keep at least $\lceil \frac{|M|-z}{2} \rceil + z$, i.e., $\lceil \frac{|M|+z}{2} \rceil$ 2-cycles in Step 2.

Step 3. Decomposition of other long cycles.

After the decomposition of 2-cycles, the other long cycles can be arbitrarily selected from the remaining graph.

The long cycles created in Step 2 are called *selected* cycles and the cycles created in Step 3 are called *arbitrary* cycles.

Our approximation algorithm for unsigned translocation problem is as follows:

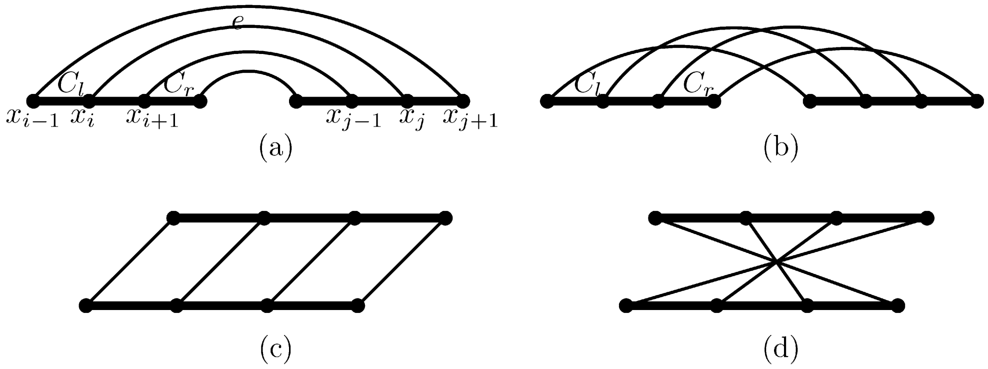

Fig. 2. The four cases of related components. Each of the related components in (a) and (b) is in one chromosome. Each of the related components in (c) and (d) is in two chromosomes.

```
Algorithm 1:
Input: $G(\boldsymbol{A}, \boldsymbol{B})$
1. Compute the cycle decomposition of $G(\boldsymbol{A}, \boldsymbol{B})$ as described before. Denote the resulting graph as $G_s^A(\boldsymbol{A}, \boldsymbol{B})$.
2. Solve the signed case using the standard algorithm.
```

Let n be the number of genes in the given genomes. $G(\boldsymbol{A}, \boldsymbol{B})$ and $F_{\boldsymbol{AB}}$ can be constructed in $O(n^2)$ time. A maximum match of $F_{\boldsymbol{AB}}$ can be found in $O(n^{\frac{3}{2}})$ time. The algorithm in [3] requires $O(n^2)$ time to compute an optimal sequence of translocations for signed case. Thus, the total time required for our approximation algorithm is $O(n^2)$.

A *minSP* $I = r(x_i), l(x_{i+1}), r(x_{i+1}), \ldots, l(x_{j-1}), r(x_{j-1}), l(x_j)$ *contains* a cycle C if all vertices of C are in $\{r(x_i), l(x_{i+1}), r(x_{i+1}), \ldots, l(x_{j-1}), r(x_{j-1}), l(x_j)\}$. A cycle C is *outside* I if no vertex of C is in $\{r(x_i), l(x_{i+1}), r(x_{i+1}), \ldots, l(x_{j-1}), r(x_{j-1}), l(x_j)\}$.

Lemma 3. *If a minSP contains a selected related* 2*-cycle in* $G_s^A(\boldsymbol{A}, \boldsymbol{B})$*, then this minSP contains at least one arbitrary cycle.*

Proof. Suppose a *minSP* I contains a selected related 2-cycle C in a related component U of $G_s^A(\boldsymbol{A}, \boldsymbol{B})$, and U contains 2-cycles $C_l, C_{l+1}, \ldots, C_r$, where C_i is related to C_{i+1}, $l \leqslant i \leqslant r-1$. C can only be in a related component U of type (a) or (b) in Fig. 2. Assume that C_l contains the leftmost vertex of U.

(1) U is of type (a). If $C = C_i$, $l \leqslant i \leqslant r-1$, then each black edge of C_{i+1} is in an arbitrary cycle of I. If $C = C_r$, then each black edge of C_{r-1} belongs to an arbitrary cycle. Since there is a grey edge connecting the leftmost and rightmost vertices of C_r, by Lemma 1, at least one black edge e of C_{r-1} (that is in an arbitrary cycle) is in I. Since I is a *minSP*, by the definition of *SP*, I contains the whole cycle that e is in.

(2) U is of type (b). If $C = C_i$, $l \leqslant i \leqslant r-1$, then the black edge of C_{i+1} between the two black edges of C_i is in an arbitrary cycle of I. If $C = C_r$, then the black edge of C_{r-1} which is between the two black edges of C_r (see Fig. 2(b)) is in an arbitrary cycle of I. □

4. Analysis of the performance ratio

In this section, we will show that the performance ratio of the algorithm is 1.75. We use several new bounds in our analysis.

Suppose that each of the given genomes has n genes and N chromosomes. Let $d(\boldsymbol{A}, \boldsymbol{B})$ denote the (optimal) translocation distance between two unsigned genomes $\boldsymbol{A}$ and $\boldsymbol{B}$, and $G_s^{\mathrm{opt}}(\boldsymbol{A}, \boldsymbol{B})$ the breakpoint graph of an optimal cycle decomposition.

4.1. 1-Cycles

In this subsection, we will show that Step 1 in the cycle decomposition algorithm always leads to a good approximation solution.

Lemma 4. *We modify* $G_s^{\mathrm{opt}}(\boldsymbol{A}, \boldsymbol{B})$ *as follows: if two vertices in* $G(\boldsymbol{A}, \boldsymbol{B})$ *are connected by a black edge and a grey edge in* $G(\boldsymbol{A}, \boldsymbol{B})$*, then we re-assign the signs of the two genes to obtain a* 1*-cycle. Assume that the resulting breakpoint graph has* c' *cycles and* s' *minSP's. We have* $d(\boldsymbol{A}, \boldsymbol{B}) \geqslant n - N - c' + s' + f^o$*, where* f^o *is the remaining index for* $G_s^{\mathrm{opt}}(\boldsymbol{A}, \boldsymbol{B})$.

Proof. Suppose that $G_s^{\mathrm{opt}}(\boldsymbol{A}, \boldsymbol{B})$ includes c cycles and s *minSP*'s. By Lemma 2, $d(\boldsymbol{A}, \boldsymbol{B}) = n - N - c + s + f^o$. Consider two vertices x_i and x_{i+1} that are neighbors in both genomes. Suppose that there are $c(i)$ cycles and $s(i)$ *minSP*'s in the breakpoint graph before x_i and x_{i+1} are processed, and there are $c'(i)$ cycles and $s'(i)$ *minSP*'s after x_i and x_{i+1} are processed. We want to show that $c' - s' - c + s \geqslant 0$. It suffices to show that $c'(i) - s'(i) - c(i) + s(i) \geqslant 0$ for each i. To form the 1-cycle, we have the following two cases.

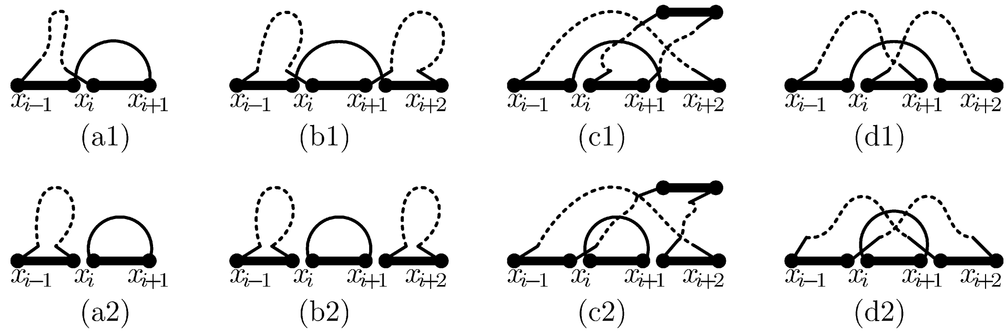
x_{i-1} x_i x_{i+1}
(a1)
x_{i-1} x_i x_{i+1} x_{i+2}
(b1)
x_{i-1} x_i x_{i+1} x_{i+2}
(c1)
x_{i-1} x_i x_{i+1} x_{i+2}
(d1)
x_{i-1} x_i x_{i+1}
(a2)
x_{i-1} x_i x_{i+1} x_{i+2}
(b2)
x_{i-1} x_i x_{i+1} x_{i+2}
(c2)
x_{i-1} x_i x_{i+1} x_{i+2}
(d2)

Fig. 3. The cases for one cycle decomposition.

(1) The sign of one of x_i and x_{i+1} is changed.

Without loss of generality, assume that the sign of x_i is changed. In Fig. 3, (a1) and (a2) show the situations before and after changing the sign of x_i. After changing the sign of x_i, the black edges $(r(x_{i-1}), l(x_i))$ and $(r(x_i), l(x_{i+1}))$ are in two different cycles. Thus, we have $c'(i) = c(i) + 1$. Moreover, the number of *minSP*'s is not increased. Thus, we have $s'(i) \leqslant s(i)$. Therefore, the lemma holds.

(2) The signs of both x_i and x_{i+1} are changed.

Since we have to change the signs of both x_i and x_{i+1} to get the 1-cycle, the grey edge $(l(x_i), r(x_{i+1}))$ must exist before changing the signs. Thus, the two black edges $(r(x_{i-1}), l(x_i))$ and $(r(x_{i+1}), l(x_{i+2}))$ are in one cycle before changing the signs. Three subcases arise. See Fig. 3(b1), (c1) and (d1). The situations after changing the signs are illustrated in Fig. 3(b2), (c2) and (d2), respectively.

In Fig. 3(b2), the number of cycles are increased by 2, i.e., $c'(i) = c(i) + 2$. The number of *minSP*'s will be increased by at most 2 (in fact, at most 1), i.e., $s'(i) \leqslant s(i) + 2$. For the cases illustrated in Fig. 3(c2) and (d2), we have $s'(i) = s(i)$. Obviously, $c'(i) \geqslant c(i)$. Thus, the lemma holds for this case. □

4.2. A lower bound

In this subsection, we give a lower bound for $d(\boldsymbol{A}, \boldsymbol{B})$. This lower bound will be used as the starting point of our analysis.

Note that every *minSP* contains at least one long cycle. A *simple minSP* (S-MSP) is a *minSP* containing *one* 2-cycle as its *unique* long cycle. By definition, a simple *minSP* is a segment of genes in a chromosome containing 1-cycle(s) in the middle of the segment and a 2-cycle containing the two black edges at the two ends of the segments. The two grey edges in the 2-cycle must be "twisted" since by Lemma 1 $(r(x_i), l(x_j))$ cannot be a grey edge for the two ending genes x_i and x_j. The whole analysis of the approximation algorithm depends heavily on the special treatment of simple *minSP*'s.

Given unsigned genomes $\boldsymbol{A}$ and $\boldsymbol{B}$, a *candidate simple minSP* (*CS*-MSP for short) is defined as an interval $I_c = x_i, x_{i+1}, \ldots, x_{i+l-1}, x_{i+l}$ containing at least *four* genes in a chromosome of $\boldsymbol{A}$ such that there is another interval of the same length $y_j, y_{j+1}, \ldots, y_{j+l}$ in a chromosome Y of $\boldsymbol{B}$ satisfying $x_i = y_j$, $x_{i+l} = y_{j+l}$ and $x_{i+k} = y_{j+l-k}$ $(1 \leqslant k \leqslant l-1)$. Any *CS*-MSP can be turned into a S-MSP by assigning proper signs to all genes in it. For convenience, we also call the unique 2-cycle in the S-MSP, the *unique* 2-*cycle* in the *CS*-MSP.

Given signed genomes $\boldsymbol{A}$ and $\boldsymbol{B}$, let $I_s = x_i, x_{i+1}, \ldots, x_{j-1}, x_j$ be a S-MSP in $G_s(\boldsymbol{A}, \boldsymbol{B})$. A cycle $C = r(x_{i-1}), l(x_i), \ldots, l(x_{j+1}), r(x_j), \ldots, r(x_{i-1})$ in $G_s(\boldsymbol{A}, \boldsymbol{B})$ containing the two black edges $(r(x_{i-1}), l(x_i))$ and $(r(x_j), l(x_{j+1}))$ on the left and right of I_s is a *removable cycle*. (See Fig. 4(a).) If there is a removable cycle C for I_s, then I_s is a *removable simple minSP* (*RS*-MSP for short).

Lemma 5. *Given a RS-MSP $I_s = x_i, x_{i+1}, \ldots, x_{j-1}, x_j$, if we change the signs of x_i and x_j, then we have*

(a) *$I_1 = -x_i, x_{i+1}, \ldots, x_{j-1}, -x_j$ is no longer a minSP*;

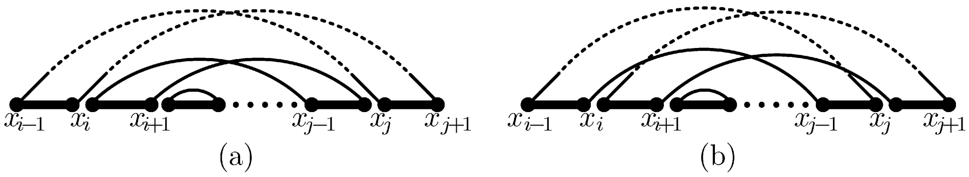

Fig. 4. The breakpoint graphs before and after a *RS*-MSP is destroyed.

(b) *the number of cycles in the new breakpoint graph remains the same and the number of minSP's is not increased.*
(c) *Each of the black edges* $(r(x_i), l(x_{i+1}))$ *and* $(r(x_{j-1}), l(x_j))$ *is in a long cycle containing a black edge that is either* $(r(x_{i-1}), l(x_i))$ *or* $(r(x_j), l(x_{j+1}))$.

Proof. Since I_s is a *RS*-MSP, it has a 2-cycle $C' = r(x_i), l(x_{i+1}), l(x_j), r(x_{j-1}), r(x_i)$ in $G_s(\boldsymbol{A}, \boldsymbol{B})$ (see Fig. 4(a)). Changing the signs of x_i and x_j destroys C (the removable cycle) and C' and creates two new cycles $C_1 = r(x_{i-1}), l(x_i), r(x_{j-1}), l(x_j), \ldots, r(x_{i-1})$ and $C_2 = r(x_i), l(x_{i+1}), r(x_j), l(x_{j+1}), \ldots, r(x_i)$ (see Fig. 4(b)). Thus, (c) holds. Moreover, the number of cycles remains the same. Since $I_1 = -x_i, x_{i+1}, \ldots, x_{j-1}, -x_j$ is no longer a *minSP* and the segment I_1 contains no *minSP*, the only possible new *minSP* created might be an old *SP* containing the segment I_1. (In this case, the old *SP* becomes a new *minSP*.) Thus, the number of *minSP*'s is not increased. □

The lower bound of $d(\boldsymbol{A}, \boldsymbol{B})$ we are going to develop is based on the modification of CS-MSP's in an optimal cycle decomposition $G_s^{\text{opt}}(\boldsymbol{A}, \boldsymbol{B})$.

Modifying an optimal cycle decomposition $G_s^{\text{opt}}(\boldsymbol{A}, \boldsymbol{B})$

Let I_c be a *CS*-MSP and $l(I_c)$ and $r(I_c)$ denote the leftmost and rightmost genes of I_c. The modification method is as follows:

```
ModificationMethod:
Input: $G_s^{\text{opt}}(\boldsymbol{A}, \boldsymbol{B})$
1. For every chromosome $X$ of $\boldsymbol{A}$.
2. Obtain possible 1-cycles as described in Step 1 of cycle decomposition.
3. For every chromosome $X$ of $\boldsymbol{A}$.
4.   Process each CS-MSP $I_c$ in $X$ from left to right:
5.     Assign proper signs to $l(I_c)$ and $r(I_c)$ to turn $I_c$ into a S-MSP $I_s$.
6.     If $I_s$ is a RS-MSP, then remove it by changing the signs of both $l(I_s)$ and $r(I_s)$.
```

Theorem 1. *c^* and s^* denote the number of cycles and number of minSP's in the new breakpoint graph after ModificationMethod. We have $d(\boldsymbol{A}, \boldsymbol{B}) \geqslant n - N - c^* + s^* + f^o$, where f^o is the remaining index for $G_s^{\text{opt}}(\boldsymbol{A}, \boldsymbol{B})$.*

Proof. Suppose that the breakpoint graph includes c' cycles and s' *minSP*'s after Steps 1 and 2 of ModificationMethod. By Lemma 4, $d(\boldsymbol{A}, \boldsymbol{B}) \geqslant n - N - c' + s' + f^o$. Consider a *CS*-MSP $I_c = x_i, x_{i+1}, \ldots, x_{j-1}, x_j$. Let I_s be the *S*-MSP obtained by assigning signs for genes in I_c properly. Suppose that there are $c'(i)$ cycles and $s'(i)$ *minSP*'s in the breakpoint graph before the ith *CS*-MSP is processed, and there are $c^*(i)$ cycles and $s^*(i)$ *minSP*'s after that. We want to show that $c^* - s^* - c' + s' \geqslant 0$. It suffices to show that $c^*(i) - s^*(i) - c'(i) + s'(i) \geqslant 0$ for each i.

(1) If obtaining I_s requires no change of signs for x_i and x_j, then $c^*(i) - s^*(i) - c'(i) + s'(i) = 0$.

(2) Consider the case where I_s is obtained by changing the sign of one of x_i and x_j.

Without loss of generality, we change the sign of x_i (see Fig. 5(b)).

Before the change of sign, the configuration is shown as Fig. 5(a). The three black edges $(r(x_i), l(x_{i+1}))$, $(r(x_{j-1}), l(x_j))$ and $(r(x_{i-1}), l(x_i))$ must be in the same cycle. After the change of the sign, $(r(x_i), l(x_{i+1}))$ and $(r(x_{j-1}), l(x_j))$ are in a 2-cycle, whereas $(r(x_{i-1}), l(x_i))$ is in another cycle. So, $c^*(i) = c'(i) + 1$.

If a new *minSP* $I_l = x_l, x_{l+1}, \ldots, x_{i-1}, x_i$ is created on the left of I_s, then $x_l, x_{l+1}, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{j-1}, x_j$ is a *SP*. Moreover, this *SP* exists as a *minSP* just before the sign of x_i is changed. Thus, $s^*(i) \leqslant s'(i) + 1$. Therefore, $c^*(i) - s^*(i) \geqslant c'(i) - s'(i)$.

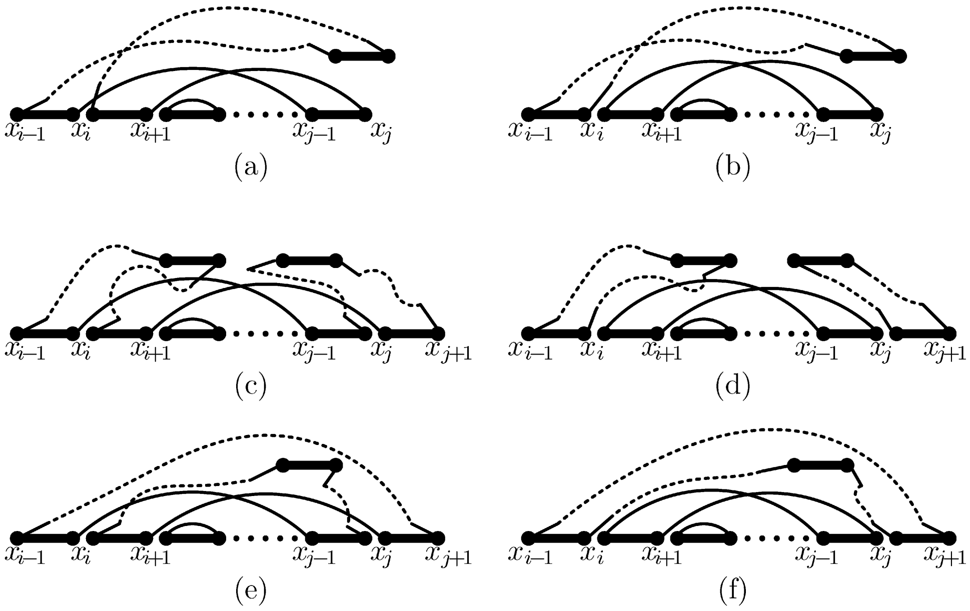

Fig. 5. Some cases of a *CS*-MSP in a breakpoint graph.

(3) If I_s is obtained by changing the signs of both x_i and x_j, then there are three cases.

Case 1. I_s is a *RS*-MSP. In this case, I_s is shown as in Fig. 4(a). Figure 4(b) shows the case before the change of signs. After the elimination of the *RS*-MSP, the case is back to the origin and thus $c^*(i) - s^*(i) - c'(i) + s'(i) = 0$.

Case 2. After changing the signs, I_s is not a *RS*-MSP, the black edges $(r(x_{i-1}), l(x_i))$ and $(r(x_j), l(x_{j+1}))$ are in two different cycles, and the black edges $(r(x_i), l(x_{i+1}))$ and $(r(x_{j-1}), l(x_j))$ are in one 2-cycle. See Fig. 5(d). Before the change of signs, these three cycles were in one cycle as shown in Fig. 5(c). Thus, $c^*(i) = c'(i) + 2$.

When I_s is obtained as in Fig. 5(d), it creates at most other two new *minSP*'s $I_l = x_l, x_{l+1}, \ldots, x_{i-1}, x_i$ and $I_r = x_j, x_{j+1}, \ldots, x_{r-1}, x_r$ that are on the left and right of I_s, respectively. If both I_l and I_r are new *minSP*'s, then I_l, I_s and I_r are three consecutive *minSP*'s. Thus, $x_l, x_{l+1}, \ldots, x_i, \ldots, x_j, \ldots, x_{r-1}, x_r$ was a *SP* before changing the signs. Obviously, this *SP* was a *minSP* since it is a merging of the three *minSP*'s I_l, I_s and I_r (by changing the signs of x_i and x_j). Thus, $s^*(i) \leqslant s'(i) + 2$. Therefore, $c^*(i) - s^*(i) \geqslant c'(i) - s'(i)$.

Case 3. I_s is not a *RS*-MSP, and the black edges $(r(x_{i-1}), l(x_i))$ and $(r(x_j), l(x_{j+1}))$ are in one cycle after changing the signs. See Fig. 5(f). Since I_s is a *S*-MSP, the black edges $(r(x_i), l(x_{i+1}))$ and $(r(x_{j-1}), l(x_j))$ are in one 2-cycle. Before changing the signs, these two cycles formed one cycle as shown in Fig. 5(e). $c^*(i) = c'(i) + 1$.

When I_s is obtained, since the black edges $(r(x_{i-1}), l(x_i))$ on the left of I_s and $(r(x_j), l(x_{j+1}))$ on the right of I_s are in one cycle, no new *minSP* is created (except I_s). Thus, $s^*(i) \leqslant s(i) + 1$. Therefore, $c^*(i) - s^*(i) - c'(i) + s'(i) \geqslant 0$. □

4.3. A key inequality

Given unsigned genomes $\boldsymbol{A}$ and $\boldsymbol{B}$, let s_c denote the number of *CS*-MSP's in $G(\boldsymbol{A}, \boldsymbol{B})$. Let c_i^* denote the number of i-cycles and s_e^* the number of *S*-MSP's in the new breakpoint graph after applying ModificationMethod.

If two *CS*-MSP's share one gene in a chromosome, then they are *adjacent*. A *CS*-MSP chain consists of a sequence of adjacent *CS*-MSP's $I_1, \ldots, I_i, I_{i+1}, \ldots, I_n$, where I_i and I_{i+1} are adjacent for $1 \leqslant i \leqslant n-1$. If a *CS*-MSP chain is not contained in any other *CS*-MSP chain, then it is a *maximum CS-MSP chain*.

Lemma 6. *For any maximum CS-MSP chain in $G(\boldsymbol{A}, \boldsymbol{B})$, either all CS-MSP's in the chain are turned into S-MSP's, or none of them is turned into a S-MSP after the ModificationMethod in Section* 4.2.

Proof. Assume a maximum *CS*-MSP chain in $G(\boldsymbol{A}, \boldsymbol{B})$ consists of *CS*-MSP's $I_1, I_2, \ldots, I_k$, where I_i and I_{i+1} are adjacent for $1 \leqslant i \leqslant k-1$. There are two cases when applying ModificationMethod.

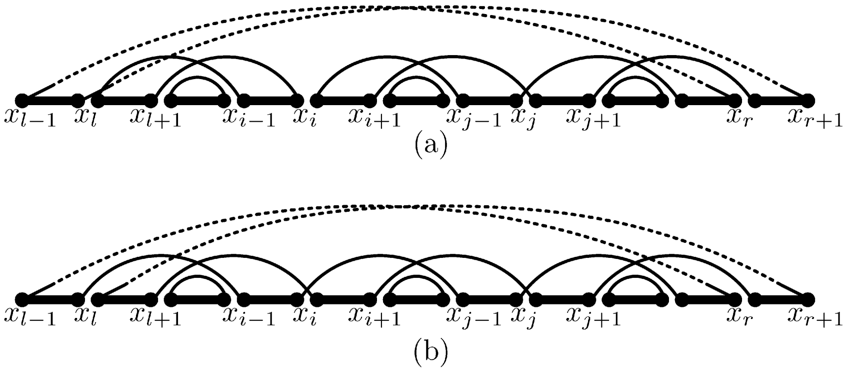

Fig. 6. Remove *RS*-MSP's on a maximum *CS*-MSP chain.

(1) I_1 is turned into a *S*-MSP, but not a *RS*-MSP. Consider $I_2 = x_i, x_{i+1}, \ldots, x_{j-1}, x_j$. When applying ModificationMethod, the black edge $(r(x_{i-1}), l(x_i))$ is in the *S*-MSP derived from I_1. Since $(r(x_{i-1}), l(x_i))$ and $(r(x_j), l(x_{j+1}))$ cannot be in one cycle, I_2 is turned into a *S*-MSP, but not a *RS*-MSP. The process goes on and all the I_k's for $k \geqslant 2$ are turned into *S*-MSP's, but not *RS*-MSP's.

(2) $I_1 = x_l, x_{l+1}, \ldots, x_{i-1}, x_i$ is turned into a *RS*-MSP. See Fig. 6(a). When I_1 is removed, the black edges $(r(x_{l-1}), l(x_l))$ and $(r(x_{i-1}), l(x_i))$ are in one cycle C_1, while the black edges $(r(x_l), l(x_{l+1}))$ and $(r(x_i), l(x_{i+1}))$ are in another cycle C_2. Assume $I_2 = x_i, x_{i+1}, \ldots, x_{j-1}, x_j$. Since grey edge $(l(x_i), r(x_{j-1}))$ exists, C_1 contains black edge $(r(x_{j-1}), l(x_j))$. Thus, C_2 contains grey edge $(l(x_{i+1}), r(x_j))$ but not $(l(x_{i+1}), l(x_j))$. Therefore, the black edge $(r(x_j), l(x_{j+1}))$ is in C_2. See Fig. 6(b). When the signs of x_i and x_j are changed, a new cycle $r(x_{i-1}), l(x_i), \ldots, l(x_{j+1}), r(x_j), \ldots, r(x_{i-1})$ is created. This implies that I_2 is turned into a *RS*-MSP. After the elimination of the *RS*-MSP, I_2 is changed back as shown in Fig. 6(b). The process goes on and we can conclude that every I_p for $p = 1, 2, \ldots, k$ is turned into a *RS*-MSP in this case. □

Theorem 2. $\sum_{i\geqslant 2}(i-1)c_i^* \geqslant 2(s_c - s_e^*)$.

Proof. If a *CS*-MSP I_c is not turned into a *S*-MSP after applying ModificationMethod, then I_c is turned into a *RS*-MSP and removed in Step 6 of ModificationMethod. First, we want to show that every black edge in the unique 2-cycle of the *CS*-MSP is in a long cycle (after ModificationMethod) containing at least one black edge that is not in any *CS*-MSP. There are two cases:

(1) $I_c = x_i, x_{i+1}, \ldots, x_{j-1}, x_j$ is not in a maximum *CS*-MSP chain. By Lemma 5, when I_s is removed, the black edges $(r(x_i), l(x_{i+1}))$ and $(r(x_{j-1}), l(x_j))$ belong to two different long cycles and each of the long cycles has a black edge that is not in a *CS*-MSP.

(2) I_c is in a maximum *CS*-MSP chain. In this case, after applying ModificationMethod, all the vertices of *CS*-MSP's are in two long cycles, one containing the black edge $(r(x_{l-1}), l(x_l))$ on the left of the chain and the other containing the black edge $(r(x_r), l(x_{r+1}))$ on the right of the chain (see Fig. 6(b)). Note that neither $(r(x_{l-1}), l(x_l))$ nor $(r(x_r), l(x_{r+1}))$ is in any *CS*-MSP.

The total number of *CS*-MSP's that are not turned into *S*-MSP's after applying ModificationMethod is $(s_c - s_e^*)$. There are $2(s_c - s_e^*)$ black edges in the unique 2-cycles of those *CS*-MSP's. Since every such black edge after modification is in a long cycle containing at least one black edge that is not in any *CS*-MSP, and the total number of black edges contained in those long cycles is at most $\sum_{i\geqslant 2} ic_i^*$, we have $\sum_{i\geqslant 2}(i-1)c_i^* \geqslant 2(s_c - s_e^*)$. □

For unsigned genomes $\boldsymbol{A}$ and $\boldsymbol{B}$, let $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ be the breakpoint graph produced by running ModificationMethod on $G_s^{\text{opt}}(\boldsymbol{A}, \boldsymbol{B})$. $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ is the breakpoint graph produced by Algorithm 1. f denotes the remaining index for $G_s^A(\boldsymbol{A}, \boldsymbol{B})$. We use $d^A(\boldsymbol{A}, \boldsymbol{B})$ to represent the translocation distance obtained by Algorithm 1. Let $d(\boldsymbol{A}, \boldsymbol{B})$ be the (optimal) translocation distance between the two unsigned genomes. Now, we are ready to show the performance ratio.

4.4. The performance ratio when $f = 0$

Assume that $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ contains z isolated 2-cycles. Let $z^{(o)}$ denote the number of isolated 2-cycles outside all *minSP*'s. Consider the *minSP*'s containing only isolated 2-cycles and 1-cycles. Let $s^{(s)}$ denote the number of (simple) *minSP*'s containing only one isolated 2-cycle. $s^{(m)}$ denotes the number of *minSP*'s containing at least two isolated 2-cycles without any selected related 2-cycle or arbitrary 2-cycle. Let $c_i^{(o)}$ be the number of arbitrary i-cycles ($i \geqslant 2$) outside all *minSP*'s in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$.

Theorem 3. *If $f = 0$, then $d^A(\boldsymbol{A}, \boldsymbol{B}) \leqslant \frac{7}{4}d(\boldsymbol{A}, \boldsymbol{B})$. That is, the performance ratio of Algorithm* 1 *is* 1.75 *if $f = 0$.*

Proof. By definition, $2s^{(m)} + s^{(s)} \leqslant z - z^{(o)}$. Thus, we have

$$s^{(m)} \leqslant \frac{z - z^{(o)} - s^{(s)}}{2}. \tag{3}$$

Suppose that $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ has s *minSP*'s. c_i ($i \geqslant 1$) denotes the number of i-cycles in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$. Similarly, c_i^* denotes the number of i-cycles in $G_s^*(\boldsymbol{A}, \boldsymbol{B})$. By Lemma 3, a *minSP* contains (at least) an isolated 2-cycle or an arbitrary cycle. Thus, there are $s - s^{(m)} - s^{(s)}$ *minSP*'s, each containing at least one arbitrary cycle. Since there are at least $\lceil \frac{|M|+z}{2} \rceil$ selected 2-cycles created in Step 2 of the cycle decomposition algorithm, the number of arbitrary cycles in *minSP*'s is less than or equal to $\sum_{i\geqslant 2} c_i - (\frac{|M|}{2} + \frac{z}{2}) - \sum_{i\geqslant 2} c_i^{(o)}$. We have

$$s - s^{(m)} - s^{(s)} \leqslant \sum_{i\geqslant 2} c_i - \left(\frac{|M|}{2} + \frac{z}{2}\right) - \sum_{i\geqslant 2} c_i^{(o)}. \tag{4}$$

Combining (3) and (4), we have

$$s \leqslant \sum_{i\geqslant 2} c_i - \sum_{i\geqslant 2} c_i^{(o)} - \frac{|M|}{2} - \frac{z^{(o)}}{2} + \frac{s^{(s)}}{2}. \tag{5}$$

By Lemma 2,

$$d^A(\boldsymbol{A}, \boldsymbol{B}) = n - N - c_1 - c_2 - \sum_{i\geqslant 3} c_i + s + f. \tag{6}$$

From Theorem 1, we have

$$d(\boldsymbol{A}, \boldsymbol{B}) \geqslant n - N - c_1^* - c_2^* - \sum_{i\geqslant 3} c_i^* + s^* + f^o. \tag{7}$$

Let $\Delta = \frac{7}{4}d(\boldsymbol{A}, \boldsymbol{B}) - d^A(\boldsymbol{A}, \boldsymbol{B})$. Since $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ and $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ contain all possible 1-cycles, $c_1 = c_1^*$. (See the definition of $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ and Step 1 of the cycle decomposition algorithm.) Since a cycle decomposition of $G(\boldsymbol{A}, \boldsymbol{B})$ contains at most $|M|$ 2-cycles, let $c_2^* = |M| - \alpha$ ($\alpha \geqslant 0$). From (7) and (6), we have

$$\begin{aligned} \Delta &= \frac{7}{4}d(\boldsymbol{A}, \boldsymbol{B}) - d^A(\boldsymbol{A}, \boldsymbol{B}) \\ &\geqslant \frac{7}{4}\left(n - N - c_1^* - c_2^* - \sum_{i\geqslant 3} c_i^* + s^* + f^o\right) - \left(n - N - c_1 - \sum_{i\geqslant 2} c_i + s + f\right) \\ &= \frac{3}{4}\left(n - N - c_1^*\right) - \frac{5}{4}c_2^* - \frac{|M| - \alpha}{2} - \frac{7}{4}\sum_{i\geqslant 3} c_i^* + \frac{7}{4}s^* + \frac{7}{4}f^o + \sum_{i\geqslant 2} c_i - s - f. \end{aligned} \tag{8}$$

From (8) and (5), we have

$$\begin{aligned} \Delta \geqslant{}& \frac{1}{4}\left(n - N - c_1^* - c_2^* - \sum_{i\geqslant 3} c_i^*\right) + \frac{1}{2}\left(n - N - c_1^* - 2c_2^* - 3\sum_{i\geqslant 3} c_i^*\right) - \frac{|M| - \alpha}{2} \\ &+ \frac{7}{4}s^* + \frac{7}{4}f^o + \sum_{i\geqslant 2} c_i^{(o)} + \frac{|M|}{2} + \frac{z^{(o)}}{2} - \frac{s^{(s)}}{2} - f. \end{aligned} \tag{9}$$

Since there are $n-N$ black edges in $G_s^*(\boldsymbol{A},\boldsymbol{B})$ and each black edge is in a cycle, we have $n-N=\sum_{i\geqslant 1} ic_i^*$. That is,

$$n-N-c_1^*-c_2^*-\sum_{i\geqslant 3}c_i^*=\sum_{i\geqslant 2}(i-1)c_i^*. \tag{10}$$

From (9) and (10), we can immediately obtain

$$\Delta\geqslant\frac{1}{4}\left(\sum_{i\geqslant 2}(i-1)c_i^*+2s^*-2s^{(s)}\right)+\frac{1}{2}\sum_{i\geqslant 4}(i-3)c_i^*+\frac{5}{4}s^*+\frac{\alpha}{2}+\frac{7}{4}f^o+\sum_{i\geqslant 2}c_i^{(o)}+\frac{z^{(o)}}{2}-f. \tag{11}$$

From Theorem 2, $\sum_{i\geqslant 2}(i-1)c_i^*\geqslant 2(s_c-s_e^*)$. Moreover, by definitions, $s_c\geqslant s^{(s)}$ and $s^*\geqslant s_e^*$. Thus, we have

$$\sum_{i\geqslant 2}(i-1)c_i^*+2s^*-2s^{(s)}\geqslant 0. \tag{12}$$

From (12), (11) becomes

$$\Delta\geqslant\frac{1}{2}\sum_{i\geqslant 4}(i-3)c_i^*+\frac{5}{4}s^*+\frac{\alpha}{2}+\frac{7}{4}f^o+\sum_{i\geqslant 2}c_i^{(o)}+\frac{z^{(o)}}{2}-f. \tag{13}$$

From the fact that all variables in (13) are non-negative, we can immediately conclude that $\Delta\geqslant 0$ when $f=0$. □

4.5. The performance ratio when $f=1$ or 2

From formula (13), we have $\Delta\geqslant 0$ when $s^*\geqslant 2$. To ensure the 1.75 performance ratio for $f=1$ or 2, we focus on the cases where $s^*=0$ or $s^*=1$.

A *spanning* edge is a grey edge whose vertices are on two chromosomes.

Lemma 7. *If $G_s^*(\boldsymbol{A},\boldsymbol{B})$ has no minSP ($s^*=0$) and $G_s^A(\boldsymbol{A},\boldsymbol{B})$ has at least one minSP, then $G(\boldsymbol{A},\boldsymbol{B})$ has at least two spanning edges.*

Proof. If $G(\boldsymbol{A},\boldsymbol{B})$ does not have any spanning edge, then $G_s^*(\boldsymbol{A},\boldsymbol{B})$ has at least one *minSP* assuming $\boldsymbol{A}$ and $\boldsymbol{B}$ are not identical. This contradicts the assumption that $G_s^*(\boldsymbol{A},\boldsymbol{B})$ has no *minSP*. Thus, there is at least one spanning edge in $G(\boldsymbol{A},\boldsymbol{B})$. Since every spanning edge must be in a cycle for any cycle decomposition, there are in fact at least two spanning edges. □

Lemma 8. *If there are spanning edges and at least one minSP in $G_s^A(\boldsymbol{A},\boldsymbol{B})$, then $G_s^A(\boldsymbol{A},\boldsymbol{B})$ has at least one arbitrary cycle outside all SP's or at least one isolated 2-cycle outside all SP's.*

Proof. Consider the cycle C containing the spanning edges in $G_s^A(\boldsymbol{A},\boldsymbol{B})$. C can be an arbitrary cycle, a selected related 2-cycle or an isolated 2-cycle. Since C contains spanning edges, C must be outside all *SP*'s.

If C is a selected related 2-cycle, then it must be in a related component U of type (c) or (d) as in Fig. 2. Assume C shares a grey edge with a related 2-cycle C' in U, then by the cycle decomposition algorithm, the two black edges of C' are in arbitrary cycle(s). Consider the two black edges in C'. Since the two ends of each of the black edges are incident to spanning edges (see Fig. 2(c) and (d)), the corresponding genes are neighbors with genes in the other chromosome. By the definition of *SP*, the black edges of C' cannot be in a *SP* in $G_s^A(\boldsymbol{A},\boldsymbol{B})$. This implies that at least one arbitrary cycle is outside all *SP*'s in $G_s^A(\boldsymbol{A},\boldsymbol{B})$. □

Lemma 9. *Consider the case that $G_s^*(\boldsymbol{A},\boldsymbol{B})$ has no minSP ($s^*=0$) and $G_s^A(\boldsymbol{A},\boldsymbol{B})$ has at least one minSP ($s\geqslant 1$). If there is a unique long cycle as a 2-cycle C outside all minSP's in $G_s^A(\boldsymbol{A},\boldsymbol{B})$, then $G_s^*(\boldsymbol{A},\boldsymbol{B})$ has at least one i-cycle for $i\geqslant 4$.*

Proof. By Lemma 7, there are at least two spanning edges in both $G_s^*(\boldsymbol{A},\boldsymbol{B})$ and $G_s^A(\boldsymbol{A},\boldsymbol{B})$. By the assumption, C contains the only two spanning edges in $G_s^A(\boldsymbol{A},\boldsymbol{B})$. Since $G_s^*(\boldsymbol{A},\boldsymbol{B})$ has no *minSP*, C does not exist in $G_s^*(\boldsymbol{A},\boldsymbol{B})$.

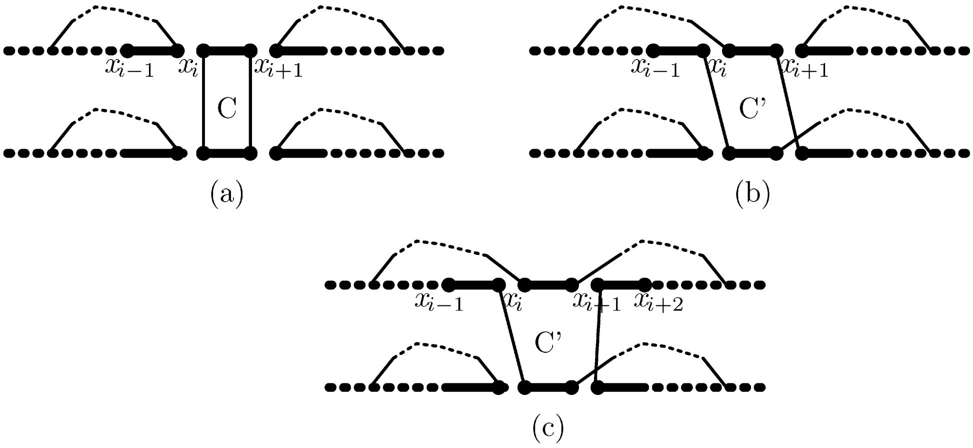

Fig. 7. (a) Only one long cycle as a 2-cycle is outside all *minSP*'s in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$. (b) $G_s^*(\boldsymbol{A}, \boldsymbol{B})$, where the sign of x_i is changed such that there are at least two grey edges in a path connecting $l(x_i)$ and $l(x_{i+1})$, since x_i and x_{i+1} are not neighbors in $\boldsymbol{B}$. (c) $G_s^*(\boldsymbol{A}, \boldsymbol{B})$, where the sign of x_i is changed such that there are at least two grey edges in a path connecting $r(x_{i-1})$ and $l(x_{i+2})$.

Assume C contains the black edge $(r(x_i), l(x_{i+1}))$ in chromosome 1 (see Fig. 7(a)) and the sign of x_i is changed in $G_s^*(\boldsymbol{A}, \boldsymbol{B})$.

In $G_s^*(\boldsymbol{A}, \boldsymbol{B})$, there is a cycle C' containing the vertex $l(x_i)$ and the only two spanning edges. (See Fig. 7(b) and (c). Since there are only two spanning edge in the graph, they must be in the same cycle.) There are two cases.

Case 1. The vertex $l(x_{i+1})$ is in cycle C'. (See Fig. 7(b).) The cycle decomposition algorithm ensures that there is no grey edge connects x_i and x_{i-1} in $G(\boldsymbol{A}, \boldsymbol{B})$. (All possible 1-cycles are kept in both $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ and $G_s^A(\boldsymbol{A}, \boldsymbol{B})$.) Thus, in cycle C', there are at least another two grey edges (other than the two spanning edges) in the cycle C' connecting the two vertices $r(x_{i-1})$ and $r(x_i)$ (on chromosome 1). Therefore, there are at least four grey edges in cycle C'.

Case 2. The vertex $r(x_{i+1})$ is in cycle C'. (See Fig. 7(c).) In this case, the grey edge $(r(x_{i-1}), l(x_{i+2}))$ does not exist in $G_s^*(\boldsymbol{A}, \boldsymbol{B})$. (*Otherwise, there is a long cycle C'' containing a grey edge $(f(x_{i-1}), f(x_{i+2}))$, where $f(x_{i-1}) \in \{l(x_{i-1}), r(x_{i-1})\}$ and $f(x_{i+2}) \in \{l(x_{i+2}), r(x_{i+2})\}$, in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$. Since the 2-cycle C containing $r(x_i)$ and $l(x_{i+1})$ is outside all minSP's and the grey edge $(f(x_{i-1}), f(x_{i+2}))$ has one end on the left of $r(x_i)$ and one end on the right of $l(x_{i+1})$, the long cycle C'' is also outside all minSP's. This contradicts the assumption that C is the only long cycle outside all minSP's in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$.*) Thus, there are at least another two grey edges (other than the two spanning edges) in the cycle C' connecting the two vertices $r(x_{i-1})$ and $l(x_{i+2})$ (on chromosome 1). Therefore, there are at least four grey edges in cycle C'. □

Lemma 10. *Suppose $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ has no minSP and $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ has an even-isolation I. If I contains a long cycle C outside all minSP's, then I contains at least one arbitrary or isolated cycle outside all minSP's.*

Proof. C can be an arbitrary cycle, an isolated 2-cycle, or a selected related 2-cycle. For the first two cases, the lemma holds immediately. If C is a selected related 2-cycle, C is in a related component U of type (a) or (b) (see Fig. 2). Assume C shares a grey edge e with a related cycle C' in U, then the two black edges of C' are in one or two arbitrary cycle(s).

(1). U is of type (a). If C' is outside C, then the arbitrary cycle(s) containing one or two of the black edges of C' cannot be in any *minSP* (due to the two grey edges of C'). By Lemma 1, at least one of the black edges of C' is in I (the two ends of e cannot be the two ending genes of the *SP I*).

Consider the case where C' is inside C. There are possibly *minSP*'s inside C. Let e' be the other grey edge of C'. The two genes x_i and x_j at the ends of edge e are neighbors in $\boldsymbol{B}$. By Lemma 1, x_i and x_j cannot be the two ending genes in the same *minSP* inside C at the same time. Now, we want to show that at least one of the black edge $(r(x_i), l(x_{i+1}))$ or $(r(x_{j-1}), l(x_j))$ is not in any *minSP*.

Without loss of generality, assume that $r(x_i)$ (or equivalently edge $(r(x_i), l(x_{i+1}))$) is in a *minSP* I_1 (x_i as an ending gene) inside C. In this case, $l(x_{j-1})$ will be the other end of I_1 since the existence of the black edge $(r(x_i), l(x_{i+1}))$ and the grey edge $e' = (x_{i+1}, x_{j-1})$. If this is true, the black edge $(r(x_{j-1}), l(x_j))$ cannot be in another *minSP*. (*A minSP* *contains at least two black edges. The vertex $r(x_{j-1})$ is not in I_1. Since C is outside all minSP's, the vertex $r(x_j)$ is also outside all minSP's.*) Thus, the arbitrary cycle containing the black edge $(r(x_{j-1}), l(x_j))$ is not in any *minSP*.

(2) U is of type (b). Let e' be the other grey edge of C'. Since e' and e are crossing with each other, the arbitrary cycle containing black edge(s) of C' cannot be in any *minSP*. (Otherwise, C is also in the same *minSP*.) □

Lemma 11. *Suppose $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ has an even-isolation I and $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ has no minSP. Let $s^{(s)}$ denote the number of S-MSP's in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ containing only one isolated 2-cycles and $|M| - \alpha$ the number of 2-cycles in $G_s^*(\boldsymbol{A}, \boldsymbol{B})$.*

(1) *If any long cycle of I is in a minSP, then $\alpha \geqslant s^{(s)}$;*
(2) *if a 2-cycle in I is outside all minSP's and the rest of long cycles in I are in minSP's, then $\alpha \geqslant s^{(s)} - 1$.*

Proof. Consider a S-MSP $I_s = r(x_i), l(x_{i+1}), r(x_{i+1}), \ldots, l(x_{j-1}), r(x_{j-1}), l(x_j)$ in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$. (See Fig. 4(a).)

If any long cycle of I is in a *minSP*, then the grey edges $(l(x_i), l(x_{j+1}))$ and $(r(x_j), r(x_{i-1}))$ do not exist in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$. (*Otherwise, the minSP I_s is inside the long cycle C containing the two black edges $(r(x_{i-1}), l(x_i))$ and $(r(x_j), l(x_{j+1}))$ and one of the grey edges $(l(x_i), l(x_{j+1}))$ or $(r(x_j), r(x_{i-1}))$ in I. (See Fig.* 4(a).) *Thus, C is not in any minSP of I. This contradicts the assumption that any long cycle in I is in a minSP.*) For the same reason, the grey edges $(l(x_i), r(x_{j+1}))$ and $(r(x_j), l(x_{i-1}))$ do not exist in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$. That is, the grey edges (x_i, x_{j+1}) and (x_{i-1}, x_j) are not in $G(\boldsymbol{A}, \boldsymbol{B})$.

After removing the S-MSP I_s in $G_s^*(\boldsymbol{A}, \boldsymbol{B})$, the two new cycles are created, one containing the three edges $(r(x_i), l(x_{i+1}))$, $(l(x_{i+1}), r(x_j))$, and $(r(x_j), l(x_{j+1}))$ and the other containing the three edges $(r(x_{i-1}), l(x_i))$, $(l(x_i), r(x_{j-1}))$, and $(r(x_{j-1}), l(x_j))$ (see Fig. 4(b) and Fig. 6(b)) in $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ are not 2-cycles. Thus, we can conclude that for each S-MSP in I, $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ does not have any 2-cycle containing black edges $(r(x_i), l(x_{i+1}))$ or $(r(x_{j-1}), l(x_j))$ in the maximum match M. Note that, by the construction of M, M has a 2-cycle containing at least one of the black edges $(r(x_i), l(x_{i+1}))$ and $(r(x_{j-1}), l(x_j))$. Moreover, $|M| - \alpha$ is the number of 2-cycles in $G_s^*(\boldsymbol{A}, \boldsymbol{B})$. Thus, $\alpha = |M| - (|M| - \alpha) \geqslant s^{(s)}$.

For the same reason, we can show that (2) if a 2-cycle in I is outside all *minSP*'s and the rest of long cycles in I are in *minSP*'s, then $\alpha \geqslant s^{(s)} - 1$. □

Let $c_i^{(oo)}$ be the number of arbitrary i-cycles ($i \geqslant 2$) outside all *SP*'s in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ and $z^{(oo)}$ be the number of isolated 2-cycles outside all *SP*'s in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$.

Lemma 12. *If $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ contains some spanning edges and at least one minSP, then*

$$\frac{1}{2}\sum_{i\geqslant 4}(i-3)c_i^* + \sum_{i\geqslant 2} c_i^{(oo)} + \frac{z^{(oo)}}{2} \geqslant 1. \tag{14}$$

Proof. If $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ contains some spanning edges, by Lemma 8, $\sum_{i\geqslant 2} c_i^{(oo)} + z^{(oo)} \geqslant 1$. If $c_i^{(oo)} \geqslant 1$ or $z^{(oo)} \geqslant 2$, then the lemma holds immediately.

Now consider the case where $c_i^{(oo)} = 0$ and $z^{(oo)} = 1$. That is, there is a unique long cycle as a 2-cycle outside all *SP*'s in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$. By Lemma 9, $\sum_{i\geqslant 4}(i-3)c_i^* \geqslant 1$. Thus we have (14) holds. □

Theorem 4. $d^A(\boldsymbol{A}, \boldsymbol{B}) \leqslant \frac{7}{4} d(\boldsymbol{A}, \boldsymbol{B})$.

Proof. We continue with the proof of Theorem 3. By formula (13), $\Delta \geqslant 0$ holds when $f = 0$ or ($f = 1$ and $s^* \geqslant 1$) or ($f = 2$ and $s^* \geqslant 2$). Thus, we only have to consider the following cases.

Case 1. $s^* = 0$ and $f = 1$. By Lemma 7, $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ contains some spanning edges. By Lemma 12, (14) holds. From (13) and (14), we know that $\Delta \geqslant 0$.

Case 2. $s^* = 1$ and $f = 2$. If $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ contains no spanning edge, then all chromosomes except the one with the even isolation contain only 1-cycles. From ModificationMethod, $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ is derived from an optimal cycle decomposition of $G(\boldsymbol{A}, \boldsymbol{B})$ with $f^0 \geqslant 1$. By formula (13), we have $\Delta \geqslant \frac{5}{4}s^* + f^0 - f \geqslant 0$. If $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ contains some spanning edges, by formulas (13) and (14), $\Delta \geqslant 0$ holds.

Case 3. $s^* = 0$ and $f = 2$. By Lemma 7, $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ contains some spanning edges. Since there are at least two *minSP*'s in the even isolation I, there are at least two long cycles in *minSP*'s of I, at least one for each *minSP*. Besides, there is at least one long cycle containing spanning edges outside I. Therefore, there are at least six black edges in long cycles of $G_s^A(\boldsymbol{A}, \boldsymbol{B})$, and they are also in long cycles of $G_s^*(\boldsymbol{A}, \boldsymbol{B})$. Now, we want to show that

$$\sum_{i\geqslant 2}(i-1)c_i^* \geqslant 4. \tag{15}$$

Assume that $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ contains exactly six black edges in long cycles. Thus, there is a unique long cycle as a 2-cycle outside I. By Lemma 9, there is at least one i-cycle ($i \geqslant 4$) in $G_s^*(\boldsymbol{A}, \boldsymbol{B})$. Thus, $\sum_{i\geqslant 2}(i-1)c_i^* \geqslant 3$. Considering the other four black edges in long cycles, we have (15) holds.

If $G_s^A(\boldsymbol{A}, \boldsymbol{B})$ contains more than six black edges in long cycles, then $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ also contains more than six black edges in long cycles.

$$\sum_{i\geqslant 2}(i-1)c_i^* = \sum_{i\geqslant 2} i \times c_i^* - \sum_{i\geqslant 2} c_i^*, \tag{16}$$

where $\sum_{i\geqslant 2} i \times c_i^*$ is the total number of black edges in long cycles in $G_s^*(\boldsymbol{A}, \boldsymbol{B})$ and $\sum_{i\geqslant 2} c_i^*$ is the total number of long cycles in $G_s^*(\boldsymbol{A}, \boldsymbol{B})$. Let $k \geqslant 3$ be an integer. If there are $2k+1$ black edges in long cycles, then the number of long cycles is at most k. In this case, we have

$$\sum_{i\geqslant 2}(i-1)c_i^* = \sum_{i\geqslant 2} i \times c_i^* - \sum_{i\geqslant 2} c_i^* \geqslant 2k+1-k = k+1 \geqslant 4. \tag{17}$$

If there are $2k+2$ black edges in long cycles, then the number of long cycles is at most $k+1$. In this case, we have

$$\sum_{i\geqslant 2}(i-1)c_i^* = \sum_{i\geqslant 2} i \times c_i^* - \sum_{i\geqslant 2} c_i^* \geqslant 2k+2-(k+1) = k+1 \geqslant 4. \tag{18}$$

Therefore, we can conclude that (15) holds. From (15), formula (11) can be transformed into

$$\Delta \geqslant \frac{\alpha - s^{(s)}}{2} + \frac{1}{2}\sum_{i\geqslant 4}(i-3)c_i^* + \sum_{i\geqslant 2} c_i^{(o)} + \frac{z^{(o)}}{2} - 1. \tag{19}$$

Let I denote the even-isolation in $G_s^A(\boldsymbol{A}, \boldsymbol{B})$. There are three subcases:

Case 3.1. All long cycles of I are in *minSP*'s. By Lemma 11, $\alpha \geqslant s^{(s)}$. By formulas (14) and (19), $\Delta \geqslant 0$.

Case 3.2. All the long cycles of I except one 2-cycle are in *minSP*'s. By Lemma 11, $\alpha \geqslant s^{(s)} - 1$. By Lemma 10, I contains at least one isolated cycle or at least one arbitrary cycle outside all *minSP*'s. Note that the above mentioned isolated cycle or arbitrary cycle is inside the *SP* I. Therefore, (14) becomes $\frac{1}{2}\sum_{i\geqslant 4}(i-3)c_i^* + \sum_{i\geqslant 2} c_i^{(o)} + \frac{z^{(o)}}{2} \geqslant \frac{3}{2}$. By formula (19), $\Delta \geqslant 0$.

Case 3.3. In I, at least one i-cycle ($i \geqslant 3$) or at least two 2-cycles of I are outside all *minSPs*. By Lemma 10, I contains at least two isolated cycles or at least one arbitrary cycle outside all *minSP*'s. Again, the above mentions cycles are inside the *SP* I. Thus, (14) becomes $\frac{1}{2}\sum_{i\geqslant 4}(i-3)c_i^* + \sum_{i\geqslant 2} c_i^{(o)} + \frac{z^{(o)}}{2} \geqslant 2$. By formula (13), $\Delta \geqslant 0$. □

Remarks. We have designed an 1.75-approximation algorithm for unsigned translocation distance. It is interesting to give an algorithm with better ratio since the gap between 1.75 and the lower bound 1.00017 [4] is still big.

Acknowledgments

Yun Cui and Lusheng Wang are fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 1196/03E]. Daming Zhu is fully supported by NSFC 60573024.

References

[1] J. Kececioglu, R. Ravi, Of mice and men: Algorithms for evolutionary distances between genomes with translocation, in: 6th ACM–SIAM Symposium on Discrete Algorithms, 1995, pp. 604–613.

[2] Sridhar Hannenhalli, Polynomial-time algorithm for computing translocation distance between genomes, in: CPM '95, 1995, pp. 162–176.

[3] Lusheng Wang, Daming Zhu, Xiaowen Liu, Shaohan Ma, An $O(n^2)$ algorithm for signed translocation, J. Comput. System Sci. 70 (2005) 284–299.

[4] Daming Zhu, Lusheng Wang, On the complexity of unsigned translocation distance, Theoret. Comput. Sci. 352 (2006) 322–328.

[5] Anne Bergeron, Julia Mixtacki, Jens Stoye, On sorting by translocation, in: RECOMB '05, 2005, pp. 615–629.

[6] S. Hannenhalli, P.A. Pevzner, Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals), in: STOC '95, 1995, pp. 178–189.

[7] H. Kaplan, R. Shamir, R.E. Tarjan, Faster and simpler algorithm for sorting signed permutations by reversals, SIAM J. Comput. 29 (3) (2000) 880–892.

[8] V. Bafna, P. Pevzner, Genome rearrangements and sorting by reversals, SIAM J. Comput. 25 (2) (1996) 272–289.

[9] D. Sankoff, J.H. Nadeau (Eds.), Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment and the Evolution of Gene Families, Series in Computational Biology, vol. 1, Kluwer Academic Press, Dordrecht, NL, 2000, pp. 225–241.

[10] D. Sankoff, N. El-Mabrouk, Genome rearrangement, in: T. Jiang, Y. Xu, Q. Zhang (Eds.), Current Topics in Computational Molecular Biology, MIT Press, 1992, pp. 132–155.

[11] L. Lovász, M.D. Plummer, Matching Theory, Annals of Discrete Mathematics, vol. 29, North-Holland, Amsterdam, 1986.