# On the class of double distance problems

Marília D. V. Braga, Leonie R. Brockmann, Katharina Klerx, and Jens Stoye

Faculty of Technology and CeBiTec, Bielefeld University, Germany

**Abstract.** This work is about comparing two genomes $\mathbb{S}$ and $\mathbb{D}$ over the same set of gene families, such that $\mathbb{S}$ is *singular* (has one gene per family), while $\mathbb{D}$ is *duplicated* (has two genes per family). Considering some underlying model, that can be simply the minimization of *breakpoints* or finding the smallest sequence of mutations mimicked by the *double-cut-and-join* (DCJ) operation, the *double distance* of $\mathbb{S}$ and $\mathbb{D}$ aims to find the smallest distance between $\mathbb{D}$ and any element from the set $2\mathbb{S}$, that contains all possible genome configurations obtained by *doubling* the chromosomes of $\mathbb{S}$. The *breakpoint* double distance of $\mathbb{S}$ and $\mathbb{D}$ can be greedily solved in linear time. In contrast, the DCJ double distance of $\mathbb{S}$ and $\mathbb{D}$ was proven to be NP-hard. The complexity space between these two extremes can be explored with the help of an intermediate family of problems, the $\sigma_k$ distances, defined for each $k \in \{2, 4, 6, ..., \infty\}$, in a way such that the $\sigma_2$ distance equals the breakpoint distance and the $\sigma_\infty$ distance equals the DCJ distance. With this class of problems it is possible to investigate the complexity of the double distance under the $\sigma_k$ distance, increasing the value $k$ in an attempt to identify the smallest value for which the double distance becomes NP-hard, indicating the point in which the complexity changes. In our more recent work we have proven that, for the particular case in which genomes can only be composed of circular chromosomes, both $\sigma_4$ and $\sigma_6$ double distances can be solved in linear time. Here we present a non-trivial extension of these results to genomes including linear chromosomes.

## 1 Introduction

A *genome* is a multiset of *chromosomes* and each chromosome is a sequence of *genes*, where the genes are classified into *families*. Considering that an *adjacency* is the oriented neighborhood between two genes in one chromosome of a genome, a simple *distance* measure between genomes is the *breakpoint* distance, that consists in quantifying their distinct adjacencies [9]. Other distance models rely on large-scale genome *rearrangements*, such as inversions, translocations, fusions and fissions, or the general *double-cut-and-join* (DCJ) operation, yielding distances that correspond to the minimum number of rearrangements required to transform one genome into another [7, 8, 10].

Our study relies on the *breakpoint graph*, a structure that represents the relation between two given genomes [1]. When the two genomes have the same set of $n_*$ genes, their breakpoint graph is a collection of cycles of even length and paths. For even $k$, let $c_k$ and $p_k$ be respectively the numbers of cycles and of paths

of length $k$. The corresponding breakpoint distance is equal to $n_* - \left(c_2 + \frac{p_0}{2}\right)$ [9]. Similarly, when the considered rearrangements are those modeled by the *double-cut-and-join* (DCJ) operation [10], the rearrangement distance is $n_* - \left(c + \frac{p_e}{2}\right)$, where $c$ is the total number of cycles and $p_e$ is the total number of even paths [2].

Both breakpoint and DCJ distances are basic constituents of a problem related to the event of a *whole genome duplication* (WGD) [6, 9]. The *double distance* of a *singular* genome $\mathbb{S}$ and a *duplicated* genome $\mathbb{D}$ aims to find the smallest distance between $\mathbb{D}$ and any element from the set $2\mathbb{S}$, that contains all possible genome configurations obtained by *doubling* the chromosomes of $\mathbb{S}$. Interestingly, it can be solved in linear time under the breakpoint distance, but is NP-hard under the DCJ distance [9]. One way of exploring the complexity space between these two extremes is to consider a $\sigma_k$ distance [5], defined to be $n_* - \left(c_2 + c_4 + \ldots + c_k + \frac{p_0+p_2+\ldots+p_{k-2}}{2}\right)$. Note that the $\sigma_2$ distance is the breakpoint distance and the $\sigma_\infty$ distance is the DCJ distance. We can then increasingly investigate the complexities of the double distance under the $\sigma_4$ distance, then under the $\sigma_6$ distance, and so on, in an attempt to identify the smallest value for which the double distance becomes NP-hard.

In our recent work, we succeeded in devising efficient algorithms for $\sigma_4$ and $\sigma_6$ if the input genomes are exclusively composed of circular chromosomes [3]. Here we close the gaps of these results by giving a solution for the double distance under $\sigma_4$ and $\sigma_6$ even if the input genomes include linear chromosomes. This work has an extended version containing detailed proofs and extra figures [4].

## 2 Background

A *chromosome* can be either linear or circular and is represented by its sequence of genes, where each *gene* is an oriented DNA fragment. We assume that each gene belongs to a *family*, which is a set of homologous genes. A gene that belongs to a family $\mathtt{X}$ is represented by the symbol $\mathtt{X}$ itself if it is read in forward orientation or by the symbol $\overline{\mathtt{X}}$ if it is read in reverse orientation. For example, the sequences $[\mathtt{1}\,\overline{\mathtt{3}}\,\mathtt{2}]$ and $(4)$ represent, respectively, a linear (flanked by square brackets) and a circular chromosome (flanked by parentheses), both shown in Fig. 1, the first composed of three genes and the second composed of a single gene. Note that if a sequence $s$ represents a chromosome $K$, then $K$ can be equally represented by the *reverse complement* of $s$, denoted by $\overline{s}$, obtained by reversing the order and the orientation of the genes in $s$. Moreover, if $K$ is circular, it can be equally represented by any circular rotation of $s$ and $\overline{s}$. Recall that a gene is an *occurrence* of a family, therefore distinct genes from the same family are represented by the same symbol.

We can also represent a gene from family $\mathtt{X}$ referring to its *extremities* $\mathtt{X}^h$ (head) and $\mathtt{X}^t$ (tail). The *adjacencies* in a chromosome are the neighboring extremities of distinct genes. The remaining extremities, that are at the ends of linear chromosomes, are *telomeres*. In linear chromosome $[\mathtt{1}\,\overline{\mathtt{3}}\,\mathtt{2}]$, the adjacencies are $\{\mathtt{1}^h\mathtt{3}^h, \mathtt{3}^t\mathtt{2}^t\}$ and the telomeres are $\{\mathtt{1}^t, \mathtt{2}^h\}$. Note that an adjacency has no orientation, that is, an adjacency between extremities $\mathtt{1}^h$ and $\mathtt{3}^h$ can be equally

represented by $\mathtt{1}^h\mathtt{3}^h$ and by $\mathtt{3}^h\mathtt{1}^h$. In the particular case of a single-gene circular chromosome, e.g. $(\mathtt{4})$, an adjacency exceptionally occurs between the extremities of the same gene (here $\mathtt{4}^h\mathtt{4}^t$).

A *genome* is then a multiset of chromosomes and we denote by $\mathcal{F}(\mathbb{G})$ the set of gene families that occur in genome $\mathbb{G}$. In addition, we denote by $\mathcal{A}(\mathbb{G})$ the multiset of adjacencies and by $\mathcal{T}(\mathbb{G})$ the multiset of telomeres that occur in $\mathbb{G}$. A genome $\mathbb{S}$ is called *singular* if each gene family occurs exactly once in $\mathbb{S}$. Similarly, a genome $\mathbb{D}$ is called *duplicated* if each gene family occurs exactly twice in $\mathbb{D}$. The two occurrences of a family in a duplicated genome are called *paralogs*. A *doubled* genome is a special type of duplicated genome in which each adjacency or telomere occurs exactly twice. These two copies of the same adjacency (respectively same telomere) in a doubled genome are called *paralogous adjacencies* (respectively *paralogous telomeres*). Observe that distinct doubled genomes can have exactly the same adjacencies and telomeres, as we show in Fig. 1, where we also give an example of a singular genome.

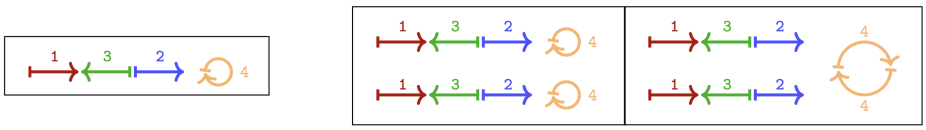


**Fig. 1.** On the left we show the singular genome $\mathbb{S} = \{[\mathtt{1}\overline{\mathtt{3}}\mathtt{2}]\,(\mathtt{4})\}$ and on the right the two doubled genomes $\{[\mathtt{1}\overline{\mathtt{3}}\mathtt{2}]\,[\mathtt{1}\overline{\mathtt{3}}\mathtt{2}]\,(\mathtt{4})\,(\mathtt{4})\}$ and $\{[\mathtt{1}\overline{\mathtt{3}}\mathtt{2}]\,[\mathtt{1}\overline{\mathtt{3}}\mathtt{2}]\,(\mathtt{44})\}$.

### 2.1 Comparing canonical genomes

Two genomes $\mathbb{S}_1$ and $\mathbb{S}_2$ are said to be a *canonical pair* when they are singular and have the same gene families, that is, $\mathcal{F}(\mathbb{S}_1) = \mathcal{F}(\mathbb{S}_2)$. Denote by $\mathcal{F}_*$ the set of families occurring in canonical genomes $\mathbb{S}_1$ and $\mathbb{S}_2$. For example, genomes $\mathbb{S}_1 = \{(\mathtt{1}\overline{\mathtt{3}}\mathtt{2})\,(\mathtt{4})\}$ and $\mathbb{S}_2 = \{(\mathtt{1}\,\mathtt{2})\,(\mathtt{3}\overline{\mathtt{4}})\}$ are canonical with $\mathcal{F}_* = \{\mathtt{1}, \mathtt{2}, \mathtt{3}, \mathtt{4}\}$.

**Breakpoint graph.** A multigraph representing the adjacencies of $\mathbb{S}_1$ and $\mathbb{S}_2$ is the *breakpoint graph* $BG(\mathbb{S}_1, \mathbb{S}_2) = (V, E)$ [1]. The vertex set $V$ comprises, for each family $\mathtt{X}$ in $\mathcal{F}_*$, one vertex for $\mathtt{X}^h$ and one vertex for $\mathtt{X}^t$. The edge multiset $E$ represents the adjacencies: for each adjacency in $\mathbb{S}_1$ (respectively $\mathbb{S}_2$) there exists one $\mathbb{S}_1$-edge (respectively $\mathbb{S}_2$-edge) in $E$ linking its two extremities.

The degree of each vertex can be 0, 1 or 2 and each connected *component* alternates between $\mathbb{S}_1$- and $\mathbb{S}_2$-edges. As a consequence, the components of $BG(\mathbb{S}_1, \mathbb{S}_2)$ can be cycles of even length or paths. An even path has one endpoint in $\mathbb{S}_1$ ($\mathbb{S}_1$-*telomere*) and the other in $\mathbb{S}_2$ ($\mathbb{S}_2$-*telomere*), while an odd path has either both endpoints in $\mathbb{S}_1$ or both endpoints in $\mathbb{S}_2$. A vertex that is not a telomere in $\mathbb{S}_1$ nor in $\mathbb{S}_2$ is said to be *non-telomeric*. In the breakpoint graph a non-telomeric vertex has degree 2. We call $i$-*cycle* a cycle of length $i$ and $j$-*path*

a path of length $j$. We also denote by $c_i$ the number of cycles of length $i$, by $p_j$ the number of paths of length $j$, by $c$ the total number of cycles and by $p_{\mathrm{e}}$ the total number of even paths. An example is given in Fig. 2(a).

**Breakpoint distance.** For canonical genomes $\mathbb{S}_1$ and $\mathbb{S}_2$ the *breakpoint distance*, denoted by $\mathrm{d}_{\mathrm{BP}}$, is defined as follows [9]:

$$\mathrm{d}_{\mathrm{BP}}(\mathbb{S}_1, \mathbb{S}_2) = n_* - \left( |\mathcal{A}(\mathbb{S}_1) \cap \mathcal{A}(\mathbb{S}_2)| + \frac{|\mathcal{T}(\mathbb{S}_1) \cap \mathcal{T}(\mathbb{S}_2)|}{2} \right), \quad \text{where } n_* = |\mathcal{F}_*|.$$

If $\mathbb{S}_1 = \{(1\overline{3}2)[4]\}$ and $\mathbb{S}_2 = \{(12)[3\overline{4}]\}$, then $\mathcal{A}(\mathbb{S}_1) \cap \mathcal{A}(\mathbb{S}_2) = \{1^t 2^h\}$, $\mathcal{T}(\mathbb{S}_1) \cap \mathcal{T}(\mathbb{S}_2) = \{4^t\}$ and $n_* = 4$, giving $\mathrm{d}_{\mathrm{BP}}(\mathbb{S}_1, \mathbb{S}_2) = 2.5$. Since a common adjacency corresponds to a 2-cycle and a common telomere corresponds to a 0-path in $BG(\mathbb{S}_1, \mathbb{S}_2)$, the breakpoint distance can be rewritten as

$$\mathrm{d}_{\mathrm{BP}}(\mathbb{S}_1, \mathbb{S}_2) = n_* - \left( c_2 + \frac{p_0}{2} \right).$$

**DCJ distance.** A *double cut and join* (DCJ) is the operation that breaks two adjacencies or telomeres[1] of a genome and rejoins the open ends in a different way [10]. A DCJ models several rearrangements, such as inversion, translocation, fission and fusion. The minimum number of DCJs that transform one genome into the other is their *DCJ distance* $\mathrm{d}_{\mathrm{DCJ}}$, that can be derived from $BG(\mathbb{S}_1, \mathbb{S}_2)$ [2]:

$$\mathrm{d}_{\mathrm{DCJ}}(\mathbb{S}_1, \mathbb{S}_2) = n_* - \left( c + \frac{p_{\mathrm{e}}}{2} \right) = n_* - \left( c_2 + c_4 + \ldots + c_\infty + \frac{p_0 + p_2 + \ldots + p_\infty}{2} \right).$$

If $\mathbb{S}_1 = \{(1\overline{3}2)[4]\}$ and $\mathbb{S}_2 = \{(12)[3\overline{4}]\}$, then $n_* = 4$, $c = 1$ and $p_{\mathrm{e}} = 2$ (see Fig. 2(a)). Consequently, their DCJ distance is $\mathrm{d}_{\mathrm{DCJ}}(\mathbb{S}_1, \mathbb{S}_2) = 2$.

**The class of $\boldsymbol{\sigma_k}$ distances.** Given $BG(\mathbb{S}_1, \mathbb{S}_2)$, for $k \in \{2, 4, 6, \ldots, \infty\}$ we denote by $\sigma_k$ the cumulative sums $\sigma_k = c_2 + c_4 + \ldots + c_k + \frac{p_0 + p_2 + \ldots + p_{k-2}}{2}$. Then the $\sigma_k$ distance of $\mathbb{S}_1$ and $\mathbb{S}_2$ is defined to be [5]:

$$\mathrm{d}_{\sigma_k}(\mathbb{S}_1, \mathbb{S}_2) = n_* - \sigma_k.$$

It is easy to see that the $\sigma_2$ distance equals the breakpoint distance and that the $\sigma_\infty$ distance equals the DCJ distance.

### 2.2 Comparing a singular and a duplicated genome

Let $\mathbb{S}$ be a singular and $\mathbb{D}$ be a duplicated genome over the same gene families, that is, $\mathcal{F}(\mathbb{S}) = \mathcal{F}(\mathbb{D})$. The number of genes in $\mathbb{D}$ is twice the number of genes in $\mathbb{S}$ and we need to somehow equalize the contents of these genomes, before searching

[1] A broken adjacency has two open ends and a broken telomere has a single one.

for common adjacencies and common telomeres of $\mathbb{S}$ and $\mathbb{D}$ or transforming one genome into the other with DCJ operations. This can be done by *doubling* $\mathbb{S}$, with a rearrangement operation mimicking a *whole genome duplication*: it simply consists of doubling each adjacency and each telomere of $\mathbb{S}$. However, when $\mathbb{S}$ has one or more circular chromosomes, it is not possible to find a unique layout of its chromosomes after the doubling: indeed, each circular chromosome can be doubled into two identical circular chromosomes, or the two copies are concatenated to each other in a single circular chromosome. Therefore, in general the doubling of a genome $\mathbb{S}$ results in a set of doubled genomes denoted by $2\mathbb{S}$. Note that $|2\mathbb{S}| = 2^r$, where $r$ is the number of circular chromosomes in $\mathbb{S}$. For example, if $\mathbb{S} = \{[1\,\overline{3}\,2]\,(4)\}$, then $2\mathbb{S} = \{\mathbb{B}_1, \mathbb{B}_2\}$ with $\mathbb{B}_1 = \{[1\,\overline{3}\,2]\,[1\,\overline{3}\,2]\,(4)\,(4)\}$ and $\mathbb{B}_2 = \{[1\,\overline{3}\,2]\,[1\,\overline{3}\,2]\,(4\,4)\}$ (see Fig 1). All genomes in $2\mathbb{S}$ have exactly the same multisets of adjacencies and of telomeres, therefore we can use a special notation for these multisets: $\mathcal{A}(2\mathbb{S}) = \mathcal{A}(\mathbb{S}) \cup \mathcal{A}(\mathbb{S})$ and $\mathcal{T}(2\mathbb{S}) = \mathcal{T}(\mathbb{S}) \cup \mathcal{T}(\mathbb{S})$.

Each family in a duplicated genome can be $\binom{\mathtt{a}}{\mathtt{b}}$-*singularized* by adding the index $\mathtt{a}$ to one of its occurrences and the index $\mathtt{b}$ to the other. A duplicated genome can be entirely singularized if each of its families is singularized. Let $\mathfrak{S}^{\mathtt{a}}_{\mathtt{b}}(\mathbb{D})$ be the set of all possible genomes obtained by all distinct ways of $\binom{\mathtt{a}}{\mathtt{b}}$-singularizing the duplicated genome $\mathbb{D}$. Similarly, we denote by $\mathfrak{S}^{\mathtt{a}}_{\mathtt{b}}(2\mathbb{S})$ the set of all possible genomes obtained by all distinct ways of $\binom{\mathtt{a}}{\mathtt{b}}$-singularizing each doubled genome in the set $2\mathbb{S}$.

**The class of $\sigma_k$ double distances.** For $k = 2, 4, 6, \ldots$, each $\sigma_k$ double distance of a singular genome $\mathbb{S}$ and a duplicated genome $\mathbb{D}$ is defined as follows [3, 9]:

$$\mathrm{d}^2_{\sigma_k}(\mathbb{S}, \mathbb{D}) = \mathrm{d}^2_{\sigma_k}(\mathbb{S}, \check{\mathbb{D}}) = \min_{\mathbb{B} \in \mathfrak{S}^{\mathtt{a}}_{\mathtt{b}}(2\mathbb{S})} \{\mathrm{d}_{\sigma_k}(\mathbb{B}, \check{\mathbb{D}})\}, \text{ where } \check{\mathbb{D}} \text{ is any genome in } \mathfrak{S}^{\mathtt{a}}_{\mathtt{b}}(\mathbb{D}).$$

Observe that $\mathrm{d}^2_{\sigma_k}(\mathbb{S}, \check{\mathbb{D}}) = \mathrm{d}^2_{\sigma_k}(\mathbb{S}, \check{\mathbb{D}}')$ for any $\check{\mathbb{D}}, \check{\mathbb{D}}' \in \mathfrak{S}^{\mathtt{a}}_{\mathtt{b}}(\mathbb{D})$.

**$\sigma_2$ (breakpoint) double distance.** The *breakpoint double distance* of $\mathbb{S}$ and $\mathbb{D}$, denoted by $\mathrm{d}^2_{\mathrm{BP}}(\mathbb{S}, \mathbb{D})$, is equivalent to the $\sigma_2$ double distance. The solution here can be found with a greedy algorithm [9]: each adjacency or telomere of $\mathbb{D}$ that occurs in $\mathbb{S}$ can be fulfilled. If an adjacency or telomere that occurs twice in $\mathbb{D}$ also occurs in $\mathbb{S}$, it can be fulfilled twice in any genome from $2\mathbb{S}$. Then,

$$\mathrm{d}^2_{\mathrm{BP}}(\mathbb{S}, \mathbb{D}) = 2n_* - |\mathcal{A}(2\mathbb{S}) \cap \mathcal{A}(\mathbb{D})| - \frac{|\mathcal{T}(2\mathbb{S}) \cap \mathcal{T}(\mathbb{D})|}{2}, \text{ where } n_* = |\mathcal{F}(\mathbb{S})|.$$

**$\sigma_\infty$ (DCJ) double distance.** For the *DCJ double distance* $\mathrm{d}^2_{\mathrm{DCJ}}$, that is equivalent to the $\sigma_\infty$ double distance, the solution space is more complex: computing the DCJ double distance of genomes $\mathbb{S}$ and $\mathbb{D}$ was proven to be NP-hard [9].

## 3 Equivalence of $\sigma_k$ double distance and $\sigma_k$ disambiguation

A nice way of representing the solution space of the $\sigma_k$ double distance is by using a modified version of the breakpoint graph [3, 9].

### 3.1 Ambiguous breakpoint graph

Given a singular genome $\mathbb{S}$ and a duplicated genome $\mathbb{D}$, their *ambiguous breakpoint graph* $ABG(\mathbb{S}, \check{\mathbb{D}}) = (V, E)$ is a multigraph representing the adjacencies of any element in $\mathfrak{S}^{\mathtt{a}}_{\mathtt{b}}(2\mathbb{S})$ and a genome $\check{\mathbb{D}} \in \mathfrak{S}^{\mathtt{a}}_{\mathtt{b}}(\mathbb{D})$. The vertex set $V$ comprises, for each family $\mathtt{X}$ in $\mathcal{F}(\mathbb{S})$, the two pairs of *paralogous vertices* $\mathtt{X}^h_{\mathtt{a}}$, $\mathtt{X}^h_{\mathtt{b}}$ and $\mathtt{X}^t_{\mathtt{a}}$, $\mathtt{X}^t_{\mathtt{b}}$. We can use the notation $\hat{u}$ to refer to the paralogous counterpart of a vertex $u$. For example, if $u = \mathtt{X}^h_{\mathtt{a}}$, then $\hat{u} = \mathtt{X}^h_{\mathtt{b}}$.

The edge set $E$ represents the adjacencies. For each adjacency in $\check{\mathbb{D}}$ there exists one $\check{\mathbb{D}}$-edge in $E$ linking its two extremities. The $\mathbb{S}$-edges represent all adjacencies occurring in all genomes from $\mathfrak{S}^{\mathtt{a}}_{\mathtt{b}}(2\mathbb{S})$: for each adjacency $\gamma\beta$ of $\mathbb{S}$, we have the *pair of paralogous edges* $\mathcal{E}(\gamma\beta) = \{\gamma_{\mathtt{a}}\beta_{\mathtt{a}}, \gamma_{\mathtt{b}}\beta_{\mathtt{b}}\}$ and its *complementary* counterpart $\widetilde{\mathcal{E}}(\gamma\beta) = \{\gamma_{\mathtt{a}}\beta_{\mathtt{b}}, \gamma_{\mathtt{b}}\beta_{\mathtt{a}}\}$. Note that $\widetilde{\widetilde{\mathcal{E}}}(\gamma\beta) = \mathcal{E}(\gamma\beta)$. The *square* of $\gamma\beta$ is then $\mathcal{Q}(\gamma\beta) = \mathcal{E}(\gamma\beta) \cup \widetilde{\mathcal{E}}(\gamma\beta)$. The $\mathbb{S}$-edges in the ambiguous breakpoint graph are therefore the squares of all adjacencies in $\mathbb{S}$. Let $a_*$ be the number of squares in $ABG(\mathbb{S}, \check{\mathbb{D}})$. Obviously we have $a_* = |\mathcal{A}(\mathbb{S})| = n_* - \kappa(\mathbb{S})$, where $\kappa(\mathbb{S})$ is the number of linear chromosomes in $\mathbb{S}$. Again, we can use the notation $\hat{e}$ to refer to the paralogous counterpart of an $\mathbb{S}$-edge $e$. For example, if $e = \gamma_{\mathtt{a}}\beta_{\mathtt{a}}$, then $\hat{e} = \gamma_{\mathtt{b}}\beta_{\mathtt{b}}$. An ambiguous breakpoint graph is shown in Fig. 2(b1).

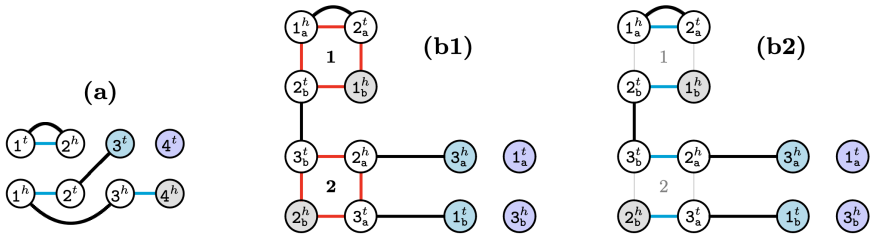


**Fig. 2.** (a) Breakpoint graph of genomes $\mathbb{S}_1 = \{(1\,2)\,[3\,\overline{4}]\}$ and $\mathbb{S}_2 = \{(1\,\overline{3}\,2)\,[4]\}$. Edge types are distinguished by colors: $\mathbb{S}_1$-edges are drawn in blue and $\mathbb{S}_2$-edges are drawn in black. Similarly, vertex types are distinguished by colors: an $\mathbb{S}_1$-telomere is marked in blue, an $\mathbb{S}_2$-telomere is marked in gray, a telomere in both $\mathbb{S}_1$ and $\mathbb{S}_2$ is marked in purple and non-telomeric vertices are white. This graph has one 2-cycle, one 0-path and one 4-path. (b1) Graph $ABG(\mathbb{S}, \check{\mathbb{D}})$ for genomes $\mathbb{S} = \{[1\,2\,3]\}$ and $\check{\mathbb{D}} = \{[1_{\mathtt{a}}\,2_{\mathtt{a}}\,\overline{3}_{\mathtt{a}}\,1_{\mathtt{b}}]\,[\overline{3}_{\mathtt{b}}\,2_{\mathtt{b}}]\}$. Edge types are distinguished by colors: $\check{\mathbb{D}}$-edges are drawn in black and $\mathbb{S}$-edges (squares) are drawn in red. (b2) Induced breakpoint graph $BG(\tau, \check{\mathbb{D}})$ in which all squares are resolved by the solution $\tau = (\{1^h_{\mathtt{a}}2^t_{\mathtt{a}}, 1^h_{\mathtt{b}}2^t_{\mathtt{b}}\}, \{2^h_{\mathtt{a}}3^t_{\mathtt{b}}, 2^h_{\mathtt{b}}3^t_{\mathtt{a}}\}\})$, resulting in one 2-cycle, two 0-paths, one 2-path and one 4-path. This is also the breakpoint graph of $\check{\mathbb{D}}$ and $\mathbb{B} = \{[1_{\mathtt{a}}\,2_{\mathtt{a}}\,3_{\mathtt{b}}], [1_{\mathtt{b}}\,2_{\mathtt{b}}\,3_{\mathtt{a}}]\} \in \mathfrak{S}^{\mathtt{a}}_{\mathtt{b}}(2\mathbb{S})$.

The elements of $2\mathbb{S}$ have the same pairs of identical linear chromosomes. Each pair corresponds to four $\mathbb{S}$-*telomeres*, that are not part of any square. The number of $\mathbb{S}$-telomeres is then $4\kappa(\mathbb{S})$. If $\kappa(\mathbb{D})$ is the number of linear chromosomes in $\mathbb{D}$, the number of $\check{\mathbb{D}}$-*telomeres* is $2\kappa(\mathbb{D})$.

### 3.2 The class of $\sigma_k$ disambiguations

*Resolving* a square $\mathcal{Q}(\cdot) = \mathcal{E}(\cdot) \cup \widetilde{\mathcal{E}}(\cdot)$ corresponds to *choosing* either $\mathcal{E}(\cdot)$ or $\widetilde{\mathcal{E}}(\cdot)$, while the complementary pair is *masked*. If we number the squares of $ABG(\mathbb{S}, \check{\mathbb{D}})$ from 1 to $a_*$, a *solution* can be represented by a tuple $\tau = (\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_{a_*})$, where each $\mathcal{L}_i$ contains the pair of paralogous edges (either $\mathcal{E}_i$ or $\widetilde{\mathcal{E}}_i$) that are chosen (kept) in the graph for square $\mathcal{Q}_i$. The graph *induced* by $\tau$ is a simple breakpoint graph, which we denote by $BG(\tau, \check{\mathbb{D}})$. Fig. 2(b2) shows an example.

Given a solution $\tau$, its $k$-score is the cumulative sum $\sigma_k$ with respect to $BG(\tau, \check{\mathbb{D}})$. The problem of finding a solution $\tau$ for $ABG(\mathbb{S}, \check{\mathbb{D}})$ so that its $k$-score is maximized is called $\sigma_k$ *disambiguation*, that is equivalent to the minimization problem of computing the $\sigma_k$ double distance of $\mathbb{S}$ and $\mathbb{D}$, therefore the complexities of solving the $\sigma_k$ disambiguation and the $\sigma_k$ double distance for any $k \geq 2$ must be the same [9]. As already mentioned, for $\sigma_2$ the double distance/disambiguation can be solved in linear time and for $\sigma_\infty$ the double distance/disambiguation is NP-hard. An optimal solution for the $\sigma_k$ disambiguation of $ABG(\mathbb{S}, \check{\mathbb{D}})$ gives its *$k$-score*, denoted by $\sigma_k(ABG(\mathbb{S}, \check{\mathbb{D}}))$. If $k < k'$, any $k$-cycle contributes for any $k'$-score, therefore $\sigma_k(ABG(\mathbb{S}, \check{\mathbb{D}})) \leq \sigma_{k'}(ABG(\mathbb{S}, \check{\mathbb{D}}))$.

**Approach for solving the $\sigma_k$ disambiguation.** Two $\mathbb{S}$-edges in $ABG(\mathbb{S}, \check{\mathbb{D}})$ are *incompatible* when they belong to the same square and are not paralogous. A cycle or a telomere-to-telomere path in $ABG(\mathbb{S}, \check{\mathbb{D}})$ is *valid* when it does not contain any pair of incompatible edges, that is, when it alternates between $\mathbb{S}$-edges and $\check{\mathbb{D}}$-edges. Any valid cycle or path can be called *piece*. Two distinct pieces in $ABG(\mathbb{S}, \check{\mathbb{D}})$ are either *intersecting*, when they share at least one vertex, or *disjoint*. Finally, a *player* is either a valid cycle whose length is at most $k$ or a valid even path whose length is at most $k-2$. Note that any player is a piece and that any solution $\tau$ of $ABG(\mathbb{S}, \check{\mathbb{D}})$ is composed of disjoint pieces.

Given a solution $\tau = (\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_i \ldots, \mathcal{L}_{a_*})$, the *switching* of its $i$-th element is denoted by $\widetilde{s}(\tau, i)$ and gives $(\mathcal{L}_1, \mathcal{L}_2, \ldots, \widetilde{\mathcal{L}_i} \ldots, \mathcal{L}_{a_*})$. A choice of paralogous edges resolving a given square $\mathcal{Q}_i$ can be *fixed* for any solution, meaning that $\mathcal{Q}_i$ can no longer be switched. In this case, $\mathcal{Q}_i$ is itself said to be *fixed*.

## 4 First steps to solve the $\sigma_k$ disambiguation

In this section we give straightforward extensions of results developed in our previous study for circular genomes [3].

### 4.1 Common adjacencies and telomeres are conserved

If $\tau$ is an optimal solution for the $\sigma_k$ disambiguation and if a player $C \in BG(\tau, \check{\mathbb{D}})$ is disjoint from any player distinct from $C$ in any other optimal solution, then $C$ must be part of all optimal solutions and is itself said to be *optimal*.

**Lemma 1 (extended from [3]).** *For any $\sigma_k$ disambiguation, all existing 0-paths and 2-cycles in $ABG(\mathbb{S}, \mathbb{D})$ are optimal.*

This lemma is a generalization of the $\sigma_2$ disambiguation and guarantees that all common adjacencies and telomeres are conserved in any $\sigma_k$ double distance, including the NP-hard $\sigma_\infty$ case. All 0-paths are isolated vertices, therefore they are selected independently of the choices for resolving the squares. A 2-cycle, in its turn, always includes one $\mathbb{S}$-edge from some square (such as square 1 in Fig. 2(b2)). From now on we assume that squares that have at least one $\mathbb{S}$-edge in a 2-cycle are fixed so that all existing 2-cycles are induced.

### 4.2 Symmetric squares can be fixed arbitrarily

Let $v$ and $\hat{v}$ be a pair of paralogous vertices in $\mathcal{Q}$. The square $\mathcal{Q}$ is said to be *symmetric* when either (i) there is a $\check{\mathbb{D}}$-edge connecting $v$ and $\hat{v}$, or (ii) $v$ and $\hat{v}$ are $\check{\mathbb{D}}$-telomeres, or (iii) $v$ and $\hat{v}$ are directly connected to $\mathbb{S}$-telomeres by $\check{\mathbb{D}}$-edges, as illustrated in Fig. 3. Note that, for any $\sigma_k$ disambiguation, a symmetric square $\mathcal{Q}$ can be fixed arbitrarily: the two ways of resolving $\mathcal{Q}$ would lead to solutions with the same score. We then assume that $ABG(\mathbb{S}, \check{\mathbb{D}})$ has no symmetric squares.

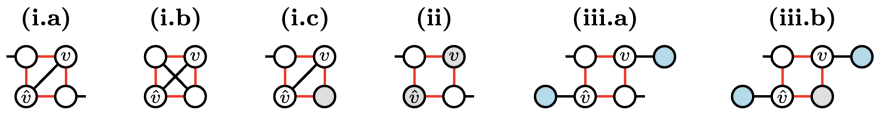


**Fig. 3.** Possible symmetric squares in the ambiguous breakpoint graph.

### 4.3 A linear time greedy algorithm for the $\sigma_4$ disambiguation

Although two valid 4-cycles can intersect with each other, since our graph is free of symmetric squares, two valid 2-paths cannot intersect with each other. Moreover, a 2-path has no $\check{\mathbb{D}}$-edge connecting squares, therefore it cannot intersect with a 4-cycle. For the $\sigma_4$ disambiguation it is then clear that, (i) any valid 2-path is always optimal and (ii) a 4-cycle that does intersect with another one is always optimal. We also know that intersecting 4-cycles are always part of co-optimal solutions [3]. An optimal solution can then be obtained greedily: after fixing squares containing edges that are part of 2-cycles, traverse the remainder of the graph and, for each valid 2-path or 4-cycle $C$ that is found, fix the square(s) containing $\mathbb{S}$-edges that are part of $C$, so that $C$ is induced. When this part is accomplished the remaining squares can be fixed arbitrarily.

### 4.4 Pruning $ABG(\mathbb{S}, \check{\mathbb{D}})$ for the $\sigma_6$ disambiguation

Players of the $\sigma_6$ disambiguation can intersect with each other and not every player is induced by at least one optimal solution. For that reason a more elaborated procedure is required, whose first step is a linear time preprocessing in

which from $ABG(\mathbb{S}, \check{\mathbb{D}})$ first all edges are removed that are incompatible with the existing 2-cycles, and then all remaining edges that cannot be part of a player [3]. This results in a $\{6\}$-*pruned* ambiguous breakpoint graph $PG(\mathbb{S}, \check{\mathbb{D}})$.

The edges that are not pruned and are therefore present in $PG(\mathbb{S}, \check{\mathbb{D}})$ are said to be *preserved*. A square that has preserved edges from distinct paralogous pairs is still ambiguous and is called a $\{6\}$-*square*. Otherwise it is resolved and can be fixed according to the preserved edges. Additionally, if none of its edges is preserved, a square is arbitrarily fixed.

The smaller $PG(\mathbb{S}, \check{\mathbb{D}})$ has all relevant parts required for finding an optimal solution of $\sigma_6$ disambiguation, therefore the 6-scores of both graphs are the same: $\sigma_6(ABG(\mathbb{S}, \check{\mathbb{D}})) = \sigma_6(PG(\mathbb{S}, \check{\mathbb{D}}))$. A clear advantage here is that the pruned graph might be split into smaller connected components, and it is obvious that the disambiguation problem can be solved independently for each one of them. Each connected component $G$ of $PG(\mathbb{S}, \check{\mathbb{D}})$ is of one of the two types [3]:

1. *Ambiguous*: $G$ includes at least one $\{6\}$-square;
2. *Resolved (trivial)*: $G$ is simply a player.

Let $\mathcal{C}$ and $\mathcal{P}$ be the sets of resolved components, so that $\mathcal{C}$ has all resolved cycles and $\mathcal{P}$ has all resolved paths. Furthermore, let $\mathcal{B}$ be the set of ambiguous components of $PG(\mathbb{S}, \check{\mathbb{D}})$. If we denote by $\sigma_6(G)$ the 6-score of an ambiguous component $G \in \mathcal{B}$, the 6-score of $PG(\mathbb{S}, \check{\mathbb{D}})$ can be computed with the formula:

$$\sigma_6(PG(\mathbb{S}, \check{\mathbb{D}})) = |\mathcal{C}| + \frac{|\mathcal{P}|}{2} + \sum_{G \in \mathcal{B}} \sigma_6(G).$$

For solving the $\sigma_6$ disambiguation the only missing part is finding, for each ambiguous component $G \in \mathcal{B}$, an optimal solution $\tau_G$ including only the $\{6\}$-squares of $G$. From now on, by $\mathbb{S}$-edge, $\mathbb{S}$-telomere, $\check{\mathbb{D}}$-edge and $\check{\mathbb{D}}$-telomere, we are referring only to the elements that are preserved in $PG(\mathbb{S}, \check{\mathbb{D}})$.

## 5 Solving the general $\sigma_6$ disambiguation in linear time

Here we present the most relevant contribution of this work: an algorithm to solve the $\sigma_6$ disambiguation in linear time, for genomes with linear chromosomes. The proofs omitted here can be found in the extended version of this work [4].

### 5.1 Intersection between players

A player in the $\sigma_6$ disambiguation can be either a $\{2,4\}$-*path*, that is a valid 2- or 4-path, or a $\{4,6\}$-*cycle*, that is a valid 4- or 6-cycle.

Let a $\check{\mathbb{D}}\mathbb{S}\check{\mathbb{D}}$-*path* be a subpath of three edges, starting and ending with a $\check{\mathbb{D}}$-edge. This is the largest segment that can be shared by two players: although there is no room to allow distinct $\{2, 4\}$-paths and/or valid 4-cycles to share a $\check{\mathbb{D}}\mathbb{S}\check{\mathbb{D}}$-path in a graph free of symmetric squares, a $\check{\mathbb{D}}\mathbb{S}\check{\mathbb{D}}$-path can be shared by at most two valid 6-cycles. Furthermore, if distinct $\check{\mathbb{D}}\mathbb{S}\check{\mathbb{D}}$-paths intersect at the same

𝔻̌-edge $e$ and each of them occurs in two distinct 6-cycles, then the 𝔻̌-edge $e$ occurs in four distinct valid 6-cycles. In Fig. 4 we characterize this exceptional situation[2], which consists in the occurrence of a *triplet*, composed of exactly three connected ambiguous squares in which at most two vertices, necessarily in distinct squares, are pruned out. In a *saturated* triplet, the squares in each pair are connected to each other by two 𝔻̌-edges connecting paralogous vertices in both squares; if a single 𝔻̌-edge is missing, that is, the corresponding vertices have outer connections, we have an *unsaturated* triplet. This structure and its score can be easily identified, therefore we will assume that our graph is free from triplets. With this condition, 𝔻̌-edges can be shared by at most two players:

**Proposition 1 (extended from [3]).** *Any 𝔻̌-edge is part of either one or two (intersecting) players in a graph free of symmetric squares and triplets.*

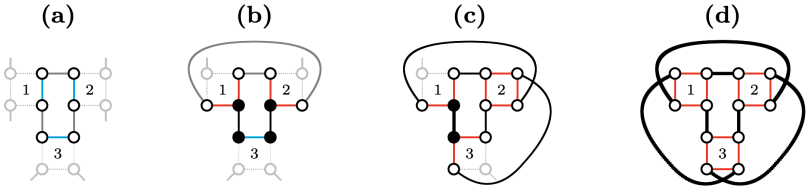


**Fig. 4.** (a) Resolved component (score = 1): a 6-cycle alternating (black) 𝔻̌- and (blue) 𝕊-edges, without intersections. (b) Two 6-cycles share one 𝔻̌𝕊𝔻̌-path composed of the two black 𝔻̌-edges with the blue 𝕊-edge in between. (c) Unsaturated triplet with score = 1: every 𝔻̌𝕊𝔻̌-path including the same 𝔻̌-edge (the thick black one) occurs in two distinct 6-cycles. The thick black 𝔻̌-edge occurs in four 6-cycles, all other black edges occur in two 6-cycles. (d) Saturated triplet with score = 2: every 𝔻̌𝕊𝔻̌-path occurs in two distinct 6-cycles, every black edge occurs in four 6-cycles.

As a consequence of Proposition 1, we know that any 𝕊-edge of a {6}-square $\mathcal{Q}$ is part of exactly one player [3]. Two {6}-squares $\mathcal{Q}$ and $\mathcal{Q}'$ are *neighbors* when a vertex of $\mathcal{Q}$ is connected to a vertex of $\mathcal{Q}'$ by a 𝔻̌-edge. For the case of circular genomes, all players are cycles. In each component $G$, since both 𝔻̌-edges inciding at the endpoints of an 𝕊-edge would induce the same {4,6}-cycle, the choice of an 𝕊-edge $e$ of a {6}-square $\mathcal{Q}$ (and its paralogous edge $\hat{e}$) would imply a unique way of resolving all neighbors of $\mathcal{Q}$, and, by propagating this to the neighbors of the neighbors and so on, all squares of $G$ would be resolved, resulting in what we called *straight solution* $\tau_G$. The score of $G$ would be given either by $\tau_G$, or by its complementary alternative $\widetilde{\tau}_G$, obtained by switching all ambiguous squares of $\tau_G$, or by both in case of co-optimality. Unfortunately, for genomes with linear chromosomes, where paths can intersect in a telomere, the procedure above no longer suffices. For that reason, we need to proceed with a further

[2] In our previous paper [3] we overlooked this particular case, that can fortunately be treated in a preprocessing. Otherwise the solution presented there is complete.

characterization of each ambiguous component $G$ of $PG(\mathbb{S}, \check{\mathbb{D}})$, allowing us to split the disambiguation of $G$ into smaller subproblems.

As we will present in the following, the solution for arbitrarily large components can be split into two types of problems, which are analogous to solving the *maximal independent set* of auxiliary subgraphs that are either simple paths or double paths. In both cases, the solutions can be obtained in linear time.

### 5.2 Intersection graph of an ambiguous component

If two {2,4}-paths intersect in their $\mathbb{S}$-telomere, this intersection must include the incident $\check{\mathbb{D}}$-edge. Therefore, when we say that an intersection occurs at an $\mathbb{S}$-telomere, this automatically means that the intersection is the $\check{\mathbb{D}}$-edge inciding in an $\mathbb{S}$-telomere. A valid 4-cycle has two $\check{\mathbb{D}}$-edges and a valid 6-cycle has three $\check{\mathbb{D}}$-edges. Besides the one at the $\mathbb{S}$-telomere, a valid 4-path has one $\check{\mathbb{D}}$-edge while a valid 2-path has none - therefore the latter cannot intersect with a {4,6}-cycle. When we say that 4-paths and/or {4,6}-cycles intersect with each other in a $\check{\mathbb{D}}$-edge, we refer to an *inner* $\check{\mathbb{D}}$-edge and not one inciding in an $\mathbb{S}$-telomere.

The auxiliary *intersection graph* $\mathcal{I}(G)$ of an ambiguous component $G$ has a vertex with weight $\frac{1}{2}$ for each {2,4}-path and a vertex with weight 1 for each {4,6}-cycle of $G$. Furthermore, if two players intersect, we have an edge between the respective vertices. The intersection graphs of all ambiguous components can be built during the pruning procedure without increasing its linear time complexity. Note that an independent set of maximum weight in $\mathcal{I}(G)$ corresponds to an optimal solution of $G$. Although in general this problem is NP-hard, in our case the underlying ambiguous component $G$ imposes a regular structure to $\mathcal{I}(G)$, allowing us to find such an independent set in linear time.

### 5.3 Path-flows in the intersection graph

A *path-flow* in $\mathcal{I}(G)$ is a maximal connected subgraph whose vertices correspond to {2,4}-paths. A *path-line* of length $\ell$ in a path-flow is a series of $\ell$ paths, such that each pair of consecutive paths intersect at a telomere. Assume that the vertices in a path-line are numbered from left to right with integers $1, 2, \ldots, \ell$. A *double-line* consists of two parallel path-lines of the same length $\ell$, such that vertices with the same number in both lines intersect in a $\check{\mathbb{D}}$-edge and are therefore connected by an edge. A 2-path has no free $\check{\mathbb{D}}$-edge, therefore a double-line is exclusively composed of 4-paths. If a path-line composes a double-line, it is *saturated*, otherwise it is a *unsaturated*. Since each 4-path of a double-line has a $\check{\mathbb{D}}$-edge intersection with another and each 4-path can have only one $\check{\mathbb{D}}$-edge intersection, no vertex of a double-line can be connected to a cycle in $\mathcal{I}(G)$.

Let us assume that a double-line is always represented with one upper path-line and one lower path-line. A double-line of length $\ell$ has $2\ell$ vertices and exactly two independent sets of maximal weight, each one with $\ell$ vertices and weight $\frac{\ell}{2}$: one includes the paths with odd numbers in the upper line and the paths with even numbers in the lower line, while the other includes the paths with even numbers in the upper line and the paths with odd numbers in the lower line.

Since a double-line cannot intersect with cycles, it is clear that at least one of these independent sets will be part of a global optimal solution for $\mathcal{I}(G)$. A maximal double-line can be of three different types:

1. *Isolated*: corresponds to the complete graph $\mathcal{I}(G)$. Here, but only if the length $\ell$ is even, the double line can be cyclic: in this case, in both upper and lower lines, the last vertex intersects at a telomere with the first vertex. Being cyclic or not, any of the two optimal local solutions can be fixed.
2. *Terminal*: a vertex $v$ located at the end of one of the two lines intersects with one unsaturated path-line. At least one of the two optimal local solutions would leave $v$ unselected; we can safely fix this option.
3. *Link*: intersects with unsaturated lines at both ends. The intersections can be:
   (a) *single-sided*: both occur at the ends of the same saturated line, or
   (b) *alternate*: the left intersection occurs at the end of one saturated line and the right intersection occurs at the end of the other.

   Let $v'$ be the outer vertex connected to a vertex $v$ belonging to the link at the right and $u'$ be the outer vertex connected to a vertex $u$ belonging to the link at the left. Let a *balanced link* be alternate of odd length, or single-sided of even length. In contrast, an *unbalanced link* is alternate of even length, or single-sided of odd length. If the link is unbalanced, one of the two local optimal solutions leaves both $u$ and $v$ unselected; we can safely fix this option. If the link is balanced, we cannot fix the solution before-hand, but we can reduce the problem, by removing the connections $uu'$ and $vv'$ and adding the connection $u'v'$. Since both $u'$ and $v'$ must be the ends of unsaturated lines, this procedure simply concatenates these two lines into a single unsaturated path-line. (See Fig. 5.) Finding a maximum independent set of the remaining unsaturated path-lines is a trivial problem that will be solved last; depending on whether one of the vertices $u'$ and $v'$ is selected in the end, we can fix the solution of the original balanced link.

### 5.4 Cycle-bubbles in the intersection graph

A *cycle-bubble* in $\mathcal{I}(G)$ is a maximal connected subgraph whose vertices correspond to {4,6}-cycles. Let $H$ be the subgraph of the underlying pruned ambiguous breakpoint graph including all edges that compose the cycles of a cycle-bubble. The optimal solution for $H$ is either the straight solution $\tau_H$ or its alternative $\widetilde{\tau}_H$ (algorithm from our previous work [3]). If both $\tau_H$ and $\widetilde{\tau}_H$ have the same score, then $H$ is said to be *balanced*, otherwise it is said to be *unbalanced*.

**Proposition 2.** *Let an ambiguous component $G$ have cycle-bubbles $H_1$, ..., $H_m$. There is an optimal solution for $G$ including, for each $i = 1, ..., m$: (1) the optimal solution for $H_i$, if $H_i$ is unbalanced; or (2) either $\tau_{H_i}$ or $\widetilde{\tau}_{H_i}$, if $H_i$ is balanced.*

*Proof.* If the whole component $G$ corresponds to one bubble $H$, the statement is clearly true. Otherwise, we need to examine the intersection with paths. One

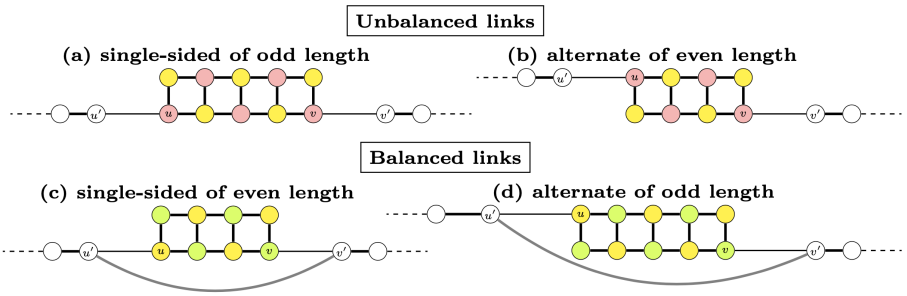


**Fig. 5.** Double-lines that are balanced and unbalanced links. The yellow solution that in cases (a-b) leaves $u$ and $v$ unselected can be fixed so that an independent set of the adjacent unsaturated path-line(s) can start at $v'$ (and $u'$). In cases (c-d) either the yellow or the green solution will be fixed later; it will be the one compatible with the solution of the unsaturated-line ending in $u'$ concatenated to the one starting in $v'$.

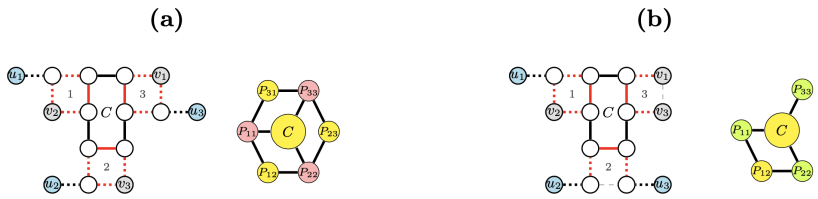


**Fig. 6.** Ambiguous and intersection graphs including a single 6-cycle $C$ (solid edges). Dotted edges are exclusive to paths and dashed gray edges are pruned out. In (a) and (b), $C$ intersects with three 4-paths $P_{11} = u_1..v_1$, $P_{22} = u_2..v_2$ and $P_{33} = u_3..v_3$. In (a), the yellow solution including $C$ also has the three 2-paths $P_{12} = u_1..v_2$, $P_{23} = u_2..v_3$ and $P_{31} = u_3..v_1$, being clearly superior. In (b), the yellow solution still has the 2-path $P_{12}$, having the same score of the green solution with three 4-paths.

critical case is a 6-cycle $C$ intersecting three 4-paths, but then there is at least one 2-path to compensate the solution including $C$ (Fig. 6). In general, the best we can get by replacing cycles by paths are co-optimal solutions [4]. □

As a consequence of Proposition 2, if a cycle-bubble is unbalanced, its optimal solution can be fixed so that the unsaturated path-lines around it can be treated separately.

**Balanced cycle-bubbles intersecting path-flows.** If a cycle-bubble $H$ is balanced and intersects with path-flows, then it requires a special treatment. The case in which the intersection involves a single cycle $C$ is also very easy, because certainly either $\tau_H$ or $\widetilde{\tau}_H$ leaves $C$ unselected and we can safely fix this option. More difficult is when the intersection involves at least two cycles. However, as we will see, here the only case that can be arbitrarily large is easy to handle. Let a cycle-bubble be a *cycle-line* when it consists of a series of valid 6-cycles, such that each pair of consecutive cycles intersect at a $\check{\mathbb{D}}$-edge.

**Proposition 3.** *Cycle-bubbles involving 9 or more cycles must be a cycle-line.*

*Proof.* A non-linear bubble reaches its "capacity" with 8 cycles, see Fig. 7. □

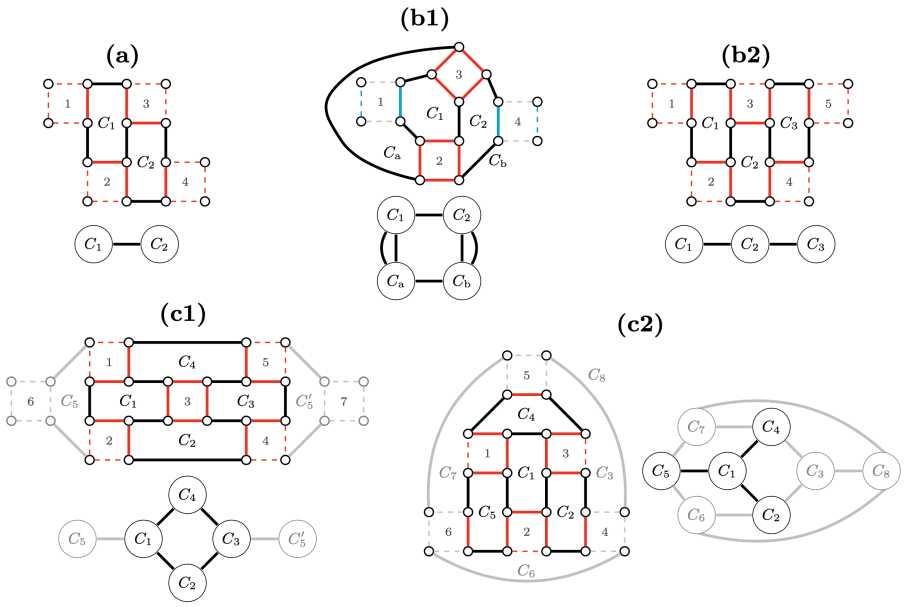


**Fig. 7.** By increasing the complexity of a bubble we quickly saturate the space for adding cycles to it. Starting with (a) a simple cycle-line of length two, we can either (b1) connect the open vertices of squares 2 and 3, obtaining a cyclic cycle-line of length 4 that cannot be extended, or (b2) extend the line so that it achieves length three. From (b2) we can obtain (c1) a cyclic cycle-line of length 4 that can be extended first by adding cycle $C_5$ next to $C_1$ and then either adding $C'_5$ next to $C_3$ or closing $C_6$, $C_7$ and $C_8$ so that we get (c2). In both cases no further extensions are possible. Note that (c2) can also be obtained by extending a cycle-line of length three and transforming it in a *star* with three branches, that can still be extended by closing $C_3$, $C_6$, $C_7$ and $C_8$.

Besides having its size limited to 8 cycles, the more complex a non-linear cycles-bubble becomes, the less space it has for paths around it. Therefore, the problem involving these bounded instances could be solved by brute force or complete enumeration. Many of the cases are shown in [4].

Our focus now is the remaining situation of a balanced cycle-line with intersections involving at least two cycles. Recall that cycles can only intersect with unsaturated path-lines. An intersection between a cycle- and a path-line is a *plug connection* when it occurs between vertices that are at the ends of both lines.

**Proposition 4.** *Cycle-lines of length at least 4 can only have plug connections.*

*Proof.* Fig. 8 shows that a cycle-line of length at least four only has "room" for intersections with path-lines next to its leftmost or rightmost cycles. □

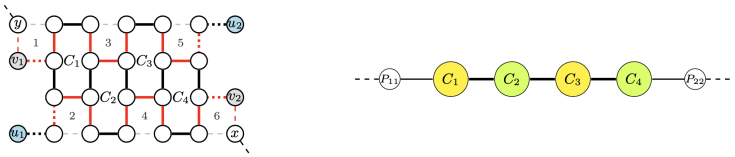


**Fig. 8.** A cycle-line of length 4 or larger only allows plug connections.

Balanced cycle-lines with two cycles can have connections to path-lines that are not plugs, but these bounded cases can be solved by complete enumeration or brute force. For arbitrarily large instances, the last missing case is of a balanced cycle-line with plug connections at both sides, called a *balanced link*. The procedure here is the same as that for double-lines that are balanced links, where the local solution can only be fixed after fixing those of the outer connections.

### 5.5 What remains is a set of independent unsaturated path-lines

If what remains is a single unsaturated path-line of even length, it can even be cyclic. In any case, an optimal solution of each unsaturated path-line can be trivially found: the one selecting all paths with odd numbers must be optimal. Fix this solution and, depending on the connections between the selected vertices of the unsaturated path-line and vertices from balanced cycle-bubbles or balanced double-lines, fix the compatible solutions for the latter ones.

## 6 Final remarks and discussion

This work is an investigation of the complexity of the double distance under a class of problems called $\sigma_k$ distances, which are between the breakpoint ($\sigma_2$) and the DCJ ($\sigma_\infty$) distance. Extending our previous results that considered only circular genomes, here we presented linear time algorithms for computing the double distance under the $\sigma_4$, and under the $\sigma_6$ distance, for inputs including linear chromosomes. Our solution relies on the ambiguous breakpoint graph.

The solutions we found so far are greedy with all players being optimal in $\sigma_2$, greedy with all players being co-optimal in $\sigma_4$ and non-greedy with non-optimal players in $\sigma_6$, all of them running in linear time. More specifically for the $\sigma_6$ case, after a pre-processing that fixes symmetric squares and triplets, at most two players share an edge. However we can already observe that, as $k$ grows, the number of players sharing a same edge also grows. For that reason, we believe that, if for some $k \geq 8$ the complexity of the $\sigma_k$ double distance is found to be NP-hard, the complexity is also NP-hard for any $k' > k$. In any case, the natural next step in our research is to study the $\sigma_8$ double distance.

Besides the double distance, other combinatorial problems related to genome evolution and ancestral reconstruction, including median and guided halving, have the distance problem as a basic unit. And, analogously to the double distance, these problems can be solved in polynomial time (but differently from the

double distance, not greedy and linear) when they are built upon the breakpoint distance, while they are NP-hard when they are built upon the DCJ distance [9]. Therefore, a challenging avenue of research is doing the same exploration for both median and guided halving problems under the class of $\sigma_k$ distances.

## References

1. Vineet Bafna and Pavel A. Pevzner. Genome rearrangements and sorting by reversals. In *Proc. of FOCS 1993*, pages 148–157, 1993.
2. Anne Bergeron, Julia Mixtacki, and Jens Stoye. A unifying view of genome rearrangements. In *Proc. of WABI 2006*, volume 4175 of *LNBI*, pages 163–173, 2006.
3. Marília D. V. Braga, Leonie R. Brockmann, Katharina Klerx, and Jens Stoye. A linear time algorithm for an extended version of the breakpoint double distance. In *Proc. of WABI 2022*, volume 242(13) of *LIPICs*, pages 1–16, 2022.
4. Marília D. V. Braga, Leonie R. Brockmann, Katharina Klerx, and Jens Stoye. Investigating the complexity of the double distance problems. arXiv 2303.04205, 2023. URL: https://arxiv.org/abs/2303.04205.
5. Cedric Chauve. Personal communication in Dagstuhl Seminar no. 18451 - Genomics, Pattern Avoidance, and Statistical Mechanics, November 2018.
6. Nadia El-Mabrouk and David Sankoff. The reconstruction of doubled genomes. *SIAM Journal on Computing*, 32(3):754–792, 2003.
7. Sridhar Hannenhalli and Pavel A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proc. of FOCS 1995*, pages 581–592, 1995.
8. Sridhar Hannenhalli and Pavel A. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46(1):1–27, 1999.
9. Eric Tannier, Chunfang Zheng, and David Sankoff. Multichromosomal median and halving problems under different genomic distances. *BMC Bioinformatics*, 10:120, 2009.
10. Sophia Yancopoulos, Oliver Attie, and Richard Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16):3340–3346, 2005.