RAIRO-Oper. Res. 57 (2023) 1045–1058
https://doi.org/10.1051/ro/2023052

RAIRO Operations Research
www.rairo-ro.org

ALGORITHMS FOR THE GENOME MEDIAN UNDER A RESTRICTED MEASURE OF REARRANGEMENT

Helmuth O.M. Silva[1], Diego P. Rubert[1], Eloi Araujo[1], Eckhard Steffen[2], Daniel Doerr[3] and Fábio V. Martinez[1,*]

Abstract. Ancestral reconstruction is a classic task in comparative genomics. Here, we study the *genome median problem*, a related computational problem which, given a set of three or more genomes, asks to find a new genome that minimizes the sum of pairwise distances between it and the given genomes. The *distance* stands for the amount of evolution observed at the genome level, for which we determine the minimum number of rearrangement operations necessary to transform one genome into the other. For almost all rearrangement operations the median problem is NP-hard, with the exception of the *breakpoint median* that can be constructed efficiently for multichromosomal circular and mixed genomes. In this work, we study the median problem under a restricted rearrangement measure called c_4*-distance*, which is closely related to the breakpoint and the DCJ distance. We identify tight bounds and decomposers of the c_4-median and develop algorithms for its construction, one exact ILP-based and three combinatorial heuristics. Subsequently, we perform experiments on simulated data sets. Our results suggest that the c_4-distance is useful for the study the genome median problem, from theoretical and practical perspectives.

Mathematics Subject Classification. 90C10, 90C27, 90C35, 90C59.

Received December 6, 2022. Accepted April 9, 2023.

1. Introduction

An important branch of research with applications in biology and medicine concerns the inference of ancestral genomes from whole genome sequencing data of living organisms. In this work, we study the problem of finding a *median* of genomes that evolve by large-scale mutations, also called *rearrangements*. These alter the order and orientation of genomic markers within and between chromosomal sequences. It is common to assume that the underlying evolutionary scenario is parsimonious, thus the minimum number of rearrangements between two genomes provides a notion of their *rearrangement distance*. Whereas pairwise rearrangement distances can be computed efficiently in some settings, finding a median of three or more genomes, *i.e.*, a genome that minimizes

Keywords. Median problem, optimization, integer linear programming, heuristics.

[1] Faculdade de Computação, Universidade Federal de Mato Grosso do Sul, Campo Grande, Brazil.

[2] Department of Mathematics, Paderborn University, Paderborn, Germany.

[3] Institute for Medical Biometry and Bioinformatics, University Hospital Düsseldorf, Heinrich Heine University Düsseldorf, Düsseldorf, Germany.

*Corresponding author: fabio.martinez@ufms.br; fabio.viduani@gmail.com

© The authors. Published by EDP Sciences, ROADEF, SMAI 2023

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the sum of rearrangement distances between itself and the given genomes, is computationally intractable even for highly simplified rearrangement distances.

The distance between two genomes depends on the chosen rearrangement operation. For instance, the number of *breakpoints* between two genomes, *i.e.*, the number of pairs of genomic markers that appear consecutive in one genome but not in the other, gives rise to a simple rearrangement distance. While, strictly speaking, the breakpoint distance underlies no rearrangement operation [6], other distances do, such as the *double-cut-and-join* (DCJ) distance [14]. A *DCJ operation* breaks a genome, represented by a set of sequences of genomic markers, at two arbitrary positions and subsequently reconnects the thus created four open ends in a new combination.

Almost all known rearrangement distances can be computed efficiently in linear time under the assumption that genomic markers appear unique in each genome [2,6,14]. However, considering one step forward, constructing a median of three genomes is NP-hard under almost all rearrangement distances, including DCJ [10], with two notable exceptions: the breakpoint distance and the closely related *single-cut-or-join* (SCJ) are tractable for multichromosomal circular and mixed genomes [5,10].

The *breakpoint graph* is a common aid in computing rearrangement distances. When comparing two genomes that are permutations of one another, the graph consists of cycles of even length. The contained number of cycles plays an essential role. For instance, the smallest cycles, *i.e.*, cycles of length 2 represent *adjacencies* in the breakpoint graph, which are the counterpart of breakpoints. Larger cycles represent the requirement of one or more DCJ operations to transform one genome into the other. Thus, to compute the breakpoint distance, cycles of length 2 are counted (and this quantity is then subtracted from the number or markers of the two genomes). Likewise, to compute the DCJ distance, cycles of any length are counted.

It is natural to also consider intermediary distance measures. In this work we study the c_4*-distance* between two genomes, which is based on the number of cycles of length up to 4 in the breakpoint graph. In other words, only those cycles are counted that require at most one DCJ operation to be transformed into adjacencies. It is important to highlight that the c_4-distance violates the triangle inequality and therefore is not a metric – unlike the breakpoint and DCJ distance. Here, we address the c_4*-median problem*, *i.e.*, the construction of a new genome that minimizes the sum of pairwise c_4-distances between it and each member of a set of three or more genomes. In particular, we are interested in finding c_4-medians for three genomes.

This work is an extension of the abstract submitted to the 7th Theoretical Computer Science Meeting (*VII ETC – Encontro de Teoria da Computação*, in portuguese) and the main new contributions are the following:

- We present properties and upper bounds to the c_4-distance (Sects. 3, 4);
- We improve the first version of the ILP, making it faster, and now it can handle instances with up to 2000 markers with neglegible gaps (Sect. 5.1);
- We develop two new combinatorial heuristics to compute the c_4-distance of given genomes (Sects. 5.2, 5.3);
- We extend the experiments with different data sets for the new algorithms (Sect. 6).

This paper is structured as follows. Section 2 provides basic definitions and notation, including the definition of the median problem. We then present bounds for the median in Section 3. Section 4 addresses the characterization of *decomposers*, which are building blocks of median genomes. In Section 5, we describe algorithms to compute the c_4-median, including one exact ILP-based and three heuristics. Experimental results on simulated instances are presented in Section 6. Section 7 concludes this paper and presents future works.

2. Preliminaries

A *marker* is an oriented DNA fragment. We denote a marker either by m or by $\overline{m}$, depending on its orientation in the DNA strand. A *chromosome* is a sequence of markers and can be linear or circular. A linear chromosome has two extremities and each one is a *telomere*. We use a string of markers to represent a chromosome and we add parentheses at the extremities of the represented string to denote a circular chromosome. A *genome* is a collection of chromosomes.

A marker m has two distinct *extremities*, called *tail* and *head*, represented by m^t and m^h, respectively. An *adjacency* in a chromosome is conformed by either the extremity of a marker adjacent to a telomere, or by a

pair of consecutive marker extremities. As an example, the adjacencies 1^t5^h, 5^t2^t, 2^h4^t, 4^h3^t, 3^h6^t and 6^h1^h define a circular chromosome. Another representation to it is $(\overline{5}\ 2\ 4\ 3\ 6\ \overline{1})$. A multichromosomal genome is a set of chromosomes such as $\{(3), \overline{5}\ 2\ 6, (4\ \overline{1})\}$, which is composed of three chromosomes, one linear and two circular.

2.1. Classical distances

One can compute a rearrangement distance between two given genomes with support of an equivalent structure known as breakpoint graph [1]. Let $\mathcal{M}$ be a set of n markers and define $\mathcal{M}_x$ the set of extremities of all markers in $\mathcal{M}$, with $|\mathcal{M}_x| = 2n$. For two genomes A and B, each one with n markers from $\mathcal{M}$, the *breakpoint graph* $BG(A, B)$ is the multigraph whose vertex set is $\mathcal{M}_x$ and the edges are of two types: A-edges and B-edges, corresponding to adjacencies in genomes A and B, respectively. This graph has vertices with degree zero, one or two, and thus it is a collection of paths and cycles. If a is the number of common non-telomeric adjacencies between A and B and t is the number of common telomeres, the *breakpoint distance* [10] between A and B is

$$\mathrm{d}_{\mathrm{BKP}}(A, B) = n - a - t/2.$$

Notice that the breakpoint distance is equivalent to the so called *single-cut-or-join (SCJ) distance* [5]. Since a is the number of nontelomeric adjacencies in $BG(A, B)$, each one of these adjacencies represents a cycle of length 2 in $BG(A, B)$ and one can denote it by $c_2 = a$. We call a cycle of length j a j-cycle.

On the other hand, if c is the number of cycles (of any length) and e is the number of paths with an even number of edges in $BG(A, B)$, the *double-cut-and-join (DCJ) distance* [14] between A and B is

$$\mathrm{d}_{\mathrm{DCJ}}(A, B) = n - c - e/2.$$

Breakpoint/SCJ and DCJ distances can be computed efficiently [2,5,10].

2.2. Multichromosomal circular genomes

Chromosomes and plasmids of single-celled organisms such as *bacteria* and *archaea*, mitochondrial DNA within eukaryotic cells, and chloroplast DNA in plants are examples of circular chromosomes/genomes and motivate the study of the circular genome median. Confining to circular chromosome also avoids a lengthy treatment of distance formulas to account for telomeric adjacencies [7,8,11]. Therefore, from now on we consider only multichromosomal genomes with circular chromosomes.

Notice that if two genomes A and B have only circular chromosomes, we have $\mathrm{d}_{\mathrm{BKP}}(A, B) = n - c_2$ and $\mathrm{d}_{\mathrm{DCJ}}(A, B) = n - c = n - c_2 - c_4 - c_6 - \ldots$, where c_j denotes the number of j-cycles in the breakpoint graph of A and B.

2.3. Median problem

Let Π be a set with $p \geq 3$ genomes, each one with n markers from $\mathcal{M}$. The *(genome) median problem* on Π asks for finding a genome Γ with n markers from $\mathcal{M}$ minimizing the pairwise distances between Γ and each genome in Π, under a rearrangement operation. If the operation is the breakpoint/SCJ, then the median can be computed in polynomial time [5]. However, for the DCJ operation, the median problem is NP-hard, even for $p = 3$ [3,10].

Motivated by the searching for where is the boundary of efficiency-hardness of the median problem, one can define a new measure. The c_4-*distance*, denoted by d_4, between Π_i and Π_j is given by $\mathrm{d}_4(\Pi_i, \Pi_j) = n - c_2 - c_4$, where n is the number of markers in $\mathcal{M}$ and in both Π_i and Π_j, and c_ℓ is the number of ℓ-cycles in $BG(\Pi_i, \Pi_j)$, $\ell \in \{2, 4\}$. As mentioned before, the triangle inequality does not hold for d_4 as we can see in a simple example: Given genomes $\Pi_1 = (1\ \overline{3}\ \overline{2}\ 4), \Pi_2 = (1\ 2\ 3\ 4)$ and $\Pi_3 = (1\ 2\ 3\ \overline{4})$, we have that

$$\mathrm{d}_4(\Pi_1, \Pi_3) = 3 \not\leq 1 + 1 = \mathrm{d}_4(\Pi_1, \Pi_2) + \mathrm{d}_4(\Pi_2, \Pi_3).$$

Let $\Pi = \{\Pi_1, \ldots, \Pi_p\}$ be a set of $p \geq 3$ genomes and Γ be a genome. The c_4-*cost* $K(\Pi, \Gamma)$ of Γ given Π is

$$K(\Pi, \Gamma) = \sum_{\Pi_i \in \Pi} \mathrm{d}_4(\Pi_i, \Gamma).$$

We say that a genome Γ is a c_4-*median* of a set of $p \geq 3$ genomes Π if Γ minimizes the c_4-cost $K(\Pi, \Gamma)$. For a given set of genomes Π, we denote by $K^\star(\Pi)$ the value of its c_4-median:

$$K^\star(\Pi) = \min\{K(\Pi, \Gamma) : \text{genomes in } \Pi \text{ and genome } \Gamma\}.$$

Thus, we can formally state the following:

Problem c_4-MEDIAN(Π)**:** given $p \geq 3$ genomes in Π, each with n markers from $\mathcal{M}$, find a genome Γ with n markers from $\mathcal{M}$ such that $K^\star(\Pi) = K(\Pi, \Gamma)$.

We are particularly interested in the simplest version of the c_4-MEDIAN problem, when $p = 3$.

2.4. Graph formulation

We can rephrase the c_4-MEDIAN in terms of graphs. We assign each extremity of a marker from the set of n markers $\mathcal{M}$ to a vertex in a graph G and thus $|V(G)| = 2n$. An adjacency in a given genome Π_i is represented as an edge in G with color i, that is, an adjacency uv in Π_i is an edge uv in G with color i. Thus, G is a p-edge-colored multigraph. Observe that G is a generalization of the breakpoint graph [1] for at least three genomes, and we call it an *extended breakpoint graph* for Π, denoted by $BG_x(\Pi) = G$.

Let Γ be a subset of unsorted pairs from $V(G)$, *i.e.*, a subset of $V(G)^{(2)} = \binom{V(G)}{2}$, such that $|\Gamma| = n$ and $e \cap f = \emptyset$ for each pair e, f in Γ. Thus, Γ is a 1-regular graph with n elements. We do not distinguish between Γ and $E(\Gamma)$.

Define $G^\Gamma := G + \Gamma$ as the multigraph such that $V(G^\Gamma) = V(G)$ and $E(G^\Gamma) = E(G) \cup \Gamma$. Hence, G^Γ is a $(p+1)$-edge-colored multigraph with a new color $p+1$. We say that a cycle in G^Γ is i-*colored* if its edges have colors alternating between i and $p+1$. We also say that a cycle in G is *bicolored* if its edges have colors alternating between i and j, $i \neq j$, $i, j \in \{1, \ldots, p\}$.

We denote by $k(G^\Gamma)$ the number of i-colored 2- and 4-cycles in G^Γ. And we denote by $k(G)$ the maximum number of i-colored 2- and 4-cycles in G^Γ for all possible 1-regular graphs Γ with vertex set $V(G)$:

$$k(G) = \max\big\{k\big(G^\Gamma\big) : \Gamma \text{ is a 1-regular graph on } V(G)\big\}.$$

Hence, we have the following problem, equivalent to c_4-MEDIAN:

Problem MAX-2/4-CYCLES(G)**:** given a p-edge-colored multigraph G with $p \geq 3$ and $|V(G)| = 2n > 0$, find a 1-regular graph Γ with vertex set $V(G)$ such that $k(G) = k(G^\Gamma)$.

See Figure 1 for an example. We are interested in the simplest version of the MAX-2/4-CYCLES problem, when $p = 3$.

3. BOUNDS

In this section we give upper bounds to MAX-2/4-CYCLES, such that the instance G is a 3-edge-colorable graph with $2n$ vertices.

We consider graphs without loops. Let G be a graph and $v, w \in V(G)$. The number of edges between v and w is the *multiplicity* of v, w, which is denoted by $\mu(v, w)$. Let e be an edge which is incident to v and w. If there is no harm of confusion, then we write $e = vw$ and we say that e is an edge of multiplicity $\mu(v, w)$. An edge of multiplicity 1 is also called a *simple edge*, and an edge of multiplicity at least 2 is called a *multiedge*. If $\mu(v, w) \leq 1$ for all $v, w \in V(G)$, then G is called a *simple graph*.

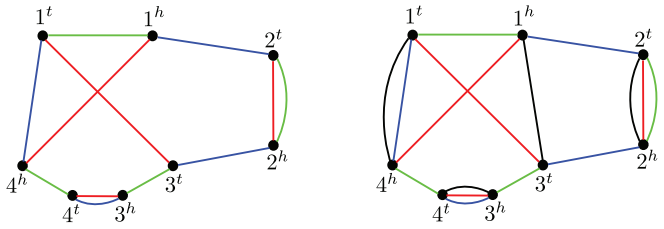


FIGURE 1. (a) Three given genomes $\Pi_1 = \{(1\ \overline{4}\ \overline{3}), (2)\}, \Pi_2 = \{(1\ 2\ 3\ 4)\}$ and $\Pi_3 = \{(1), (2), (3), (4)\}$ for the c_4-MEDIAN and its equivalent given graph G for the MAX-2/4-CYCLES. (b) An optimal solution $\Gamma = \{(1\ 3\ 4), (2)\}$ with 7 cycles: 5 cycles of length 2 (two 1-colored cycles (red), two 2-colored (blue) and one 3-colored (green)) plus 2 cycles of length 4 (one 1-colored cycle (red) and one 2-colored (blue)). Therefore, $K^\star(\Pi) = K(\Pi, \Gamma) = 3 \cdot 4 - 7 = 5$ and $k(G^\Gamma) = k(G) = 7$.

A 3-regular graph is also called a *cubic graph*. A cubic graph is 3-edge-colorable if its edges set can be partitioned into three perfect matchings, which are also called the *color classes* of a 3-edge-coloring of G.

Let G be a 3-edge-colorable cubic graph of order $2n$ and Γ be a color class of G. Then G^Γ has n i-colored 2-cycles and hence, $k(G) \geq n = \frac{|V(G)|}{2}$.

Note that every edge of G is contained in at most one i-colored cycle of G^Γ and that an i-colored 2-cycle contains precisely one edge of G and an i-colored 4-cycle contains precisely two edges of G. Let K_2^3 be the unique cubic graph with two vertices.

Proposition 1. *Let G be a connected cubic 3-edge-colorable graph. Then $k(G) \leq \frac{3}{2}|V(G)|$. Furthermore, $k(G) = \frac{3}{2}|V(G)|$ if and only if $G = K_2^3$.*

Proof. The first part follows directly from the c_4-distance definition. If $G = K_2^3$, then $k(G) = 3$. For the other direction, choose Γ such that $k(G^\Gamma) = k(G)$. If $k(G) = \frac{3}{2}|V(G)|$, then it follows by the remarks above that G^Γ has i-colored 2-cycles only. Hence, $G = K_2^3$. □

Theorem 2. *Let m be a positive integer and $G \neq K_2^3$ be a connected 3-edge-colorable cubic graph. If G has m multiedges, then $k(G) \leq |V(G)| + \lfloor \frac{m}{2} \rfloor$. Furthermore, the bound is tight.*

Proof. Since K_2^3 is the unique connected cubic graph which contains an edge e with $\mu(e) = 3$, it follows that G has m edges of multiplicity 2. Let $|V(G)| = 2n > 2$.

Let Γ be a 1-regular graph with vertex set $V(G)$. For $i \in \{1, 2\}$ let $E_i(G) = \{e : e \in E(G) \text{ and } \mu(e) = i\}$, and $m_i = |E_i(G) \cap \Gamma|$. Since $m_1 + m_2$ counts the number of edges in a subset of Γ, it follows that $m_1 + m_2 \leq n$. Furthermore, $m_2 \leq m$.

The graph G^Γ has $2m_2 + m_1$ many i-colored 2-cycles, which cover $2m_2 + m_1$ edges of G. Since every i-colored 4-cycle contains precisely two edges of G and each edge of G is in at most one i-colored cycle, it follows that there are at most $\frac{1}{2}(3n - (2m_2 + m_1))$ i-colored 4-cycles. Hence,

$$k(G^\Gamma) \leq 2m_2 + m_1 + \frac{1}{2}(3n - (2m_2 + m_1)) = \frac{1}{2}(m_1 + m_2) + \frac{1}{2}m_2 + \frac{3}{2}n \leq 2n + \tfrac{1}{2}m_2.$$

Since $m_2 \leq m$ and $k(G^\Gamma)$ is an integer, it follows that $k(G^\Gamma) \leq 2n + \lfloor \frac{m}{2} \rfloor = |V(G)| + \lfloor \frac{m}{2} \rfloor$. And since Γ was chosen arbitrarily, the first statement of the theorem is proved.

To see that the bound is tight, check Figure 2. □

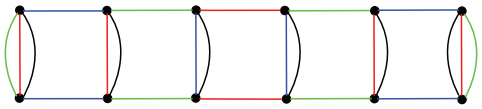


FIGURE 2. A linear ladder, *i.e.*, a connected cubic 3-edge-colorable graph G with $|V(G)| = 2n = 12$, edges with colors 1 (red), 2 (blue) and 3 (green), $m = 2$ multiedges, and a 1-regular graph Γ (black edges) such that $k(G^\Gamma) = k(G) = 13 = 2n + \lfloor \frac{m}{2} \rfloor$.

Let $n \geq 2$ and P_n be a path on vertices $v_1, \ldots, v_n$ in this order and let P'_n be a copy of P_n. Let $\mathcal{L}(n)$ be the cubic graph obtained from P_n and P'_n by adding the edges $v_i v'_i$ for $i \in \{1, \ldots, n\}$ and duplicating edges $v_1 v'_1$ and $v_n v'_n$. We call a graph a *linear ladder* if it is isomorphic to $\mathcal{L}(n)$ for an integer $n \geq 2$. See Figure 2 for an example.

Let G_4^3 be the connected cubic graph on 4 vertices which has multiedges. Note that $k(G_4^3) = 5 = |V(G_4^3)| + \lfloor \frac{m}{2} \rfloor$. Therefore, the bound of Theorem 2 is tight for this graph. However, this graph can be characterized by its c_4-median.

Proposition 3. *Let G be a connected cubic 3-edge-colorable graph. Then $k(G) = \frac{5}{4}|V(G)|$ if and only if $G = G_4^3$.*

Proof. Let G be of order $2n$. It follows with Theorem 2, that $m = n$. Hence, the edges of multiplicity 2 form a perfect matching in G. Thus, G is obtained from an even cycle by doubling every second edge. Now it easy to see that G_4^3 is the only graph with $k(G) = \frac{5}{2}n$. The other direction is trivial. □

Corollary 4. *Let G be a connected 3-edge-colorable cubic graph. If $|V(G)| > 4$, then $k(G) < \frac{5}{4}|V(G)|$.*

As we will see in Section 4, a special type of ladder introduced below, called circular ladder, is a strong decomposer and this is a motivation to show the next results on simple graphs. That is, for simple graphs we can prove some better bounds. For $m \in \{0, 2\}$ there are infinitely many graphs which attain the bound of Theorem 2. Maybe, it is true that the family of Figure 2 characterizes the graphs with maximum $k(G)$ and two multiedges. It is unclear whether such graphs exist for $m > 2$. It could be that if a connected cubic graph G has more than two multiedges, then $k(G) < 2n + \lfloor \frac{m}{2} \rfloor$.

Let $n \geq 3$ and C_n be a cycle on vertices $v_1, \ldots, v_n$ in this order and let C'_n be a copy of C_n. Let $L_1(n)$ be the cubic graph obtained from C_n and C'_n by adding the edges $v_i v'_i$ for $i \in \{1, \ldots, n\}$ and $L_2(n)$ be the cubic graph which is obtained from $L_1(n) - \{v_n v_1, v'_n v'_1\}$ by adding the edges $v_n v'_1$ and $v'_n v_1$. Graph $L_2(n)$ is also called a Möbius ladder.

We call a graph a *circular ladder* if it is isomorphic to K_4 or to $L_1(n)$ or $L_2(n)$ for an integer $n \geq 3$.

Theorem 5. *If G is a connected simple 3-edge-colorable cubic graph, then $k(G) \leq |V(G)|$. Furthermore, $k(G) = |V(G)|$ if and only if G is a circular ladder.*

Proof. The first part follows from Theorem 2, since G is simple. If G is a circular ladder, then $k(G^\Gamma) = 2n = |V(G)|$ for $\Gamma = \{v_i v'_i : i \in \{1, \ldots, n\}\}$.

Let Γ be a 1-regular graph on $V(G)$ such that $k(G^\Gamma) = k(G)$. For $j \in \{2, 4\}$ let c_j be the number of i-colored j-cycles. We have $k(G^\Gamma) = c_2 + c_4 = 2n$. Consequently, $c_4 = \frac{1}{2}(3n - c_2)$ and therefore, $c_2 = c_4 = n$. Hence, Γ is a perfect matching of G. Let $e \in \Gamma$, say $e = vw$, and let v_1, v_2 and w_1, w_2 be the two other neighbors of v and w, respectively. The edges vv_i and ww_i cannot be in a 2-cycle since G is simple and Γ is a matching. Thus, each of them is in an i-colored 4-cycle, and therefore, Γ induces a perfect matching on $G[\{v, v_1, v_2, w, w_1, w_2\}]$; say $v_1 w_1, v_2 w_2 \in \Gamma$. It follows that $v_1 w_1, v_2 w_2 \in E(G)$, since Γ is a perfect matching of G. If $|V(G)| = 2n = 4$, then $G = K_4$. If $|V(G)| = 2n > 4$, then G is isomorphic to a circular ladder on $2n$ vertices. □

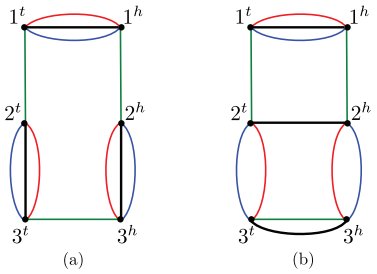


FIGURE 3. A connected cubic 3-edge-colorable graph G with two optimal c_4-medians: (a) with six (all) 2-cycles and (b) with two 2-cyles and four 4-cycles. Notice that 2^t3^t and 2^h3^h are decomposers but not strong decomposers.

4. Decomposers

A popular strategy for constructing solutions to median problems is to decompose G into smaller parts and then identify partial solutions thereof. These are subsequently integrated into a complete median [4, 9–13]. In the following, we make use of notation from [12] to characterize such partial solutions for Max-2/4-Cycles.

Let G be a 3-edge-colorable cubic graph and Γ be a 1-regular graph with vertex set $V(G)$ such that $k(G) = k(G^\Gamma)$. For any vertex-induced subgraph $H \subset G$, Γ is *H-crossing* if and only if it contains an edge $uv \in \Gamma$ such that $|\{u, v\} \cap V(H)| = 1$. Conversely, a vertex-induced subgraph $H \subset G$ is a *decomposer* if and only if there exists a 1-regular graph Γ with vertex set $V(G)$ such that $k(G) = k(G^\Gamma)$ and Γ is not H-crossing. And H is a *strong decomposer* if every 1-regular graph Γ with $k(G) = k(G^\Gamma)$ is not H-crossing. From Propositions 1, 3, and Theorem 5 directly follows:

Corollary 6. *K_2^3, G_4^3, and any linear and circular ladder are strong decomposers.*

Decomposers of related problems. In dissecting the computational problem of finding c_4-medians, a straightforward question is whether some decomposers of its related problems, the breakpoint and the DCJ median, are also decomposers of a c_4-median. Here, we look at simple decomposers, in particular K_2^2, a graph of two vertices connected by two distinctly colored parallel edges, and K_2^3, a graph of two vertices connected by parallel edges of all three distinct colors.

Proposition 7 ([9])**.** *K_2^2 and every connected component in G is a decomposer. Further, K_2^3 is a strong decomposers of the breakpoint median of three.*

Proposition 8 ([12, 13])**.** *K_2^2 is a decomposer and K_2^3 is a strong decomposer of the DCJ median of three.*

While K_2^3 is a strong decomposer of the c_4-median as shown above, we observe that K_2^2 may be a decomposer, as we can see in Figure 3.

Adequate subgraphs [12] are a family of decomposers of the DCJ median of three genomes: A subgraph $H \subset G$ is an *adequate subgraph* if $k(H) \geq \frac{3}{4}|V(H)|$.

Proposition 9. *Adequate subgraphs are not decomposers of* Max-2/4-cycles.

Proof. Figure 3 is a trivial counterexample. A more illustrative example is the following: Cycles of four vertices $v_1, \ldots, v_4$ are adequate subgraphs [12]. Figure 4 depicts a counterexample where all c_4-medians are H-crossing for the highlighted cycle H. □

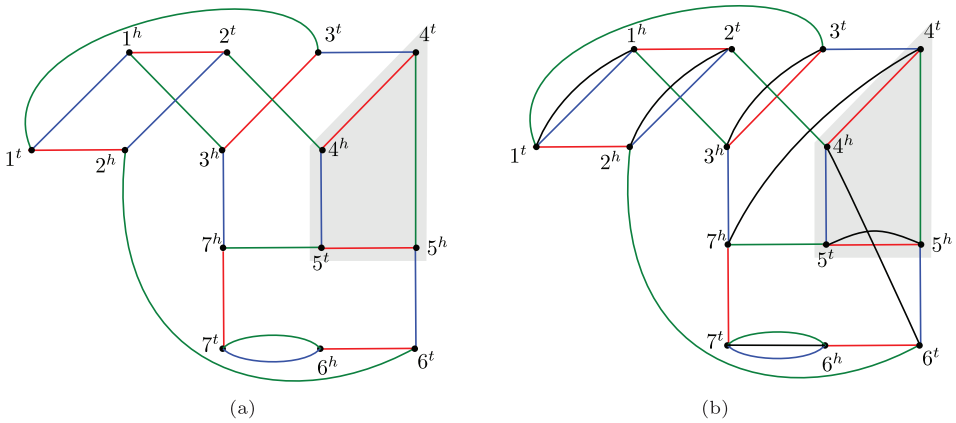


FIGURE 4. (a) 3-edge-colorable cubic graph G with embedded adequate subgraph highlighted in gray and (b) graph G^Γ with $k(G) = k(G^\Gamma) = 12$.

5. ALGORITHMS

In this section we present four algorithms for the MAX-2/4-CYCLES: one exact ILP-based algorithm and three greedy heuristics.

Suppose we have an instance of MAX-2/4-CYCLES: a graph G with $2n$ vertices, $n > 0$, and a set of three 1-regular graphs $\Pi = \{\Pi_1, \Pi_2, \Pi_3\}$ such that each Π_i has vertex set $V(G)$. Saying in other words, Π_i has n edges, *i.e.*, it is a perfect matching in G. Algorithms in this section receive G and Π and build and return an 1-regular graph Γ in G maximizing the number of 2- and 4-cycles in G^Γ.

5.1. ILP formulation

Our exact ILP algorithm translates the minimization formula for the c_4-median problem in a straightforward way. See Algorithm 1.

Algorithm 1. ILP for computing the c_4-median.

$$
\begin{array}{lrcll}
\min & 3n - \displaystyle\sum_{\pi \in E} c_\pi \; y_\pi - \sum_{\pi,\sigma \in E, \pi \neq \sigma} c_{\pi,\sigma} \; x_{\pi,\sigma} & & & \\
\text{subject to} & \displaystyle\sum_{\pi \sim v} y_\pi & = & 1 \quad \forall\, v \in V & \text{(C.01)} \\
& x_{\pi,\sigma} & \leq & y_\pi \\
& x_{\pi,\sigma} & \leq & y_\sigma
\end{array}
\quad \Big\} \quad \forall\, \{\pi,\sigma\} \in F \qquad \text{(C.02)}
$$

$$
\begin{array}{lrcll}
\text{and} & y_\pi & \in & \{0,1\} \quad \forall\, \pi \in E & \text{(D.01)} \\
& x_{\pi,\sigma} & \in & \{0,1\} \quad \forall\, \pi,\sigma \in E, \pi \neq \sigma & \text{(D.02)}
\end{array}
$$

We establish that one, and only one, edge in the solution is chosen for each possible v in V_i and if the edge π in E is chosen, then the binary variable y_π receives 1, otherwise it receives 0 (constraint (C.01) and binary variable (D.01)). If the pair of distinct edges π and σ is chosen in E, then the binary variable $x_{\pi,\sigma}$ receives 1. Otherwise, $x_{\pi,\sigma}$ gets 0 (constraint (C.02), binary variable (D.02)). Finally, we maximize the amount of 2- and 4-cycles with the variables c_π and $c_{\pi,\sigma}$, respectively, which correspond to the ILP objective function.

Observe that the choices of edges conforming 2-cycles are linked to the counter c_π when an edge is chosen to be part of the solution. And the choice of edge pairs for 4-cycles in the variable $c_{\pi,\sigma}$ it is performed when the

pair is in E_i, with $i \in \{1, 2, 3\}$. The constraint is divided into two parts, since this speeds up the process in the search for the optimal solution.

Lastly, observe that we have $O(n^2)$ constraints and binary variables in the Algorithm 1.

5.2. Induced bicolored cycles

The basic idea behind Algorithm 2 is the following. We take two of the perfect matchings Π_i and Π_j from Π in G, with $i, j, \in \{1, 2, 3\}$ and $i \neq j$, and build an induced graph $H_{i,j} = G[\Pi_i \cup \Pi_j]$. Observe that $H_{i,j}$ is a collection of bicolored cycles, with alternating i- and j-colored edges. For each bicolored cycle C in $H_{i,j}$, we add edges $e = uv$ in Π_k to C such that both extremities u and v are vertices of C, obtaining the subgraph C' which is a linear ladder. Notice that C' may have vertices of degree either 2 or 3. Then we find a set of edges $\Gamma_{i,j}$ that maximizes the number of 2- and 4-cycles in C'. We repeat this process to each pair $i, j \in \{1, 2, 3\}$, $i \neq j$, and return the best of three.

Algorithm 2. Induced bicolored cycles.

Input: Graph G with perfect matchings Π_1, Π_2, Π_3
Output: 1-regular graph Γ maximizing $k(G^\Gamma)$

1: **for** each pair $i, j \in \{1, 2, 3\}, i \neq j$ **do**
2: build an induced graph $H_{i,j}$ from genomes Π_i, Π_j
3: $\Gamma_{i,j} \leftarrow \emptyset$
4: **for** each cycle C in $H_{i,j}$ **do**
5: find a set of edges A' such that $C \cup A'$ is a linear ladder and maximizes the number of 2- and 4-cycles in G
6: $\Gamma_{i,j} \leftarrow \Gamma_{i,j} \cup A'$
7: **return** Γ such that $\Gamma = \Gamma_{i,j}$ for some pair $i, j \in \{1, 2, 3\}, i \neq j$, and $k(G^\Gamma)$ is minimum

The running time of line 5 is $O(n)$ and thus Algorithm 2 can be implemented in $O(n^3)$ time, as shows a simple inspection of the nested loops.

5.3. Shrinking adjacencies

Shrinking adjacencies is a greedy strategy to find an 1-regular graph Γ with vertex set $V(G)$ of a given graph G defined by three 1-regular graphs Π_1, Π_2 and Π_3 each one with vertex set $V(G)$.

Remember that G has $2n$ vertices and $3n$ edges. Let u, v be vertices in $V(G)$. Let u_i and v_i be vertices in $V(G)$ such $u_i \neq v$, and vv_i, uu_i are i-colored edges. For each color $i \in \{1, 2, 3\}$, let $G' := G + u_iv_i - uv$ and set color i for the new edge u_iv_i. We call the edge uv a *shrinked adjacency* in G. Notice that G' is a 3-edge-colorable graph. Moreover, notice that a shrinking procedure removes three to six edges and adds zero to three new edges from G to G', depending on how many edges with endpoints u and v there exist in G. Thus, the number of edges in G' is $3n - 3$. Algorithm 3 implements this idea.

Algorithm 3. Shrinking adjacencies.

Input: 3-edge-colorable cubic graph G obtained by the perfect matching in $\Pi = \{\Pi_1, \Pi_2, \Pi_3\}$
Output: 1-regular graph Γ with vertex set $V(G)$ obtained by edges chosen by the shortest cycle criterion

1: **while** there exists an edge in G **do**
2: choose an edge uv in $E(G)$ according to the shortest cycle criterion
3: let u_i and v_i be vertices in $V(G)$ such that $u_i \neq v$, and vv_i, uu_i are i-colored edges
4: **return** uv plus the result of the recursive call of shrinking adjacencies for $G + u_iv_i - uv$

The criterion in the line 2 of Algorithm 3 is described below.

Shortest cycle criterion: for $h \geq 1$, let G_h be a 3-edge-colorable graph given as an instance of h-th recursive call of Algorithm 3.

Notice that each edge in a 3-edge-colorable graph belongs to exactly two bicolored cycles. Thence, in order to describe the shortest cycle criterion, consider a quadruple $(vl(e), rl(e), sh(e), lg(e))$, for each edge $e = uv \in E(G_h)$ such that $vl(e) \in \{0, 1\}$ denotes the contribution value of the edge e; $rl(e) = \texttt{T}$ denotes an edge in G, and $rl(e) = \texttt{F}$ otherwise; and $sh(e)$ and $lg(e)$ are the lengths of the two bicolored cycles whose edge e belongs to, with $sh(e) \leq lg(e)$ (longest and shortest cycles).

Initially, in graph $G = G_1$, $vl(e) = 1$ and $rl(e) = \texttt{T}$ for each edge $e \in E(G)$. Algorithm 3 chooses an i-colored edge $e \in E(G_h)$ according to the following. Suppose that $e = uv$ and u_i, v_i are vertices such that $u_i \neq v$, $e_u = uu_i$ and $e_v = vv_i$ are i-colored edges. When vertices u and v are removed from G_h then, for each i, we define an i-colored edge $e_i = u_iv_i$, set $rl(e_i) = \texttt{F}$ and $vl(e_i) = 1$ if $rl(vv_i) = rl(uu_i) = \texttt{T}$; otherwise, $vl(e_i) = 0$. Notice that removing vertices u and v implies in removing edges which, in turn, implies that the quadruple attributes of all remaining edges, of cycles where those edges belong to, must be updated. This means that in each recursive call $O(n)$ edges must have their attributes updated.

Considering the quadruple attributes, we define a total order $\preceq$ on the set of edges of G_h. Given two edges e_1 and e_2, we say that $e_1 \preceq e_2$ if one and only one of the following conditions holds:

(1) $vl(e_1) > vl(e_2)$, or
(2) $vl(e_1) = vl(e_2)$, $rl(e_1) = \texttt{T}$ and $rl(e_2) = \texttt{F}$, or
(3) $vl(e_1) = vl(e_2)$, $rl(e_1) = rl(e_2)$, $sh(e_1) < sh(e_2)$, or
(4) $vl(e_1) = vl(e_2)$, $rl(e_1) = rl(e_2)$, $sh(e_1) = sh(e_2)$ and $lg(e_1) \leq lg(e_2)$.

We say that $e \in E(G_h)$ is an *optimal edge* for the shortest cycle criterion if $e \preceq g$ for each $g \in E(G_h)$. In each recursion call of Algorithm 3, an optimal edge is chosen.

Finally observe that, in each recursive call, we have to find an edge $e = uv$ according to the shortest cycle criterion, which takes $O(n)$ time, and we have to remove u and v from G_h, which takes $O(1)$ time. Then, we have to update the quadruple attributes of all edges in cycles involved in this operation and it can be performed in $O(n)$ time. Thus, Algorithm 3 spends time $O(n)$ in each recursive call and therefore its running time is $O(n^2)$.

5.4. Edge scores

Let $\mathcal{R}$ be the set of *reliable edges*, and an edge e in G belongs to $\mathcal{R}$ if $\mu(e) \geq 2$. Let Γ be a 1-regular graph with vertex set $V(G)$. Define the *score* s of an edge uv in Γ as $s(uv) = t + \frac{1}{2}f$, where t is the number of i-colored 2-cycles and f is the number of i-colored 4-cycles such that uv belongs to them in G^Γ, with $i \in \{1, 2, 3\}$. It is easy to see that $0 \leq t + f \leq 3$. Let $s(\Gamma) := \sum_{uv \in \Gamma} s(uv)$ and observe that $s(\Gamma) = k(G^\Gamma)$. The two edges in Γ of an i-colored 4-cycle in G^Γ are called *sibling edges*. Figure 1b shows G^Γ such that $s(1^t4^h) = \frac{3}{2}, s(1^h3^t) = 1, s(3^h4^t) = 2$, and $s(2^t2^h) = \frac{5}{2}$.

Afterwards, for each edge uv in Γ, define its *cycle potential* λ in G^Γ as $\lambda(uv) = \frac{1}{2}(\mu(uv) + 3) - s(uv)$. The cycle potential $\lambda(uv)$ of an edge uv in Γ represents the possibility of collaboration of uv in other 2- and 4-cycles in G^Γ. Referring again to the graph G^Γ in Figure 1b, we have $\lambda(1^t4^h) = \lambda(1^h3^t) = \lambda(3^h4^t) = \frac{1}{2}$ and $\lambda(2^t2^h) = 0$.

Algorithm 4 starts choosing reliable edges and arbitrary remaining edges to be part of an initial solution, obtaining a 1-regular graph Γ with vertex set $V(G)$. Then, it computes score and cycle potential for each edge in Γ. The next step is trying to increase the score of an edge, and to decrease the cycle potential of a small subset of edges as well, through local changes. Let uv be an edge in Γ such that $\lambda(uv) > 0$. Subsequently, for $i \in \{1, 2, 3\}$, there is at least one pair of i-colored edges in G, say uu_1 and vv_1, such that $u_1v_1 \notin \Gamma$. Since Γ is a perfect matching, u_1 and v_1 are saturated vertices and let u_1u_2 and v_1v_2 be these edges in Γ. Now, the algorithm removes u_1u_2, v_1v_2 and adds u_1v_1, u_2v_2 to Γ and the score and cycle potential of the edge uv must be updated, as well as for the sibling edges of the removed edges u_1u_2 and v_1v_2. Additionally, the score and cycle potential of the new edges u_1v_1 and u_2v_2 must be computed. Algorithm allows this operation if and only if u_1u_2 and v_1v_2 are not reliable edges. It repeats this process while the sum of the scores of all edges increases from one step to the next and there is an edge with positive cycle potential.

Algorithm 4. Edge scores.

Input: 3-edge-colorable cubic graph G obtained by the perfect matchings in $\Pi = \{\Pi_1, \Pi_2, \Pi_3\}$
Output: 1-regular graph Γ with vertex set $V(G)$ such that $s(\Gamma)$ is maximum

1: let $\mathcal{R}$ be the set of reliable edges of G
2: let Γ be a 1-regular graph comprised of $\mathcal{R}$ and arbitrary remaining edges
3: compute $s(uv), \lambda(uv)$ for each edge uv in Γ
4: **if** there exists an edge uv in Γ such that $\lambda(uv) > 0$ **then**
5: let uu_1, vv_1 be i-colored edges, $u_1v_1 \notin \Gamma$, $i \in \{1, 2, 3\}$
6: let u_1u_2, v_1v_2 be edges in Γ
7: **if** $u_1u_2, v_1v_2 \notin \mathcal{R}$ **then**
8: $\Gamma = \Gamma + \{u_1v_1, u_2v_2\} - \{u_1u_2, v_1v_2\}$
9: update $s(uv), \lambda(uv)$
10: update s, λ for sibling edges of $u_1u_2, v_1v_2, u_1v_1, u_2v_2$
11: compute $s(u_1v_1), \lambda(u_1v_1)$ and $s(u_2v_2), \lambda(u_2v_2)$
12: repeat lines 4–11 while $s(\Gamma)$ can be increased without removing edges in $\mathcal{R}$
13: **return** Γ

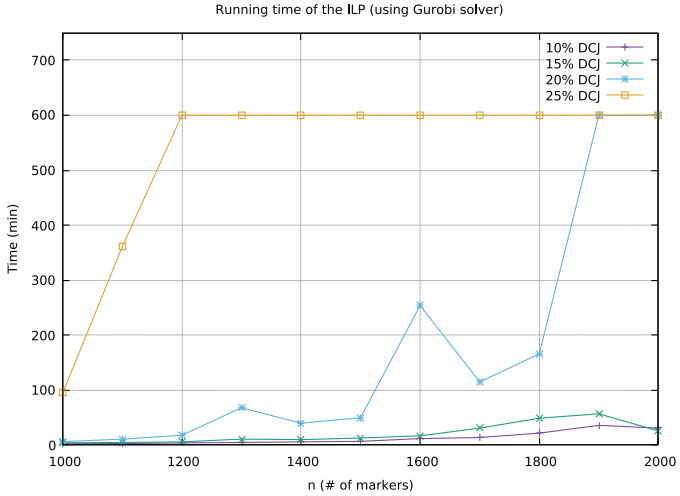


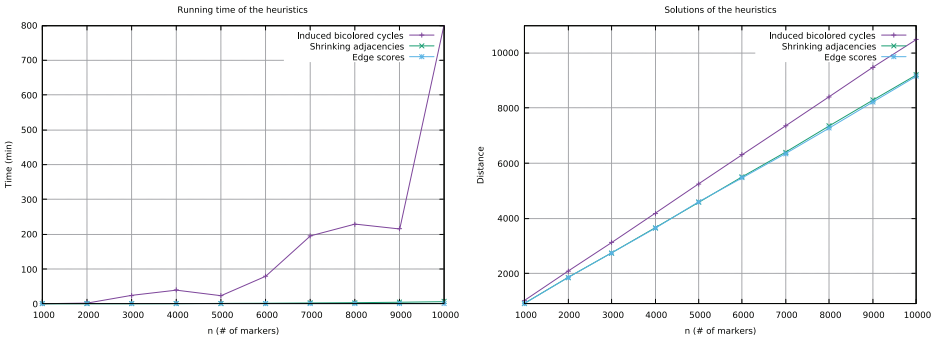
FIGURE 5. For multiple genome sizes and k values, running times of the ILP solver in minutes.

Observe that the search in line 4, for an edge in Γ with cycle potential positive, takes $O(n)$ time. Besides that, each line from 5 to 11 can be performed in constant time. Since $s(\Gamma)$ can be increased $O(n)$ times, we have that the running time of Algorithm 4 is $O(n^2)$.

6. EXPERIMENTS AND PERFORMANCE EVALUATION

We simulated multiple genomes in order to (i) sketch the boundaries where our ILP (Algorithm 1) can perform in reasonable time whilst providing an acceptable accuracy, and (ii) evaluate the quality and running times of our heuristics (Algorithms 2–4). Experiments were run using 3.6 GHz CPUs. We implemented the heuristics in Python 3 and used Gurobi 9.0.2 as ILP solver with 8 cores.

The simulated instances were built as follows. Given a root genome with n markers, a *descendant trio* is a set of three genomes, each one generated by simulating independently $\frac{n}{100} \cdot k$ random DCJs in the root genome (*i.e.*, k is a percentage of the root genome size n ranging in $\{10, 15, 20, 25\}$).



(A) (B)

FIGURE 6. For genomes with number of markers ranging in 1000 to 10 000 and $k = 25$, (a) running times and (b) c_4-distances of the heuristics.

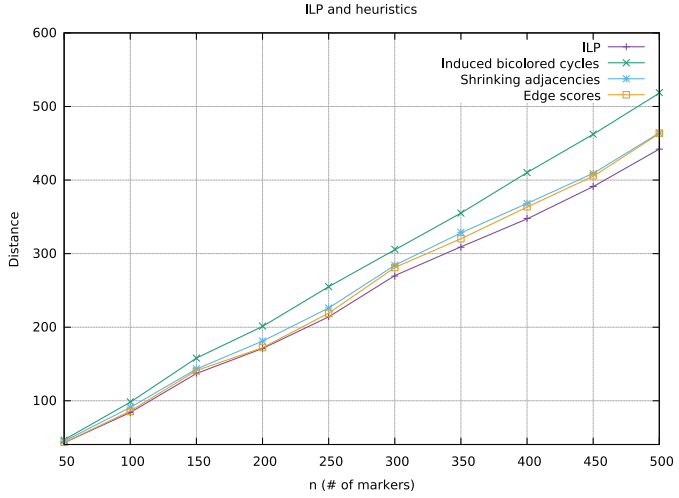


FIGURE 7. Optimal distances obtained by the exact ILP algorithm *versus* distances obtained by the heuristics, for genome sizes in $\{50, 100, 150, \ldots, 500\}$ and $k = 25$.

In order to evaluate the performance of Algorithm 1 for large genomes, we generated root genomes with $1000, 1100, 1200, \ldots, 2000$ markers distributed in several circular chromosomes and then for each root genome, four descendent trios ranging k in $\{10, 15, 20, 25\}$. In the solver execution command, we set the time limit to 10 h. For this dataset, the solver exceeded the time limit for genomes with $n > 1100$ markers when $k = 25$. See Figure 5.

Second, in order to stress the heuristics and evaluate their performance, we simulated datasets with large genomes, from 1000 to 10 000 markers and $k = 25$. Figure 6 shows running times and c_4-distances for Algorithms 2–4. Algorithm 4 always returned the smallest c_4-distances, followed by Algorithm 3 with a difference of 0.71%, on average.

Finally, after performing experiments with ILP and heuristics, we compare the returned distances. Due to the ILP limitations, we first generate root genomes with $50, 100, 150, \ldots, 500$ markers spread over several circular chromosomes and then, for each root genome, four descending triplets with $k = 25$. We then carried out comparisons of the results obtained by the three heuristics with the optimal distances obtained by the ILP.

Figure 7 shows that the "Edge scores" heuristic returns distances closer to the optimal ones with a difference of 2.8% on average. Then the heuristics "Shrinking adjacencies" and "Induced bicolored cycles" have, respectively, differences of 5.5% and 15.9% on average in comparison to the optimal distances.

7. Conclusion

In this work we study the genome median problem under the c_4-*distance*, a restricted measure of rearrangement. We show bounds and properties concerning this measure, and establish connections with previous works on the breakpoint and the DCJ median problems. We also develop algorithms, one exact ILP-based and three combinatorial heuristics, which allow us to perform experiments on simulated data sets and to provide many insights about the problem. Moreover, this work offers many perspectives for future research as detailed below.

From the theoretical perspective, the computational complexity of the problem is still open, although we conjecture it is NP-hard. Additionally, there is room to deepen the relationships between this and Xu's work [12], especially with respect to decomposers and adequate subgraphs.

Algorithms proposed in this work give a practical perspective to the problem and allow to compare their results to those for the breakpoint and the DCJ median, and we will do so in future work. Particularly, the bounds obtained in Section 3 can give support to design a strategy to speed up Algorithm 1, such as a branch and bound algorithm. On the other hand, these bounds can help us to establish approximations factors to the developed heuristics (Algorithms 2–4). Besides that, a real data analysis, of single-celled organisms or mitochondrial DNA of more complex organisms should give us more information about the behaviour of this measure in practice. Furthermore, we can possibly extend this work to multichromosomal linear genomes, using a classic approach to deal with *capping* [7, 8, 11], in which we can safely transform linear into circular chromosomes and preserve the distances.

Acknowledgements. We want to thank Jens Stoye and Cedric Chauve for introducing this topic and for all the fruitful discussions about it.

References

[1] V. Bafna and P.A. Pevzner, Genome rearrangements and sorting by reversals. *SIAM J. Comput.* **25** (1996) 272–289.
[2] A. Bergeron, J. Mixtacki and J. Stoye, A unifying view of genome rearrangements, in Proc. of WABI. Vol. 4175 of *LNBI*. Springer Berlin Heidelberg (2006) 163–173.
[3] A. Caprara, The reversal median problem. *INFORMS J. Comput.* **15** (2003) 93–113.
[4] D. Doerr, M. Balaban, P. Feijão and C. Chauve, The gene family-free median of three. *Algorithm Mol. Biol.* **12** (2017) 1–14.
[5] P. Feijão and J. Meidanis, SCJ: a breakpoint-like distance that simplifies several rearrangement problems. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **8** (2011) 1318–1329.
[6] G. Fertin, A. Labarre, I. Rusu, E. Tannier and S. Vialette, Combinatorics of Genomes Rearrangements. The MIT Press (2009).
[7] S. Hannenhalli and P. Pevzner, Transforming men into mice (polynomial algorithm for genomic distance problem), in Proc. of FOCS 1995. IEEE (1995) 581–592.
[8] G. Jean and M. Nikolski, Genome rearrangements: a correct algorithm for optimal capping. *Inf. Process. Lett.* **104** (2007) 14–20.
[9] J. Kováč, On the complexity of rearrangement problems under the breakpoint distance. *J. Comput. Biol.* **21** (2013) 1–15.
[10] E. Tannier, C. Zheng and D. Sankoff, Multi-chromosomal median and halving problems under different genomic distances. *BMC Bioinf.* **10** (2009) 1–15.
[11] A.W. Xu, DCJ median problems on linear multichromosomal genomes: graph representation and fast exact solutions, in Proc. of RECOMB-CG. Vol. 5817 of *LNCS*. Springer Berlin Heidelberg (2009) 70–83.
[12] A.W. Xu, A fast and exact algorithm for the median of three problem: a graph decomposition approach. *J. Comput. Biol.* **16** (2009) 1369–1381.
[13] A.W. Xu and D. Sankoff, Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem, in Proc. of WABI. Volume 5251 of *LNBI*. Springer (2008) 25–37.
[14] S. Yancopoulos, O. Attie and R. Friedberg, Efficient sorting of genomic permutations by translocation, inversion and block interchanges. *Bioinformatics* **21** (2005) 3340–3346.

Please help to maintain this journal in open access!

This journal is currently published in open access under the Subscribe to Open model (S2O). We are thankful to our subscribers and supporters for making it possible to publish this journal in open access in the current year, free of charge for authors and readers.

Check with your library that it subscribes to the journal, or consider making a personal donation to the S2O programme by contacting subscribers@edpsciences.org.

More information, including a list of supporters and financial transparency reports, is available at https://edpsciences.org/en/subscribe-to-open-s2o.