APPROXIMATION ALGORITHMS FOR THE MEDIAN PROBLEM IN THE BREAKPOINT MODEL

Itsik Pe'er
Department of Computer Science
Tel Aviv University
izik@math.tau.ac.il

Ron Shamir
Department of Computer Science
Tel Aviv University
shamir@math.tau.ac.il

Abstract

The problem of genome rearrangement is a central problem in computational molecular biology. When trying to determine the evolutionary history of three species it gives rise to the median problem between permutations. Recently, the model of breakpoint analysis was suggested for tackling this problem. We consider the median problem for signed permutations, counting breakpoints to measure genomic distance. We give polynomial time approximation algorithms for this problem, attaining an approximation ratio of $\frac{7}{6}$.

1 Introduction

The study of genome rearrangements has drawn a lot of attention in recent years. Large amounts of genomic data on various organisms become available rapidly, and for the first time they enable a large scale study of evolutionary relations between species by comparing the order in which common genes appear along their chromosomes. Changes in gene order are much less frequent than point mutations, and therefore, in principle, one can elucidate the evolutionary history of speciation more precisely and further backwards in time using gene orders. This is done by comparing gene orders in the studied species and reconstructing the sequences of genome rearrangements that have transformed the ancestral genome species into the contemporary species. Particularly, for two extant species, one wishes to explain the lineages leading to them from their common ancestral species in the most parsimonious manner, i.e., by as few rearrangement events as possible.

When restricting the discussion to only one chromosome, and assuming all genes occur exactly once per genome, gene orders are modeled as permutations. When taking into account the orientation of each gene, the objects discussed are *signed* permutations (see e.g., Hannenhalli and Pevzner (1999)). In that case, a reversal operation on a segment of a permutation inverts not only the segment order, but also the signs of the elements in the segment. The complexity of the problem has been well studied under the assumption that all rearrangement events are *reversals* (inversions of a chromosome segment). This problem, of finding the *reversal distance* between two permutations, is already NP-hard in the case where gene orders are given as unsigned permutations (Caprara, 1999b). However, the problem is polynomial if the permutations are signed (Hannenhalli and Pevzner, 1999; Berman and Hannenhalli, 1996; Kaplan et al., 2000). Other models of the genome (Hannenhalli, 1996; Kececioglu and Ravi, 1995) or of the assumed events were also studied (Hannenhalli and Pevzner, 1995; Hannenhalli, 1996; Sankoff and Nadeau, 1996; Bafna and Pevzner, 1998; DasGupta et al., 1998). When studying more than two species the key problem arising is to reconstruct the phylogenetic tree achieving minimal total distance along tree branches (most parsimonious), given only permutations (gene orders) at the leaves (contemporary species). When the topology of the tree is restricted to a star, this problem is known as the *median problem* (Sankoff et al., 1996; DasGupta et al., 1998) which is known to be NP-hard already for three species, counting reversals (Caprara, 1999a).

Recently, Sankoff and Blanchette (1997) introduced a new model for studying the reconstruction of evolutionary tree of more than two species. They argued forcefully that the reversal distance and similar distance measures have certain weakness that renders them inappropriate for studying complex trees, and suggested a simpler criterion, merely counting, for every two permutations connected by a branch of the tree, the number of *breakpoints* between them, i.e., how many pairs of adjacent elements in one permutation are not adjacent in the other. This *breakpoint distance* can be easily computed for a single branch, for signed and unsigned permutations.

Sankoff and Blanchette (1997) studied the *median problem for breakpoints*, and developed several efficient heuristics for the problem. This problem was subsequently shown to be NP-complete as well (Pe'er and Shamir, 1998; Bryant, 1998). We provide constant factor polynomial approximation algorithms for the signed model.

The paper is organized as follows: Section 2 provides necessary notation and definitions. Section 3 presents a very straightforward $\frac{6}{5}$-approximation algorithm. Section 4 presents a more subtle algorithm and tighter analysis, achieving an approximation ratio of $\frac{7}{6}$. In section 5 we derive an approximation algorithm for the median problem on four species.

2 Preliminaries

All graphs in this paper are finite, and unless specifically indicated otherwise, undirected. Self loops are not allowed, but parallel edges are: The number of edges between two vertices u, v may be 0,1,2 or 3. When this number is greater than one, we refer to the multiple edges as a single entity uv, called a *double* or *triple* edge, respectively. For a set of $2n$ vertices, a *perfect matching* is a set of n edges, incident on every vertex. A *Hamiltonian cycle* is a set of edges forming one cycle, that passes once through each vertex.

We briefly review standard terminology of genome rearrangements (see also Pe'er and Shamir (1998)). We denote a signed permutation π on n elements by the n-tuple $(\pi(1), \pi(2), \ldots, \pi(n))$, with $\pi(i) \in \{\pm 1, \ldots, \pm n\}$, and $|\pi(i)| \neq |\pi(j)|$ for $i \neq j$. We rescale a signed permutation on n elements to an unsigned permutation on $2n$ elements, with each positive element $\pi(j) = i$ mapped to the pair $\pi(2j-1) = 2i-1, \pi(2j) = 2i$ and each negative element $\pi(j) = -i$ mapped to the pair $\pi(2j-1) = 2i, \pi(2j) = 2i-1$. We further expand an unsigned permutation of $1, \ldots, 2n$ by defining $\pi(0) = 0$, and $\pi(2n+1) = 2n+1$. We call this transformation of a signed n element permutation into an unsigned $2n+2$ elements the *standard augmentation.*

For an unsigned permutation π, $\pi(i)$ and $\pi(i+1)$ are said to be *successive* in π. For a signed permutation π', standardly augmented to an unsigned permutation π, $\pi(2i)$ and $\pi(2i+1)$ are said to be *successive* in π'.

Let π, σ be signed permutations of n elements. A pair of successive elements in π which are also successive in σ is called an *adjacency.* A pair of successive elements in π which are not successive in σ is called a *breakpoint* in π w.r.t. σ. The number of breakpoints in π w.r.t. σ is denoted by $d(\pi, \sigma)$.

For example, the standard augmentation of the signed permutation $\tau = (+1, -3, -2)$ is $0, 1, 2, 6, 5, 4, 3, 7$, while that of $\rho = (+2, +3, -1)$ is $0, 3, 4, 5, 6, 2, 1, 7$. The breakpoints in τ w.r.t ρ are $\{0, 1\}$ and $\{3, 7\}$, thus $d(\tau, \rho) = 2$.

The **(signed) median problem for breakpoints** is defined as follows:
Instance: Three (signed) permutations π_0, π_1, π_2.
Goal: Find a (signed) permutation $\widehat{\pi}$ s.t. $\sum_i d(\pi_i, \widehat{\pi})$ is minimal.
This paper analyzes only with an equivalent representation of this problem from a graph theoretic approach, which we now explain.

Let $V = \{v_i\}_{i=0}^{2n-1}$ be a set of $2n$ vertices. We call the perfect matching $M_b = \{(v_{2i}v_{2i-1})\}_{i=0}^{n-1}$ the *base matching* on V (subscripts for V are calculated modulo $2n$). A perfect matching M on V is called a *Hamiltonian matching* (w.r.t. M_b), if $M_b \cup M$ forms a Hamiltonian cycle.

For a Hamiltonian matching M, the edge set $M_b \cup M \setminus \{v_0 v_{2n-1}\}$ forms a Hamiltonian path. We now define a signed permutation $signed(M)$, according to the order, and the direction in which M_b edges are traversed along this path, from v_0 to v_{2n-1}. Fix $signed(M)_j$ to be i $(-i)$, if the j-th M_b edge in this traversal is from v_{2i-1} to v_{2i} (from v_{2i} to v_{2i-1}, respectively). Observe, that for every signed permutation π on $n-1$ elements, denoting its standard augmentation into an unsigned permutation on $2n$ elements by π', the edge set $hmatch(\pi) =$

$\{v_0 v_{\pi'(1)}, v_{\pi'(2)} v_{\pi'(3)}, \dots, v_{\pi'(2n-4)} v_{\pi'(2n-3)}, v_{\pi'(2n-2)} v_{2n-1}\}$ is a Hamiltonian matching with respect to the base matching on $2n$ vertices.

Let π, σ be two signed permutations on $n-1$ elements. Each adjacency between them corresponds to an edge common to both $hmatch(\pi)$ and $hmatch(\sigma)$ and vice versa.

Corollary 2.1 *(Pe'er and Shamir, 1998) Let π and σ be signed permutations on $n-1$ elements. Then $bp(\pi, \sigma) = n - |hmatch(\pi) \cap hmatch(\sigma)|$*

The problem of **Consensus of 3 Hamiltonian Matchings** (***C3HM*** for short) is defined as follows:
Instance: Three Hamiltonian matchings M_0, M_1, M_2 w.r.t. the base matching M_b on a set V of $2n$ vertices. For any Hamiltonian matching M, define $w(M) \equiv \sum_i |M \setminus M_i|$.
Goal: Find a Hamiltonian matching $\widehat{M}$ w.r.t. M_b, with minimal $w(\widehat{M})$.

We extend the weight function to edges, by defining $w(e) = |\{i | e \notin M_i\}|$. Then for a Hamiltonian matching M:

$$w(M) = \sum_{e \in M} w(e) \tag{1}$$

Proposition 2.2 *(Pe'er and Shamir, 1998) C3HM and the signed permutations median problem are equivalent.*

Formally: the instance (V, M_b, M_0, M_1, M_2) of the C3HM problem has a solution of value k iff the instance $(signed(M_0), signed(M_1), signed(M_2))$ of the signed permutations median problem has a solution of value k. Moreover, the mapping between the solutions of these two problems guarantees that approximating one of them with a ratio R implies an R-approximation to the other as well.

For a C3HM instance, we call $G(V, M_b \cup M_0 \cup M_1 \cup M_2)$ the *consensus graph*. W.l.o.g., we assume there are no triple edges in G: Otherwise, such an edge trivially belongs to an optimal consensus Hamiltonian matching, and can be dropped, contracting its incident vertex pairs to one pair. In such a case, we can solve the contracted instance, and the approximation ratio of our solution would only improve upon re-insertion of the triple edge.

Let $E^1(G)$ (respectively, $E^2(G)$) be the set of single (double) edges of G. Let $E^0(G)$ be the set of all edges of the complete graph that are not in G. Let $\bar{G}$ be the complete graph on V, with no parallel edges and edge weights $w : E(\bar{G}) \mapsto \{-\infty, 1, 2, 3\}$, defined as follows:

$$w(e) = \begin{cases} 3 & \text{if } e \in E^0 \\ 2 & \text{if } e \in E^1 \\ 1 & \text{if } e \in E^2 \\ -\infty & \text{if } e \in M_b \end{cases}$$

$(\bar{G}, w)$ can be considered an instance of the Traveling Salesperson Problem (TSP). Solving the original C3HM instance is equivalent to solving TSP on $(\bar{G}, w)$ (Sankoff and Blanchette, 1997, 1998), with the weight $-\infty$ replaced by a finite negative constant with large absolute value.

Let G_2 be the subgraph of G whose edge set is $E^2 \cup M_b$. Since each vertex is incident on exactly one M_b edge, and on at most one E^2 edge, G_2 is a collection of paths and cycles with alternating E_2-M_b edges. We partition E^2 edges to a set C of *cycle edges* and a set P of *path edges*, according to their role in G_2. If G_2 is one $2n$-long cycle, then the solution is trivial (E^2 is then an optimal solution) so we ignore this case in the rest of the paper. Assume G_2 is a collection of n_c cycles and n_p paths. The *length* of a cycle or path is the number of E^2 edges in it. Let l_c (l_p) denote the total length of all the cycles (paths, respectively). A pair of vertices connected by an edge of M_b is considered a path of length 0. All the cycles contain a total of l_c edges of M_b, while all the paths contain a total of $l_p + n_p$ edges of M_b. Observe that a Hamiltonian matching cannot include all the E^2 edges of a cycle, therefore

$$r \equiv |E^2| - n_c = l_p + l_c - n_c \tag{2}$$

is an upper bound on the number of E^2 edges used by a solution. Note that

$$n - r = n_p + n_c \tag{3}$$

3 A Simple Approximation for C3HM

3.1 Outline

In this section we first prove that the optimal median solution value w_{opt} satisfies $w_{opt} \geq 2n - r$. We also provide two simple polynomial approximation algorithms, $\mathcal{A}$ and $\mathcal{B}$, with guaranteed solution values of $w_{\mathcal{A}} \leq 2n - \frac{2r}{3}$ and $w_{\mathcal{B}} \leq 3n - 2r$. The approximation ratio of algorithm $\mathcal{A}$ is bounded by $\frac{2n-\frac{2r}{3}}{2n-r}$, which is an increasing function of r, while the approximation ratio of algorithm $\mathcal{B}$ is bounded by $\frac{3n-2r}{2n-r}$, which decreases when r increases. The threshold value of r, in which these two bounds meet, is $r_t = \frac{3n}{4}$. By applying both algorithms $\mathcal{A}$ and $\mathcal{B}$, and choosing the better solution, we get a guaranteed solution value $w_{approx} \leq \min(2n - \frac{2r}{3}, 3n - 2r)$. The worst approximation ratio guarantee we thus get is for $r = r_t$, and then

$$\frac{w_{approx}}{w_{opt}} \leq \frac{\min(2n - \frac{2r}{3}, 3n - 2r)}{2n - r} \leq \frac{3n - 2 \cdot \frac{3n}{4}}{2n - \frac{3n}{4}} = \frac{6}{5} \tag{4}$$

3.2 A Bound on w_{opt}

Let M be an optimal solution. Since $M \cup M_b$ is a single cycle, for each cycle C_i in G_2, there is at least one E^2 edge not in M. Hence, $|M \cap (\cup_i C_i)| \leq l_c - n_c$, and $|M \cap E^2| \leq l_p + l_c - n_c = r$. It follows that:

$$w_{opt} \geq 1 \cdot r + 2 \cdot (n - r) = 2n - r \tag{5}$$

3.3 Approximation Algorithm $\mathcal{A}$

Let $d_i = |M_i \cap E^2|$. Then $d_0 + d_1 + d_2 = 2|E^2| = 2(r + n_c)$. Therefore there is some i for which $d_i \geq \frac{2(r+n_c)}{3}$. Our approximate solution is simply M_i. The value of that solution satisfies:

$$w_{\mathcal{A}} = \sum_{e \in M_i} w(e) = 1 \cdot d_i + 2 \cdot (n - d_i) = \leq 2n - \frac{2(r + n_c)}{3} \leq 2n - \frac{2r}{3} \qquad (6)$$

Note that M_i is the input Hamiltonian matching, whose sum of disagreement distances from the two other input Hamiltonian matchings is minimal. Such an approximation for the median problem is known Sankoff et al. (1996) for any kind of metric, attaining an approximation ratio $\frac{4}{3}$, based on the triangle inequality. Eq [6] provides a better bound on the performance of this approximation, when the metric at hand is the number of breakpoints.

3.4 Approximation Algorithm $\mathcal{B}$

For each cycle C_i in G_2, arbitrarily choose some edge e_i, and denote $E' = E^2 \setminus \{e_i\}_{0 \leq i < n_c}$. The graph $H(V, E' \cup M_b)$ is a collection of disjoint paths. Concatenate these paths to a Hamiltonian cycle (which passes through all the edges of M_b), by connecting path endpoints arbitrarily. Let E'' be the set of those arbitrarily added edges. $E' \cup E''$ is our approximated solution. As $|E'| = r$ and $|E''| = n - r$, we conclude

$$w_{\mathcal{B}} = \sum_{e \in E' \cup E''} w(e) \leq 1 \cdot |E'| + 3 \cdot |E''| \leq 3n - 2r \qquad (7)$$

4 Better Approximation

4.1 Outline

The bound in Eq [5] can be attained by an optimal solution if and only if such a solution employs the maximum number, r, of E^2 edges, and all the rest of the edges are E^1 edges. In this ideal scenario:

- All path edges are used.
- All cycle edges are used, except for one edge per cycle, breaking it into a path.
- The paths and broken cycles are chained by E^1 edges.

In such a case none of the paths is broken, and every cycle is broken only once. Whenever the ideal bound in Eq [5] is not attained by a solution, the discrepancy between the solution and the ideal bound can be attributed to deviations from the ideal scenario, namely, to edges not of the above specified types, called *non-ideal* edges. We call such a discrepancy *waste*, and charge non-ideal edges for this waste.

In this section we formalize these notions, distinguishing between several kinds of waste, according to the non-ideal edges charged for it. We devise a better approximation algorithm, which tries to conform with the ideal scenario as much as possible, attempting to minimize the usage of non-ideal edges, thus reducing the waste. In particular, we try to chain paths and (broken) cycles by E^1 edges whenever possible. More specifically:

- We use E^1 edges which comprise a matching on a set of cycles and connected components of paths. These edges chain broken cycles and paths, without any implied waste.
- We greedily choose E^1 edges which are not charged for more waste.

Furthermore, we observe that sometimes waste is inevitable. We characterize and quantify some of the inevitable waste, thus obtaining a better bound for the optimal solution. The new algorithm and the new bound guarantee a $\frac{7}{6}$ approximation ratio.

4.2 Some More Definitions

Each vertex v in $V(G) = V(G_2)$ is either a cycle vertex in G_2, or an internal path vertex, or a path endpoint. In these cases, we say that v is a *c-vertex*, an *i-vertex*, or a *p-vertex*, respectively. Let I be the set of all i-vertices and p-vertices, and let $M_b[I]$ be the set of M_b edges that belong to paths of G_2. We distinguish between six categories of edges in $E^0 \cup E^1 \cup E^2$ according to the vertices they are incident on: E_{cc}, E_{ii}, E_{pp}, E_{ci}, E_{cp} and E_{ip}. For $i = 0, 1, 2$, define $E^i_{cc} = E^i \cap E_{cc}$, and similarly for the rest of the edge categories. We define the subgraph $G_{pp}(I, E^2_{ii} \cup E^1_{pp} \cup M_b[I])$ of G, including all the paths of G_2, and all the single edges of G connecting their endpoints (see figure 1, top middle).

Construct a bipartite graph $H_{cp}(A \cup B, F_{cp})$, with $A = \{a_v | \text{p-vertex } v\}$ and $B = \{b_i | \text{cycle } C_i\}$; for each p-vertex v and cycle C_i, whenever the set $X_{iv} = \{uv \in E^1_{cp} | u \in C_i\}$, F_{cp} includes the edge $a_v b_i$, which is the *representative* of all X_{iv} edges (see figure 1, bottom right). Define also the graph $H_{cc}(A \cup B, F_{cp} \cup F_{cc})$, with F_{cc} being the set of all representatives $b_i b_j$ of edges $uv \in E^1_{cc}$, with $u \in C_i$ and $v \in C_j$. Define $H'_{cp}(A' \cup B, F'_{cp})$, with a vertex a'_S for each connected component S in G_{pp}. Set F'_{cp} to include all representatives $a'_S b_i$ of edges $uv \in E^1_{cp}$, for which $u \in S$, $v \in C_i$. Let M_{cc}, M_{cp},M'_{cp} be maximum matchings of H_{cc}, H_{cp}, and H'_{cp} respectively (see figure 1, bottom right). Surely, $|M_{cc}| \geq |M_{cp}| \geq |M'_{cp}|$. We use M_{cp} and M'_{cp} only for analyzing performance, while M_{cc} is actually computed by our algorithm.

For a Hamiltonian matching M, recall that $w(M) \geq (2n-r)$, by Eq [5]. Define the *waste* of M as the gap between this bound and the actual weight:

$$waste(M) \equiv w(M) - (2n - r)$$

Substituting Eq [1] and Eq [2], we obtain:

$$waste(M) = |E^2 \setminus M| - n_c + |E^0 \cap M| \tag{8}$$

Let the *i-waste* of a Hamiltonian matching M be

$$iwaste(M) \equiv 2|M \cap (E^0_{ii} \cup E^1_{ii})| + |M \cap (E_{ci} \cup E_{ip})|$$

Claim 4.1 *A Hamiltonian matching M with i-waste$(M) = 2x$, uses no more than $l_p - x$ path edges.*

Proof: Define $W_I = \{\text{i-vertex } v | \exists e(v) = uv \in (E^0 \cup E^1) \cap M\}$, and observe that $iwaste(M) = |W_I|$. Each $v \in W_I$ has a path edge $e_2(v) \in E^2_{ii}$ incident on it. Since $e_2(v) \neq e(v)$, and they both share a vertex, surely $e_2(v) \notin M$. Therefore:

$$|M \cap E^2_{ii}| = |E^2_{ii}| - |E^2_{ii} \setminus M| \leq l_p - |\{e_2(v) | v \in W_I\}| \leq l_p - \frac{|W_I|}{2} = l_P - x$$

■

Let the *c-waste* of a Hamiltonian matching M be

$$cwaste(M) \equiv 2|M \cap (E^1_{cc} \cup E^0_{cc})| + |M \cap (E_{ci} \cup E_{cp})| - 2n_c$$

Claim 4.2 *A Hamiltonian matching M with c-waste$(M) = 2y$ uses no more than $l_c - y - n_c$ cycle edges.*

Proof: Define $W_C = \{\text{c-vertex } v | \exists e(v) = uv \in (E^0 \cup E^1) \cap M\}$, and observe that $cwaste(M) = |W_C| - 2n_c$. Each $v \in W_C$ has a cycle edge $e_2(v) \in E^2_{cc}$ incident on it. Since $e_2(v) \neq e(v)$, and they both share a vertex, surely $e_2(v) \notin M$. Therefore:

$$|M \cap E^2_{cc}| = |E^2_{cc}| - |E^2_{cc} \setminus M| \leq l_c - |\{e_2(v) | v \in W_C\}| = l_c - y - n_c$$

■

Let the *o-waste* of a Hamiltonian matching M be $owaste(M) \equiv 2|M \cap E^0|$. Eq [8], claim 4.1, and claim 4.2 imply:

Corollary 4.3 $2waste(M) = cwaste(M) + owaste(M) + iwaste(M)$

4.3 Algorithm $\mathcal{C}$

For a set X of edges, and an edge e, we define the *neighborhood* of e in X, $N_X(e)$, to be the set of all edges in X sharing a vertex with e. Algorithm $\mathcal{C}$ is as follows (refer to Figure 2 for a step-by-step application of the algorithm to the example of Figure 1):

1. Initialize $M, \tilde{M} \leftarrow \phi$.
2. Construct H_{cc} and find a maximum matching M_{cc}, in it. For each edge xy in M_{cc}, if both x,y represent cycles, add to $\tilde{M}$ an edge incident on vertices of these cycles, respectively. Otherwise, x represents a cycle C_x, while y represents a p-vertex v_y in G. In this case, add to $\tilde{M}$ an edge incident on v_y and on a vertex of C_x.

3. Find a forest $F \subseteq E^2_{ii} \cup E^1_{pp}$, s.t. $F \dot\cup M_b$ is a maximum weight spanning forest of G_{pp}.
4. Set $F \leftarrow F \setminus \cup_{e \in \tilde{M}} N_F(e)$.
5. While $F \neq \phi$ do:
   (a) Choose a leaf u of F, with the edge $uv \in F$.
   (b) Add uv to M.
   (c) Set $F \leftarrow F \setminus N_F(uv)$, middle row).
6. Add $\tilde{M}$ to M.
7. For each cycle C_i, add to M all the cycle edges of C_i, except one edge. If C_i is matched by M_{cc}, then choose the omitted edge to be incident on the vertex in $M \cap C_i$.
8. Add arbitrary edges to M to complete a Hamiltonian matching.

4.4 Correctness

Claim 4.4 *Algorithm C produces a Hamiltonian matching*

Proof: It suffices to show that until step 8, M satisfies:

1. M is a matching.
2. $M \cup M_b$ includes no cycles.

- **Edges added in step 5b**:
  Denote the set of all of the edges added to M in step 5b by M_{5b}. Since $M_{5b} \subseteq F$ and $F \cup M_b$ is a forest in G_{pp}, $M_{5b} \cup M_b$ is guaranteed not to form a cycle with path edges, nor with cycle edges (which are never connected to G_{pp} vertices). Furthermore, step 5c makes sure no added edge is incident on a vertex of a previously chosen edge, thus M_{5b} is a matching.
- **Edges added in step 6**:
  All edges added in step 6 have at least one vertex incident on a cycle, all such cycles are distinct, and until step 7 no cycle edges have been considered. Hence, adding $\tilde{M} \subseteq E^1_{cc} \cup E^1_{cp}$ to M (step 6) does not create cycles in $M \cup M_b$. Since no c-vertices were considered till step 6, and $\tilde{M}$ is a matching, no degree 2 c-vertices are created by this step. Step 4 makes sure that no G_{pp} vertices receive degree 2 by adding $\tilde{M}$ to M.
- **Edges added in step 7**:
  Each cycle C_i is connected to the rest of $M \cup M_b$ by at most one edge, added to $\tilde{M}$ in step 2, and merged into M in step 6. Therefore, the path of C_i edges added in step 7 creates no cycles in $M \cup M_b$. C_i has at most one edge e sharing a vertex with an edge in $M \setminus C_i$, and e is never added to M, thus step 7 does not create any vertices of degree 2 in M.

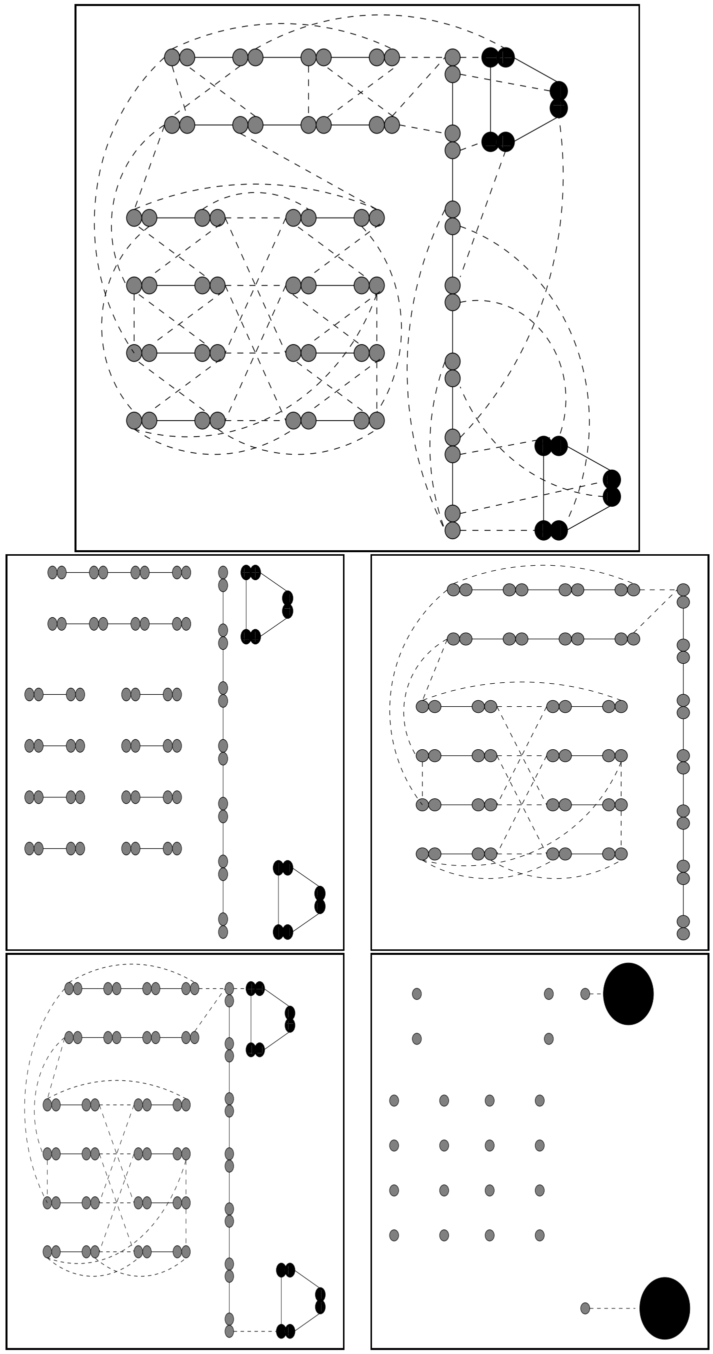

Figure 1: Top left: This is a C3HM instance graph which we shall use for our examples. Solid lines denote double (E^2) edges. Dashed lines denote single (E^1) edges. E^0 edges are omitted. Instead of drawing M_b edges, each pair of adjacent circles denotes a pair of M_b-matched vertices. The c-vertices appear in bold. Top middle: G_2. Top left: G_{pp}. Bottom left: G_{pp}, along with cycle edges and E^1_{cp} edges (in our example, $E^1_{cc} = \emptyset$). Bottom right: H_{cp}. The large nodes denote the cycles C_i. Since in our example $E^1_{cc} = \emptyset$, this is also H_{cc}. Furthermore, since this graph contains only two edges with no common endpoint, $M_{cc} = M_{cp} = F_{cp}$.

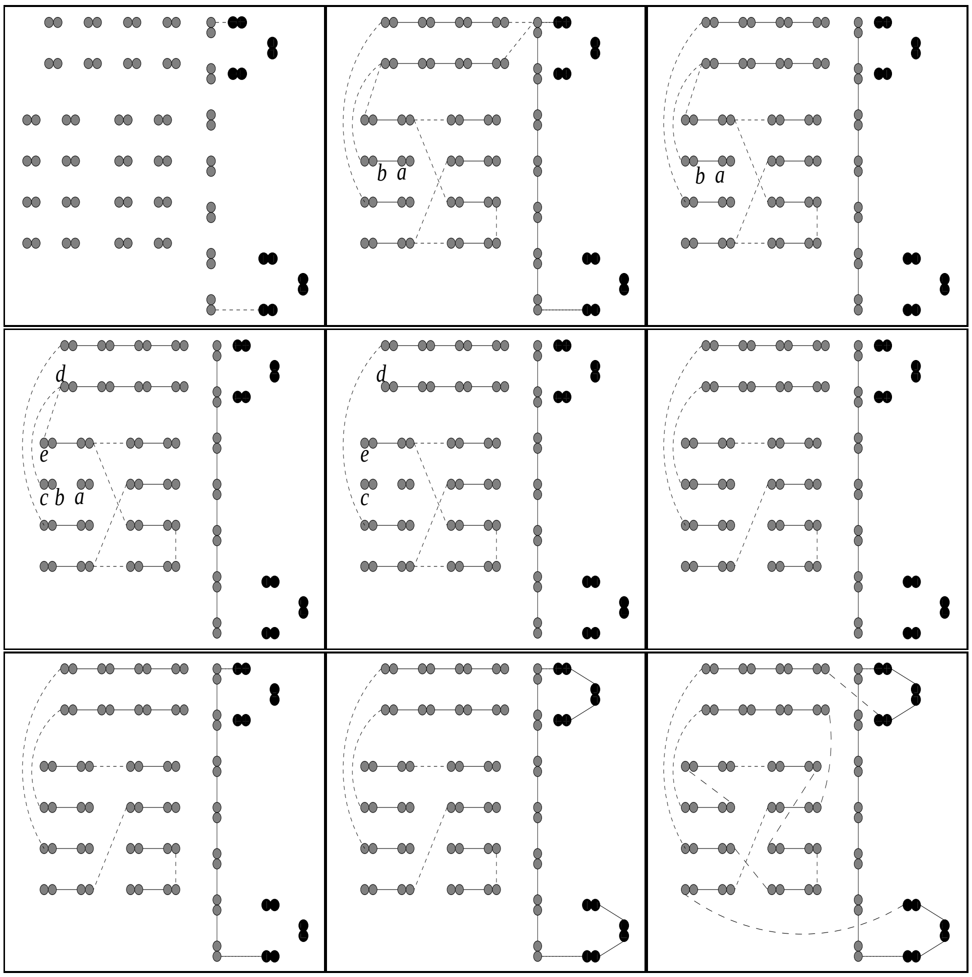

Figure 2: Top left: $\tilde{M}$. Top middle: F, after step 3. $\tilde{M}$ edges are dotted. Top right: F, after discarding the edges and neighboring edges of $\tilde{M}$ (in step 4). Middle-left: F, after the first loop iteration, adding the edge ab to M. Middle: F, after the second loop iteration, adding the edge cd to M, discarding the edge de. Middle-right: M at the end of step 5. All path edges are in M. Bottom-left: M, after adding $\tilde{M}$ (end of step 6). Bottom-middle: M, after adding cycle edges (end of step 7). Bottom-right: The final Hamiltonian matching. The edges added in step 8 appear as longer dashes line.

$\blacksquare$

4.5 A Better Bound on the Optimal Solution

Before analyzing Algorithm C, we provide a better lower bound on the value of the optimal solution, using the terms defined in Section 4.3. Denote by F_0 the forest F when it is first constructed in step 3. Let k be the number of connected components in $F_0 \cup M_b$. Let $m = |M_{cc}|$, $m' = |M'_{cp}|$.

Claim 4.5 *For $k > 2m'$, and any Hamiltonian matching M: $waste(M) \geq \frac{k}{2} - \frac{3m'}{4}$*

Proof: Let M be a Hamiltonian matching, and let the matching M' be a maximum cardinality subset of $E^1_{cp} \cap M$, such that no two edges of M' are incident on vertices of the same connected component of $F_0 \cup M_b$, nor of the same cycle in G_2. M' is a matching, and each of its edges has a unique representative in H'_{cp}. These representatives compose a matching of equal size in H'_{cp}, therefore $|M'| \leq m'$. Let $\mathcal{P}^-$ be the set of all connected components in $F_0 \cup M_b$, unmatched by M'. Obviously, $|\mathcal{P}^-| = k - |M'|$. Let U be the set of all vertices of connected components in $\mathcal{P}^-$, with an edge of M incident on them. The Hamiltonian cycle $M \cup M_b$ must enter and leave each component in $\mathcal{P}^-$ through a U vertex, thus:

$$|U| \geq 2|\mathcal{P}^-| \geq 2k - 2m'$$

Denote the edge of M incident on u by uv_u. Let $\mathcal{C}^+$ be the set of all G_2 cycles matched by M', and let $W \subseteq U$ be a maximal set of vertices x, with each v_x being in a distinct cycle in $\mathcal{C}^+$. Note that none of the vertices in U may be connected by edges of M to cycles outside $\mathcal{C}^+$.

$$|W| \leq |\mathcal{C}^+| = |M'| \leq m'$$

therefore $|U \setminus W| \geq 2k - 3m'$. We charge every $u \in U \setminus W$ for some increase in $waste(M)$, as follows:

- If $uv_u \in E^0$, then $owaste(M)$ increases by 1 on account of u. We charge u for this increase.
- If either u or v_u is an i-vertex, then $iwaste(M)$ increases by at least 1 on the account of uv_u, charging u for half of this increase.
- Otherwise, $uv_u \in E^1_{cp}$. v_u must be on a G_2 cycle C_i in $\mathcal{C}^+$, or else adding uv_u to M' contradicts its maximality. Every such C_i has already one vertex with an (uncharged) M edge incident to it, the edge of M' connected to C_i. Furthermore, C_i has another vertex with an uncharged edge M edge incident to it, connecting it to a vertex of W, otherwise u can be added to W, contradicting its maximality. Hence, if d_i denotes the number of C_i vertices with M edges incident on them, then the sum $\sum_i (d_i - 2)$ increases by 1 on account of v. This sum equals $cwaste(M)$, and we charge u for this increase.

To conclude, u is charged for increasing any of $owaste(M)$, $iwaste(M)$, or $cwaste(M)$, by at least $\frac{1}{2}$. By Corollary 4.3, this implies an increase in $waste(M)$ by at least $\frac{1}{4}$, and no such increase in $waste(M)$ is charged for more than once, hence:

$$waste(M) \geq \frac{|U \setminus W|}{4} \geq \frac{k}{2} - \frac{3m'}{4}$$

■

4.6 Performance Analysis of Algorithm $\mathcal{C}$

Since $F_0 \cup M_b$ is a maximum weight panning forest, F_0 contains all path edges. Throughout the algorithm $F \subseteq F_0$, and for a path edge e, always $N_F(e) \subseteq \{e\}$. Therefore, no such edge is being discarded in step 5c or in step 4, prior to adding it to M.

Connecting a total of n_p paths, F_0 contains also exactly $n_p - k$ edges of E^1_{pp}. No more than $2m$ edges are discarded from F in step 4. Since for every edge added to M in step 5b, at most two edges are discarded from F in step 5c, the total number of E^1_{pp} edges in M is at least $m + \lceil \frac{n_p - k - 2m}{3} \rceil$. M contains exactly r edges of E^2, therefore:

$$\begin{aligned}
w(M) &\leq 1 \cdot r + 2 \cdot (m + \lceil \frac{n_p - k - 2m}{3} \rceil) + 3 \cdot (n - r - (m + \lceil \frac{n_p - k - 2m}{3} \rceil)) \\
&\leq 3n - 2r - m - \frac{n_p - k - 2m}{3} \\
&= 3n - 2r - \frac{n - r - n_c - k + m}{3} \\
&= \frac{8n}{3} - \frac{5r}{3} + \frac{n_c + k - m}{3}
\end{aligned}$$

This performance guarantee gives an approximation ratio which increases whenever r decreases, meeting the performance guarantee of Eq [6] at the threshold r_t value for r, for which

$$\begin{aligned}
\frac{8n}{3} - \frac{5r_t}{3} + \frac{n_c + k - m}{3} &= 2n - \frac{2r_t}{3} - \frac{2n_c}{3} \\
\frac{2n}{3} + n_c + \frac{(k - m)}{3} &= r_t
\end{aligned}$$

We apply both algorithms $\mathcal{A}$, and $\mathcal{C}$, choosing the better solution. This guarantees an approximation ratio of $2n - \frac{2r_t}{3} - \frac{2n_c}{3}$. We distinguish two cases:

case 1: If $n_c + m - k \geq 0$, then:

$$\begin{aligned}
\frac{w_{approx}}{w_{opt}} &\leq \frac{2n - \frac{2r_t}{3} - \frac{2n_c}{3}}{2n - r_t} \\
&= \frac{2n - \frac{4n}{9} - \frac{2n_c}{3} + \frac{2(m-k)}{9} - \frac{2n_c}{3}}{2n - \frac{2n}{3} - n_c + \frac{(m-k)}{3}} \\
&= \frac{28n - 24n_c + 4(m-k)}{24n - 18n_c + 6(m-k)} \\
&\leq \frac{28n - 24n_c + 4(m-k) + 3n_c + 3(m-k)}{24n - 18n_c + 6(m-k)} = \frac{7}{6}
\end{aligned}$$

case 2: If $k > n_c + m \geq 2m$, then the bound of Claim 4.5 applies, in particular:

$$w_{opt} \geq 2n - r_t + \frac{k}{2} - \frac{3m'}{4} \geq 2n - r_t + \frac{k - m - n_c}{7}$$

$$\begin{aligned}
\frac{w_{approx}}{w_{opt}} &\leq \frac{2n - \frac{2r_t}{3} - \frac{2n_c}{3}}{2n - r_t + \frac{k-m-n_c}{7}} \\
&= \frac{2n - \frac{4n}{9} - \frac{2n_c}{3} + \frac{2(m-k)}{9} - \frac{2n_c}{3}}{2n - \frac{2n}{3} - n_c + \frac{(m-k)}{3} - \frac{m-k}{7} - \frac{n_c}{7}} \\
&= \frac{98n - 84n_c + 14(m-k)}{84n - 72n_c + 12(m-k)} = \frac{7}{6}
\end{aligned}$$

5 Extension to Four Species

We now extend the approximation algorithm to handle trees with four leaves. We shall in fact give a general technique to transform an approximation algorithm of the median problem for triplets into an approximation algorithm for the Steiner problem on four leaves.

Theorem 5.1 *For any metric $\lceil$ and $\epsilon < 1$, a polynomial $(1-\epsilon)$-approximation algorithm of the median problem for triplets yields a polynomial $(1.25 + 0.75\epsilon)$-approximation algorithm for the Streiner problem on four leaves.*

Proof: Consider a norm $\lceil$ over a space $\mathcal{S}$. For triplet of points $\tau = (a, b, c), a, b, c \in \mathcal{S}$, their distance to a point $x \in \mathcal{S}$ is $d(x, \tau) = \lceil(a, x) + \lceil(b, x) + \lceil(c, x)$. Define a *median*, $med(\tau)$, of a triplet $\tau \in \mathcal{S}^3$ to be $argmin_x d(x, \tau)$. The *minimum Steiner tree* on a set $S \in \mathcal{S}$ of points is a tree $T = (V, E)$ whose leaf set is S, minimizing $\sum_{uv \in E} \lceil(u, v)$. Finding the median is a restriction of the Steiner tree problem to only three input points. In the sequel we write uv instead of $\lceil(u, v)$, for short.

Suppose there exists a polynomial algorithm which given $\tau \in \mathcal{S}^3$ finds a point $\widehat{med}(\tau)$ such that $d(\widehat{med}, \tau) \leq (1+\epsilon)d(med(\tau), \tau)$. We shall devise an algorithm, which given four points $Q = \{a, b, a', b'\}$ finds a Steiner tree of weight at most $(1.25 + 0.75\epsilon)$ times the optimal tree.

Define τ_x to be the triplet consisting of the four elements a, b, a', b' except x. We apply the median approximation algorithm on each τ_x, obtaining $\hat{m_x} = \widehat{med}(\tau_x)$. For each $x, y \in Q$, we now define the weight of the 4-species tree T_{xy}, formed by taking m_x as median of τ_x, and connecting x to the leaf $y \neq x$:

$$w(T_{xy}) = xy + d(\hat{m_x}, \tau_x) \tag{9}$$

Algorithm D will compute T_{xy} for all x, y and pick T_{xy} for which $w(T_{xy})$ is minimal. Let z and w be the elements of $Q \setminus \{x, y\}$, and let T^*_{xy} be the optimal tree with topology separating x, y from z, w, and let m_{xy} and m_{zw} be its internal vertices that are adjacent to these pairs, respectively. We observe that:

$$\begin{aligned}
w(T_{xy}) &= xy + d(\hat{m_x}, \tau_x) \\
&\leq xm_{xy} + ym_{xy} + (1+\epsilon)d(med(\tau_x), \tau_x) \\
&\leq xm_{xy} + ym_{xy} + (1+\epsilon)d(m_{zw}, \tau_x) \\
&= xm_{xy} + ym_{xy} + (1+\epsilon)zm_{zw} + (1+\epsilon)wm_{zw} + (1+\epsilon)ym_{zw} \\
&\leq xm_{xy} + ym_{xy} + (1+\epsilon)zm_{zw} + (1+\epsilon)wm_{zw} + (1+\epsilon)ym_{yx} + (1+\epsilon)m_{zw}m_{yx} \\
&= xm_{xy} + (2+\epsilon)ym_{xy} + (1+\epsilon)zm_{zw} + (1+\epsilon)wm_{zw} + (1+\epsilon)m_{zw}m_{xy}
\end{aligned}$$

Hence:

$$\min\{w(T_{xy}), w(T_{yx})\} \leq \frac{w(T_{xy}) + w(T_{yx})}{2} \leq (1.5 + 0.5\epsilon)(ym_{xy} + xm_{xy}) + (1+\epsilon)(zm_{zw} + wm_{zw}) + (1+\epsilon)m_{zw} \tag{10}$$

obtaining, for $W_{xy} = \{w(T_{xy}), w(T_{yx}), w(T_{zw}), w(T_{wz})\}$, that:

$$\begin{aligned}
\min W_{xy} &\leq (1.25 + 0.75\epsilon)(ym_{xy} + xm_{xy} + zm_{zw} + wm_{zw}) + (1.25 + 0.75\epsilon)m_{zw}m_{xy} \\
&\leq (1.25 + 0.75\epsilon)(ym_{xy} + xm_{xy} + zm_{zw} + wm_{zw} + m_{zw}m_{xy}) \\
&= (1.25 + 0.75\epsilon)w(T^*_{xy})
\end{aligned}$$

Finally,

$$w_{\mathcal{D}} = \min_{xy} T_{xy} \leq (1.25 + 0.75\epsilon)min_{xy}w(T^*_{xy}) = (1.25 + 0.75\epsilon)w_{opt} \tag{11}$$

proving the approximation ratio.

■

Corollary 5.2 *For the breakpoint distance measure, Algorithm D is a $\frac{11}{8}$-approximation for the Steiner problem on four genomes.*

Acknowldgments

I. P. was supported by the Clore foundation scholarship. R. S. was supported in part by the Israel Science Foundation formed by the Israel Academy of Sciences and Humanities.

References

Bafna, V. and Pevzner, P. A. (1998). Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240.

Berman, P. and Hannenhalli, S. (1996). Fast sorting by reversal. In Hirschberg, D. S. and Myers, E. W., editors, *Combinatorial Pattern Matching, 7th Annual Symposium*, volume 1075 of *Lecture Notes in Computer Science*, pages 168–185, Laguna Beach, California. Springer.

Bryant, D. (1998). The complexity of the breakpoint median problem. Technical Report CRM-2579, Centre de recherches mathématiques, Université de Montréal.

Caprara, A. (1999a). Formulations and complexity of multiple sorting by reversals. In Istrail, S., Pevzner, P. A., and Waterman, M., editors, *Proceedings of the 3rd Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 84–93, Lyon, France. ACM Press.

Caprara, A. (1999b). Sorting permutations by reversals and Eulerian cycle decompositions. *SIAM Journal on Discrete Mathematics*, 12(1):91–110.

DasGupta, B., Jiang, T., Kannan, S., Li, M., and Sweedyk, E. (1998). On the complexity and approximation of syntenic distance. *Discrete Applied Mathematics*, 88:59–82.

Hannenhalli, S. (1996). Polynomial-time algorithm for computing translocation distance between genomes. *Discrete Applied Mathematics*, 71:137–151.

Hannenhalli, S. and Pevzner, P. A. (1995). Transforming men into mice (polynomial algorithm for genomic distance problem). In *36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, pages 581–592, Los Alamitos. IEEE Computer Society Press.

Hannenhalli, S. and Pevzner, P. A. (1999). Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46(1):1–27.

Kaplan, H., Shamir, R., and Tarjan, R. E. (2000). A faster and simpler algorithm for sorting signed permutations by reversals. *SIAM Journal on Computing*, 29(3):880–892.

Kececioglu, J. D. and Ravi, R. (1995). Of mice and men: Algorithms for evolutionary distances between genomes with translocation. In *Proceedings of the 6th Annual Symposium on Discrete Algorithms*, pages 604–613, New York, NY, USA. ACM Press.

Pe'er, I. and Shamir, R. (1998). The median problem for breakpoints is NP-complete. Technical Report TR98-071, Electronic Colloqium of Computer Science (ECCC).

Sankoff, D. and Blanchette, M. (1997). The median problem for breakpoints in comparative genomics. *Computing and Combinatorics, 3rd Ann. Int. Conf. COCOON 97*, 1276:251–263.

Sankoff, D. and Blanchette, M. (1998). Multiple genome rearrangements and breakpoint phylogeny. *Journal of Computational Biology*, 5:555–570.

Sankoff, D. and Nadeau, J. H. (1996). Conserved synteny as a measure of genomic distance. *Discrete Applied mathematics*, 71:247–257.

Sankoff, S., Sundaram, G., and Kececioglu, J. D. (1996). Steiner points in the space of genome rearrangements. *IJFCS: International Journal of Foundations of Computer Science*, 7.