Electronic Colloquium on Computational Complexity, Report No. 71 (1998)

The median problems for breakpoints are NP-complete

Itsik Pe'er* Ron Shamir†

November 1998

Abstract

The breakpoint distance between two n-permutations is the number of pairs that appear consecutively in one but not in the other. In the median problem for breakpoints one is given a set of permutations and has to construct a permutation that minimizes the sum of breakpoint distances to all the original ones. Recently, the problem was suggested as a model for determining the evolutionary history of several species based on their gene orders. We show that the problem is already NP-hard for three permutations, and that this result holds both for signed and for unsigned permutations.

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 they make possible for the first time a large scale study of evolutionary relations among species by comparing the order of appearance of common genes in 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 gene rearrangement events that led from the ancestral genome species to the current species.

When restricting the discussion to only one chromosome, and assuming genes occur only once, genomes are modeled as permutations. When taking into account the orientation of the genes, the objects discussed are *signed* permutations. The case where all rearrangement events are reversals (inversion of a chromosome segment) has been studied intensively in recent years. For two species, this problem, of finding the *reversal distance* between two permutations, is already NP-hard in case the gene orders are given as unsigned permutations [4],

*Supported by Eshkol scholarship from the Ministry of Science and Technology, Israel

†Research supported in part by a grant from the Ministry of Science and Technology, Israel

ISSN 1433-8092

but it is polynomial if the permutations are signed [9, 2, 11]. Finding a tree minimizing the total number of reversals for three signed permutations is already NP-hard [3]. Other models of the genome [8, 12] or of the assumed events were also studied [10, 8, 14, 1, 6]. When studying more than two species the key problem arising is to reconstruct the phylogenetic tree achieving minimal total distance along edges (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* [5, 6].

Recently, Sankoff and Blanchette [13] 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 weaknesses that render them inappropriate for studying complex trees, and suggested a simpler criterion of *breakpoint distance*, i.e., merely counting the number of breakpoints between every two permutations connected by an edge of the tree. This number can be easily computed for a single edge, for signed and unsigned permutations. Sankoff and Blanchette studied *the median problem for breakpoints*, and developed several efficient heuristics for the problem. However, the computational complexity of this problem was not determined. Here we settle that question by showing that the problem is NP-hard already for three permutations. The hardness result applies both to the signed and to the unsigned model. In a companion manuscript we give a constant factor polynomial approximation algorithm for the signed model.

The paper is organized as follows: Section 2 recalls the standard notation in the field, which we follow. Section 3 defines further terminology used throughout the paper. Section 4 defines a problem which is equivalent to the median problem for breakpoints, for 3 signed permutations, and proves these problems are NP-complete. Section 5 extends the hardness result also to unsigned permutations.

2 Preliminaries

Let d be a measure of genomic distance, namely an integer distance function on the space of (signed) permutations. The (signed) median problem is defined as follows:

Instance: Three (signed) permutations π_0, π_1, π_2, and an integer k
Question: Does there exist a (signed) permutation $\widehat{\pi}$ s.t. $\sum_i d(\pi_i, \widehat{\pi}) \leq k$?

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 by defining $\pi(0) = 0$, and $\pi(n+1) = n+1$. We call this transformation of a

signed n element permutation into an unsigned $2n+2$ elements the *standard augmentation* (see also [9]). Note that the range of the standard augmentation is a set of permutations which is closed under composition.

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'.

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 of π w.r.t. σ is denoted by $bp(\pi,\sigma)$. Trivially, $bp(\pi,\sigma) = bp(\sigma,\pi)$.

For example, let $\pi' = (1,-3,-2)$ be a signed permutation on three elements. It is rescaled to the 6-permutation $(1,2,6,5,4,3)$, and it's standard augmentation is the 8-permutation $\pi = (0,1,2,6,5,4,3,7)$. The only breakpoints in π w.r.t. the identity permutation σ are $(2,6)$ and $(3,7)$, thus $bp(\pi,\sigma) = 2$.

All graphs in this paper are finite, and unless specifically indicated otherwise, undirected. They do not contain self loops, but may contain parallel edges: single, double or triple edges. 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 $2n$ edges, forming one cycle passing through all the vertices.

3 Definitions

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 $2n$).

Let M_b be the base matching on the set V of $2n$ vertices. A perfect matching M on V is called a *Hamiltonian matching* w.r.t. M_b (or simply a Hamiltonian Matching) if $M_b \cup M$ forms a Hamiltonian cycle. For a Hamiltonian matching M, let $unsigned(M)$ be the permutation on $\{0,1,\ldots,2n-1\}$, denoting the order of appearance of the vertices in V along the cycle $M_b \cup M$, starting from v_0, and ending on v_{2n-1}. Clearly, $2i-1$ and $2i$ are consecutive in $unsigned(M)$, for each $0 < i \leq n$, hence, there exists a unique signed permutation denoted $signed(M)$ on $n-1$ elements, whose standard augmentation into an unsigned permutation on $2n$ elements is $unsigned(M)$. Furthermore, for every signed permutation π on $n-1$ elements, its standard augmentation into an unsigned permutation on $2n$ elements is π', and the edge set $hmatch(\pi) = \{v_0 v_{\pi'(1)}, v_{\pi'(2)}v_{\pi'(3)}, \ldots, v_{\pi'(2n-4)}v_{\pi'(2n-3)}, v_{\pi'(2n-2)}v_{2n-1}\}$ is a Hamiltonian matching w.r.t. 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, on the other hand, each such common edge corresponds to such an adjacency.

Corollary 3.1 *Let π and σ be signed permutations on $n-1$ elements. Then $bp(\pi,\sigma) = n - |hmatch(\pi) \cap hmatch(\sigma)|$*

4 The Consensus of 3 Hamiltonian Matchings problem

4.1 Problem Definition

We define the Consensus of 3 Hamiltonian Matchings(*C3HM* for short) problem 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, and an integer w_{target}. M_b, M_0, M_1, M_2 define a weight function w for any Hamiltonian matching M, setting $w(M) \equiv \sum_i |M \cap M_i|$.
Question: Does there exist a Hamiltonian matching $\widehat{M}$ w.r.t. M_b, s.t.

$$w(\widehat{M}) \geq w_{target}$$

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

4.2 Equivalence to the Median Problem

Proposition 4.1 *The instance* $(V, M_b, M_0, M_1, M_2, w_{target})$ *of the C3HM problem and the instance* $(signed(M_0), signed(M_1), signed(M_2), 3|M_b| - w_{target})$ *of the signed permutations median problem are equivalent.*

Proof: Follows directly from corollary 3.1. ■

Note that the transformation in the other direction, i.e., from signed permutations median to C3HM, is also immediate.

4.3 NP-completeness

Theorem 4.2 *C3HM is NP-complete.*

Proof: Membership in NP is trivial.

We prove NP-hardness by reduction from the Hamiltonian cycle problem, restricted to 3-regular graphs. This problem is well known to be NP-complete [7]. Let $G(N, A)$ be a simple 3-regular input graph, with $n = |N|$. We construct, in polynomial time, an instance $I = (V, M_b, M_0, M_1, M_2, w_{target})$ of C3HM, of linear size in n, s.t. I is a "yes" instance iff G admits a Hamiltonian path. To avoid confusion, we refer to G as a graph on the set N of *nodes* with the set A of *arcs*, reserving the terms *vertex* and *edge* for the graph constructed as a C3HM instance.

We first give an overview of the reduction. We build a node component for each node of G. There are edges between these components, some of them correspond to the arcs of G. Our construction maps Hamiltonian cycles in G to Hamiltonian cycles $\widehat{M} \cup M_b$ in the

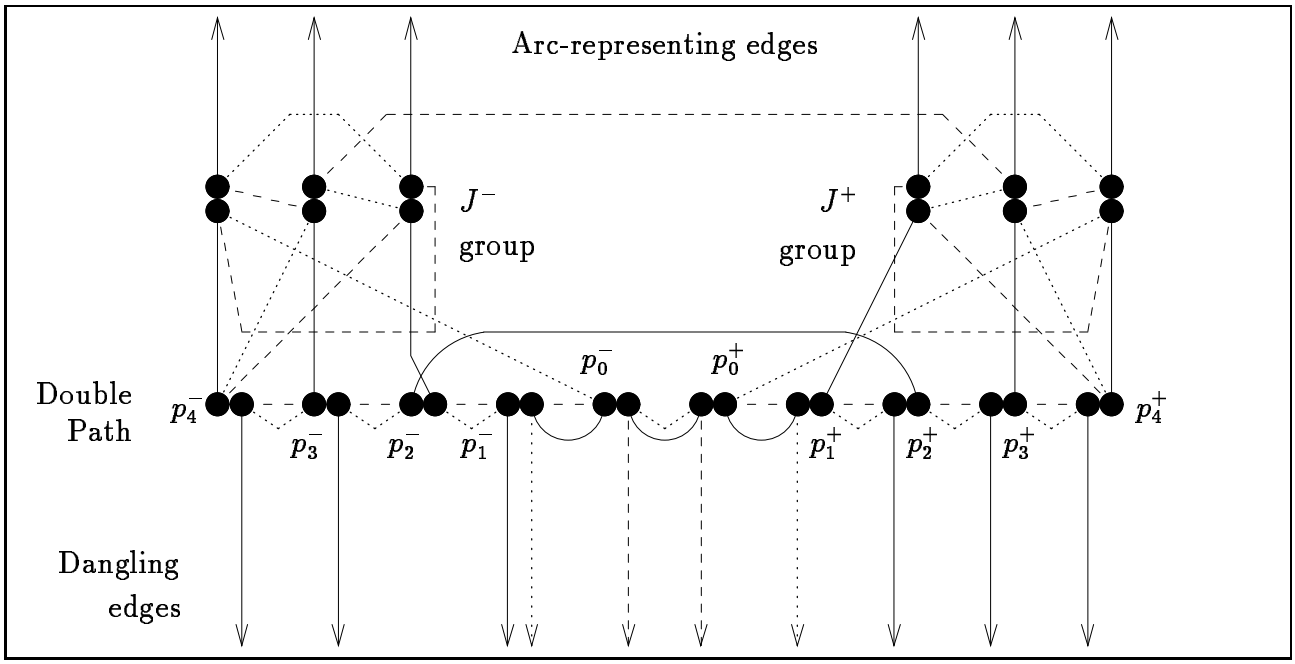

Figure 1: The whole node component. Each pair of tangent discs denotes a pair of vertices $u\tilde{u}$ (matched by M_b). In such a pair $\tilde{u}$ is the bottom disc (in the J groups) or the disc closer to the center (along the double path) Solid, dotted and dashed lines denote edges of M_0, M_1, and M_2 respectively. Arrows downwards denote dangling edges, while arrows upwards denote arc-representing edges.

constructed graph, with $\widehat{M}$ being a large-weight Hamiltonian matching. We can force specific properties of such Hamiltonian cycles in our construction, and urge them to pass through certain edges, by using double edges, with which $\widehat{M}$ must concur, in order to maximize its weight. In this way, we construct, for each node, a component all of whose vertices must be visited *consecutively* by $\widehat{M} \cup M_b$, and the order in which the different node components are visited by $\widehat{M} \cup M_b$ corresponds to the order of the nodes along a Hamiltonian cycle in G. We make sure that a Hamiltonian cycle $\widehat{M} \cup M_b$ in our construction can visit two node components consecutively if and only if the two corresponding nodes in G are adjacent.

We now start describing our construction in detail. For each node $\nu \in N$, we build a node-component $C(\nu)$ of 32 vertices. See figure 1 for the outline of a full component. We will now describe the construction of the node-component. We explicitly specify the vertices of such a component, and some of the edges. The other edges in the constructed graph are either:

- *Arc-representing*, and will be explicitly specified in a later stage.
- *Dangling* edges, which will be constructed implicitly. When describing the node component, we only specify:
  - On which vertices are the dangling edges incident.
  - To which of the three matchings these edges belong.

The component $C(\nu)$ is composed of two (not fully symmetrical) halves, whose vertices and vertex-groups will be superscripted with $+$ and $-$, respectively.

The 32 vertices in $C(\nu)$ are divided into four groups:

- A half-path $P^+(\nu)$ of 10 vertices $\{p_i^+(\nu)\}_{i=0}^4$ and $\{\widetilde{p_i^+}(\nu)\}_{i=0}^4$.
- A half-path $P^-(\nu)$ of 10 vertices $\{p_i^-(\nu)\}_{i=0}^4$ and $\{\widetilde{p_i^-}(\nu)\}_{i=0}^4$.
- A junction $J^+(\nu)$ of 6 vertices $\{j_i^+(\nu)\}_{i=0}^2$ and $\{\widetilde{j_i^+}(\nu)\}_{i=0}^2$.
- A junction $J^-(\nu)$ of 6 vertices $\{j_i^-(\nu)\}_{i=0}^2$ and $\{\widetilde{j_i^-}(\nu)\}_{i=0}^2$.

Note that vertices in our construction come in pairs $(u(\nu), \widetilde{u}(\nu))$, where $u(\nu)$ and $\widetilde{u}(\nu)$ are always in the same component half and in the same group. Our base matching M_b is simply the set of all edges $u(\nu)\widetilde{u}(\nu)$. Whenever the node ν or the $+/-$ superscript are clear from context, we omit them.

The edges in a component C include:

- Double (*path*) edges:
  - Between the component half paths : $\widetilde{p_0^+}\widetilde{p_0^-} \in M_0 \cap M_1$.

– In each half path: $p_0\widetilde{p_1} \in M_0 \cap M_2$, and for each $i = 1, 2, 3$: $p_i\widetilde{p_{i+1}} \in M_1 \cap M_2$.

We call the path in $M_b \cup \cup_i M_i$ between $p_4^+(\nu)$ and $p_4^-(\nu)$, using the path edges, the *double path* of ν.

- *Bypassing edges*: The edges $\widetilde{j_1}p_3 \in M_0$ and $\widetilde{j_0}p_0 \in M_1$ for each half, and the edges $\widetilde{j_2^+}p_1^+, \widetilde{j_2^-}p_2^-, p_2^+p_2^- \in M_0$.

- *Junction* edges (see Figure 2):
  - The edges $p_4\widetilde{j_i} \in M_i$ for each $0 \leq i \leq 2$ and for each half.
  - The edges $j_2\widetilde{j_0}, j_0\widetilde{j_1} \in M_2$ and $j_1\widetilde{j_2} \in M_1$ for each half.
  - The edges $j_0j_2 \in M_1$ for each half, and the edge $j_1^+j_1^- \in M_2$.

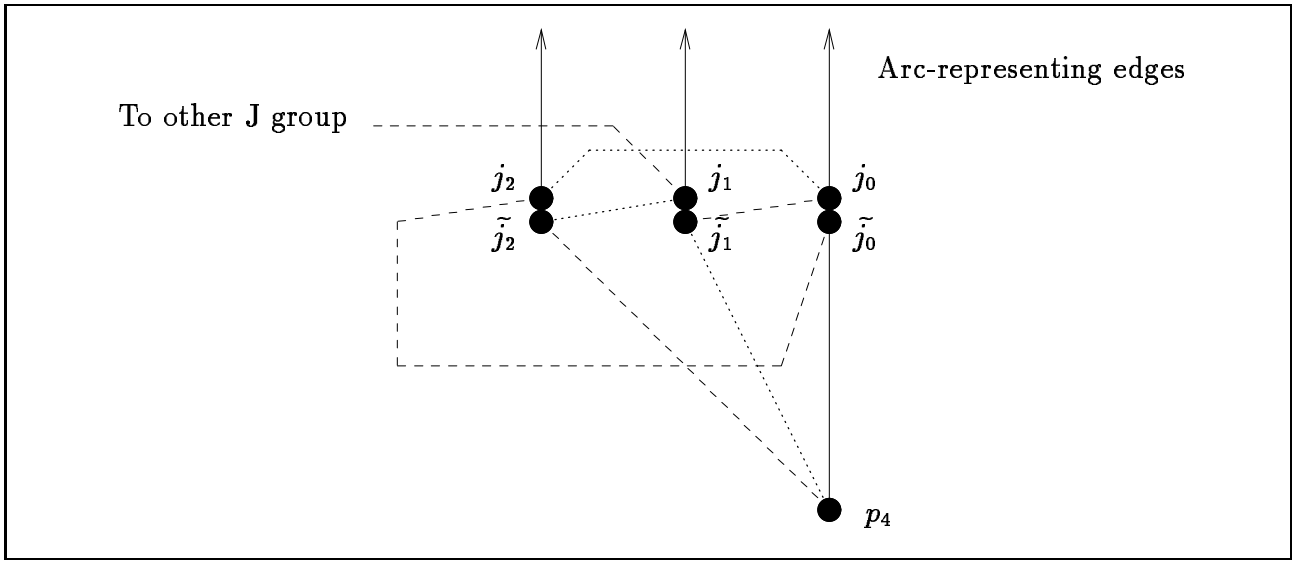

Figure 2: Zoom in on one J group. Only junction edges are drawn.

- *Dangling edges*: As explained, at this stage we only specify the incidence of a dangling edge, without explicitly specifying the edges.
  - Dangling edges of M_0 are incident on $\widetilde{p_4}$, $\widetilde{p_3}$ for each half, and to $\widetilde{p_2^+}$ and $\widetilde{p_1^-}$.
  - Dangling edges of M_1 are incident on $\widetilde{p_1}$ for each half.
  - Dangling edges of M_2 are incident on $\widetilde{p_0}$ for each half.

We now describe the arc-representing edges. For each node $\nu \in N$, arbitrarily number its three arcs $a_0(\nu), a_1(\nu), a_2(\nu)$. For an arc $\nu\nu' = a_i(\nu) = a_{i'}(\nu') \in A$ add the edge $j_i^+(\nu)j_{i'}^-(\nu')$ to M_0. Note that the edge $j_{i'}^+(\nu')j_i^-(\nu)$ will also be added to M_0.

We will now describe some simple properties of our construction so far:

Property 4.3 *Each bypassing or dangling edge is incident on at least one vertex along the double path, which is not an endpoint of this path.*

Property 4.4 *For each matching M_i, let M_i' be the subset of M_i excluding all dangling edges Observe that each of $M_b \cup M_1'$ and $M_b \cup M_2'$ is a collection of $|N|$ (disjoint) paths, and each such path passes through all the vertices of some node component $C(\nu)$ (see Figure 3 and Figure 4).*

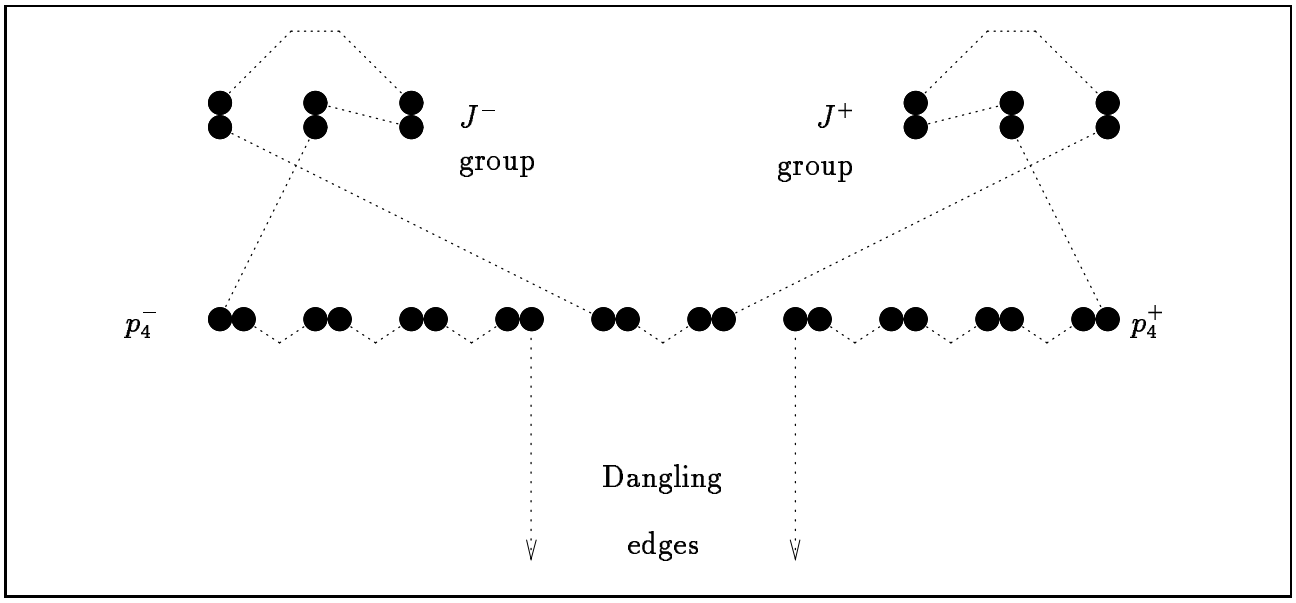

Figure 3: $M_b \cup M_1$ forms a path in each component.

Denote the paths described in property 4.4 $PATH_1(\nu)$ and $PATH_2(\nu)$, respectively. Both endpoints of such a path are vertices where dangling edges are incident on and both are along the double path of ν, one in $P^+(\nu)$ (called *head*) and the other in $P^-(\nu)$ (called *tail*)

Property 4.5 *$M_b \cup M_0'$ is a collection of $2|A|$ (disjoint) paths, each passing through one arc-representing edge $j_i^-(\nu)j_{i'}^+(\nu')$ (see Figure 5).*

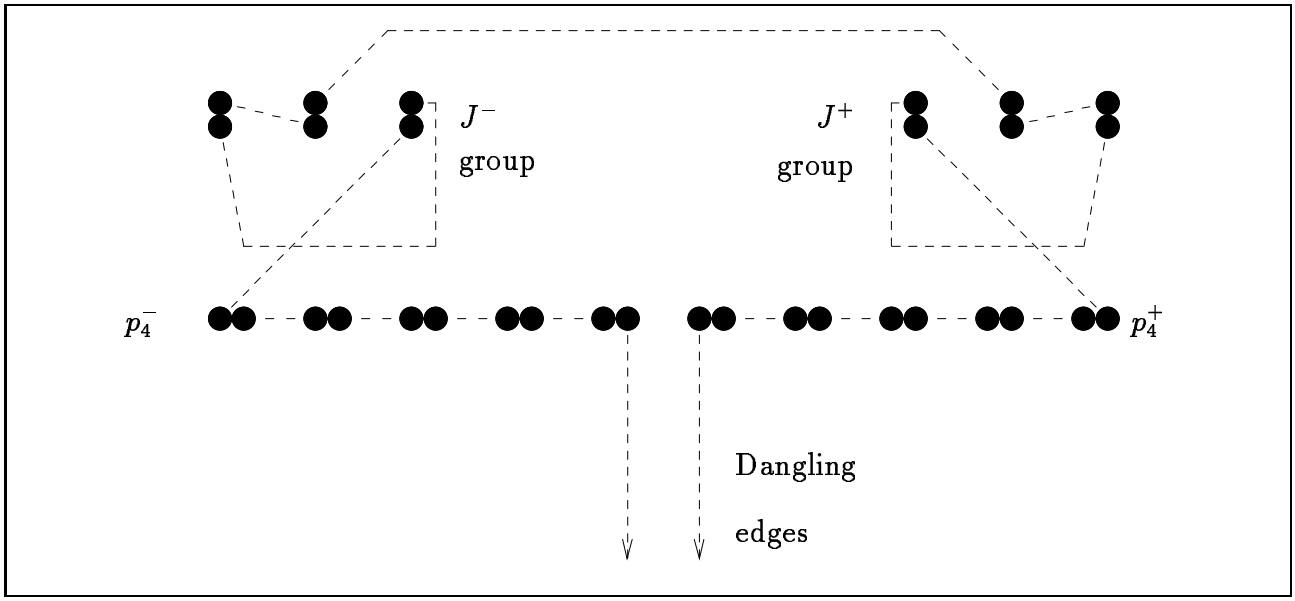

Figure 4: $M_b \cup M_2$ forms a path in each component.

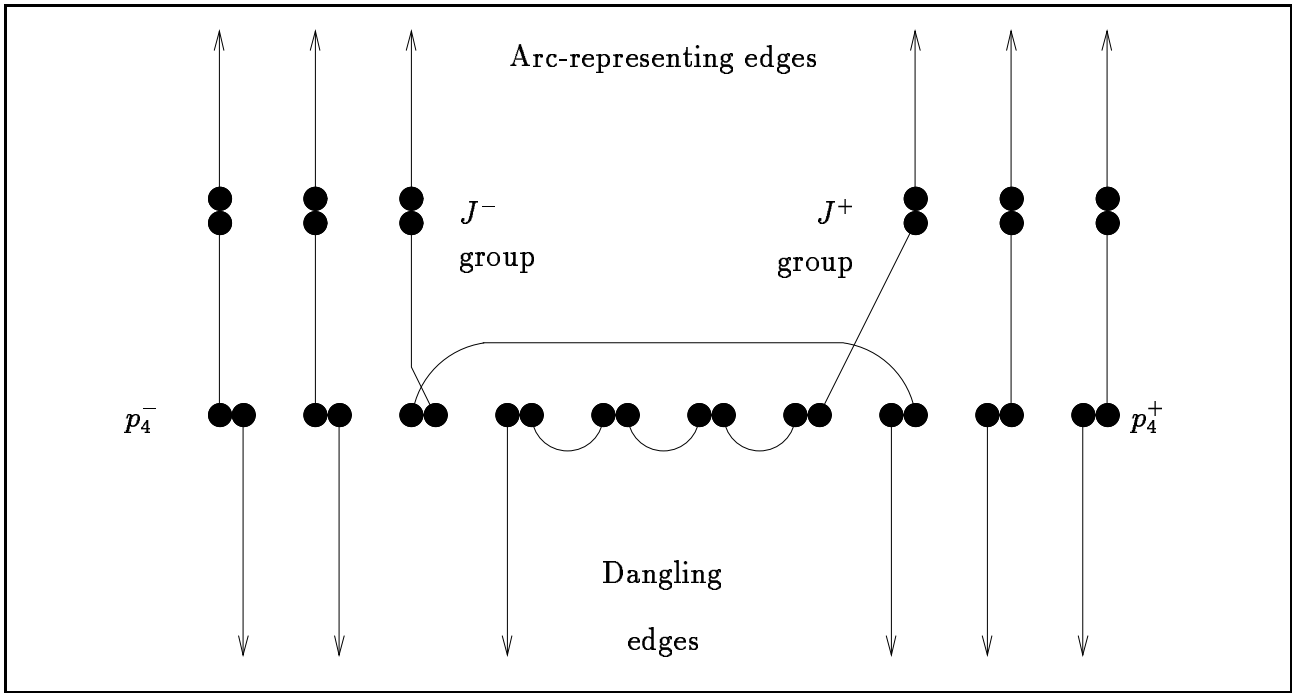

Figure 5: $M_b \cup M_0$ forms a path in through each arc-representing edge.

Denote the paths described in property 4.5 $PATH_0(\nu, \nu')$. This path has its endpoints on two vertices along the double paths of ν and ν', respectively, with a dangling edge incident on each of them. The lengths of such paths may vary between 4 and 8 vertex pairs.

We now describe how to connect the vertices on which dangling edges are incident, thus constructing the dangling edges themselves. Number the nodes $\nu_0, \nu_1, \ldots, \nu_{n-1}, \nu_n = \nu_0$ in an arbitrary order. For each $i = 1, 2$ and $k = 0, \ldots, n-1$, add to M_i a dangling edge connecting the tail of each $PATH_i(\nu_k)$ to the head of $PATH_i(\nu_{k+1})$.

Let $D(N, \vec{A})$ be a directed graph formed by replacing every arc $\nu\nu'$ of A by a pair of anti-parallel directed arcs $\nu\nu'$ and $\nu'\nu$. Clearly, D is Eulerian. Let T be an arbitrary directed Euler tour of D. For every two consecutive directed arcs $\nu\nu'$ and $\nu'\nu''$ in T, the paths $PATH_0(\nu\nu')$ and $PATH_0(\nu'\nu'')$ each have one endpoint in $C(\nu')$. Connect these two endpoints by a M_0 edge. All the dangling edges are constructed by this process, since every node ν' is visited exactly 3 times by T, with each of the six dangling edges endpoints in $C(\nu')$ visited once.

Note that no dangling edge thus constructed is parallel to any other edge in any M_i. Therefore, each dangling edge is single, and no double or triple edges are generated when determining the dangling edges.

We set w_{target} to be $25n$.

Clearly, the reduction is polynomial. We prove now the validity of our construction.

Claim 4.6 *I is an instance of C3HM.*

Proof: M_b is a perfect matching. For each $i = 0, 1, 2$, every vertex in V is incident to exactly one edge of each M_i, so also M_i is a perfect matching. The construction of the dangling edges connects all the paths $\{PATH_1(\nu) | \nu \in N\}$ into one Hamiltonian cycle, whose edges are exactly $M_b \cup M_1$. Similarly, $M_b \cup M_2$ is a Hamiltonian cycle. Also, the construction of the dangling edges connects all the paths $\{PATH_0(\nu\nu') | \nu\nu' \in A\}$ into one Hamiltonian cycle, whose edges are exactly $M_b \cup M_0$. Hence, M_i is a Hamiltonian matching for $i = 0, 1, 2$. ■

Claim 4.7 *If there is a Hamiltonian cycle in G, then I is a "yes" instance.*

Proof: Assume there exists a Hamiltonian cycle h in G. We construct a matching $\widehat{M}$ as follows:

- Include all 9 double path edges of each node component (4 of each half, and one connecting the two halves).
- Choose an arbitrary orientation for h. Let $\nu\nu' = a_i(\nu) = a_{i'}(\nu')$ be an arc of h in this orientation. For each such arc add to $\widehat{M}$ the following seven edges:
  - Edges in J^+ and P^+: $p_4^+(\nu)\widetilde{j_{i+1}^+}(\nu)$, $j_{i+1}^+(\nu)\widetilde{j_{i+2}^+}(\nu)$, $j_{i+2}^+(\nu)\widetilde{j_i^+}(\nu)$.
  - Edges in J^- and P^-: $p_4^-(\nu')\widetilde{j_{i'+1}^-}(\nu')$, $j_{i'+1}^-(\nu')\widetilde{j_{i'+2}^-}(\nu')$, $j_{i'+2}^-(\nu')\widetilde{j_{i'}^-}(\nu')$.

– The edge $j_i^+(\nu)j_{i'}^-(\nu')$.

Subscripts for junction vertices are calculated modulo 3. Compare Figure 6 and Figure 2.

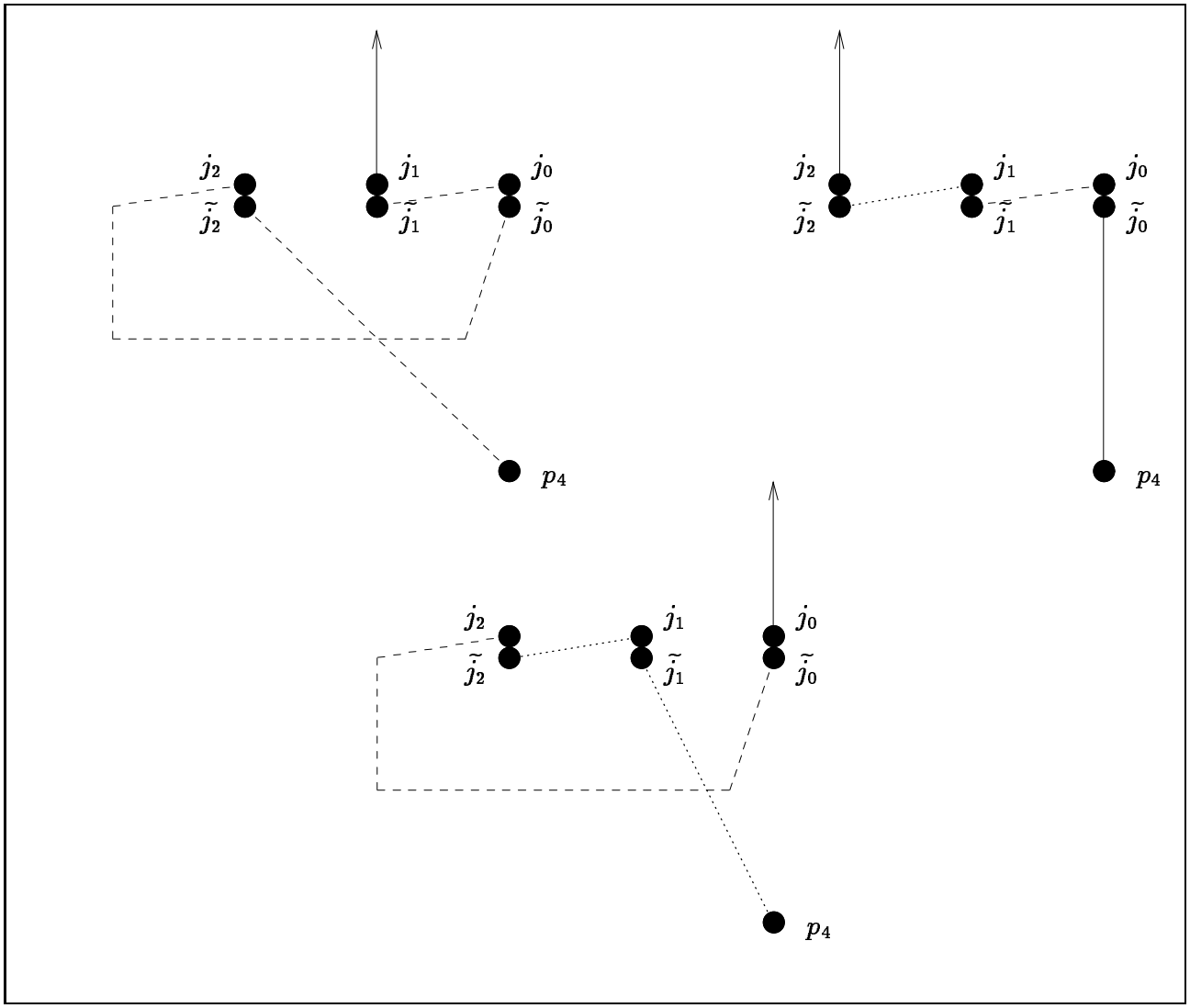

Figure 6: The three possible ways we use to connect the vertices of a junction group by a Hamiltonian matching.

Clearly, $\widehat{M}$ is a matching. Since h is Hamiltonian, $\widehat{M} \cup M_b$ is a Hamiltonian cycle. For the double edges in $\widehat{M}$, $w(e) = 2$. For the other edges in $\widehat{M}$, $w(e) = 1$. It follows that $w(\widehat{M}) = 2 \cdot 9n + 7n = 25n$ ■

Claim 4.8 *If I is a "yes" instance, then there is a Hamiltonian cycle in G.*

Proof: Let $\widehat{M}$ be a Hamiltonian matching in I, with $w(\widehat{M}) \geq 25n$. We call a Hamiltonian matching M in I *proper* if all the vertices of each node component appear consecutively along the cycle $M \cup M_b$. We now prove that $\widehat{M}$ is proper.

Since there are exactly $16n$ edges in $\widehat{M}$, and there are only $9n$ double edges in I, $\widehat{M}$ must contain all the double edges, and furthermore, for every other edge $e \in \widehat{M}$, $w(e) = 1$, i.e., $e \in M_i$ for some i. It follows that $\widehat{M} \cup M_b$ contains the double path of each ν. Furthermore, according to property 4.3, every bypassing or dangling edge is incident on an internal vertex of such a path, $\widehat{M}$ contains no bypassing edges, nor dangling edges (See Figure 7).

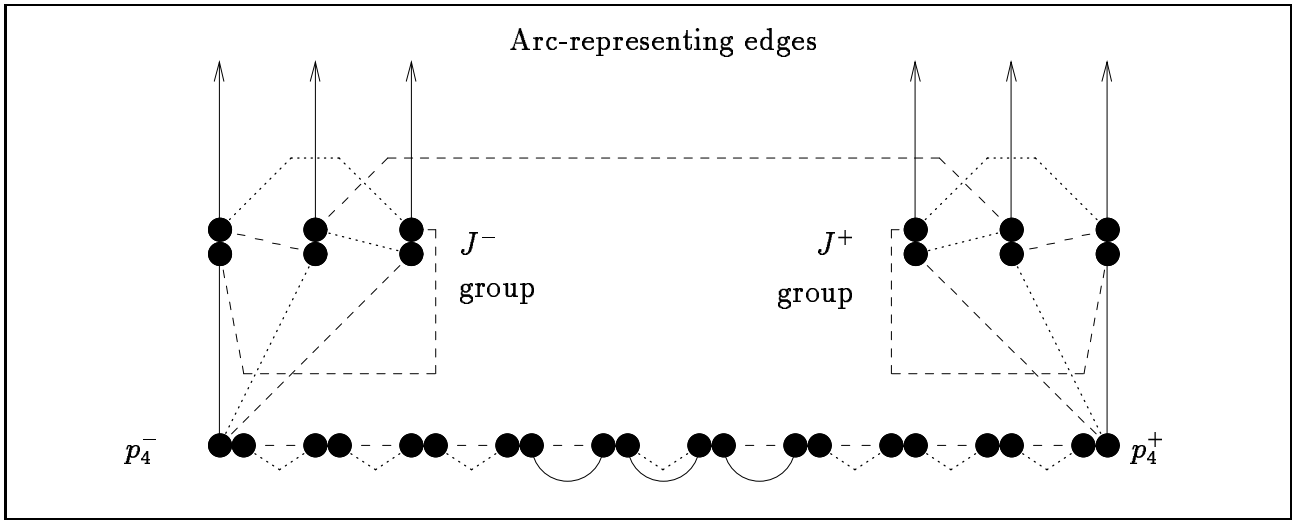

Figure 7: The component without the bypassing edges and the dangling edges. A proper Hamiltonian matching must pass through the six J^+ vertices consecutively, and immediately "after" (direction is arbitrary) the P vertices.

Fix some ν. $\widehat{M}$ must contain one edge incident on $p_4^+(\nu)$. Since this edge is a member of some Hamiltonian matching M_i, it is $p_4^+(\nu)\widetilde{j_i^+}(\nu)$, for some $0 \leq i \leq 2$ (see Figure 6). Since $\widehat{M}$ must contain no other edge incident on $p_4^+(\nu)$, the edges of the two other Hamiltonian matchings incident on $p_4^+(\nu)$ are not in $\widehat{M}$. Formally, for $i' \neq i$, $p_4^+(\nu)\widetilde{j_{i'}^+}(\nu) \notin \widehat{M}$. Now we can examine the vertex $\widetilde{j_{i'}^+}(\nu)$. There must be some non-bypassing edge in $\widehat{M}$ incident on it, and the only possible candidate is $j_{i'-1}^+(\nu)\widetilde{j_{i'}^+}(\nu)$ (see Figure 2), therefore $j_{i'-1}^+(\nu)\widetilde{j_{i'}^+}(\nu) \in \widehat{M}$.

Hence, the path through $J^+(\nu)$ $j_{i+2}, \widetilde{j_{i+2}}, j_{i+1}, \widetilde{j_{i+1}}, j_i^+, \widetilde{j_i}p_4$ is contained in $\widehat{M} \cup M_b$. The argument for a path passing through $J^-(\nu)$ is similar, proving that $\widehat{M}$ is proper.

Let $C(\nu), C(\nu')$ be two components visited consecutively by $\widehat{M} \cup M_b$. There must be an edge e connecting the two components, and since $\widehat{M} \cup M_b$ contains no dangling edges, e

must be an arc-representing edge (all other edges are incident on two vertices in the same component). According to our construction, there must be an arc in G between ν and ν'.

Let $h = \nu_0, \nu_1, \ldots, \nu_{n-1}, \nu_0$ be the order in which components are visited by $\widehat{M} \cup M_b$. It follows that h is a Hamiltonian cycle in G.

■

Claims 4.6, 4.7, and 4.8 complete the proof of Theorem 4.2. ■

Corollary 4.9 *The signed permutations median problem is NP-complete.*

Proof: Immediate from Theorem 4.2 and Proposition 4.1. ■

Corollary 4.10 *Finding a tree and corresponding signed permutations in internal nodes so as to minimize overall breakpoint distance along tree edges is NP-hard, both if the tree topology is known and if it is unknown.*

5 Unsigned Permutations

We map each unsigned permutation π on the elements $1, 2, \ldots, n-2$ to the Hamiltonian cycle on n vertices $hcycle(\pi) = v_0, v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n-2)}, v_{n-1}, v_0$. This is a 1-1 mapping, whose range is all the Hamiltonian cycles on n vertices passing through the *compulsory* edge $v_{n-1}v_0$. We call such cycles *image* cycles. For two image cycles, $hcycle(\pi)$ and $hcycle(\sigma)$, a non-compulsory edge in $hcycle(\pi) \cap hcycle(\sigma)$, maps to an adjacency between π and σ, and vice versa.

The unsigned permutations median problem formulates as the Consensus of 3 Hamiltonian Cycles (C3HC for short) problem:
Instance: Three image cycles C_0, C_1, C_2 on n vertices, and an integer w_{target}.
Question: Does there exist an image cycle $\hat{C}$, s.t. $w(\hat{C}) = \sum_i |\hat{C} \cap C_i| \geq w_{target}$.

Theorem 5.1 *C3HC (and therefore also the unsigned permutations median problem) is NP-complete.*

Proof: (sketch): The proof is very similar to the proof of Theorem 4.2. We use a similar construction to reduce the problem of Hamiltonian cycle on 3-regular graphs passing through a specific edge (actually, the Hamiltonian path problem), to C3HC. The only differences are:

- We set $C_i = M_i \cup M_b$, thus replacing every edge $e \in M_b$ with a triple edge $e \in C_0 \cap C_1 \cap C_2$.
- We number the vertices choosing a triple edge to be the compulsory edge.
- We increase w_{target} by the total weight of the triple edges and set it to $25n + 3 \cdot 16n = 73n$.

Cardinality considerations of the weight of a consensus image cycle make sure that this cycle passes through all triple edges, and the rest of the proof is similar. ■

Proof:(Other option): We reduce signed permutations median to unsigned permutations median. Let $I' = (\pi'_0, \pi'_1, \pi'_2, d_{target})$ be a signed permutations median problem instance with each π'_i being a signed permutation on n elements. Let π_i be the $2n+2$ element signed permutation obtained by the standard augmentation of π'_i. We claim that the unsigned permutations median problem instance $I = (\pi_0, \pi_1, \pi_2, d_{target})$ is a "yes" instance iff I' is.

For every two signed permutations σ'_1, σ'_2, whose standard augmentations are, respectively, σ_1, σ_2, $bp(\sigma_1, \sigma_2) = bp(\sigma'_1, \sigma'_2)$. Therefore it remains only to be shown that:

Claim 5.2 *If τ is an unsigned permutation satisfying $\sum_i bp(\tau, \pi_i) \leq d_{target}$, then there is also some unsigned permutation σ satisfying $\sum_i bp(\sigma, \pi_i) \leq d_{target}$, with a signed permutation σ' whose standard augmentation is σ.*

Proof: We prove this by induction on the number $p(\tau)$ of pairs $2i-1, 2i$ of non-successive elements in τ.

Clearly, for $p(\tau) = 0$, $\sigma = \tau$ is a standard augmentation of some unsigned σ'.

For positive k, assume correctness for all permutations ρ with $p(\rho) < k$, and suppose $p(\tau) = k$. It suffices to find a permutation ρ satisfying $\sum_i bp(\rho, \pi_i) \leq \sum_i bp(\tau, \pi_i)$, with $p(\rho) < k$. Let $C_0, C_1, C_2, w_{target}$ be the C3HC instance corresponding to I'. Define $w(e) = |\{i | e \in C_i\}|$. Note that for each vertex v:

$$\sum_u w(uv) = 6 \tag{1}$$

We call every edge $e = v_{2i-1}v_{2i}$ a *parity* edge. Note that every parity edge e has $w(e) = 3$.

Then there is a Hamiltonian cycle $C_\tau = v_0, v_{\tau_1}, \ldots, v_{\tau_{2n}}, v_{2n+1}$ with weight $w(C_\tau) \geq w_{target}$, and with $n+1-k$ parity edges. Fix some i for which the parity edge $v_{2i-1}v_{2i}$ is not contained in C_τ. $2i-1$ and $2i$ are not successive in τ, i.e. $\tau_j = 2i-1, \tau_{j'} = 2i$, and $|j-j'| > 1$. Let $x = v_{\tau_{j-1}}$, $y = v_{2i-1}$, $z = v_{\tau_{j+1}}$ and $x' = v_{\tau_{j'-1}}$, $y' = v_{2i}$, $z' = v_{\tau_{j'+1}}$. $w(yy') = 3$, so according to equation 1 $w(xy) + w(yz) + w(x'y') + w(y'z') \leq 6$, either $w(xy) + w(x'y') \leq 3$ or $w(yz) + w(y'z') \leq 3$. In the first case, define $C_\rho = (C_\tau \setminus \{xy, x'y'\}) \cup \{xx', yy'\}$ and in the latter define $C_\rho = (C_\tau \setminus \{yz, y'z'\}) \cup \{zz', yy'\}$. In either case, C_ρ is an image cycle, and $w(C_\rho) \geq w(C_\tau)$. The corresponding permutation ρ is as required. ■

Now it is clear that I is a "yes" instance iff I' is, completing the proof of theorem 5.1 ■

In a forthcoming companion paper, we give several constant-factor approximation algorithms to the signed problem.

References

[1] V. Bafna and P. Pevzner. Sorting permutations by transpositions. In *Proceedings of the 6th Annual Symposium on Discrete Algorithms*, pages 614–623. ACM Press, January 1995.

[2] P. Berman and S. Hannenhalli. Fast sorting by reversal. In *Proc. Combinatorial Pattern Matching (CPM)*, pages 168–, 1996. LNCS 1075.

[3] A. Caprara. Formulations of complexity of multiple sorting by reversals. Technical Report OR-97-15, University of Bologna, Bologna, Italy, 1997.

[4] A. Caprara. Sorting by reversals is difficult. In *Proceedings of the First International Conference on Computational Molecular Biology*, pages 75–83, New York, January19–22 1997. ACM Press.

[5] G. Sundaram D. Sankoff and J. Kececioglu. Steiner points in the space of genome rearrangements. In *International journal on Foundations of Computer Science, World scientific*, volume 7. 1996.

[6] B. DasGupta, T. Jiang, S. Kannan, M. Li, and Z. Sweedyk. On the complexity and approximation of syntenic distance. In *Proceedings of the First International Conference on Computational Molecular Biology*, pages 99–108, New York, January19–22 1997. ACM Press.

[7] M. R. Garey, D. S. Johnson, and R. E. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM J. Computing*, 5:704–714, 1976.

[8] S. Hannenhalli. Polynomial algorithm for computing translocation distance between genomes. *Discrete Applied Mathematics*, 71:137–151.

[9] S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 178–189, Las Vegas, Nevada, 29 May–1 June 1995.

[10] S. Hannenhalli and P. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problems. In *Proc. IEEE Symp. of the Foundations of Computer Science*, 1995.

[11] H. Kaplan, R. Shamir, and R. E. Tarjan. Faster and simpler algorithm for sorting signed permutations by reversals. to appear in SIAM Journal of Computing (Preliminary version in Proceedings of the eighth annual ACM-SIAM Symposium on Discrete Algorithms 1997 (SODA 97), pages 344–351).

[12] J. Kececioglu and R. Ravi. Physical mapping of chromosomes using unique probes. In *Proc. sixth annual ACM-SIAM Symp. on Discrete Algorithms (SODA 95)*, pages 604–613. ACM Press, 1995.

[13] D. Sankoff and M. Blanchette. The median problem for breakpoints in comparative genomics. In *Proc. COCOON '97, Lecture Note in Computer Science*, volume 1276, pages 251–163, New York, 1997. Springer-Verlag.

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

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'