A Fast Method for Large-scale Multichromosomal Breakpoint Median Problems *

Maryam Haghighi
Department of Mathematics and Statistics
University of Ottawa
Ottawa, Canada
mhagh051@uottawa.ca

Sylvia Boyd
School of Information and Technology
University of Ottawa
Ottawa, Canada
sylvia@site.uottawa.ca

ABSTRACT

We provide a computationally realistic mathematical framework for the NP-hard problem of the multichromosomal breakpoint median for linear genomes that can be used in constructing phylogenies. A novel approach is provided that can handle both signed and unsigned cases of the multichromosomal breakpoint median problem. Our method provides an avenue for incorporating biological assumptions (whenever available) such as the number of chromosomes in the ancestor, and thus, it can be tailored to obtain a more biologically-relevant picture of the median. We demonstrate the usefulness of our method by performing an empirical study on both simulated and real data with a comparison to other methods.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics; G.2.3 [**Discrete Mathematics**]: Applications; G.2.2 [**Discrete Mathematics**]: Graph theory—*graph algorithms*

General Terms

Algorithms

Keywords

Breakpoint median problem, multichromosomal median problem, travelling salesman problem

ACM-BCB '11, August 1-3, Chicago, IL, USA

1. INTRODUCTION

Constructing phylogenies can be very challenging, and therefore, restricted versions of the problem are often studied. One of the most common restrictions is to find one common ancestor of several given genomes. This problem can be modeled by using the notion of a median, where one searches for a genome that is close to several other genomes. Once we have a method to find the median, we can then use it in an iterative manner to construct the evolutionary tree (see [16] for examples of such construction). A fundamental question in building phylogenies is how far apart two species are from each other. This idea can be captured under the notion of *distance* between two genomes. There are several popular measures for distance, such as breakpoint, reversal, double cut-and-join, etc.. For measuring distance, we focus on breakpoint distance as it provides an intuitive link between the order of genes in each genome and how far two genomes could be from each other. It has been argued that the reversal distance is a more biologically accurate representation of what can happen in nature [5]. However, there are some disadvantages, such as the fact that finding the reversal distance for unsigned genomes is NP-hard, whereas the breakpoint distance can be easily computed in both signed and unsigned cases [10].

In general, eukaryotic cells (cells with nuclei) have several linear chromosomes (as opposed to prokaryotic cells which have circular chromosomes). Most of the research on the median problem involves the simplified unichromosomal case. The multichromosomal median problem is less studied due to its conceived theoretical difficulty compared to the unichromosomal case. However, the linear multichromosomal median problem is the more accurate model for the eukaryotes, which includes all complex organisms. In this paper our focus is on finding solutions to the linear multichromosomal breakpoint median problem (BMP). To the best of our knowledge, our method is the first practical method that provides solutions for both signed and unsigned linear multichromosomal BMP with the ability of considering several possibilities for the number of chromosomes in the desired solution.

The unichromosomal BMP is NP-hard for signed and unsigned genomes ([6] and [14]). In 2009, Tannier, Zheng and Sankoff ([19]) showed that for the case of multichromosomal genomes where circular and mixed genomes are allowed, the BMP can be solved in polynomial time. They also showed that if we only allow linear chromosomes, the

median problem becomes NP-hard. In this paper we focus on the case where only linear chromosomes are allowed, i.e. the NP-hard case. As previously mentioned, the assumption of allowing only linear chromosomes is the relevant case for all eukaryotes. In this analysis, we consider multichromosomal genomes where each gene is present exactly once in every genome. We first define the framework in which we can mathematically represent the median problem. Sankoff and Blancehtte [16] gave a reduction of the unichromosomal BMP to another well-known problem, the Travelling Salesman Problem (TSP). In this paper we extend this method and provide a novel approach for a transformation from the multichromosomal BMP to the multiple salesmen TSP. We study the case for both signed and unsigned linear genomes. Then, we apply a second transformation from the multiple salesmen TSP to the usual TSP. The subject of several books in the past few years, TSP is arguably the most intensively studied problem in combinatorial optimization [12]. Therefore, a transformation of multichromosomal BMP to an instance of TSP opens the door to the vast knowledge and tools available for solving TSP which we can apply for finding the median. In particular, we take advantage of a software package called Concorde for solving TSP [2]. We demonstrate the usefulness of our method by presenting the results of an empirical study on both simulated and real-world data, with a comparison to another method.

2. DEFINITIONS AND BACKGROUND

We first present the definitions for unsigned genomes. An *unsigned gene* g is a sequence of DNA where the orientation is unknown. A set of unsigned genes form an unsigned *genome*. We can represent an unsigned genome on n genes by a string of unsigned integers $1, 2, \ldots, n$ which represents the ordering of the genes in the genome. This string can be broken into segments, representing the *chromosomes* of the genome. Chromosomes can be *circular* or *linear*. Circular chromosomes have a circular gene ordering (which will be represented with brackets around the segment). A linear chromosome has two extremities, called the *telomeres* of the chromosome. For example, G = (3 6 10 1) | (2 4 5) | (7 9 8) represents an unsigned genome G on 10 genes with three circular chromosomes, where the chromosome are separated from each other by a vertical line. H = 1 2 4 3 has one linear chromosome with 4 unsigned genes. Gene 1 and gene 3 are the telomeres of H. If all of the chromosomes in a genome are circular, the genome is called *circular*, and if all of the chromosomes are linear, the genome is called *linear*. If some chromosomes are linear and some are circular, the genome is called *mixed*.

Given an unsigned genome A, we say two genes are *adjacent* in A if they are adjacent in the gene ordering. For example, in G above, 2 and 4 are adjacent, 7 and 8 are adjacent, and 3 and 10 are not adjacent. In H above, 4 and 3 are adjacent, 1 and 3 are not.

Next we present the definitions for signed genomes. In this case, the orientation of each gene is known, and a gene g is a sequence of DNA with two extremities called the tail and the head, denoted by g_t and g_h respectively. A *signed genome* is a sequence of oriented genes. As in the unsigned case, we can represent a signed genome on n genes by a string of integers $1, 2, \ldots, n$ broken into segments representing the chromosomes, where each integer will be given a sign (+ or -) representing the orientation of the gene. In this notation, we let $+g$ represent the gene g in the orientation $g_h g_t$, and we let $-g$ represents the gene in the orientation $g_t g_h$. For example, consider the following signed genome C with 5 genes:

$$C = (-3 - 4 + 1) \ | \ +5 - 2$$

Genome C has two chromosomes, one is circular and one is linear. We could also represent C by writing each gene as its ordered tail and head extremities:

$$C = (3_t 3_h \ 4_t 4_h \ 1_h 1_t) \ | \ 5_h 5_t \ 2_t 2_h$$

Note that for each linear or circular chromosome, there are two equivalent strings where one is obtained from the other by reversing the order and switching the signs of all the genes.

For signed genomes, the adjacencies are not defined on the genes, but instead on the extremities of the genes, namely the heads and tails of the genes. We say an extremity of gene u and an extremity of gene v are adjacent if they are adjacent in the ordering of the genome. Thus, there are four possible adjacencies between two genes u and v depending on the direction of the genes: $u_h v_h$, $u_h v_t$, $u_t v_h$, or $u_t v_t$. The telomeres are sets that contain one element, i.e. u_h, u_t, v_h or v_t.

For example, in the genomes C above, the adjacenies are 3_t and 1_t, 3_h and 4_t, 4_h and 1_h, and 5_t and 2_t. The telomeres of the genome are the gene extremities at the ends of linear chromosomes. For example, in the genome C above, 5_h and 2_h are telomeres.

We now define breakpoint and breakpoint distance, for both signed and unsigned genomes. Consider two genomes A and B on the same set of n genes, where both are either signed or both are unsigned. If two genes (or two gene extremities, in the case of signed genomes) are adjacent in A, but not in B, then we say they determine a *breakpoint*. Note that the usual notion of breakpoint distance is defined on unichromosomal genomes, where the *breakpoint distance* between A and B is defined as the number of breakpoints in A (or B). This can be calculated as $d(A, B) = n - a(A, B)$ for two circular genomes, where $a(A, B)$ represents the number of common adjacencies between genomes A and B.

As described in [19], the breakpoint distance between two (signed or unsigned) multichromosomal genomes G and H on the same set of n genes can be defined as

$$d(G, H) = n - a(G, H) - \frac{e(G, H)}{2} \qquad (1)$$

where $a(G, H)$ is the number of common adjacencies between G and H, and $e(G, H)$ is the number of common telomeres of G and H. Such a definition counts one breakpoint for a fusion or a fission of two linear chromosomes (for two unsigned multichromosomal genomes, if each genome contains a linear chromosome with exactly one gene g, then

g contributes twice to the number of common telomeres of the two genomes). Note that we can equivalently think of the breakpoint distance between two multichromosomal genomes G and H as the number of breakpoints between G and H plus half the number of times a gene g is a telomere in one of the two genomes, but not the other.

As an example of using Equality (1), consider the genome C from before, and the following genome D with 3 chromosomes on 5 genes:

$D = (-3\ -4)\ \mid\ -5\ \mid\ -1\ -2$

Then $a(C, D) = 1$, $e(C, D) = 2$, and $d(C, D) = 3$.

Given three genomes A, B, and C, the *breakpoint median problem* (BMP) is the problem of finding a genome M, called the *median*, such that the sum of the breakpoint distances between M and each other genome is minimized.

One of the most well-known problems in combinatorial optimization is the *Travelling Salesman Problem (TSP)* (see [12] for background on the TSP). Consider a complete weighted graph. A *Hamilton cycle* is a cycle of the graph that visits every vertex exactly once. A minimum-weight Hamilton cycle is a Hamilton cycle such that the sum of the edge-weights of the cycle is minimized. An optimal solution to the TSP calls for finding such a minimum-weight Hamilton cycle. Due to the wide range of applications and its theoretical appeal, there is a huge amount of research on the TSP (see [11] and [12] for example).

The TSP is known to be NP-hard. Currently the best k-approximation algorithm known for the TSP, when the costs satisfy the triangle inequality, is the algorithm due to Christofides [8] for which $k = \frac{3}{2}$. However, for the general TSP, a k-approximation algorithm for any constant k would imply P=NP, and thus it is considered highly unlikely to exist [12].

A generalization of the well-known TSP is to consider multiple salesmen. A *Multiple Travelling Salesman Problem* (mTSP) is an assignment of exactly m salesmen to a set of vertices of a graph such that all salesmen start and end their journey at a fixed vertex called the *depot* and each other vertex gets visited exactly once by exactly one salesman. The goal of the mTSP is to minimize the total cost of all of the routes. We define the *r-to-m Multiple Travelling Salesman Problem* (rmTSP) as a variation of the mTSP in which at least r and at most m salesmen are used. Due to several real-life applications, the mTSP has been the subject of several studies [4].

3. TRANSFORMING THE MULTICHROMOSOMAL MEDIAN PROBLEM TO MTSP

3.1 Median problem for unsigned genomes

In this section we describe our solution method for solving the linear multichromosomal BMP in the case where the genomes are unsigned. In 1997, Sankoff and Blanchette [16] showed that the unichromosomal circular BMP can be reduced to the TSP in the case of three unsigned genomes A, B and C over a set of n genes. In [17] the same authors used a branch-and-bound method to solve this TSP. Although useful for many problems in combinatorial optimization, branch-and-bound is not usually an effective method for solving the TSP.

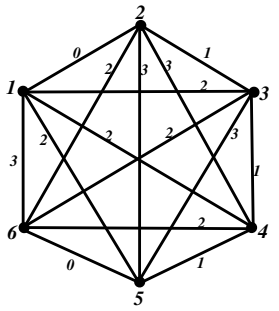

Figure 1: BP-median graph constructed for three circular genomes A = (1 2 6 5 4 3), B = (2 1 4 5 6 3), and C = (1 2 3 4 6 5). A breakpoint median is M = (1 2 3 6 5 4).

The transformation in [16] is as follows. Let G be a complete undirected graph of n vertices such that each vertex represents one gene. For each edge uv let $adj(uv)$ be equal to the number of times the genes corresponding to u and v are adjacent (do not form a breakpoint) in genomes A, B, and C (so $adj(uv)$ can be 0, 1, 2, or 3). Let the weight of the edge uv, $w(uv)$, be equal to $3 - adj(uv)$. Then the solution to the TSP for the weighted graph G traces out a permutation of $\{1, 2, \ldots, n\}$ that provides an optimal solution to the BMP. We call the weighted graph G the *BP-median graph*. Note that in this method, if we use an edge uv in the final TSP solution, this corresponds to genes u and v being adjacent in the median genome M. The weight $w(uv)$ in the BP-median graph (i.e. $3 - adj(uv)$) represents the number of times u and v are not adjacent in three genomes A,B and C. Thus using the edge uv in the TSP solution contributes exactly $w(uv)$ to the sum $d(M, A) + d(M, B) + d(M, C)$ which we are trying to minimize, i.e. the sum of the distance between the median and the other genomes.

Since the goal of the TSP is to find a minimum cost Hamilton cycle, the solution given by the TSP optimizes the use of adjacencies available in the three genomes. Hence, starting at any vertex of a cycle that is a TSP solution for the problem, we will have a genome that is the breakpoint median for the given three genomes. Figure 1 provides the BP-median graph for a simple example of three circular unichromosomal genomes A = (1 2 6 5 4 3), B = (2 1 4 5 6 3), and C = (1 2 3 4 6 5). The edge weights are assigned by using $w(uv) = 3 - adj(uv)$. It can be seen from the example that an optimal TSP solution is 1236541 with the minimum cost equal to 6. Therefore, a breakpoint median for this problem is M = (1 2 3 6 5 4), and $d(M, A) + d(M, B) + d(M, C) = 6$.

We first provide a transformation from the unsigned linear multichromosomal BMP to an instance of an rmTSP.

Construct the graph G for the genomes the same way as described for the BP-median graph, i.e. G is a complete graph

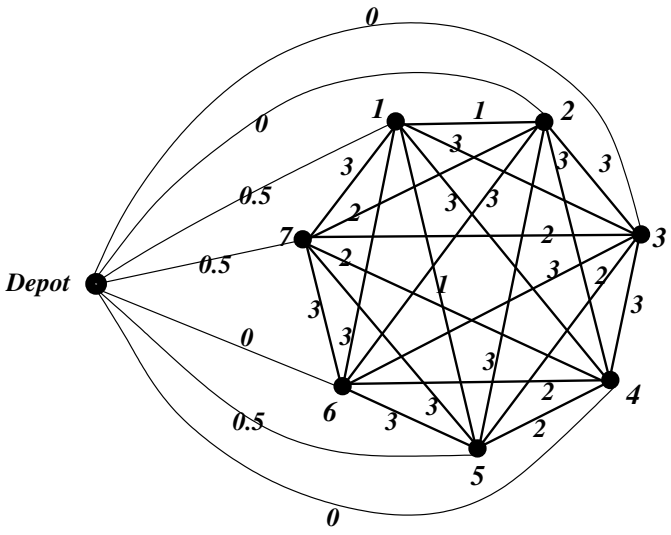

Figure 2: An example of transforming a linear multichromosomal BMP to an rmTSP with one depot. The three genomes are $A = 1\ 2 \mid 3 \mid 4\ 5 \mid 6 \mid 7$, $B = 1\ 5\ 3 \mid 2\ 7 \mid 4\ 6$, and $C = 3\ 7\ 4 \mid 6 \mid 5\ 1\ 2$. The chromosomes in each genome are separated by vertical lines.

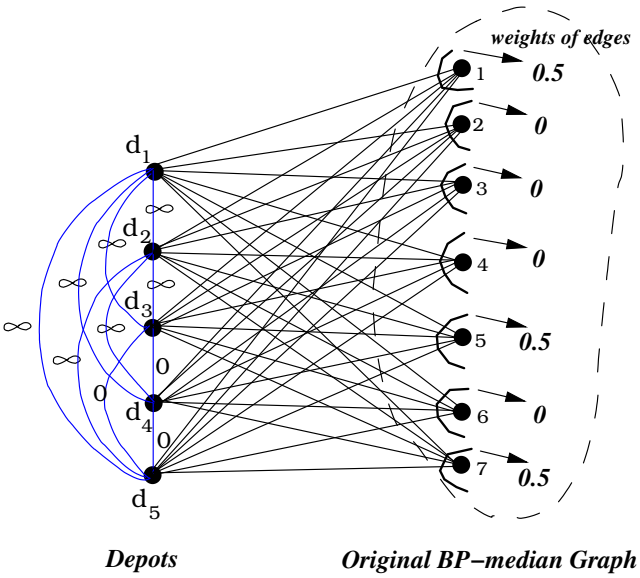

Figure 3: Transforming an rmTSP to TSP for r=3, m=5. The original BP-median graph is the graph presented in Figure 2 (without the depot). The edges between the depots and the original graph are grouped together based on their endpoint in the original graph. Each group has the same weight written besides the arrow.

of n vertices corresponding to n genes where the weight of an edge uv is equal to $3 - adj(uv)$. Then add a vertex d called the *depot* and add an edge between d and every other vertex g of the graph. Let $\tau(g)$ be equal to the number of times g is a telomere in one of the three genomes. Then, the edge dg has a weight $w(dg) = \frac{3-\tau(g)}{2}$.

Figure 2 provides an example of such a transformation for the following three linear multichromosomal genomes A,B and C, where the chromosomes are separated by vertical lines:

$A = 1\ 2 \mid 3 \mid 4\ 5 \mid 6 \mid 7$,

$B = 1\ 5\ 3 \mid 2\ 7 \mid 4\ 6$,

and $C = 3\ 7\ 4 \mid 6 \mid 5\ 1\ 2$.

An rmTSP solution to this graph corresponds to a set of say C cycles, with $r \leq C \leq m$, starting and ending at the depot such that each non-depot vertex belongs to exactly one cycle, with the total cost minimized. Once we obtain such a solution, we can delete the depot and have C disjoint paths (r$\leq C \leq$ m) covering all vertices of the graph such that the overall cost is minimized. Each path will correspond to a linear chromosome in the solution to the multichromosomal BMP. We claim this is the solution to the BMP, and that this solution has cost equal to the cost of the rmTSP solution. To see this, first consider an edge uv in the rmTSP graph where u and v are gene (non-depot) vertices. As we described before for the unichromosomal circular BMP, if we use the edge uv in the final rmTSP solution, this corresponds to u and v being adjacent in the final BMP solution, and the rmTSP solution contributes exactly $w(uv) = 3 - adj(uv)$ to the BMP solution cost $d(M, A) + d(M, B) + d(M, C)$ that we are trying to minimize. Next, consider an edge Dg in the rmTSP graph, where D is a depot vertex and g is a gene (non-depot) vertex. The weight $w(Dg) = \frac{3-\tau(g)}{2}$ represents half the number of times g is not a telomere in A, B, or C. If we use the edge Dg in the final rmTSP solution, this corresponds to gene g being a telomere in the final BMP solution, and the rmTSP solution contributes exactly $w(Dg) = \frac{3-\tau(g)}{2}$ to the BMP solution value of $d(M, A) + d(M, B) + d(M, C)$.

Next we show that an rmTSP can be transformed to a TSP. Several methods for such a transformation exist (see [4] for a survey of the results). We provide the most straight-forward transformation in this context.

ALGORITHM 1. *A transformation of rmTSP to TSP*

- *Include m copies of the depot, d_1,...,d_m.*
- *Add edges from each depot to all non-depot vertices $v_1, \ldots, v_n$. An edge $d_i v_j$ has the same weight as its corresponding edge dv_j in the original graph.*
- *Add an edge between each pair of depots.*
- *Assign the weight ∞ to the edges between $d_1, \ldots, d_{r-1}$ depots and from these $r-1$ depots to all other depots.*
- *Assign the weight 0 to the edges between $d_r, \ldots, d_m$ depots.*

- *All other edges in the graph (edges with $v_1, \ldots, v_n$ vertices as endpoints) have weights equal to the weights assigned as per the original BP-median graph.*

- *Find a TSP solution for this expanded graph.*

Figure 3 provides a sketch of the transformation mentioned in Algorithm 1 for $r = 3$ and $m = 5$.

In order to minimize the total cost, a TSP solution to the expanded graph explained above will be forced to not choose the edges with weight ∞. Therefore, a TSP tour visits at least $r - 1$ depots (the $r - 1$ depots with costs ∞ between them) and for the remaining depots, it will require between 1 and $m - r + 1$ visits. Thus it provides a solution to the rmTSP problem.

A TSP solution on this graph visits every vertex exactly once while minimizing the total sum of the edges it goes through. Therefore, those edges with weight ∞ are not included in the TSP solution. So for the $r - 1$ depots for which the incident edges from depots all have value infinity (d_1 and d_2 in Figure 3), a TSP solution must visit those depots by using two edges that have one end in the original BP-median graph. The other depots (d_3, d_4 and d_5 in Figure 3) may get visited by using the edges between them (edges of weight 0), or the edges with one end in the original BP-median graph, or a combination of these two types of edges. Therefore, a TSP solution has to visit the subgraph containing the depots (the left side of Figure 3) at least r times and at most m times. So we have obtained a solution to the rm-TSP.

By combining the transformation from the unsigned linear multichromosomal BMP to an instance of an rmTSP and Algorithm 1, we have reduced the unsigned linear multichromosomal BMP to a TSP. Once a solution to the TSP with r to m visits to the depots is obtained, we delete the depots and the adjacent edges. Deleting the depots will result in at least r and at most m disjoint paths covering all vertices exactly once. A TSP solution has minimized the total cost on the edges. Therefore, after removal of the depots, this solution traces a set of minimum cost disjoint paths covering the vertices of the graph. Each of the these disjoint paths gives the sequence of genes in one linear chromosome of the median. So we have obtained a median with at least r and at most m linear chromosomes.

For example, if we consider our previous example of A, B and C, our optimal TSP tour for the TSP problem shown in Figure 3 is d_1 3 7 4 d_5 d_3 2 d_2 5 1 d_4 6 , which gives the optimal median solution $M = 3\ 7\ 4 \mid 2 \mid 5\ 1 \mid 6$.

It should be noted that there is a one-to-one correspondence between the solutions of the multichromosomal BMP and the TSP formed. As explained above, every linear multichromosomal BMP can be solved by transforming it into a TSP. On the other hand, every optimal TSP tour provides an ordering that minimizes the total cost of the edges of the graph and therefore, it provides the order of genes in a multichromosomal breakpoint median. Also, the length of the TSP tour is equal to the sum of the breakpoint distances between the median and the other genomes.

3.1.1 Special cases for various numbers of chromosomes

Note that our solution method for the linear multichromosomal BMP allows us to choose the range (r to m) of the number of linear chromosomes which we will allow in the BMP solution. Thus, we can extend our method to include several special cases of the median problem that may be useful in real-world applications. For example, the common ancestor of the grass family in 52.5 million years ago has 12 chromosomes, while the children have anywhere from 5 to 12 chromosomes: sorghum has 10, rice has 12, Brachypodium distachyon has 5 and wheat lineage has 7 chromosomes [15]. Therefore, in such examples it may be appropriate for the algorithm to set a maximum for the number of the chromosomes of the ancestral genome, where that maximum is defined by some parameters or biological data such as the largest number of chromosomes in the children. In such cases, we would use an rmTSP with $r = 1$ (or $r =$ minimum number of chromosomes required) and m equals the maximum.

Other cases, such as enforcing the ancestor to have exactly k chromosomes might be desired in cases when k is given by some other methods (statistical, laboratory, etc.), but the sequence of genes in the median is unknown. In this case we would use $r = m = k$ in the rmTSP model.

3.1.2 Linear unichromosomal BMP

We must point out that if we only add one depot (set $r = m = 1$ in our transformation), we are using exactly one salesman. This corresponds to having one chromosome, i.e. the unichromosomal BMP.

There is a key difference between how the unichromosomal BMP for a linear versus a circular chromosome is modeled that often goes unnoticed. The transformation provided in [16] from the unichromosomal BMP to the TSP is valid for circular unichromosomal genomes. Though not specifically mentioned in the literature, it is commonly assumed that the case for linear unichromosomal genomes is similar to the circular case, in the sense that the circular median can be found and then "cut" to obtain a linear median. However, this assumption is not necessarily true. The main problem is that once a circular median in the form of a TSP tour is obtained, it is unclear where to "cut" the tour to obtain a path corresponding to a linear median. Moreover, an optimal TSP tour corresponding to a solution to the circular BMP, may not contain the optimal solution to the linear case (an optimal path covering all vertices of the graph exactly once) at all. For instance, in the example shown in Figure 1, a circular median is $M = (1\ 2\ 3\ 6\ 5\ 4)$ with an optimal cost of 6. This is an optimal solution to the circular unichromosomal BMP and an optimal TSP tour. The optimal solution to the linear unichromosomal BMP for this example is $P = 1\ 2\ 3\ 4\ 5\ 6$ with cost 3. However, it is impossible to obtain an optimal path P from M. Indeed, the best path that can be obtained from M by removing one edge, would have a cost of at least 4, which is higher that the cost of P. Therefore, the optimal solution M to the circular unichromosomal BMP may not yield to an optimal solution for the linear version of the problem.

To fix this problem, we propose the following solution. Use our method for the linear multichromosomal BMP, with the restriction that there is exactly one chromosome in each genome. Then, the unichromosomal case is just a special case of the multichromosomal BMP. Specifically, add a depot vertex to the BP-median graph, and add the edges between the depot and all other vertices with the edge weight of $\frac{3-\tau(g)}{2}$. Solve the TSP on this extended graph where an optimal tour starts and ends at the depot. Then remove the depot to obtain a path of minimum cost that goes through all vertices. This path corresponds to the sequence of the genes in the solution for the linear unichromosomal BMP.

3.2 Median problem for signed genomes

The transformation described in the previous section considers unsigned genomes. In this section we provide a method to transform a signed unichromosomal circular BMP to a TSP instance and we then generalize the result to include the linear multichromosomal case, and thereby reduce the linear multichromosomal breakpoint median problem on signed genomes to a TSP. Recall that the breakpoint distance between two signed multichromosomal genomes can be defined using Equality(1), and that in signed genomes, there are four possible adjacencies between two genes u and v depending on the direction of the genes: $u_h v_h$, $u_h v_t$, $u_t v_h$, or $u_t v_t$. The telomeres are sets that contain one element, i.e. u_h, u_t, v_h or v_t.

Consider the signed BMP on three unichromosomal circular genomes A, B, and C where each genome has n (signed) genes. We construct a *signed BP-median graph* G with $3n$ vertices. Each gene g in the chromosome corresponds to three vertices g_h (representing the head of the gene), g_t (representing the tail of the gene) and g_m in the graph. Both edges $g_h g_m$ and $g_m g_t$ have weights equal to zero. All the other edges with g_m as an endpoint have weights equal to infinity. Also, all edges between the head and the tail of the same gene, $g_h g_t$, have weights equal to infinity. See Figure 4. A positive (negative) gene in a genome indicates that in a Hamilton cycle of G, the head (tail) is visited before the tail (head). All other edges are formed as in the BP-median graph, and the weights are similarly assigned based on the adjacencies in the signed genomes: for example, the weight $w(g_t f_h)$ on the edge between the tail of gene g and the head of gene f is $3 - adj(g_t f_h)$ where $adj(g_t f_h)$ is the number of times tail of g and head of f are adjacent.

The existence of a vertex g_m between the head and the tail of each gene g is necessary to ensure that an optimal tour for the TSP includes g_h if g_t is picked (and vice-versa) by travelling through g_m. Otherwise, if g_m is not included in the model, it is possible to have a case where one endpoint of a gene is visited and it is not followed by the other endpoint. Note that the edges of weight ∞ in the signed BP-Median graph G ensure that the edges $g_t g_m$ and $g_m g_t$ will be included in the optimal TSP solution for every gene g.

Similar to the unsigned case, a solution to the TSP on the signed BP-median graph provides a solution to the BMP simply by taking the vertices in the order they are visited in the TSP solution, ignoring the g_m vertices. For example, in Figure 4, the TSP tour $1_t 1_m 1_h 2_h 2_m 2_t 3_t 3_m 3_h 1_t$ corresponds to the genome $(1_t 1_h\ 2_h 2_t\ 3_t 3_h)$.

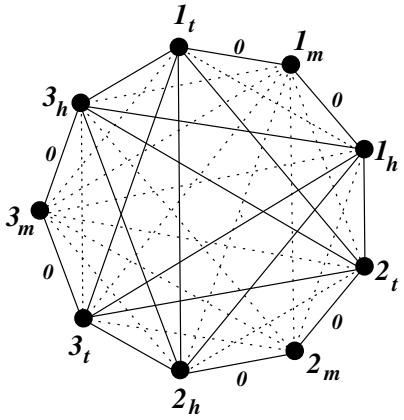

Figure 4: An example of transforming a circular unichromosomal signed BMP for 3 genes to a TSP on 9 vertices: the solid unlabelled edges will receive weights according to the adjacencies, and the dotted edges have infinity as their weight.

In the multichromosomal case, similar to what was described previously for the unsigned case, the problem can be transformed to an rmTSP.

ALGORITHM 2. *Transformation of a signed multichromosomal BMP to an rmTSP*

- *Add a depot vertex d.*
- *Add an edge between the depot d and all vertices g corresponding to the heads and tails, such that the weight of such edge is equal to* $w(dg) = \frac{3-\tau(g)}{2}$ *where* $\tau(g)$ *is the number of times g is a telomere in one of the chromosomes of the three genomes.*
- *Add an edge of weight* ∞ *between the depot and all vertices* g_m*.*
- *Find a solution to the rmTSP on this graph.*

A solution to the rmTSP described above has at least r and at most m visits to the depot. Let P be such a solution. After deleting the depot, we are left with at least r and at most m disjoint paths covering all (non-depot) vertices of the graph. Once again, if we ignore the g_m vertices, each path corresponds to a chromosome in the solution to the signed multichromosomal BMP. By combining Algorithm 1 and Algorithm 2 a multichromosomal BMP for signed genomes can be transformed to a TSP.

Similar to Section 3.1.1 for unsigned genomes, our proposed method for signed genomes can handle different requirements on the number of linear chromosomes.

As mentioned in [13], since typically the number of salesmen m for an rmTSP is much less than the number of vertices n of the graph, the addition of m depots and therefore expanding the graph to $n+m$ vertices is not a substantial increase, when

$m << n$. Also, in theory, the mTSP on $n+m$ vertices should be an easier problem to solve than the TSP on n vertices [13]. From a biological point of view, this is also accurate, since the m depots correspond to m chromosomes, and typically, the number of chromosomes is far less than n, the number of genes.

4. COMPUTATIONAL EXPERIMENTS

We tested our method on randomly generated datasets. For this purpose, we randomly generated sets of genomes with various numbers of chromosomes. To simulate the occurrence of breakpoints, randomly selected genes in each genome were swapped. The number of gene swaps corresponding to the occurrence of breakpoints were assigned by setting a ratio $k = \frac{m}{n} \times 100$ where m is the number of breakpoints compared to the identity, and n is the total number of genes present in the genome. We performed our tests for a range of ratios between 10% and 90%, for the number of genes ranging from 10 to 10,000. We also tested this method on both signed and unsigned genomes with various numbers of chromosomes: A fixed chromosome number for all genomes, and a lower and upper bound for the number of chromosomes allowed in each genome. After generating these random datasets, we created the BP-median graph for each test. We then used our transformation to add the depots to the graph based on the number of chromosomes, and found the input for the rmTSP. The next step was to apply the second algorithm to tranform the rmTSP to a TSP. Once we obtained an instance of the TSP corresponding to our linear multichromosomal BMP, we applied Concorde [2] to solve the TSP. Concorde is a sophisticated TSP solver that combines many of the most recent advances on TSP research to find exact or provably nearly-exact solutions for a TSP. It uses the linear programming formulation of the TSP and applies advanced branch-and-cut steps to move towards the optimal solution. Concorde starts from a feasible solution (not necessarily optimal) and applies cutting-plane methods to move towards the optimal solution. It also finds a lower bound on the value of the optimal solution. If a solution is obtained with value equal to the lower bound, then it is an optimal solution. If the optimal solution cannot be reached, Concorde provides the best found solution and a bound indicating its worst-case distance from optimality. The output from Concorde is in the form of a sequence of vertices that gives the TSP tour. Based on our method, we can translate this tour back into the sequence of the genes present in the median. We used the C programming language, and all the tests were performed on an Intel Xeon 3.2 GHz with 3.2 GB of memory and the GCC compiler for Linux. The reported results are for $k = 30\%$ and they are averages over 5 runs for each sample type of the simulated data. There was no noticable difference in performance time and optimality gap within various ratios. The optimality gap is calculated as the worst-case difference, as a percentage, between the reported median and an optimal solution (i.e. the difference, as a percentage, between the reported median value and the lower bound found by Concorde on the optimal solution value). Note that an optimality gap of 0 indicates that an optimal solution was found. Table 1 provides a summary of the results of applying our method for the multichromosomal BMP. The average time to find the optimal median in the case of 3 chromosomes in each genome was 127.54 seconds. Note that we were successful in obtaining an optimal median in all cases where $n < 10000$. For $n \geq 10000$, the worst-case accuracy of 7% gap from the optimal solution was for 25000 genes per genome spread over 23 chromosomes.

For real-world datasets we tested our method on the human, cat, and mouse gene data available from [20]. This dataset consists of 114 common genes (markers) contained in several chromosomes. Our method was able to find an optimal breakpoint median of this dataset in less than one minute. The parameters chosen were $r = 19$ and $m = 23$ (The number of chromosomes for cat, mouse and human are 19, 20, and 23, respectively). An optimal median was found with 20 chromosomes.

As mentioned in other studies (for example see [23]) and to the best of our knowledge, aside from our current work, there are no other large-scale linear multichromosomal median solvers for breakpoint distance. We should mention that there are other tools available for finding the median of multichromosomal genomes using other distance functions. These tools include MGR [5], GRIMM [20], MGRA [1], and ASMedian [21]. Among these packages, MGR and GRIMM are primarily based on reversal distance and can only handle genomes of significantly smaller size (in the range of 100 or fewer genes). MGRA can handle larger data, however, it is based on a scenario called "2-break" that is used to approximate other rearrangement scenarios such as reversals. ASMedian, solves the median problem under Double Cut-and-Join (DCJ) distance. A recent version of ASMedian, called ASMedian-linear [22] has been developed for linear chromosomes under DCJ distance and tested on large datasets of up to 5000 genes, again using the DCJ distance measure. The reported running times for ASMedian-linear are very fast. One drawback is that it is possible to have circular chromosomes in the solution found by ASMedian-linear. As mentioned, none of these tools are based on breakpoint distance and therefore, we were not able to compare them to our method.

4.1 Unichromosomal breakpoint median problem

The *circular unichromosomal breakpoint median problem* can be defined in cases where each genome has only one circular chromosome. There has been more studies on the unichromosomal case compared to the multichromosomal median problem. Therefore, we decided to test our method for the unichromosomal case as well so we could compare its performance with other available packages, given that we know of no other methods for large-scale linear multichromosomal breakpoint median problem. For this purpose, we focused on the circular unichromosomal BMP for signed and unsigned genomes. This problem is known to be NP-hard ([6], [7]). In Section 3 we discussed the transformation proposed in [16] for transforming the unichromosomal BMP for unsigned genomes, as well as our adaptation of this method for signed genomes, into an instance of TSP. We use this transformation and apply the TSP solver Concorde [2] to obtain a solution to the TSP, and then translate this solution to the median. We tested our method on both real-world and synthetic datasets of 3 to 10,000 circular unichromosomal genomes, each containing 10 to 10,000 genes for both signed and unsigned cases. For all of our test data, we were able to obtain an optimal solution. The time it took for our method

Number of Genes Per Genome	Number of Chromosomes In Each Genome	Time (seconds)	Median Score	Optimality Gap
10	3	0	15	0%
100	3	5.12	182	0%
500	3	18.49	873	0%
1000	3	42.17	1802	0%
5000	3	125.19	9247	0%
10000	3	574.26	5274	0%
10	$3 \leq n \leq 5$	0	18	0%
100	$3 \leq n \leq 5$	9.07	192	0%
500	$3 \leq n \leq 5$	28.31	943	0%
1000	$3 \leq n \leq 5$	189.06	2403	0%
5000	$3 \leq n \leq 5$	170.41	10684	0%
10000	$3 \leq n \leq 5$	681.01	14278	2%
25000	n=23	1091.72	31361	7%

Table 1: Performance for the multichromosomal breakpoint median problem on simulated data sets with a ratio of 30%, averages over 5 runs for each sample type of the simulated data. The optimality gap is calculated as the difference, as a percentage, between the reported median and the lower bound found by Concorde on the value of an optimal solution.

to reach optimality ranged from less than 1 second to about 5 minutes for the largest dataset on an Intel Xeon 3.2 GHz with 3.2 GB of memory running Linux. The times reported are averages over 5 runs for each sample type of the simulated data. The optimal median is found using the optimal TSP solution from Concorde. Total average time over all samples for our method was 39.38 seconds.

For real-world datasets, we tested our method on the human, fruit fly, and sea urchin mtDNA data which was used in [18]. This dataset contained 33 signed genes over the three genomes. We also tested the Campanulaceae cpDNA dataset which was considered a more challenging dataset studied in [9]. This is a dataset of 13 genomes each with 105 genes. In both cases, our method was able to find optimal solutions in a matter of seconds.

A well-known genomic median solver is GRAPPA [3]. We compared the performance of our method to GRAPPA. For smaller genomes (less than 60 genes per genome) GRAPPA was faster than our method, however, once the genome size exceeded 60 genes, our method outperformed GRAPPA in terms of speed. For $n \geq 1000$, GRAPPA was unable to find any median even after one hour of running time. Also, in terms of accuracy, our method obtained provably optimal results for all test cases, whereas GRAPPA was, on average, 20% away from optimal in the test cases where it was applied and able to find a solution. Table 2 provides the results for the unichromosomal BMP.

5. CONCLUSIONS

In this paper, a novel approach for solving the breakpoint median problem on signed and unsigned multichromosomal genomes is presented. The focus of our framework is on the NP-hard problem of finding a multichromosomal breakpoint median for linear genomes. Our method is based on constructing a complete graph that has all the genes as vertices and the edge weights representing the breakpoints. We then obtain a multiple salesman TSP by adding a depot. Next, we converted this graph into an rmTSP. The parameters r and m (minimum and maximum number of salesmen/chromosomes) can be used to capture different biological assumptions on the number of chromosomes in a common ancestor. By setting these parameters, we can use our method to test hypotheses on the number of chromosomes of a common ancestor, even when the known genomes or the available data provide less chromosomes. It should be noted that the parameters r and m only need to be specified if desired. If such information is unknown or we do not want to restrict our solutions to particular parameters, we can still use the same method without any restrictions. In such cases, we can set r to be the minimum number of chromosomes (say 1) and m to be the maximum number of chromosomes possible.

Also, our method can be easily extended to include more than 3 genomes. For example, if we want to find the median of K genomes, it suffices to set the weights as $w(gh) = K - adj(gh)$ instead of $3 - adj(gh)$ for genes g and h, and $w(dg) = \frac{K-\tau(g)}{2}$ instead of $\frac{3-\tau(g)}{2}$ for a depot d and a gene g in our TSP transformation.

The other advantage of this method is its computational efficiency in terms of both the running time and also accuracy of finding optimal results. In terms of the running time, the presented method allowed us to compute the median in realistic time limits on datasets which can appropriately resemble real-world data. In all cases up to 5000 genes an optimal solution is found. For datasets of size 10,000 the optimality gap is 2% and in the largest dataset that was tested, $n = 25000$ with 23 chromosomes, the worst case difference between the solution we found and the optimal solution value is 7% and this solution was found in less than 20 minutes.

One drawback of this method is the assumption of containing equal gene content in the genomes. This is a very common assumption in the field in order to reduce the general problem into manageable datasets. Ideally, models should

Number of genes per genome	Time for our method (sec.)	Time for GRAPPA (sec.)	Median Score with our method	Median Score with GRAPPA	Optimal Median
10	0.01	0	11	14	11
20	0.05	0	33	43	33
30	0.07	0.02	44	56	44
50	0.18	0.04	76	91	76
60	0.33	0.28	96	104	96
70	0.49	3.2 min	113	121	113
80	0.76	12.5 min	137	148	137
90	1.08	48.7 min	142	153	142
100	1.34	$>$ 60 min	168	172	168
150	2.02	$>$ 60 min	259	267	259
200	3.88	$>$ 60 min	351	352	351
500	10.74	$>$ 60 min	861	869	861
1000	28.12	$>$ 60 min	1743	N/A	1743
1500	22.27	$>$ 60 min	2640	N/A	2640
2000	39.31	$>$ 60 min	3485	N/A	3485
3000	54.07	$>$ 60 min	5259	N/A	5259
5000	83.54	$>$ 60 min	8719	N/A	8719
10000	328.14	$>$ 60 min	10751	N/A	10751

Table 2: Our method versus GRAPPA: unichromosomal breakpoint median problem. The results are for simulated data with a ratio of 30%. The times reported are averages over 5 runs for each sample type of the simulated data.

allow for unequal numbers of genes in the genomes. We are planning to extend our method to include cases where unequal gene content can be considered, and we believe that such an extension of our framework can be done. For future work, including several copies of the genes may also lead to a more realistic model.

We should point out that finding a median is a combinatorial optimization problem that is only a rough approximation of a common ancestor. With more genomic data becoming available and different biological hypotheses tested on known common ancestors, it would be very interesting to compare the results of the median obtained by our method to the sequence of a known common ancestor. Such comparison could shed light on appropriate parameters and constraints that need to be considered in computing medians in the future.

6. ACKNOWLEDGEMENTS

We would like to thank the anonymous referees of this paper for their helpful comments.

7. REFERENCES

[1] Alekseyev, M.A., and Pevzner, P.A.: Breakpoint graphs and ancestral genome reconstructions, Genome Research, 19:943-957, 2009.

[2] Applegate, D., Bixby, R., Chvatal, V., and Cook, W.: Concorde TSP Solver, www.tsp.gatech.edu/concorde

[3] Bader, D.A., Moret, B.M.E., Warnow, T., Wyman, S.K., and Yan., M.: GRAPPA (Genome Rearrangements Analysis under Parsimony and other Phylogenetic Algorithms), www.cs.unm.edu/ moret/GRAPPA/

[4] Bektas, T.: The multiple Traveling Salesman Problem: an overview of formulations and solution procedures, Elsevier-Omega 34, 209-219, 2006.

[5] Bourque, G., and Pevzner, P.A.: Genome-scale evolution: reconstructing gene orders in ancestral species, Genome Research, 12:26-36, 2002, http://grimm.ucsd.edu/MGR/

[6] Bryant, D.: The complexity of the breakpoint median problem, technical report CRM-2579, Centre de Recherches de Mathematiques, Universite de Montreal, 1998.

[7] Caprara, A.: Additive bounding, worst-case analysis and the breakpoint median problem, SIAM Journal on Optimization, 13:508-519, 2002.

[8] Christofides, N.: Worst case analysis of a new heuristic for the traveling salesman problem, Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, 1976.

[9] Cosner, M.E., Jansen, R.K., Moret, B.M.E., Raubeson, L.A., Wang, L.S., Warnow, T., and Wyman, S.: An empirical comparison of phylogenetic methods on chloroplast gene order data in Campanulaceae, Comparative Genomics (DCAF-2000), 104-115, 2000.

[10] Fertin, G., Labarre, A., Rusu, I., Tannier, E., and Vialette, S.: Combinatorics of genome rearrangements, The MIT Press, 2009.

[11] Gutin, G., and Punnen, A.P. (eds.): The traveling salesman problem and its variations, Kluwer, Dordrecht, 2002.

[12] Lawler, E., Lenstra, J., Rinnooy Kan, A., and Shmoys, D.: The traveling salesman problem–a guided tour of combinatorial optimization, Wiley, Chichester, 1985.

[13] Orloff, C.S.: Routing a fleet of m vehicles to/from a central facility, Networks, 4, 147-162, 1974.

[14] Pe'er, I., and Shamir, R.: The median problems for breakpoints are NP-complete, Electronic Colloquium on Computational Complexity, technical reports 71, 1998.

[15] Salse, J., Bolot, S., Throude, M., Jouffe, V., Piegu, B., Quraishi, U.M., Calcagno, T., Cooke, R., Delseny, M., and Feuillet, C.: Identification and characterization of shared duplications between rice and wheat provide new insight into grass genome evolution, The Plant Cell, 20:11-24, 2008.

[16] Sankoff, D., and Blanchette, M.: The median problem for breakpoints in comparative genomics, in Proceedings of the

Third International Computing and Combinatorics Conference, T.Jiang and D.T.Lee eds, Lecture Notes in Computer Science 1276:251-263, Springer-Verlag, 1997.

[17] Sankoff, D., and Blanchette, M.: Multiple genome rearrangement and breakpoint phylogeny, Journal of Computational Biology, 5:555-570, 1998.

[18] Sankoff, D., Sundaram, G., and Kececioglu, J.: Steiner points in the space of genome rearrangements, International Journal of Foundations of Computer Science 7:1-9, 1996.

[19] Tannier, E., Zheng, C., and Sankoff, D.: Multichromosomal median and halving problems under different genomic distances, BMC Bioinformatics, 10:120, 2009.

[20] Tesler,G.: GRIMM: genome rearrangements web server, Bioinformatics, 18-3: 492-493, 2002, http://grimm.ucsd.edu/GRIMM/

[21] Xu, A.W., and Sankoff, D.: Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem, Lecture Notes in Computer Science, 5251:25-37, 2008.

[22] Xu, A.W.: The median problems on linear multichromosomal genomes: Graph representations and fast exact solutions, Journal of Computational Biology, 17-9:1195-1211, 2010.

[23] Zhang, M., Arndt, W., and Tang, J.: An exact median solver for the DCJ distance, Proceedings of the 14th Pacific Symposium on Biocomputing, 138-149, 2009.