Adaptive randomized rounding in the big parsimony problem

Sunil Chopra* Eunseok Kim† Sangho Shim‡

May 1, 2019

Abstract

A phylogenetic tree is a binary tree where each node represents a sequence of the states and all the input sequences are represented at the leaf nodes. Given sequences of the states of the same length, the big parsimony problem constructs the most parsimonious phylogenetic tree along with labeling the internal nodes at the maximum parsimony. The big parsimony problem is known to be NP-hard. We describe randomized rounding methods that allow us to obtain good solutions.

Our first randomized rounding method starts with a fractional optimal solution to the LP-relaxation of an integer linear programming formulation of the big parsimony problem, and repeats randomized rounding based on this fractional solution, which we refer to as fixed randomized rounding without changing the fractional solution. Solutions obtained using the fixed randomized rounding approach are superior to the best solutions obtained using branch-and-bound with GUROBI and can be obtained quicker.

We then describe an adaptive randomized rounding approach where the underlying fractional solution changes based on the best integer solution observed so far and produces solutions that are superior to the fixed randomized rounding approach.

1 Introduction

The maximum parsimony method is the most widely used sequence-based tree reconstruction method. In science, the principle of maximum parsimony is to use the simplest and the most parsimonious explanation of an observation. In phylogenetic analysis, the maximum parsimony problem is to find a phylogenetic tree that explains a given set of aligned sequences using a minimum number of "evolutionary events". The maximum parsimony problem is called the *big parsimony problem*, which is distinguished from the small parsimony problem to just find the most parsimonious sequences at the internal nodes of a given phylogenetic tree on the given set of aligned sequences. The big parsimony problem is NP-hard gaining notoriety in complexity for several decades, while the small parsimony problem can be solved by Fitch's algorithm in polynomial time.

A phylogenetic tree is a tree interpreting an input set of sequences from the genomes of evolutionarily related organisms, where each node represents a sequence of the states and all the input sequences are represented at the leaf nodes. It has been defined as a binary tree because evolutionary events such as mutation or speciation are understood to split a lineage into two parts, not

*Kellogg School of Management, Northwestern University, Evanston, IL 60208, USA,
†Rutgers Business School, Rutgers University, Piscataway Twp, NJ 08854, USA,
‡School of Engineering, Mathematics and Science, Robert Morris University, Moon Twp, PA 15108, USA, shim@rmu.edu

three or more [2, 5, 6, 8, 10, 13, 19]. We thus consider only binary phylogenetic trees throughout this paper.

In order to formulate the big parsimony problem, we precisely define a rooted binary tree and a few terms which are frequently used in phylogeny. Let $L \subseteq V$ be the set of the *leaf nodes* or *leaves*. The leaf nodes represent existing genes, populations or species. A *rooted binary* tree $D = (V \cup \{0\}, A)$ is a tree with the *root* node 0 of degree 2, all the other non-leaf nodes of degree 3 and the leaf nodes of degree 1. The root node 0 is the ancestral node in a phylogenetic tree. The non-leaf nodes of degree 3 are called the *internal* nodes representing hypothetical ancestors. There is only one path from the root 0 to a node in the tree D and we can give each edge in A the direction outward from the root. The directed edge is called an *arc*. We can create an unrooted binary tree by contracting the edge between the root node and one of its two neighbors. An unrooted binary tree $G = (V, E)$ is a tree with all internal nodes $v \in V \setminus L$ of degree 3. The number of internal nodes is $|V \setminus L| = |L| - 2$ because the degree sum across the nodes implies

$$3|V \setminus L| + |L| = 2|E| = 2(|V| - 1) = 2(|V \setminus L| + |L| - 1) = 2|V \setminus L| + 2|L| - 2.$$

We see that $|A| = |E| + 1 = |V|$. Let $l = |L|$ denote the number of leaf nodes. We can number the nodes in $V \setminus L = \{1, ..., l-2\}$ and $L = \{l-1, ..., 2l-2\}$ so that each arc $(u, v) \in A$ may satisfy $u < v$. Then, the arc $(0, 1) \in A$. In order to formulate parsimony problems, it is more convenient to consider a rooted binary tree than an unrooted binary tree. Throughout this paper, we consider only rooted binary trees $T = (V \cup \{0\}, A)$ with identifying the root node 0 as the copy of internal node 1.

Definition 1 (Character, extension, parsimony value)**.**

- Let K be a finite set of *character states* (or simply *states*). In a deoxyribonucleic acid (DNA), the states are 4 nucleobases $K = \{A, T, G, C\}$. When morphological data is analyzed, binary characters are also often relevant [5, 7, 12, 13, 18, 19] and we assume that there are only two states $K = \{0, 1\}$ throughout this paper.

- A *character* on leaves L over states K is any function s from L into K.

- A function $x : V \cup \{0\} \to K$ such that $x(u) = s(u)$ for $u \in L$ is said to be an *extension* of s. We identify the states of the root node 0 and node 1 in an extension x; *i.e.*, $x(0) = x(1)$. The *restriction* of x on L is denoted by $x|_L : L \to K$; *i.e.* $x|_L(u) = x(u)$ for $u \in L$.

- Let $pv(T, x) = |\{(u, v) \in A : x(u) = x(v)\}|$. The *parsimony value* of s on T is the maximum number of arcs $(u, v) \in A$ where both end nodes are of the same state; *i.e.*,

$$pv(T, s) = \max_x \{pv(T, x) : x|_L = s, x(0) = x(1)\}.$$

The Hamming distance of (u, v) is the number of sites where u and v are labeled by different states. Let $s = (s_1, ..., s_m)$ be a sequence of characters s_j for *sites* $j \in [m] = \{1, ..., m\}$ on L. We need only examine one character at a time (i.e., we determine the solution for site 1, then we work on site 2, etc.) Thus, the parsimony value on the multiple characters (s_j) subtracts the sum of the Hamming distances on the tree arcs from $m|V|$ which is the number of sites times the number of tree edges. We simply denote $pv(T, s)$ by $pv(T)$ when the sequence of characters $s = (s_j)$ is clear. The parsimony principle finds the binary tree that requires the fewest evolutionary changes which is measured by the Hamming distance, the number of character changes (mutations) along the evolutionary tree. Throughout this paper, we use the parsimony value as a measure of the maximum parsimony.

The big parsimony problem (BPP) aims to find the rooted binary tree topology at the maximum parsimony. It is known in [1, 7] to be NP-hard:

Problem 1 (Big Parsimony Problem (BPP))**.** *Given a sequence $s = (s_1, ..., s_m)$ of characters, determine the binary tree topology T at the maximum parsimony*

$$pv(s) = \max_T \max_x \{pv(T, x) : x|_L = s, x(0) = x(1)\}.$$

Since the BPP is NP-hard, it might not be tractable to obtain the exact optimal solution. We develop a randomized rounding method with a fixed probability distribution and then an adaptive randomized rounding method that is shown to work better than the previous method.

Our first randomized rounding method starts with a feasible fractional solution to the LP-relaxation of the ILP formulation of the BPP. We show that any randomized rounding of the fractional solution can be used to construct a solution to the BPP. Repeated randomized rounding based on this fractional solution thus helps us identify a good quality solution to BPP. We refer to this approach as *fixed randomized rounding* because the underlying fractional solution does not change. Our computational experiments show that solutions obtained using fixed randomized rounding are superior to the best solutions obtained using branch-and-bound with GUROBI (limited to 7200 seconds) and can be obtained quicker.

We observe, however, that fixed randomized rounding produces solutions to BPP whose objective function values are concentrated. We are able to show that as the size of the problem instances grows, the coefficient of variation (the ratio of the standard deviation to the mean) of the objective function values obtained through fixed randomized rounding converges to 0. This finding indicates that rounding using a fixed fractional solution has some limitations.

To improve the outcome of randomized rounding, we develop an adaptive randomized rounding method where the underlying fractional solution changes based on the best integer solution observed so far. Our computational experiments show that adaptive randomized rounding produces solutions that are superior to fixed randomized rounding.

In Section 2, we develop an integer linear programming formulation of the big parsimony problem, and show that flipping a fair coin is an optimal solution to the LP-relaxation. In Section 3, fixed randomized rounding is shown to provide good solutions that are much better than the incumbent solutions produced by a branch-and-bound procedure of a professional integer linear programming software (GUROBI) within 7,200 seconds. In Section 4, we describe an adaptive randomized rounding method and provide computational results that show that adaptive randomized rounding gives much better solutions to BPP than fixed randomized rounding. Our computational experiments on several real biological data which are large mtDNA and Y-chromosome instances confirm this finding.

2 ILP formulation of the big parsimony problem

Consider a *rooted binary* tree $T = (V \cup \{0\}, A)$ with leaf nodes L. We can number the nodes in $V \setminus L = \{1, ..., |L|-2\}$ and $L = \{|L|-1, ..., 2|L|-2\}$ so that each arc $(u, v) \in A$ satisfies $u < v$. We then construct the directed graph $\bar{D} = (\bar{V}, \bar{A})$ by adding all arcs (u, v) where $u, v \in V$ and $u < v$ and not both u and v are leaf nodes. $\bar{D}$ is defined by

$$\begin{aligned}\bar{V} &= \{0\} \cup V = \{0\} \cup (V \setminus L) \cup L = \{0\} \cup \{1, ..., |L| - 2\} \cup \{|L| - 1, ..., 2|L| - 2\},\\ \bar{A} &= \{(u, v) : u < v,\ u \in \{0\} \cup (V \setminus L),\ \text{and}\ v \in V\}\,.\end{aligned}$$

That is, $\bar{D}$ is defined by constructing directed arcs from each node u that is not a leaf to each larger numbered node v. We refer to $\bar{D}$ as *binary tree search space* because it contains all possible rooted binary tree topologies.

Now we develop an ILP formulation of the BPP over the binary tree search space $\bar{D}$ employing *tree variables* $z(u, v)$, *node variables* $x_j(v, t)$ and *edge variables* $y_j(u, v)$ at each site $j = 1, ..., m$. For the set of tree arcs $A \subset \bar{A}$ of a binary tree $T = (\bar{V}, A)$, we use T and A interchangeably. For a binary tree $T \subset \bar{A}$ over the binary tree search space $\bar{D}$, $z(u, v) = 1$ if $(u, v) \in T$ and $z(u, v) = 0$ if $(u, v) \in \bar{A} \setminus T$. (If a binary tree T needs to be distinguished from other binary trees, we may use z^T for z.)

Before formulating the binary trees z, we need some more definitions. For $B \subset \bar{A}$, let $z(B) = \sum_{(u,v)\in B} z(u, v)$. For $U \subset \bar{V}$, let $\delta^-(U)$ denote the set of arcs of $\bar{A}$ incoming to U (i.e. arcs $(u, v) \in \bar{A}$ with $u \notin U, v \in U$). For simplicity, we write $\delta^-(v)$ instead of $\delta^-(\{v\})$. On the other hand, let $\delta^+(U)$ denote the set of arcs of $\bar{A}$ outgoing from U (i.e. arcs (u, v) with $u \in U, v \notin U$). For simplicity, we write $\delta^+(u)$ instead of $\delta^+(\{u\})$.

To formalize the parsimony value, our ILP formulation of the BPP includes that of the graph partition problem introduced by Chopra and Rao [3]. The node variables $x_j(v, t) \in \{0, 1\} \subset [0, 1]$ indicate a sequence of extensions $x = (x_1, ..., x_m)$; *i.e.*,

$$x_j(v, t) = \mathbf{1}[x_j(v) = t] \text{ for } v \in V \cup \{0\}, t \in K \text{ and } j = 1, ..., m$$

where $\mathbf{1}[\text{event}] = 1$ if [event] is true, and it is zero otherwise. Vector x denotes both the vector of node variables and the sequence of extensions. Likewise we can use vector $s = (s_j(u, t))$ to indicate a sequence of characters $s = (s_j)$. An edge variable $y_j(u, v) \in \{0, 1\} \subset [0, 1]$ indicates that both end nodes u, v of edge $uv \in E$ are of the same state; *i.e.*,

$$y_j(u, v) = \mathbf{1}[x_j(u) = x_j(v)] \text{ for } (u, v) \in A \text{ and } j = 1, ..., m.$$

We will call the edge variables as the *parsimony value variables* interchangeably.

Given a sequence of characters $s = (s_j)$, the boundary conditions first fix the states of the leaves:

$$x_j(v, t) = s_j(v, t) \text{ for } v \in L, t \in K \text{ and } j = 1, ..., m. \tag{1}$$

We add the following additional boundary conditions to identify node 1 with the root node 0:

$$x_j(0, t) = x_j(1, t) \text{ for } t \in K \text{ and } j = 1, ..., m, \tag{2}$$

$$y_j(0, 1) = 1 \text{ for } j = 1, ..., m. \tag{3}$$

With the boundary conditions (1)-(3), an ILP formulation of the BPP over the binary tree search space $\bar{D}$ is

$$\max \quad \sum_{j=1}^{m} \sum_{(u,v)\in\bar{A}} y_j^z(u,v) \tag{4}$$

$$s.t. \quad y_j^z(u,v) \le z(u,v) \text{ for } (u,v)\in\bar{A} \text{ and } j=1,...,m, \tag{5}$$

$$y_j^z(u,v) \le y_j(u,v) \text{ for } (u,v)\in\bar{A} \text{ and } j=1,...,m, \tag{6}$$

$$z(\delta^+(v)) = 2 \text{ for } v\in\bar{V}\setminus L, \tag{7}$$

$$z(\delta^-(v)) = 1 \text{ for } v\in V=\bar{V}\setminus\{0\}, \tag{8}$$

$$\sum_{t\in K} x_j(v,t) = 1 \text{ for } v\in\bar{V} \text{ and } j=1,...,m, \tag{9}$$

$$\left.\begin{array}{r} x_j(u,t)-x_j(v,t)+y_j(u,v)\le 1 \\ -x_j(u,t)+x_j(v,t)+y_j(u,v)\le 1 \end{array}\right\} \text{ for } (u,v)\in\bar{A}, t\in K \text{ and } j=1,...,m, \tag{10}$$

$$x_j(u,t)+x_j(v,t)-y_j(u,v)\le 1 \text{ for } (u,v)\in\bar{A}, t\in K \text{ and } j=1,...,m. \tag{11}$$

The out-degree constraints (7) and the in-degree constraints (8) define a binary tree z. Constraints (10) imply that both end nodes u,v of an arc (u,v) will have a same state if $y_j(u,v)=1$. Constraints (11) imply the other way around. Note that the linking variables $y_j^z(u,v)=1$ only if (u,v) is a tree arc (*i.e.*, $z(u,v)=1$) and the j-th states of u,v are same (*i.e.*, $y_j(u,v)=1$).

In case of binary characters (*i.e.*, $K=\{0,1\}$), we have an immediate fractional solution to the LP-relaxation. The *uniformly fractional sequence of extensions* $\dot{x}$ given by

$$\dot{x}_j(u,t) = 1/2 \text{ for } u\notin L, t\in K \text{ and } j=1,...,m,$$

is extended to an optimal solution $(\dot{x},\dot{y},\dot{z},\dot{y}^z)$ to the LP-relaxation (4)-(11) as follows: For $(u,v)\in\bar{A}$,

$$\begin{aligned} \dot{y}_j(u,v) &= 1 \text{ if } v\notin L, \\ &= 1/2 \text{ if } v\in L. \end{aligned}$$

If l is even, set $\dot{z}(0,1)=\dot{z}(0,2)=1$ and for $i=1,...,(l-2)/2$,

$$\begin{aligned} &\dot{z}(2i-1,4i-1)=\dot{z}(2i-1,4i)=\dot{z}(2i-1,4i+1)=\dot{z}(2i-1,4i+2)=1/2, \\ &\dot{z}(2i,4i-1)=\dot{z}(2i,4i)=\dot{z}(2i,4i+1)=\dot{z}(2i,4i+2)=1/2. \end{aligned}$$

If l is odd, set $\dot{z}(0,1)=\dot{z}(1,2)=1$ and $\dot{z}(0,3)=\dot{z}(0,4)=\dot{z}(1,3)=\dot{z}(1,4)=1/2$, and for $i=1,...,(l-3)/2$,

$$\begin{aligned} &\dot{z}(2i,4i+1)=\dot{z}(2i,4i+2)=\dot{z}(2i,4i+3)=\dot{z}(2i,4i+4)=1/2, \\ &\dot{z}(2i+1,4i+1)=\dot{z}(2i+1,4i+2)=\dot{z}(2i+1,4i+3)=\dot{z}(2i+1,4i+4)=1/2. \end{aligned}$$

The other $\dot{z}$-variables are set to be zero. Then, $y_j^z=\dot{z}, j=1,...,m$, leads to the optimal value $m|V|$ of the LP-relaxation (4)-(11). We refer to $(\dot{x},\dot{y},\dot{z},\dot{y}^z)$ as the *uniformly fractional solution.* Table 1 describes $\dot{z}$ for $l=6$. Note that the sum of entries in a row is 2 and the sum of entries in a column is 1. In the next section we use randomized rounding on the uniformly fractional solution of the LP-relaxation to obtain good solutions to BPP.

Given a tree T and a sequence of characters $s=(s_j)$, the uniformly fractional sequence of extensions $\dot{x}$ yields the lower bound of the parsimony value of T which was first identified by Steel [19].

$u \setminus v$	1	2	3	4	5	6	7	8	9	10
0	1	1	0	0	0	0	0	0	0	0
1	x	0	1/2	1/2	1/2	1/2	0	0	0	0
2	x	x	1/2	1/2	1/2	1/2	0	0	0	0
3	x	x	x	0	0	0	1/2	1/2	1/2	1/2
4	x	x	x	x	0	0	1/2	1/2	1/2	1/2

Table 1: For $l = 6$, $y_j^z(u, v) = \dot{z}(u, v) = \dot{z}(\text{row}, \text{column})$. (x = no entry)

Proposition 1. *Given a tree T of l leaves and a sequence of m characters $s = (s_j)$, the parsimony value of T is lower bounded by $\left(\frac{3l}{2} - 2\right) m$; i.e.,*

$$pv(T) \geq m(|V| - l/2) = m(2l - 2 - l/2) = m\left(\frac{3l}{2} - 2\right). \tag{12}$$

Assign a common state to all the internal nodes in uniform distribution. More precisely, $X_j(u, t) = X_j(v, t)$ for $u, v \in \bar{V} \setminus L, t \in K, j \in 1, ..., m$, and

$$P(X_j(v, 0) = 1) = P(X_j(v, 1) = 1) = \dot{x}_j(v, 0) = \dot{x}_j(v, 1) = 1/2.$$

(An uppercase letter is used to denote a random variable.) The expected value of the parsimony value is $\dot{y}_j(u, v)$ on each arc (u, v) at each position j. They induce the lower bound $m(l/2 + (l - 2))$ in (12) which converges to 3/4 of the total number $m|V| = m(2l - 2)$ of tree arcs; *i.e.*,

$$pv(T) \geq \frac{3}{4} m|V| \tag{13}$$

for every binary tree T.

3 Fixed randomized rounding approach

In this section we develop a randomized rounding procedure to obtain a good quality solution to BPP that uses the uniformly fractional solution to the LP-relaxation obtained in the previous section. Before we describe the procedure, we define the maximum binary tree problem:

Problem 2 (Maximum Binary Tree Problem (MBTP))**.** *Given a weight function $w : \bar{A} \to R$ on the arcs of the binary tree search space $\bar{D} = (\bar{V}, \bar{A})$, identify the binary tree $T = (\bar{V}, A)$ of the maximum weight*

$$\sum_{(u,v) \in A} w(u, v).$$

Our randomized rounding procedure repeats the following four steps:

1. The first step is to randomly round the x-variables in the uniformly fractional solution. Given that $|K| = 2$, randomized rounding assigns each of the x-variables to one of the two elements in $K = \{0, 1\}$. We refer to this assignment as the *pre-assignment* x^{Pre}.

2. For this assignment of x-variables we construct the implied parsimony value variables $y^{\text{Pre}}(u,v)$ for each arc (u,v). These values are then used to obtain weights $w\left(x^{\text{Pre}}\right)$, where

$$w\left(x^{\text{Pre}}\right)(u,v) = \sum_{j=1}^{m} y_j^{\text{Pre}}(u,v) \text{ for } (u,v) \in \bar{A}. \tag{14}$$

3. These weights are then used to obtain the maximum weight binary tree. Solve the MBTP with objective $w\left(x^{\text{Pre}}\right)$. The optimal solution is the binary tree T^{Pre} at the maximum parsimony value

$$pv\left(T^{\text{Pre}}, x^{\text{Pre}}\right) = \max_{T} pv\left(T, x^{\text{Pre}}\right)$$

which we call the *pre-value*.

4. Over the fixed binary tree $T^{\text{Final}} = T^{\text{Pre}}$, Fitch's algorithm then produces the sequence of extensions x^{Final} at the maximum parsimony value

$$pv\left(T^{\text{Final}}, x^{\text{Final}}\right) = \max_{x|_L = s} pv\left(T^{\text{Final}}, x\right),$$

which we call the *final value*.

We refer to the procedure as *fixed randomized rounding* because it uses the same uniformly fractional solution for each repetition of randomized rounding. Observe that MBTP can be formulated as follows:

$$\begin{aligned} \max \quad & \sum_{(u,v)\in \bar{A}} w(u,v)z(u,v) \\ \text{s.t.} \quad & z(\delta^+(v)) = 2 \text{ for } v \in \bar{V} \setminus L \\ & z(\delta^-(v)) = 1 \text{ for } v \in V = \bar{V} \setminus \{0\} \\ & z \geq 0. \end{aligned} \tag{15}$$

In fact, the formulation is *totally unimodular*; *i.e.*, all the square sub-matrices have determinant 0, 1 or -1. The optimal solution to its LP-relaxation is an integer solution, and the MBTP can be solved in polynomial time. For more details of total unimodularity, the readers may refer to Cook et al. [4].

Theorem 2. *The ILP formulation (15) of the MBTP is totally unimodular.*

Proof. Let the system of linear equations in (15) be simply written as $Bz = b$, and let B' denote a square sub-matrix of B. By mathematical induction on the size of B', we show that every square sub-matrix B' of B has determinant 0, 1 or -1. A 1×1 square sub-matrix trivially has determinant 0, 1 or -1. We now assume that any square sub-matrix of size $\leq q-1$ has determinant 0, 1 or -1 for $q \geq 2$. We only need to show that a square sub-matrix B' of size q has determinant 0, 1 or -1.

A column of B corresponds to an arc $(u,v) \in \bar{A}$ having two 1's, one at a row corresponding to $\delta^+(u)$ and the other corresponding to $\delta^-(v)$. If B' has a column of at most one non-zero entry (=1), the $(q-1)\times(q-1)$ square sub-matrix B'' of B' given by deleting the column and

		B&B (= 7200 sec.)		Fixed Random. Round. (10000 trials no.)						
nTax	nChar	ObjVal	**sec.**	Break	**sec.**	no.	Max	sec.	no.	total sec.
10	100	1517	**976**	1519	**13**	84	1520	218	1404	1550
10	200	3001	**4562**	3002	**34**	111	3008	1314	4286	3066
10	300	4490	**5721**	4491	**110**	240	4499	3306	7230	4572
10	400	5988	**4195**	5988	**29**	48	6015	5373	8839	6078
10	500	7477	**5931**	7478	**2**	2	7499	1323	1742	7596
10	600	8953	**6080**	8968	**15**	16	8984	2326	2552	9116
10	700	10341	**7200**	10421	**1**	1	10481	4995	4509	11078
10	800	11454	**6435**	11942	**1**	1	12004	4060	3305	12283
10	900	12471	**6979**	13390	**1**	1	13452	11598	8476	13683
10	1000	13810	**6592**	14884	**2**	1	14963	4958	3255	15232
20	100	3154	**6991**	3163	**1**	1	3193	2219	4311	5147
20	200	5701	**6400**	6318	**1**	1	6361	6737	6580	10239
20	300	7025	**7200**	9478	**2**	1	9553	11291	7407	15243
20	400	9366	**7200**	12618	**2**	1	12700	1658	814	20364
20	500	11663	**7200**	15775	**3**	1	15863	14995	5916	25347
20	600	16898	**7200**	18991	**3**	1	19051	28385	9341	30388
20	700	16363	**7200**	22063	**4**	1	22171	6485	1825	35533
20	800	18963	**7200**	25249	**4**	1	25377	13896	3411	40738
20	900	25376	**7200**	28343	**5**	1	28493	23376	5107	45772
20	1000	28092	**7200**	31604	**5**	1	31695	30773	6009	51212

Table 2: Comparison of branch-and-bound and 10,000 trials of fixed randomized rounding

the row of the non-zero entry has determinant 0, 1 or -1 by induction hypothesis and therefore $\det(B') \in \{0, \det(B''), -\det(B'')\}$ is 0, 1 or -1.

We assume that every column of B' has two 1's. Then, the determinant of B' is shown to be zero, because adding the rows of $\delta^+(u)$ to the first row and subtracting the rows of $\delta^-(v)$ from the first row make the first row vanish, completing the proof. □

Since the MBTP can be solved very quickly, our randomized rounding method can be solved relatively fast.

3.1 The Power of Fixed Randomized Rounding

In this section we present computational results comparing our fixed randomized rounding approach with branch-and-bound using GUROBI. The computational experiment verifies that fixed randomized rounding outperforms the incumbent solutions produced by the branch-and-bound (B&B) procedure of GUROBI. Table 2 compares results of 10,000 trials of fixed randomized rounding with the incumbent solution (the best integer solution) to the ILP formulation (4)-(11) on the generated instances within 2 hours of the branch-and-bound (B&B) procedure by GUROBI. We use Python 3.6 as the programming language and Gurobi 8.0 as the ILP solver. We also use them to solve an MBTP in the computational experiments throughout this paper. Our computational experiments are carried out on a machine with 64GB of RAM running on a 3.6GHz processor.

		$pv\left(T^{\text{Pre}}, x^{\text{Pre}}\right)$		$pv\left(T^{\text{Final}}, x^{\text{Final}}\right)$				
nTax	nChar	**STD**	Mean	**STD**	Mean	c.v.	Max	dev./Mean
10	100	**13.4**	1039.0	**6.4**	1499.4	0.004268	1520	0.013739
10	200	**18.9**	2026.5	**7.9**	2981.8	0.002649	3008	0.008787
10	300	**23.1**	3004.9	**11.7**	4460.8	0.002623	4499	0.008563
10	400	**26.7**	3978.4	**13.5**	5964.5	0.002263	6015	0.008467
10	500	**30.1**	4950.0	**15.2**	7449.2	0.002040	7499	0.006685
10	600	**32.8**	5919.0	**17.7**	8929.6	0.001982	8984	0.006092
10	700	**36.1**	6886.9	**17.1**	10423.7	0.001640	10481	0.005497
10	800	**38.6**	7852.4	**15.3**	11948.1	0.001281	12004	0.004679
10	900	**40.6**	8818.5	**18.7**	13385.2	0.001397	13452	0.004991
10	1000	**42.8**	9783.2	**21.8**	14886.8	0.001464	14963	0.005119
20	100	**16.8**	2213.9	**10.6**	3155.8	0.003359	3193	0.011788
20	200	**24.2**	4273.4	**14.5**	6301.5	0.002301	6361	0.009442
20	300	**30.2**	6305.9	**17.5**	9485.0	0.001845	9553	0.007169
20	400	**34.9**	8327.5	**18.4**	12630.4	0.001457	12700	0.005511
20	500	**39.1**	10338.9	**20.5**	15788.3	0.001298	15863	0.004731
20	600	**42.5**	12345.0	**25.7**	18954.2	0.001356	19051	0.005107
20	700	**45.9**	14347.6	**25.1**	22084.5	0.001137	22171	0.003917
20	800	**49.3**	16344.6	**28.2**	25257.5	0.001117	25377	0.004731
20	900	**52.5**	18340.5	**28.4**	28385.7	0.001001	28493	0.003780
20	1000	**55.4**	20332.6	**29.8**	31582.4	0.000944	31695	0.003565

Table 3: Statistics of fixed randomized rounding in Table 2

In Table 2, nTax is the number of leaf nodes $|L|$ and nChar is the number of characters in our problem. ObjVal is the best incumbent objective function value obtained using branch-and-bound within 7,200 seconds. Break is the first value from fixed randomized rounding that is better than the best branch-and-bound value under ObjVal. From Table 2 observe that fixed randomized rounding very quickly obtains a better solution than branch-and-bound is able to obtain in 7,200 seconds. Max represents the best solution obtained after 10,000 trials of randomized rounding. It is interesting to observe that while our randomized rounding approach quickly does better than branch-and-bound, many replications of fixed randomized rounding do not significantly improve the solution value.

The reason for the failure of fixed randomized rounding to significantly improve the solution is evident in our computational results in Table 3. For each problem instance, Mean represents the average solution value and STD the standard deviation of solution values from 10,000 trials using randomized rounding (for both the pre as well as the final values). Observe the very low coefficient of variation c.v., which seems to get smaller as problem size grows.

Max represents the maximum value obtained after 10,000 randomized rounding trials and dev./Mean is given by

$$\text{dev./Mean} = \frac{\text{Max} - \text{Mean}}{\text{Mean}}.$$

The low values of dev./Mean indicate that running 10,000 randomized rounding trials does not significantly improve the solution obtained. This is to be expected given the low c.v. obtained from our trials. In the next section we identify a potential reason for the concentration of solutions obtained as a result of fixed randomized rounding. Then in Section 4 we come up with an adaptive randomized rounding procedure that improves on the results observed in Table 2.

3.2 Concentration analysis on the pre-value from fixed randomized rounding

In this section we show that the pre-values resulting from randomized rounding of the uniformly fractional solution have a small standard deviation $O(\sqrt{|V|m})$. Since the same final value may result from multiple pre-values, it is intuitive that

$$\mathrm{STD}\left[pv\left(T^{\mathrm{Final}}, x^{\mathrm{Final}}\right)\right] < \mathrm{STD}\left[pv\left(T^{\mathrm{Pre}}, x^{\mathrm{Pre}}\right)\right] \in O(\sqrt{|V|m}).$$

The intuition is verified by the experiment in Table 3. Recall Proposition 1 that a lower bound of the final value $pv\left(T^{\mathrm{Final}}, x^{\mathrm{Final}}\right)$ is

$$\approx \frac{3}{4}|V|m.$$

The coefficient of variation of $pv\left(T^{\mathrm{Final}}, x^{\mathrm{Final}}\right)$ is

$$\frac{\mathrm{STD}}{\mathrm{Mean}} \in \frac{O(\sqrt{|V|m})}{\Omega(|V|m)} = O\left(\frac{1}{\sqrt{|V|m}}\right),$$

and converges to zero as the size of problem instance grows.

The random variable X^{Pre} and the observation x^{Pre} are simply denoted by X and x without superscripts. (The superscripts of X^{Final} or x^{Final} will not be omitted.) Let f and F denote the number and the set of the pairs of node u and position j where x-values are fractional; *i.e.*,

$$f = |F| = \left|\left\{(j,u) : 0 < x_j^{LP}(u,t) < 1 \text{ for some } t \in K\right\}\right|. \tag{16}$$

We simply denote the random variables $X_j(u)$ with a fractional probability by $X_i, i = 1, ..., f$, in a linear order, and those with 0/1-probability by $X_{i+1}, ..., X_N$, where $N = m|V \setminus L| = m(l-2)$. If $x^{LP} = \dot{x}$, then $f = N = m(l-2)$.

Let $pv\,(X)$ be the maximum value of the MBTP with the parsimony value function $w = w\,(X)$; *i.e.*,

$$pv\,(X) = \max_T pv\,(T, X)\,.$$

In this section, we prove large deviation bounds of the form

$$P\left(pv(X) - E[pv(X)] \geq 3\lambda\sqrt{f}\right) \leq e^{-2\lambda^2}.$$

We also induce a bound for the standard deviation of the pre-value which provides a range of the pre-value (and so the final value) in this randomized rounding.

Lemma 3 (McDiarmid [15])**.** *Let $X = (X_1, ..., X_f)$ be a family of independent random variables with X_i taking values in a set A_i for each i. Suppose that the real-valued function g defined on $\prod_{i=1}^f A_i$ satisfies the following Lipschitz condition,*

$$\left|g(x) - g(x')\right| \leq c_i \tag{17}$$

whenever the vectors x and x' differ only in the i-th co-ordinate. Then for any $t \geq 0$,

$$P\left(g(X) - E[g(X)] \geq t\right) \leq e^{-2t^2/\sum_{i=1}^{f} c_i^2}, \tag{18}$$

$$P\left(g(X) - E[g(X)] \leq -t\right) \leq e^{-2t^2/\sum_{i=1}^{f} c_i^2}. \tag{19}$$

We induce large deviation bounds from (18) with $g = pv$, $A_j = K$, $t = 3\lambda\sqrt{f}$ and $c_i = 3$ for all $i = 1, ..., f$ generalizing those introduced by Steel, Goldstein and Waterman [20].

Theorem 4 (Tail Inequality)**.** *Let f denote the number of tuples of nodes and positions where x-variables are fractional in the optimal solution to the LP-relaxation. Then,*

$$P\left(pv(X) - E[pv(X)] \geq 3\lambda\sqrt{f}\right) \leq e^{-2\lambda^2}, \tag{20}$$

$$P\left(pv(X) - E[pv(X)] \leq -3\lambda\sqrt{f}\right) \leq e^{-2\lambda^2}, \tag{21}$$

and for each $p > 0$

$$E\left[\left|\frac{pv(X) - E[pv(X)]}{3\sqrt{f}}\right|^p\right] \leq 2p\int_0^{\infty} \lambda^{p-1} e^{-2\lambda^2} d\lambda. \tag{22}$$

It implies that

$$\frac{V[pv(X)]}{9f} \leq 1. \tag{23}$$

Proof. First we verify that the parsimony value pv satisfies the Lipschitz condition

$$|pv(X_1, ..., X_{i-1}, X_i, X_{i+1}, ..., X_f) - pv(X_1, ..., X_{i-1}, X_i', X_{i+1}, ..., X_f)| \leq 3$$

for all $X_i' \in K$. Suppose that $pv(X_1, ..., X_{i-1}, X_i, X_{i+1}, ..., X_f) \geq pv(X_1, ..., X_{i-1}, X_i', X_{i+1}, ..., X_f)$ over tree topologies T and T', respectively. Over T, changing X_i to X_i' can decrease the anti-score by at most 3 over the edges for the parent and two children. By symmetry, the other way around is also true.

Using the independence of fractional variables $X_1, ..., X_f$, (20) and (21) follow by applying the tail inequalities (18) and (19) introduced by McDiarmid [15]. Since (20) and (21)imply

$$P\left(|pv(X) - E[pv(X)]| \geq 3\lambda\sqrt{f}\right) \leq 2e^{-2\lambda^2},$$

(22) now follows from

$$E\left[W^p\right] = p\int_0^{\infty} \lambda^{p-1} P(W > \lambda) d\lambda$$

for any $W \geq 0$. □

Behaving nice like the sum of independent random variables, the variance of the pre-value is upper bounded by $O(f)$ (not $O(f^2)$). Since the standard deviation of $pv\left(T^{\text{Final}}, x^{\text{Final}}\right)$ is intuitively smaller than that of $pv\left(T^{\text{Pre}}, x^{\text{Pre}}\right)$, it is most likely that

$$\text{STD}\left[pv\left(T^{\text{Final}}, x^{\text{Final}}\right)\right] \in O\left(\sqrt{|V|m}\right)$$

and the coefficient of variation of $pv\left(T^{\text{Final}}, x^{\text{Final}}\right)$ converges to 0. Theorem 4 provides a theoretical justification for why fixed randomized rounding may find a good solution early but is unable to significantly improve the integer solution obtained over many randomized trials.

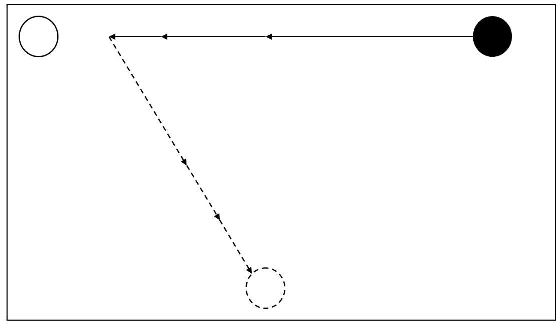

Figure 1: 2-step procedure of adaptive randomized rounding

4 Adaptive randomized rounding method changing the distribution

In this section, we present an adaptive randomized rounding procedure that addresses the concentration of values when randomized rounding is performed on the uniformly fractional solution. Our adaptive procedure changes the underlying fractional solution used for randomized rounding whenever a better integer solution is identified.

4.1 2-step procedure for adaptive randomized rounding

At each trial $i \geq 1$, the adaptive randomized rounding procedure performs randomized rounding based on $x^{LP(i)}$ after moving the underlying fractional solution from the current fractional solution $x^{LP(i-1)}$ toward x^* of the best known integer solution (z^*, x^*):

$$x^{LP(i)} = (1 - \alpha(i))x^{LP(i-1)} + \alpha(i)x^*,$$

assuming that a better integer solution $x^{\text{Final}(i)}$ is near the line segment $\left[x^{LP(i-1)}, x^*\right]$. Since the LP-relaxation of the ILP formulation of the big parsimony problem is convex, the line segment is contained in the LP-relaxation. Converging to x^*, the randomized rounding based on $x^{LP(i)}$ produces the pre-value $pv\left(T^{\text{Pre}(i)}, x^{\text{Pre}(i)}\right)$ and the final value $pv\left(T^{\text{Final}(i)}, x^{\text{Final}(i)}\right)$ together approach to $pv\left(T^*, x^*\right)$ and the MBTP pushes up the final value $pv\left(T^{\text{Final}(i)}, x^{\text{Final}(i)}\right)$ as well as the pre-value $pv\left(T^{\text{Pre}(i)}, x^{\text{Pre}(i)}\right)$. We now describe the procedure in detail.

Using exponential smoothing, we develop a 2-step procedure to change the underlying fractional solution used for randomized rounding:

Step 0: Initialization. The initial fractional solution is the uniformly fractional solution $x^{LP(0)} = \dot{x}$ and the initial smoothing constant is $\alpha(1) = 1/2$. At the initial trial $i = 0$, an integer solution x^* is obtained after the randomized rounding based on $\dot{x}$. Set $i = 1$ and go to Step 1.

Step 1: Adjusting the smoothing constant. The first step moves the current fractional solution between $\dot{x}$ and x^* adjusting the smoothing constant α.

- Criterion to go to Step 2: If the observed parsimony value $pv\left(T^{\text{Final}(i)}, x^{\text{Final}(i)}\right)$ is better than the best known parsimony value $pv\left(T^*, x^*\right)$, the procedure updates the best integer solution

$x^* = x^{\text{Final}(i)}$ and the best parsimony value $pv\left(T^*, x^*\right) = pv\left(T^{\text{Final}(i)}, x^{\text{Final}(i)}\right)$. The speed of the move seems to be appropriate. Fix the current smoothing constant $\alpha = \alpha(i)$. Set $i \leftarrow i+1$ and go to Step 2. The solid dot represents $\dot{x}$, the hollow dot represents the current best known integer solution, and the dotted hollow dot represents the new best integer solution.

- Major decrease of α: If $pv\left(T^{\text{Final}(i)}, x^{\text{Final}(i)}\right)$ is equal to $pv\left(T^*, x^*\right)$, $x^{LP(i)}$ seems to arrive close to x^* too fast. Slow down the speed of the move. The smoothing constant α makes a major decrease to $\alpha(i+1) = \alpha(i)/2$. The fractional solution comes back to $x^{LP(i)} = \dot{x}$. Set $i \leftarrow i+1$ and Step 1 repeats.

- Minor increase of α: If $pv\left(T^{\text{Final}(i)}, x^{\text{Final}(i)}\right)$ is strictly worse than $pv\left(T^*, x^*\right)$, $x^{LP(i)}$ seems to move too slow. Speed up the move a little bit faster. The smoothing constant α makes a minor increase to $\alpha(i+1) = 2^{\lfloor \log_2 \alpha(i) \rfloor} + \alpha(i)/2$ such as $1/2 + 1/4$ to $1/2 + 1/4 + 1/8$, and Step 1 goes on (without coming back to $x^{LP(i)} = \dot{x}$).

Step 2: Fixed smoothing constant. The second step moves the fractional solution from the current fractional solution $x^{LP(i-1)}$ toward x^* of the best known integer solution (z^*, x^*) without changing the smoothing constant α adjusted in Step 1.

- Criterion to go back to Step 1: If $pv\left(T^{\text{Final}(i)}, x^{\text{Final}(i)}\right)$ hit $pv\left(T^*, x^*\right)$ for a run of say 10 times of trials i, the current probability distribution is too close to $pv\left(T^*, x^*\right)$ and the fractional solution comes back to $x^{LP(i)} = \dot{x}$ initializing $\alpha(i+1) = 1/2$. Set $i \leftarrow i+1$ and go back to Step 1.

- If a better integer solution is observed, the procedure updates $pv\left(T^*, x^*\right)$ with the new best one. Set $i \leftarrow i+1$ and Step 2 repeats.

- If the observed solution is strictly worse than the best known integer solution, Step 2 just moves on setting $i \leftarrow i+1$.

In Table 4, we compare the 10,000 trials of fixed randomized rounding of Table 2 with 10,000 trials of the adaptive randomized rounding. Under Random. Round., Max represents the maximum value obtained using fixed randomized rounding. Under Adaptive Random. Round., Break represents the first value obtained that is larger than the best solution obtained using fixed randomized rounding. Observe that for each problem in Table 4, adaptive randomized rounding was able to obtain a better solution in relatively few iterations. Max represents the best solution obtained using adaptive randomized rounding.

Table 5 contains results that demonstrate that adaptive randomized rounding produces results that are unlikely to have been produced from fixed randomized rounding using the uniformly fractional solution. Norm. gives the R-square for the distribution of final values with a normal distribution. The high R-square indicates that the distribution of final values is normal when randomly rounding the uniformly fractional solution.

dev./σ represents the number of standard deviations between the best value from 10,000 randomized rounding trials and the mean (the Z-value). Under fixed randomized rounding from the uniformly fractional solution, observe that this value is generally less than 3.8. At a z-value of 3.8, we would expect 1 trial in 10,000 at this value or higher. In other words, fixed randomized rounding using the uniformly fractional solution behaves as expected using the normal distribution.

		Fixed Random. Round.			Adaptive Random. Round.				
nTax	nChar	Max	**no.**	total sec.	Break	**no.**	Max	no.	total sec.
10	100	1520	**1404**	1550	1524	**765**	1524	765	1505
10	200	3008	**4286**	3066	3010	**200**	3010	200	3182
10	300	4499	**7230**	4572	4502	**179**	4517	9677	4736
10	400	6015	**8839**	6078	6019	**2475**	6019	2475	6320
10	500	7499	**1742**	7596	7501	**85**	7510	93	7789
10	600	8984	**2552**	9116	8989	**1057**	8993	1263	9469
10	700	10481	**4509**	11078	10487	**112**	10489	1960	10976
10	800	12004	**3305**	12283	12007	**1032**	12007	1032	12681
10	900	13452	**8476**	13683	13455	**198**	13457	200	14010
10	1000	14963	**3255**	15232	14965	**259**	14974	1514	15635
20	100	3193	**4311**	5147	3197	**86**	3219	3833	5338
20	200	6361	**6580**	10239	6365	**470**	6375	2115	10430
20	300	9553	**7407**	15243	9558	**387**	9612	9182	16101
20	400	12700	**814**	20364	12702	**125**	12742	2550	20807
20	500	15863	**5916**	25347	15866	**134**	15896	636	26246
20	600	19051	**9341**	30388	19052	**482**	19135	9237	31548
20	700	22171	**1825**	35533	22195	**208**	22229	3512	36867
20	800	25377	**3411**	40738	25379	**4054**	25443	6662	42180
20	900	28493	**5107**	45772	28495	**260**	28502	8621	47294
20	1000	31695	**6009**	51212	31699	**497**	31747	7137	53241

Table 4: Comparison of Fixed Randomized Rounding (the 10,000 trials in Table 2) and Adaptive Randomized Rounding (10,000 trials)

		Fixed Random.			Adaptive Random.	
nTax	nChar	Norm.	Max	**dev./σ**	Max	**dev./σ**
10	100	0.95	1520	**3.21**	1524	**3.84**
10	200	0.96	3008	**3.30**	3010	**3.56**
10	300	0.96	4499	**3.28**	4517	**4.80**
10	400	0.96	6015	**3.74**	6019	**4.03**
10	500	0.96	7499	**3.29**	7510	**4.00**
10	600	0.96	8984	**3.07**	8993	**3.58**
10	700	0.96	10481	**3.34**	10489	**3.81**
10	800	0.96	12004	**3.65**	12007	**3.84**
10	900	0.95	13452	**3.57**	13457	**3.83**
10	1000	0.95	14963	**3.49**	14974	**4.00**
20	100	0.96	3193	**3.52**	3219	**5.96**
20	200	0.95	6361	**4.11**	6375	**5.06**
20	300	0.96	9553	**3.88**	9612	**7.25**
20	400	0.96	12700	**3.79**	12742	**6.08**
20	500	0.96	15863	**3.65**	15896	**5.25**
20	600	0.95	19051	**3.77**	19135	**7.03**
20	700	0.96	22171	**3.44**	22229	**5.75**
20	800	0.96	25377	**4.24**	25443	**6.57**
20	900	0.96	28493	**3.78**	28502	**4.09**
20	1000	0.96	31695	**3.77**	31747	**5.52**

Table 5: Statistics for the experiments in Table 4

In contrast, all but two values of dev./σ when using adaptive randomized rounding are larger than 3.8 with most being larger than 4 and many larger than 5. At a z-value of 5, it is extremely unlikely that the solution using adaptive randomized rounding could have been obtained by chance. The results in Table 5 clearly indicate that adaptive randomized rounding is superior and provides solutions that are unlikely to be obtained by randomized rounding using the uniformly fractional solution.

In the next section we confirm these findings on real biological data.

4.2 Computational experiments on biological data

Tables 6 and 7 show the results on real biological data from computational experiments of 10,000 trials of fixed randomized rounding based on the uniformly fractional solution $\dot{x}$ and 10,000 trials of adaptive randomized rounding in the 2 step procedure. The real biological data were used by Sridhar, Lam, Blelloch, Ravi and Schwartz [18] to construct non-binary phylogenetic trees. We failed to solve the instance of the largest number (nTax = 395) of the leaf nodes because of out of memory (OOM). In Table 6, observe that adaptive randomized rounding obtains a better solution very quickly in all instances but one.

The results in Table 7 confirm the finding in Table 5 that adaptive randomized rounding is superior to fixed randomized rounding using the uniformly fractional solution.

			Fixed Random.		Adaptive Random.	
Data Set	nTax	nChar	Max	**no.**	Break	**no.**
human mtDNA [17]	13	390	9310	**1106**	9311	**1844**
chimp chrY [21]	15	98	2642	**1059**	2643	**40**
bacterial [16]	17	1510	48193	**1406**	48199	**8**
chimp mtDNA [21]	24	1041	47806	**2444**	47807	**16**
human mtDNA [14]	33	405	25853	**3847**	25855	**7**
human mtDNA [22]	40	52	3953	**8154**	3956	**6**
human chrY [11]	150	49	14433	**6136**	14503	**4**
human mtDNA [9]	395	830	OOM	OOM	OOM	OOM

Table 6: Fixed Randomized Rounding (10,000 trials) vs. Break of Adaptive Randomized Rounding (10,000 trials) on a Selection of Nonrecombining Data Sets [18]

			Fixed Random.		Adaptive Random.			**ratioP**
Data Set	**nTax**	nChar	Max	dev./σ	Max	no.	dev./σ	**RR/AR**
human mtDNA [17]	**13**	390	9310	3.37	9311	1844	3.77	**4.69**
chimp chrY [21]	**15**	98	2642	3.73	2645	48	4.11	**4.95**
bacterial [16]	**17**	1510	48193	3.48	48224	91	4.28	**27.27**
chimp mtDNA [21]	**24**	1041	47806	2.87	47822	614	4.34	**291.78**
human mtDNA [14]	**33**	405	25853	3.66	25877	9727	7.34	$\mathbf{1.18 \times 10^9}$
human mtDNA [22]	**40**	52	3953	3.50	3982	567	8.41	∞
human chrY [11]	**150**	49	14433	3.96	14585	1488	13.68	∞
human mtDNA [9]	**395**	830	OOM	OOM	OOM	OOM	OOM	OOM

Table 7: Ratio (ratioP) of P-values of Max in Fixed Randomized Rounding (10,000 trials) vs. Adaptive Randomized Rounding (10,000 trials) on a Selection of Nonrecombining Data Sets [18]

5 Conclusion

We show that randomized rounding can be an effective method to solve the big parsimony problem. A fixed randomized rounding procedure using the uniformly fractional solution is shown to provide much better solutions than using the integer linear programming software GUROBI. We then develop an adaptive randomized rounding procedure that is seen to be superior (in a statistically significant manner) when compared to fixed randomized rounding using the uniformly fractional solution. Our method can be a reasonable approach when looking for good solutions to the big parsimony problem.

Acknowledgement. The authors would like to thank R. Ravi and Russell Schwartz for providing biological data [18].

References

[1] N. Alon, B. Chor, F. Pardi and A. Rapoport, Approximate maximum parsimony and ancestral maximum likelyhood, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **7**:1 (2010) 183–187.

[2] D. Catanzaroa,R. Aringhieric, M.D. Summad and R. Pesenti, A branch-price-and-cut algorithm for the minimum evolution problem, *European Journal of Operational Research* **244** (2015) 753–765.

[3] S. Chopra and M.R. Rao, The partition problem, *Mathematical Programming* **59** (1993) 87–115.

[4] W. Cook, W.H. Cunningham, W.R. Pulleyblank and A. Schrijver, Combinatorial Optimization, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley and Sons, Inc., NY, 1998.

[5] M. Fischer, M. Galla, L. Herbst and M. Steel, The most parsimonious tree for random data, *Molecular Phylogenetics and Evolution* **80** (2014) 165168.

[6] W. Fitch, Toward defining the course of evolution: Minimum change for a specific tree topology, *Systematic Zoology* **20**: 4 (1971) 406–416.

[7] L.R. Foulds and R.L. Graham, The Steiner problem in phylogeny is NP-complete, *Advances in Applied Mathematics* **3** (1982) 43–49.

[8] D. Haws, T. Hodge and R. Yoshida, Phylogenetic tree reconstruction: geometric approaches. In: Robeva, R., Hodge, T. (eds.) Mathematical Concepts and Methods in Modern Biology: Using Modern Discrete Models, pp. 307342. Elsevier, Dublin (2013)

[9] A. Helgason, G. Palsson, H.S. Pedesen, E. Angulalik, E.D. Gunnarsdottir, B. Yvgvadottir and K. Stefansson, mtDNA Variation in Inuit Populations of Greenland and Canada: Migration History and Population Structure, *Am. J. Physical Anthropology* **130** (2006) 123-134.

[10] M.D. Hendy and D. Penny, A framework for the quantitative study of evolutionary trees, *Systematic Zoology* **38**: 4 (1989) 297–309.

[11] International HapMap Consortium, The International HapMap Project, *Nature* **426**, www.hapmap.org (2005) 789–796.

[12] L. Kannan and W. Wheeler, Exactly computing the parsimony scores on phylogenetic networks using dynamic programming, *Jounal of Computational Biology* **21**: 4 (2017) 303–319.

[13] S. Kelk and M. Fischer, On the complexity of computing MP distance between binary phylogenetic trees, *Annals of Combinatorics* **21**: 4 (2017) 573–604.

[14] C.J. Lewis, R. Tito, B. Lizarraga and A. Stone, Land, Laguage and Loci: mtDNA in Native Americans and the Genetic History of Peru, *Am. J. Physical Anthropology* **127** (2005) 351-360.

[15] C. McDiarmid, On the method of bounded differences. In *Surveys in Combinatorics*, Proceedings, Norwic, 1989, London Math. Soc. Lecture Note Ser. 141, Cambridge Univ. Press, Cambridge, 148-188.

[16] M. Merimaa, M. Liivak, E. Heinaru, J. Truu and A. Heinaru, Functional Co-Adaption of Phenol Hydroxylase and Catechol 2,3-Dioxygenase Genes in Bacteria Possessing Different Phenol and p-Cresol Degradation Pathways, *Proc. Eighth Symp. Bacterial Genetics and Ecology (BAGECO '05)* **31** (2005) 185-212.

[17] S. Sharma, A. Saha, E. Rai, A. Bhat and R. Bamezai, Human mtDNA Hypervariable Regions, HVR I and II, Hint at Deep Common aterial Founder and Subsequent Maternal Gene Flow in Indian Population Groups, *Am. J. Human Genetics* **50** (2005) 497-506.

[18] S. Sridhar, F. Lam, G.E. Blelloch, R. Ravi, and R. Schwartz, Mixed integer linear programming for maximum-parsimony phylogeny inference, *IEEE/ACM Transaction on Computational Biology and Bioinformatics* **5** (2008) 323-331.

[19] M.Steel, Distributions on bicoloured binary trees arising from the principle of parsimony, *Discrete Applied Mathematics* **41** (1993) 245-261.

[20] M.Steel, L. Goldstein and M.S. Waterman, A central limit theorem for the parsimony length of trees, *Advances in Applied Probability* **28** (1996) 1051-1071.

[21] A.C. Stone, R.C. Griffiths, S.L. Zegura and M.F. Hammer, High Levels of Y-Chromosome Nucleotide Diversity in the Genus Pan, *Proc. Nat'l Academy of Sciences* **99** (2002) 43-48.

[22] T. Wirth, X. Wang, B. Linz, R.P. Novick, J.K. Lum, M. Blaser, G. Morelli, D. Falush and M. Achtman, Distinguishing Human Ethnic Groups by Means of Sequences from Helicobacter Pylori: Lessons from Ladakh, *Proc. Nat'l Academy of Sciences* **101** (2004) 4746-4751.