Compact Mixed Integer Linear Programming models to the Minimum Weighted Tree Reconstruction Problem

Bernard Fortz, Olga Oliveira, Cristina Requejo

▶ **To cite this version:**

Bernard Fortz, Olga Oliveira, Cristina Requejo. Compact Mixed Integer Linear Programming models to the Minimum Weighted Tree Reconstruction Problem. European Journal of Operational Research, 2017, 256 (1), pp.242 - 251. 10.1016/j.ejor.2016.06.014 . hal-01410554

HAL Id: hal-01410554
https://inria.hal.science/hal-01410554

Submitted on 4 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Compact Mixed Integer Linear Programming models to the Minimum Weighted Tree Reconstruction Problem *

Bernard Fortz[1], Olga Oliveira[2], and Cristina Requejo[3]

[1] *Department of Computer Science, Université Libre de Bruxelles, B-1050 Brussels, Belgium and INOCS, INRIA Lille Nord-Europe, France. (bernard.fortz@ulb.ac.be)*

[2] *Department of Mathematics, University of Aveiro, 3810-193 Aveiro, Portugal. (olga.oliveira@ua.pt)*

[3] *Department of Mathematics, University of Aveiro, 3810-193 Aveiro, Portugal. (crequejo@ua.pt)*

April 4, 2017

Abstract

The Minimum Weighted Tree Reconstruction (MWTR) problem consists of finding a minimum length weighted tree connecting a set of terminal nodes in such a way that the length of the path between each pair of terminal nodes is greater than or equal to a given distance between the considered pair of terminal nodes. This problem has applications in several areas, namely, the inference of phylogenetic trees, the modeling of traffic networks and the analysis of internet infrastructures. In this paper, we investigate the MWTR problem and we present two compact mixed-integer linear programming models to solve the problem. Computational results using two different sets of instances, one from the phylogenetic area and another from the telecommunications area, show that the best of the two models is able to solve instances of the problem having up to 15 terminal nodes.

***Keywords*:** mixed integer linear programming, distance matrix, tree realization, topology discovery, routing topology inference, minimum evolution problem, balanced minimum evolution problem

1 Introduction

We address the problem of reconstructing a weighted tree $T = (V, E)$ by knowing only pairwise distances d_{ij}, for all $i, j \in V_t$, between a set of terminal nodes $V_t \subset V$. Given a set V_t of n terminal nodes and an $n \times n$ distance matrix D, find a tree $T = (V, E)$ spanning $V = V_t \cup V_a$, where V_a is a subset of $n-2$ additional nodes, and associate edge weights $w_e,\ e \in E$, such that the weight of the unique path P_{ij} between any two terminal nodes i and $j \in V_t$ is at least d_{ij}, i.e. $\sum_{e \in P_{ij}} w_e \geq d_{ij}$, and such that the total sum of the edge weights $\sum_{\{i,j\} \in E} w_{ij}$ is minimized. This problem is the Minimum Weighted Tree Reconstruction (MWTR) Problem.

The MWTR problem is a specific version of the distance realization problem (which is a graph realization problem), namely a tree realization problem for a distance matrix. Several authors studied the tree realization problem for a distance matrix and this class of combinatorial problems

*Published in *European Journal of Operational research*, Volume 256, Issue 1, 1 January 2017, Pages 242–251.

was proved to be NP-complete [10, 15, 25]. Fiorini and Joret [28] and Catanzaro et al. [6] discuss the NP-hardness of two related optimization problems, the minimum evolution problem (MEP) and the balanced minimum evolution problem (BMEP). These two problems are very well known distance realization problems from the computational biology area.

The MWTR arises in a number of areas, such as telecommunications and computational biology. All the applications we consider have in common the notion of a *graph realization* of a distance matrix and more specific of a tree realization of an additive distance matrix. A distance matrix has a tree realization, if it is an additive matrix and can be embedded into a tree. Abdi [1] mentions several applications from these "*additive trees that are used to represent objects as leaves such that the distance on the tree between two leaves reflects the similarity between the objects*". Probably the most well know application is, in computational biology, the reconstruction of phylogenetic trees [5, 26, 37]. There are also applications in psychology [13, 14, 30, 50, 55] to represent cognitive processes or proximity and similarity relations. There are applications in information security for the detection and recognition of documents duplications [16, 30]. And there are applications in telecommunications, namely in network tomography to discover the logical underlying network [10, 12, 18, 47] and the routing topology of a network [2, 10, 18, 34]. In this work we use data from the two applications, the phylogenetics application and the network application, that will be described.

An application of the MWTR includes in computational biology the reconstruction of phylogenetic trees. A phylogenetic tree represents the evolutionary relationships of a set of species, where terminal nodes represent the observed species, additional nodes represent common ancestors, edges represent the evolutionary relationships between pairs of nodes and edge weights represent the quantification of this evolutionary relationship [26, 43, 44]. It is worth noting that phylogenetic trees allow understanding of the evolutionary history of species and can assist in the development of vaccines [29] and the study of biodiversity [40]. In computational biology inferring a phylogenetic tree is one of the steps of the phylogenetic reconstruction. As part of such inference is the determination of the tree topology and the determination of the branch lengths, which requires further analysis methods. Such analysis methods includes parsimony methods, distance methods, and probabilistic methods arising from the maximum likelihood approach. These methods make specific assumptions about evolution. Distance methods exploit the existence of a measure of dissimilarity (also called distance) among pairs of species and aim at determining the tree topology together with branch lengths. Examples of distance methods assumptions are the minimum evolution principle and the balanced minimum evolution principle. In the minimum evolution principle the sum of all branch lengths is minimized [44]. This is the principle used in MEP. The balanced minimum evolution principle is a new version of the minimum evolution principle. This new version was first introduced by Pauplin [49] to simplify tree length computations. Within this principle sibling subtrees have equal weight, as opposed to the standard version where all terminal nodes have the same weight and thus the weight of a subtree is equal to the number of its terminal nodes. This is the principle used in BMEP.

Another application of the MWTR includes in telecommunications the inference of the underlying network. The reconstruction of the telecommunications network includes the determination of the underlying network topology as part of the network tomography process. The internet is a collection of interconnected networks, and its topology is unknown because of its decentralized

and unregulated growth. In network tomography the knowledge of the underlying routing within some (unknown) network topology is important to develop more sophisticated and ambitious traffic control protocols and dynamic routing algorithms. Ni and Tatikonda [46] propose "*a general framework for designing and analyzing topology and link performance inference algorithms using ideas and tools from phylogenetic inference in evolutionary biology. The framework is built upon additive metrics. Under an additive metric the path metric (path length) is expressed as the summation of the link metrics (link lengths) along the path. The basic idea is to use (estimated) distances between the terminal nodes (end hosts) to infer the routing tree topology and link metrics*". Chung et al. [10] investigate distance realization problems, a class of problems which, in their words, arise in the study of internet data traffic models and refer the construction of trees to infer the logical network topology.

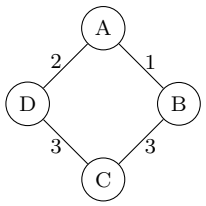
There are several techniques to reconstruct a network topology. For instance, one such technique can use ICMP (Internet Control Message Protocol) commands such as ping and traceroute [17, 19, 33]. However, these techniques require the cooperation of all the internal network devices, which is, most of the time, not possible, due to political and security issues. Another way to infer the routing network topology is to use tomographic techniques. These techniques only use end-to-end network measurements, such as loss measurements or delay variance. The end-to-end network measurements can be obtained using multicast [21, 22, 47] or unicast probing [11, 24, 51]. With this limited information only the logical topology can be inferred. The logical topology is obtained from the physical topology, representing only the physical devices on the network where traffic branching occurs and joining all the connections between these devices by a single logical link. The key idea is to use measurements at pairs of receivers to identify the logical topology defined by the branching points between paths to different receivers. In network design the terminal nodes represent the receivers, additional nodes represent physical devices where traffic branching occurs, edges represent relationships between pair of nodes and edges weights represent the quantification of some evolving connection property, like the delay or packet losses, for which end-to-end network measurements can be obtained.

As already mentioned, all these applications have in common the notion of a graph realization of a distance matrix [1]. Over the years several authors studied the characteristics of the distance matrix and its graph and tree realization [23, 35, 39, 52, 53, 54, 56, 57].

Hakimi and Yau [36] were the first to refer, in 1965, the graph realization problem of a distance matrix and presented an algorithm for the special case where the realization of the matrix is a tree. Consider a $n \times n$ positive, symmetric matrix D, where the diagonal elements are zero, and every entry $d_{ij} > 0$, $\forall i, j \in V_t, i \neq j$, represents the distance between terminal node i and terminal node j. In addition, if the elements of the matrix satisfy the *triangle inequality* $d_{ij} \leq d_{ik} + d_{kj}$, $\forall i, j, k \in V_t$, then the matrix D is called a *distance matrix* and has a *graph realization* [38, 56]. The graph realization problem [10] can be stated as follows. Given a distance matrix D of dimension n, representing distances between a set of n objects, determine a edge-weighted connected graph $G = (V, E)$ with node set V, $|V| > n$, containing a subset $V_t \subset V$, with $|V_t| = n$ terminal nodes (each node representing an object), and the value d_{ij} in matrix D satisfying $d^G_{ij} = d_{ij}$, where d^G_{ij} denotes the length of the shortest path in G between terminal node i and terminal node j, $i, j \in V_t$. If such a graph exists then the distance matrix D has a realization. This means a metric given by the distance matrix D must be embedded into

	A	B	C	D
A	0	1	2	4
B	1	0	3	3
C	2	3	0	3
D	4	3	3	0

(a) Distance matrix.



(b) Graph realization.

Figure 1: A distance matrix and its graph realization.

a graph. Associated with this existence problem is the optimization problem that determines a graph such that the sum of all edge weights of G is minimized. Among all the realizations of the distance matrix D, the one having the minimum total length is optimal [10].

For a general distance matrix an optimal realization exists [20], however computing optimal realizations is hard, even for a small number of terminal nodes [10]. Chung et al. [10] formulate several versions of the distance matrix realization problem, mention relevant results, discuss their algorithmic implications, present approximation results and several heuristics. Because the realization problem is hard even to approximate, Chung et al. [10] and Farach et al. [25] introduce a *weak realization* of a distance matrix. A distance matrix D has a weak realization if there is a graph G whose node set contains node set V_t and the distance d_{ij}^G in the graph G between nodes i and j is greater than or equal to d_{ij}. For a given distance matrix with n terminal nodes, the distance, in the graph to be built, between two terminal nodes must be the minimal possible. Such a graph which is a weak realization for D must be a tree [10]. Since the weak realization is a tree, for a set of n terminal nodes, there can be at most $n-2$ additional nodes of degree exactly three. Even when restricting the topologies to binary unrooted trees, there are $(2n-5)!! = \frac{(2n-5)!}{2^{n-3}(n-3)!}$ different possible topologies [9, 44]. Hence there are finitely many topologies for a weak realization for D, and for each topology the problem of determining a weak realization can be formulated as a linear programming problem. To the best of our knowledge Cantanzaro et al. [6, 7] are the only authors to present (mixed integer) linear programming models for solving the problem.

It is worth noting that although a distance matrix has a graph realization, not all distance matrices describe trees. To describe a tree the distance matrix must also be *additive* [3], i.e. satisfy the four-point condition. In Figure 1 we present a distance matrix and its graph realization. This distance matrix has no tree realization.

Distances that fit exactly on a tree can be characterized by the following *four-point condition* that generalizes the triangle inequality (take $k = \ell$)

$$d_{ij} + d_{k\ell} \leq \max\{d_{ik} + d_{j\ell}, d_{i\ell} + d_{jk}\}, \qquad \forall i, j, k, \ell \in V_t. \tag{1.1}$$

As a consequence of this condition, on any quartet i, j, k, ℓ we have that of the three sums $d_{ij}+d_{k\ell}$, $d_{ik}+d_{j\ell}$ and $d_{i\ell}+d_{jk}$, the largest two are equal. Distance matrices with this property are called *additive*, because the weights on the paths along the tree add up to the values in the distance matrix. The distance matrix represented in Figure 1 does not satisfy the four-point condition, it is not additive. Buneman [3] proved that a distance matrix can be represented by a tree if it is additive. For a tree with 4 terminal nodes, i, j, k, ℓ (e.g. Figure 2), it can be checked that the

four-point condition (1.1) holds

$$w_{i1} + w_{1j} + w_{k2} + w_{2\ell} \leq$$

$$\max\{w_{i1} + w_{12} + w_{2k} + w_{j1} + w_{12} + w_{2\ell},\ w_{i1} + w_{12} + w_{2\ell} + w_{j1} + w_{12} + w_{2k}\}.$$

with w_{ij} being the weights associate to each edge in the tree solution.

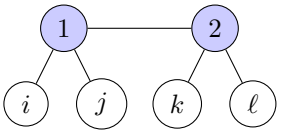


Figure 2: A possible topology for a tree with 4 terminal nodes.

The constructed trees can either be rooted or unrooted. We consider unrooted trees and all the contents in this work can be adapted to a rooted tree. Therefore, and for modeling purposes, we force the tree structure to be a unrooted binary tree, all the additional nodes to have degree exactly three and the terminal nodes to be the leaves. Restricting ourselves to such topologies will help us to deal with this combinatorial problem which is a weak realization problem. Enforcing this tree structure is not a restriction to the problem as any tree can be converted into a rooted binary tree, and vice-versa. Byun and Yoo [4] present an algorithm to convert a tree into a binary tree by including dummy nodes. The root node can then be added at the middle point in the longest path of the tree. If this point is in the edge $\{i, j\}$, remove edge $\{i, j\}$, add the dummy root node r and edges $\{i, r\}$ and $\{j, r\}$, one new edge inherits the weight of edge $\{i, j\}$ and the other new edge has weight 0. If this point is in the node i, consider the edge $\{i, j\}$ connecting two nodes and proceed as before.

In this work we contribute with two compact mixed integer linear programming (MILP) formulations of the MWTR problem. Both formulations use flows to help with the definition of the tree topology, however each with a different underlying reconstructing idea. One formulation, the Path-weight formulation, produces a tree solution and the other formulation, the Path-edges formulation, constructs a balanced tree. A balanced tree is a tree with minimal diameter. The matrix defined using the obtained branch weights is such that its submatrix corresponding to the terminal nodes dominates the given distance matrix. The Path-weight formulation produces a tree solution to the MWTR problem, using additional flow variables to ensure connectivity. This formulation is inspired from Model 4 in Catanzaro et al. [7]. As reported in [7], the linear relaxation of this model has an objective value of zero. We strengthen the formulation by adding valid inequalities that cut off the zero-objective solutions.

The Path-edges formulation produces a balanced tree solution of the MWTR problem and flows are defined between every pair of terminal nodes. In Catanzaro et al. [8] a balanced tree is also considered. However, we use multicommodity flows to deal with connectivity and path lengths. Instead of minimizing the total sum of all edge weights, the objective function deals with the minimization of the total tree length. Comparatively to the formulations presented in [8], formulation Path-edges solves more instances to optimality and, for the instances not solved to optimality, obtains a better lower bound value in smaller computational time. This paper aims

at developing tighter compact MILP formulations for the MWTR problem, without requiring the development of specialized algorithms. Our aim is not to compete with the state-of-the-art algorithms which is a very specialized branch-and-cut-and-price algorithm based on a formulation with an exponential number of variables.

In Section 2, we present and discuss two mixed integer linear programming formulations for the MWTR problem and include valid equalities and inequalities which improve the performance of the second formulation. Section 3 is devoted to computational results and Section 4 concludes the paper.

2 Formulations

In this section we present two Mixed Integer Linear Programming formulations of the MWTR problem. Consider a tree $T = (V, E)$ spanning the set of nodes $V = V_a \cup V_t$. V_a is the set of additional or internal nodes and V_t is the set of terminal or external nodes. The pairwise distances between terminal nodes in V_t are given in distance matrix D. The tree topologies we consider are such that the additional nodes of the tree all have degree three. Any tree can be transformed into a tree where every additional node has degree three by adding "dummy" nodes and edges, as described in [4]. When $|V_t| = n$, an unrooted tree has $n-2$ additional nodes and $2n-3$ edges. Without loss of generality, let $V_a = \{1, \ldots, n-2\}$ be the additional node set and $V_t = \{n-1, \ldots, 2n-2\}$ be the terminal node set.

In feasible trees, the length of a path varies between the length of a caterpillar tree (the most imbalanced tree) and the length of the most balanced tree. In a caterpillar tree all additional nodes are consecutively connected forming the central path (e.g. Figure 3). Thus all the additional nodes, but two, are connected to exactly one terminal node, and those two additional nodes are connected exactly to two terminal nodes. A binary balanced tree has the minimum possible maximum height (depth) and is such that the heights of the left and right subtrees differ by at most one and the left and the right subtrees are balanced. A balanced tree is a binary tree having the smallest diameter (e.g. Figure 4) and therefore the shortest path length with consecutive additional nodes. This tree has the n terminal nodes pairwise connected to $\lfloor \frac{n}{2} \rfloor$ additional nodes, if n is even (e.g. Figure 4, on the right), and if n is odd $n-1$ terminal nodes pairwise connected to $\lfloor \frac{n}{2} \rfloor$ additional nodes and 1 terminal node connected to an additional node (e.g. Figure 4, on the left). Eliminating the terminal nodes from this balanced tree we obtain a tree with only additional nodes. The degree of the nodes of this new tree are distributed as follows: $\lfloor \frac{n}{2} \rfloor$ nodes have degree one, $\lceil \frac{n}{2} \rceil$ -$\lfloor \frac{n}{2} \rfloor$ nodes have degree two and $n-2-\lceil \frac{n}{2} \rceil$ nodes have degree three. The shortest path with consecutive additional nodes is obtained by binding consecutively a node with degree one, to the nodes with degree three, to the node of degree two, if it exits, and to a node with degree one. The obtained path uses $1 + (n-2-\lceil \frac{n}{2} \rceil) + (\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor) + 1 = n - \lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{2} \rceil$ additional nodes. Therefore any feasible solution contains a path with at least $\lceil \frac{n}{2} \rceil$ consecutive additional nodes.

Additional nodes can be arbitrarily interchanged in a solution to our problem, without modifying the tree topology. This introduces a lot of symmetry in the problem. As any feasible solution to the problem contains a path with at least $\lceil \frac{n}{2} \rceil$ consecutive additional nodes, the numbering of additional nodes is arbitrary. To break some of that symmetry, we can assume this

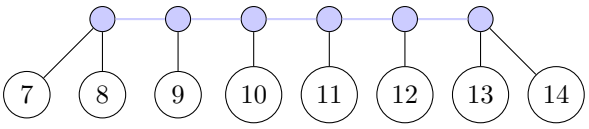


Figure 3: A caterpillar tree. A possible topology for a tree with 8 terminal nodes.

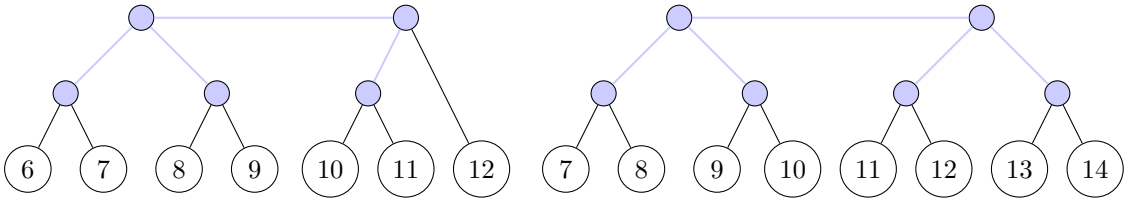


Figure 4: Two balanced trees. On the left, a possible topology for a tree with 7 terminal nodes. On the right, a possible topology for a tree with 8 terminal nodes.

path is composed of edges $\{i, i+1\}$ for additional nodes $i = 1, \ldots, \lceil \frac{n}{2} \rceil - 1$, and force these edges to be present in any solution.

Both formulations use the following variables. Binary variables x_{ij}, $i \in V_a$, $j \in V$, $i < j$ indicate whether edge $\{i, j\}$ belongs to the tree solution, while continuous variables $w_{ij} \geq 0$ represent the weight associated to edge $\{i, j\}$.

2.1 Path-weight formulation

We seek a tree T and associated weights w_e, $e \in T$, that provide a weak realization of distance matrix D. Consider the path P_{ij} that connects terminal nodes i and j, $i, j \in V_t$. By definition of a weak realization,

$$\sum_{e \in P_{ij}} w_e \geq d_{ij} \tag{2.1}$$

must hold. To ensure this, consider additional continuous variables u_{ij}, for all $i, j \in V$, $i \neq j$, which indicate the length of the path between nodes i and j.

In order to impose connectivity several approaches can be used. Usual approaches consists either in the inclusion of the subtour elimination inequalities

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, S \subset V, S \neq \emptyset, |S| > 1 \tag{2.2}$$

or in the inclusion of the cut-set inequalities

$$\sum_{i \in S, j \in S^c} x_{ij} \geq 1, S \subset V, S \neq \emptyset, 0 \in S. \tag{2.3}$$

The linear relaxation of both models provide the same bound, however the number of inequalities increase exponentially with the size of the model. It is well known that in order to ensure connectivity/prevent circuits, instead of using one of the families of inequalities (2.2) and (2.3) with an exponential number of inequalities, one can use compact extended formulations [41]. The most common are derived using either the well-known Miller-Tucker-Zemlin inequalities [42, 31]

or using stronger multicommodity flow formulations [32, 41]. In the formulation presented below we use multicommodity flows. Fixing additional node 1 as the root of the flow, we introduce binary flow variables y_{ij}^k for all $k \in V_t$, $i \in V_a$, $j \in V$ with $i < j$, indicating whether edge $\{i, j\}$ is used from i to j in the path from root node 1 to terminal node k.

Let $d_{max} := \max\{d_{ij} : i, j \in V_t\}$. The formulation that minimizes the total edges weights and reconstructs an unrooted tree for the MWTR problem is as follows.

Path-weight formulation

$$min \quad \sum_{i \in V_a} \sum_{\substack{j \in V \\ j>i}} w_{ij}$$

subject to

$$\sum_{i \in V_a} \sum_{\substack{j \in V \\ j>i}} x_{ij} = 2n - 3 \tag{2.4}$$

$$\sum_{\substack{j \in V \\ j>i}} x_{ij} + \sum_{\substack{j \in V_a \\ i>j}} x_{ji} = 3, \qquad \forall i \in V_a \tag{2.5}$$

$$\sum_{\substack{i \in V_a \\ j>i}} x_{ij} = 1, \qquad \forall j \in V_t \tag{2.6}$$

$$x_{i,i+1} = 1 \qquad \forall i \in V_a,\ i = 1, ..., (\lceil n/2 \rceil - 1) \tag{2.7}$$

$$\sum_{\substack{j \in V_a \cup \{k\} \\ j \neq 1}} y_{1j}^k = 1, \qquad \forall k \in V_t \tag{2.8}$$

$$\sum_{i \in V_a} y_{ik}^k = 1, \qquad \forall k \in V_t \tag{2.9}$$

$$\sum_{\substack{i \in V_a \\ i<j}} y_{ij}^k = \sum_{\substack{i \in V_a \cup \{k\} \\ i>j}} y_{ji}^k, \qquad \forall k \in V_t, \forall j \in V_a \setminus \{1\} \tag{2.10}$$

$$y_{ij}^k \leq x_{ij}, \qquad \forall k \in V_t, \forall i \in V_a, \forall j \in V \setminus \{1\}, j > i \tag{2.11}$$

$$u_{ij} \geq d_{ij} \qquad \forall i, j \in V_t, i < j \tag{2.12}$$

$$u_{ij} \geq w_{ij} \qquad \forall i \in V_a, \forall j \in V_t, i < j \tag{2.13}$$

$$d_{max} x_{ij} \geq w_{ij} \qquad \forall i \in V_a, \forall j \in V, i < j \tag{2.14}$$

$$w_{ij} \geq u_{ij} - d_{max}(1 - x_{ij}) \qquad \forall i \in V_a, \forall j \in V, i < j \tag{2.15}$$

$$w_{ij} \geq u_{ik} - u_{jk} - d_{max}(1 - x_{ij}) \qquad \forall i \in V_a, \forall j, k \in V, i < j, j < k \tag{2.16}$$

$$w_{ij} \geq u_{jk} - u_{ik} - d_{max}(1 - x_{ij}) \qquad \forall i \in V_a, \forall j, k \in V, i < j, j < k \tag{2.17}$$

$$w_{ij} \geq u_{ik} - u_{kj} - d_{max}(1 - x_{ij}) \qquad \forall i \in V_a, \forall j, k \in V, i < k, k < j \tag{2.18}$$

$$w_{ij} \geq u_{kj} - u_{ik} - d_{max}(1 - x_{ij}) \qquad \forall i \in V_a, \forall j, k \in V, i < k, k < j \tag{2.19}$$

$$w_{ij} \geq u_{ki} - u_{kj} - d_{max}(1 - x_{ij}) \qquad \forall i, k \in V_a, \forall j \in V, i < j, k < i \tag{2.20}$$

$$w_{ij} \geq u_{kj} - u_{ki} - d_{max}(1 - x_{ij}) \qquad \forall i, k \in V_a, \forall j \in V, i < j, k < i \tag{2.21}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in V_a, \forall j \in V, i < j \tag{2.22}$$

$$y_{ij}^k \in \{0, 1\} \qquad \forall k \in V_t, \forall i \in V_a, \forall j \in V, i < j \tag{2.23}$$

$$w_{ij} \geq 0 \qquad \forall i \in V_a, \forall j \in V, i < j \tag{2.24}$$

$$u_{ij} \geq 0 \qquad \forall i, j \in V, i < j \tag{2.25}$$

Constraints (2.4)-(2.7) define a spanning tree with all additional nodes with degree three and all terminal nodes with degree one. The cardinality constraints (2.4) ensure that there are $2n-3$ edges in the solution. Constraints (2.5) ensure that all additional nodes have degree three. Constraints (2.6) ensure that all terminal nodes have degree one. As already mentioned, to reduce the symmetry, we fix a path between the first additional nodes with constraints (2.7). Constraints

(2.8)-(2.10) are flow conservation constraints and constraints (2.11) are linking constraints between flow variables y_{ij}^k and topology variables x_{ij}. Constraints (2.12) - (2.21) link edge weight variables w_{ij} and path length variables u_{ij}. Constraints (2.12) ensure a weak realization of the distance matrix. Constraints (2.13) establish a lower bound for variables u_{ij}, associated to the length of the path between an additional node i and an external node j, to be the value of the corresponding (edge) weight w_{ij}. Constraints (2.14) fix w_{ij} to 0 when edge $\{i, j\}$ does not belong to the tree. Together with (2.13), constraints (2.15) impose that w_{ij} is equal to u_{ij} if edge $\{i, j\}$ belongs to the tree. Constraints (2.16)-(2.21) impose the triangle inequality for any order of the nodes i, j and k. The index specifications included in the constraints (2.16)-(2.21) improve the performance of our formulation as they impose an order on the variables index. Constraints (2.22) to (2.25) are integrality and non-negativity constraints.

Our formulation strengthens the Flow Model proposed in [7] by including lower bounds on the w_{ij} variables.

2.2 Path-edges formulation

Pauplin [49] developed a method to directly calculate the sum of all weights of a tree (the length of the tree) without having to explicitly determine its edge-weights. According to this method the tree length is given by $\sum_{i,j\in V_t} d_{ij}\ 2^{-z_{ij}}$ where z_{ij} indicates the number of edges in the path P_{ij} between terminal nodes i and j. When $d_{ij} = \sum_{e\in P_{ij}} w_e$, we have $\sum_{i,j\in V_t} d_{ij}\ 2^{-z_{ij}} = \sum_{i,j\in V_t} \sum_{e\in P_{ij}} w_e\ 2^{-z_{ij}}$. As the weight w_e appears as many times as the number of paths P_{ij} to which the edge e belongs to, it holds $\sum_{i,j\in V_t} d_{ij}\ 2^{-z_{ij}} = \sum_{e\in E} w_e$ and for a weak realization we have $\sum_{i,j\in V_t} d_{ij}\ 2^{-z_{ij}} \leq \sum_{e\in E} w_e$ and $\min \sum_{i,j\in V_t} d_{ij}\ 2^{-z_{ij}} \leq \min \sum_{e\in E} w_e$.

Using binary variables p_{ij}^ℓ, for all $i, j \in V_t$, $i < j$ and $\ell \in \{2, 3, ..., (n-1)\}$, specifying the number of edges of a path P_{ij} between terminal nodes i and j, the expression $\sum_{i,j\in V_t} d_{ij}\ 2^{-z_{ij}}$ can be linearized. This relation and linearization process was already used in [8]. The binary decision variables p_{ij}^ℓ indicating whether the path P_{ij} connecting terminal node i to terminal node j has (exactly) ℓ edges. Therefore $\sum_{i,j\in V_t} d_{ij}\ 2^{-z_{ij}} = \sum_{i,j\in V_t} d_{ij}\ \sum_{\ell=2}^{n-1} 2^{-\ell} p_{ij}^\ell$. Variables p_{ij}^ℓ have the same interpretation as variables x_{ij}^ℓ in [8].

Besides these path variables p_{ij}^ℓ, the binary topology variables x_{ij} and the weight variables w_{ij}, we consider flow variables. The binary flow variables $f_{ij}^{k\ell}$, for all $i, j \in V_a \cup \{k, \ell\}, k, \ell \in V_t$, $i \neq j$ and $k < \ell$, indicate whether the flow traverses the edge $\{i, j\}$ belonging to the path connecting terminal node k to terminal node ℓ in the direction from node i to node j.

The formulation that specifies the number of edges of a path between terminal nodes and reconstructs an unrooted tree for the MWTR problem is as follows.

Path-edges formulation

$$min \quad \sum_{i \in V_t} \sum_{\substack{j \in V_t \\ j>i}} d_{ij} \sum_{\ell=2}^{n-1} 2^{-\ell} \cdot p_{ij}^{\ell}$$

subject to

$$\sum_{i \in V_a} \sum_{\substack{j \in V \\ j>i}} x_{ij} = 2n-3 \tag{2.26}$$

$$\sum_{i \in V_a} x_{ij} = 1, \qquad \forall j \in V_t \tag{2.27}$$

$$\sum_{\substack{j \in V \\ j>i}} x_{ij} + \sum_{\substack{j \in V_a \\ j<i}} x_{ji} = 3, \qquad \forall i \in V_a \tag{2.28}$$

$$x_{i,i+1} = 1 \qquad \forall i \in V_a,\ i = 1, ..., (\lceil n/2 \rceil - 1) \tag{2.29}$$

$$\sum_{j \in V_t} x_{1j} = 2 \tag{2.30}$$

$$\sum_{j \in V_t} x_{(n-2)j} = 2 \tag{2.31}$$

$$\sum_{j \in V_t} x_{ij} \leq 2 \qquad \forall i \in V_a \tag{2.32}$$

$$\sum_{i \in V_a} f_{ki}^{k\ell} = 1 \qquad \forall k, \ell \in V_t, k < \ell \tag{2.33}$$

$$\sum_{j \in \{\ell\} \cup V_a \setminus \{i\}} f_{ij}^{k\ell} - \sum_{j \in \{k\} \cup V_a \setminus \{i\}} f_{ji}^{k\ell} = 0 \qquad \forall i \in V_a, k, \ell \in V_t, k < \ell \tag{2.34}$$

$$\sum_{i \in V_a} f_{i\ell}^{k\ell} = 1 \qquad \forall k, \ell \in V_t, k < \ell \tag{2.35}$$

$$\sum_{h \in \{\ell\} \cup V_a \setminus \{i\}} f_{jh}^{k\ell} - f_{ij}^{k\ell} \geq 0 \qquad \forall i \in V_a \cup \{k\}, j \in V_a, k, \ell \in V_t, k < \ell \tag{2.36}$$

$$f_{ij}^{k\ell} + f_{ji}^{k\ell} \leq x_{ij} \qquad \forall i, j \in V, \forall k, \ell \in V_t, i < j, k < \ell \tag{2.37}$$

$$\sum_{\ell=2}^{n-1} p_{ij}^{\ell} = 1 \qquad \forall i, j \in V_t, i < j \tag{2.38}$$

$$2 + \sum_{i \in V_a} \sum_{\substack{j \in V_a \\ j \neq i}} f_{ij}^{k\ell} = \sum_{i=2}^{n-1} i \cdot p_{k\ell}^{i} \qquad \forall k, \ell \in V_t, k < \ell \tag{2.39}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in V_a, \forall j \in V, i < j \tag{2.40}$$

$$p_{ij}^{\ell} \in \{0,1\} \qquad \forall \ell \in \{2,3,...,n-1\}, \forall i, j \in V_t, i < j \tag{2.41}$$

$$f_{ij}^{k\ell} \in \{0,1\} \qquad \forall i, j \in V_a \cup \{k, \ell\} \forall k, \ell \in V_t, i \neq j, k < \ell \tag{2.42}$$

Constraint (2.26) is the tree cardinality constraint and establishes that the number of edges in the tree is $2n-3$. Constraints (2.27) establish that all the terminal nodes have degree one and constraints (2.28) force the additional nodes degree to be three. As above, we fix a path of additional nodes with constraints (2.29). Since the tree is unrooted, we know that there are two additional nodes which are adjacent to two terminal nodes. Therefore, to reduce symmetry, constraints (2.30) and (2.31) enforce those two additional nodes to be node 1 and node $(n-2)$.

Constraints (2.32) impose that an additional node is connected to, at most, two terminal nodes. Constraints (2.33)–(2.35) are flow conservation constraints. Constraints (2.36) establish that if the flow sent from terminal node k to terminal node ℓ passes through edge $\{i, j\}$, in the direction from node i to node j, then the flow passes through, at least, one edge between node j and a node different than node i. Constraints (2.37) are linking constraints between variables and impose that there can be no flow in edge $\{i, j\}$ if it does not belong to the tree. Constraints (2.38) impose that variables p_{ij}^{ℓ} assume value one for exactly one ℓ in $\{2, 3, ..., (n-1)\}$ corresponding to the number of edges in the path between terminal nodes i and j. Constraints (2.39) relate variables $f_{ij}^{k\ell}$ with variables p_{ij}^{ℓ}. Using variables $f_{ij}^{k\ell}$ we know exactly the number of edges in the path between terminal nodes k and ℓ. Using variables $p_{k\ell}^{i}$ we also know exactly the number of edges in the path between terminal nodes k and ℓ and that number must be exactly the same. Constraints (2.40), (2.41) and (2.42) are the integrality constraints.

This second formulation uses the same idea in [8] to obtain the total tree edges length by the formulation designated by the authors as the Path-Length-4-Point (PL4). In Section 3 we will compare the formulations presented in this section with this PL4 formulation.

After having reconstructed a unrooted tree with this formulation, the weights have to be assigned to the tree edges. This is accomplished by solving the following simple linear program.

$$
\begin{aligned}
& min \quad \sum_{i \in V_a} \sum_{\substack{j \in V \\ j>i}} w_{ij} \\
& subject\ to \\
& \quad \sum_{i \in V_a} \sum_{\substack{j \in V \\ i<j}} w_{ij} (f_{ij}^{k\ell} + f_{ji}^{k\ell}) \geq d_{k\ell} \qquad \forall k, \ell \in V_t, k < \ell \\
& \quad w_{ij} \geq 0 \qquad \forall i, j \in V, i < j
\end{aligned}
$$

Using the flow variables, the path between each pair of terminal nodes is exactly known. This information is used to associate weights to the edges such that the total sum of the edges weights is minimized and the tree length between every pair of terminal nodes dominates (is greater than) the corresponding distance from the distance matrix D.

In a tree there is exactly one path between every pair of terminal nodes and the path variables p_{ij}^{ℓ} state there are ℓ, for some unique $\ell \geq 2$, edges in the path between terminal nodes i and j. Therefore $\sum_{\ell=2}^{n-1} \ell p_{ij}^{\ell}$ is the number of edges in the path between terminal nodes i and j. By summing the number of edges of all paths and taking into account that some edges belong to more than one path we must have

$$
2 \sum_{i \in V_t} \sum_{\substack{j \in V_t \\ j>i}} \sum_{\ell=2}^{n-1} 2^{-\ell}\ \ell\ p_{ij}^{\ell} = 2n - 3, \tag{2.43}
$$

as the number of edges in the tree is $2n - 3$. This is already stated in the formulation through constraint (2.26), however this equality reinforces this condition using the path variables.

Huffman codes are optimal path-length sequences whose corresponding rooted binary tree determines a code. When producing optimal Huffman codes the key idea are these rooted binary trees represented as sequences of ascending path-lengths. A sequence of n path-lengths represents a binary tree with n leaves where each leaf represents a symbol in the code. Parker and Ram [48]

characterize these path-length sequences by establishing that these binary trees obey the property established by the Kraft equality, a special case of the Kraft inequality. These path-lengths can be compared to the distances from the tree realization problem. Therefore the nontrivial property of the path-length sequences in a rooted binary tree characterized with the Kraft equality can be borrowed by the MWTR problem and the following equality can be established:

$$\sum_{\ell=2}^{n-1} \sum_{\substack{j \in V_t \\ j>i}} 2^{-\ell} p_{ij}^{\ell} + \sum_{\ell=2}^{n-1} \sum_{\substack{j \in V_t \\ j<i}} 2^{-\ell} p_{ji}^{\ell} = \frac{1}{2} \qquad \forall i \in V_t \tag{2.44}$$

The inclusion of the two equalities (2.43) and (2.44) improved the performance of the Path-edges formulation.

Beside these two equalities the following valid inequalities presented in [8] can also be included.

$$\sum_{\substack{j \in V_t \\ i<j}} p_{ij}^{n-1} + \sum_{\substack{j \in V_t \\ j<i}} p_{ji}^{n-1} \leq 2 \sum_{\substack{j \in V_t \\ i<j}} p_{ij}^{\ell} + 2 \sum_{\substack{j \in V_t \\ j<i}} p_{ji}^{\ell} \qquad \forall i \in V_t, \forall \ell \in \{2, 3, ..., n-2\} \tag{2.45}$$

$$\sum_{i \in V_t} \sum_{\substack{j \in V_t \\ j>i}} p_{ij}^{n-1} \leq 4 \tag{2.46}$$

$$\sum_{\substack{j \in V_t \\ i<j}} \sum_{q=2}^{\ell} 2^{\ell-q} p_{ij}^{q} \leq 2^{\ell-1} - 1 \qquad \forall i \in V_t, \forall \ell \in \{2, 3, ..., \lfloor \frac{n}{2} \rfloor\}, n > 2^{\ell-1} + 1 \tag{2.47}$$

Inequalities (2.45) state that if a tree has a path of length $n-1$ then it also has a path of length $n-2, n-3, \ldots, 2$, inequality (2.46) indicates that a tree has at most four paths of length $n-1$ and inequalities (2.47) are a consequence of the Kraft equality.

3 Computational Results

Computational results will assess the quality of the Linear Programming (LP) solutions obtained with each formulation from Section 2 and the best lower and upper bounds achieved. The computational tests were performed on an Intel(R) Core(TM) i7-3770 CPU 3.40 GHz processor and 16.0 Gb of RAM. We present computational results for instances of the problem with a number of terminal nodes varying between 5 and 15, for a total of 55 instances.

We use two sets of data instances, one set coming from a phylogenetics application, and the other one from a networking application. The first set of instances is available from `http://pubsonline.informs.org/doi/\-suppl\-/10.1287/ijoc.1110.0455` [8]. From this set we use three phylogenetic distance matrices, matrices M391, Primate and M887, with $t = 15$, $t = 12$ and $t = 15$ taxa, respectively, and for each we vary the number of terminal nodes (taxa) between 5 and t, obtaining 30 instances. The data for the second set of instances were generated using the network-level simulator NS-3 [45]. We performed three simulations named Sim7, Sim15 and Sim20 with $t = 7$, $t = 15$ and $t = 15$ terminal nodes, varying the number of terminal nodes between 5 and t obtaining 25 instances.

The two formulations from Section 2 were implemented using the Mosel language and solved with FICO Xpress 7.1 [27] (Xpress-IVE 1.23.00 64 bit, Xpress-Optimizer 23.01.03 and Xpress-Mosel 3.4.0). Path-edges formulation used together with the valid equalities (2.43) and (2.44)

and inequalities (2.45), (2.46), (2.47) presented at the end of Section 2 is designated Path-edges$^+$ formulation. We compare the performance of the two formulations introduced in Section 2, Path-weight formulation and Path-edges formulation, and the Path-edges$^+$ formulation with the formulation PL4 from [8]. Our implementation of PL4 considers all the valid inequalities presented in [8].

The computational results are summarized in Tables 1 and 2 in which the first column, labelled **Matrix**, indicates the name of the matrix instance used and the second column, labelled $|V_t|$, indicates the size of the instance. The third, forth and fifth columns concern the results of the Path-weight formulation, from the sixth to the eleventh columns the results of the Path-edges formulation are presented, from the twelfth to the seventeenth columns are the results of the Path-edges$^+$ formulation and the eighteenth, nineteenth, twentieth and twenty-first columns concern PL4 formulation. The columns labelled **T** show the execution time, in seconds, used to solve the instance and having a maximum runtime of 7200 seconds and for the Path-edges formulation and Path-edges$^+$ formulation, the columns labelled $\mathbf{T_w}$ shows the execution time of the linear program solved to assign the weights. The columns labelled **W** and **DZ** present the optimum value obtained or the best value obtained having a runtime limit of 7200 seconds, where DZ stands for $\sum_{k\in V_t}\sum_{\substack{\ell\in V_t\\ \ell>k}} d_{k\ell}\sum_{i=2}^{n-1} 2^{-i}\cdot p^i_{k\ell}$ and W stands for $\sum_{i\in V_a}\sum_{\substack{j\in V\\ j>i}} w_{ij}$. The columns labelled **GAP** present the LP solution gap which is obtained as follows: $GAP = \dfrac{UB - PL}{UB} \times 100$, where UB represents the best upper bound value obtained (or the optimum value) within the runtime of 7200 seconds and PL represents the value of the corresponding linear programming relaxation. The columns labelled $\mathbf{GAP_{LB}}$ present the lower bound gap and is obtained as follows: $GAP_{LB} = \dfrac{UB - LB}{UB} \times 100$, where UB represents the best upper bound value obtained (or the optimum value) and LB the best lower bound value obtained within the runtime of 7200 seconds.

Table 1: Computational results for data from the phylogenetics application.

Matrix	$\|V_t\|$	Path-weight formulation			Path-edges formulation						Path-edges$^+$ formulation						PL4			
		T	**W**	**GAP$_{LB}$**	**T**	**DZ**	**GAP**	**GAP$_{LB}$**	**T$_w$**	**W**	**T**	**DZ**	**GAP**	**GAP$_{LB}$**	**T$_w$**	**W**	**T**	**DZ**	**GAP**	**GAP$_{LB}$**
M391	5	0.219	0.0486	0	0.109	0.0473	31.52	0	0	0.0486	0	0.0473	0	0	0.031	0.0487	0.577	0.0473	0	0
	6	0.499	0.0583	0	0.499	0.057	58.78	0	0.016	0.0591	0.078	0.057	0.34	0	0.031	0.0591	33.243	0.057	0.08	0
	7	2.278	0.0626	0	2.793	0.0607	73.4	0	0.015	0.0634	1.919	0.0607	0.69	0	0.031	0.0634	328.021	0.0607	0.37	0
	8	73.975	0.0693	0	58.141	0.0672	82.69	0	0.063	0.0698	6.49	0.0672	0.75	0	0.109	0.0698	> 7200	–	–	–
	9	1003.18	0.0948	0	735.182	0.0923	90.14	0	0.14	0.0951	20.155	0.0923	0.6	0	0.172	0.0951	> 7200	–	–	–
	10	> 7200	0.1061	60.77	> 7200	0.1027	94.29	36.08	0.655	0.1068	728.225	0.1027	1.04	0	7.691	0.1068	> 7200	–	–	–
	11	> 7200	0.1326	83.44	> 7200	0.1289	96.78	59.67	18.658	0.134	2675.67	0.1276	1.18	0	14.929	0.1337	> 7200	–	–	–
	12	> 7200	0.1386	–	> 7200	0.1377	98.21	95.64	33.509	0.1449	> 7200	1.322	1.21	0.84	77.532	0.1378	> 7200	–	–	–
	13	> 7200	0.1555	–	> 7200	0.1537	98.98	98.18	59.075	0.1658	> 7200	0.1445	1.8	1.73	63.882	0.1523	> 7200	–	–	–
	14	> 7200	0.1607	–	> 7200	0.1603	99.42	99.1	108.95	0.1711	> 7200	0.1481	2.34	2.27	306.4	0.1562	> 7200	–	–	–
	15	> 7200	0.1733	–	> 7200	0.1694	99.66	99.65	278.866	0.1813	> 7200	0.156	3.82	3.75	417.815	0.1647	> 7200	–	–	–
Primate	5	0.25	0.843	0	0.218	0.0823	31.91	0	0	0.0843	0	0.0823	0	0	0	0.0843	1.42	0.0823	0	0
	6	0.531	0.1059	0	0.452	0.1023	61.19	0	0.016	0.1059	0.031	0.1023	0	0	0.032	0.1059	67.907	0.1023	0	0
	7	1.841	0.1272	0	1.732	0.1232	72.54	0	0.031	0.1272	0.406	0.1232	0.37	0	0.046	0.1272	507.906	0.1232	0	0
	8	25.943	0.128	0	23.041	0.1229	80.3	0	0.063	0.128	2.294	0.1229	0.38	0	0.094	0.128	> 7200	–	–	–
	9	228.697	0.1296	0	211.802	0.1246	87.11	0	0.14	0.1296	17.768	0.1246	3.03	0	0.188	0.1296	> 7200	–	–	–
	10	> 7200	0.1297	6.5	> 7200	0.122	91.98	10.62	0.531	0.1298	107.593	0.122	2.45	0	19.16	0.1298	> 7200	–	–	–
	11	> 7200	0.17	72.41	> 7200	0.1652	95.88	52.35	15.787	0.1762	1.356.95	0.1625	2.3	0	34.413	0.17	> 7200	–	–	–
	12	> 7200	0.2076	96.68	> 7200	0.2019	97.62	76.03	29.125	0.212	> 7200	0.1959	2.23	1.54	75.426	0.2058	> 7200	–	–	–
M887	5	0.219	0.1033	0	0.218	0.0972	29.56	0	0	0.1033	0	0.0971	0	0	0	0.1033	0.125	0.0972	0	0
	6	0.499	0.1157	0	0.546	0.1098	57.62	0	0.015	0.1201	0.5	0.1098	1.88	0	0.078	0.1201	64.788	0.1098	0.42	0
	7	2.121	0.1364	0	1.965	0.1292	71.56	0	0.016	0.1364	1.263	0.1292	1.52	0	0.047	0.1364	608.917	0.1292	0	0
	8	63.945	0.1756	0	41.917	0.1619	82.73	0	0.063	0.1763	5.975	0.1619	1.52	0	0.094	0.1763	> 7200	0.188	14.14	14.01
	9	742.375	0.1823	0	487.235	0.1685	89.37	0	0.125	0.186	24.102	0.1685	1.7	0	0.172	0.186	> 7200	–	–	–
	10	> 7200	0.2026	50.69	> 7200	0.1859	93.75	30.69	0.609	0.2059	220.412	0.1859	2.11	0	18.954	0.2059	> 7200	–	–	–
	11	> 7200	0.2113	81.2	> 7200	0.196	96.50	0	60.25	0.2168	1960.88	0.1938	2.57	0	15.038	0.2135	> 7200	–	–	–
	12	> 7200	0.2184	–	> 7200	0.2058	98.95	94.65	33.041	0.2259	> 7200	0.1967	2.71	2.59	72.961	0.2179	> 7200	–	–	–
	13	> 7200	0.2395	–	> 7200	0.2292	98.94	97.84	56.098	0.2562	> 7200	0.2172	4.4	4.31	146.999	0.238	> 7200	–	–	–
	14	> 7200	0.2512	–	> 7200	0.2519	99.45	99.09	136.344	0.2829	> 7200	0.2304	8.1	8.03	252.018	0.2543	> 7200	–	–	–
	15	> 7200	0.2787	–	> 7200	0.2602	99.66	99.65	207.262	0.2922	> 7200	0.2664	14.34	14.29	953.521	0.304	> 7200	–	–	–

Table 2: Computational results for data from the networking application.

Matrix	$\|V_t\|$	Path-weight formulation			Path-edges formulation						Path-edges$^+$ formulation						PL4			
		T	**W**	**GAP$_{LB}$**	**T**	**DZ**	**GAP**	**GAP$_{LB}$**	**T$_w$**	**W**	**T**	**DZ**	**GAP**	**GAP$_{LB}$**	**T$_w$**	**W**	**T**	**DZ**	**GAP**	**GAP$_{LB}$**
	5	0.22	0.4043	0	0.296	0.4043	28.13	0	0	0.4043	0.016	0.4043	0	0	0.015	0.4043	0.16	0.4043	0	0
Sim7	6	0.28	0.4348	0	0.858	0.4347	49.88	0	0	0.4348	0.031	0.4347	0	0	0.296	0.4348	23.845	0.4347	0	0
	7	1.872	0.5055	0	2.808	0.5054	67.91	0	0.031	0.5055	0.047	0.5054	0	0	0.031	0.5055	4181.93	0.5054	0	0
	5	0.3	0.1893	0	0.515	0.1893	34.45	0	0	0.1893	0	0.1893	0	0	0.015	0.1893	0.124	0.1893	0	0
	6	0.26	0.2756	0	1.138	0.2756	57.34	0	0.016	0.2756	0.047	0.2756	0	0	0.015	0.2756	118.092	0.2756	0	0
	7	1.286	0.3013	0	1.748	0.3016	66.47	0	0.016	0.3013	0.53	0.3013	1.55	0	0.047	0.3013	8.056	0.3013	0	0
	8	31.692	0.3778	0	18.189	0.3778	79.30	0	0.063	0.3778	9.672	0.3778	2.69	0	0.125	0.3778	> 7200	–	–	–
Sim15	9	309.517	0.3932	0	122.663	0.3932	86.48	0	0.156	0.3932	13.307	0.3932	2.61	0	0.203	0.3932	> 7200	–	–	–
	10	> 7200	0.4286	33.44	4726.98	0.4286	91.81	0	0.406	0.4286	155.048	0.4286	3.16	0	18.58	0.4286	> 7200	–	–	–
	11	> 7200	0.626	60.31	> 7200	0.626	95.94	51.79	13.478	0.626	1625.24	0.626	2.22	0	32.926	0.626	> 7200	–	–	–
	12	> 7200	0.722	94.74	> 7200	0.727	97.64	94.99	30.311	0.7322	> 7200	0.722	1.93	1.73	67.579	0.722	> 7200	–	–	–
	13	> 7200	0.8085	–	> 7200	0.7918	98.54	97.44	38.142	0.8188	> 7200	0.7474	7.09	2.45	82.571	0.7474	> 7200	–	–	–
	14	> 7200	0.8845	–	> 7200	0.9796	99.26	98.82	146.531	1.0875	> 7200	0.8335	14.16	12.65	281.939	0.8742	> 7200	–	–	–
	15	> 7200	0.9804	–	> 7200	1.1557	99.61	99.37	172.099	1.2949	> 7200	0.9859	15.31	14.72	823.027	1.0418	> 7200	–	–	–
	5	0.203	0.299	0	0.328	0.299	0	0	0	0.299	0	0.299	0	0	0	0.299	0.109	0.299	0	0
	6	0.187	0.3394	0	0.796	0.3394	57.39	0	0.015	0.3394	0.031	0.3394	0	0	0.031	0.3394	0.842	0.3394	0	0
	7	1.825	0.3751	0	1.747	0.3751	70.65	0	0.31	0.3751	0.14	0.3751	3.72	0	0.031	0.3751	8.973	0.3751	3.24	0
	8	46.722	0.4564	0	31.496	0.4564	81.78	0	0.078	0.4564	8.112	0.4564	3.53	0	0.094	0.4564	> 7200	–	–	–
Sim20	9	581.085	0.5174	0	376.226	0.5173	88.40	0	0.172	0.5174	32.199	0.5174	3.3	0	0.218	0.5174	> 7200	–	–	–
	10	> 7200	0.5736	48.81	> 7200	0.5736	92.98	23.97	0.609	0.5736	434.664	0.5736	2.89	0	19.078	0.5736	> 7200	–	–	–
	11	> 7200	0.609	69.97	> 7200	0.609	95.72	49.69	18.938	0.609	1652.79	0.609	2.72	0	15.35	0.609	> 7200	–	–	–
	12	> 7200	0.6347	98.44	> 7200	0.6373	97.55	76.05	36.208	0.645	> 7200	0.6244	2.83	0	85.488	0.6244	> 7200	–	–	–
	13	> 7200	0.7515	–	> 7200	0.7657	98.71	98.39	70.06	0.7824	> 7200	0.7301	2.79	2.17	146.952	0.7301	> 7200	–	–	–
	14	> 7200	0.8481	–	> 7200	0.9027	99.33	98.88	104.567	0.9674	> 7200	0.7915	3.22	2.7	732.016	0.7967	> 7200	–	–	–
	15	> 7200	1.0057	–	> 7200	1.0633	99.65	99.64	282.391	1.1268	> 7200	0.9012	4.26	3.82	958.435	0.914	> 7200	–	–	–

The optimum solution within the time limit imposed is obtained in the following cases:

- by the Path-weight formulation for all instances with $n < 10$ terminal nodes and for instance Sim15 with $n = 10$;
- by the Path-edges formulation for all instances with $n < 10$ terminal nodes;
- by the Path-edges$^+$ formulation for all instances with $n < 12$ terminal nodes;
- by the PL4 formulation for all instances with $n < 8$ terminal nodes.

To better check the improvements achieved we display in Table 3 the number of instances, among the total of 55 instances, for which the optimal solution was obtained, within the time limit imposed, when the model indicated is used.

Path-weight	Path-edges	Path-edges$^+$	PL4
28	29	38	18

Table 3: Number of instances solved within the time limit imposed.

The Path-edges$^+$ formulation solves the instances substantially faster than the other formulations. In Figure 5, we compare the computational times in the form of a profile graph that displays the number of solved instances in a given time by each formulation.

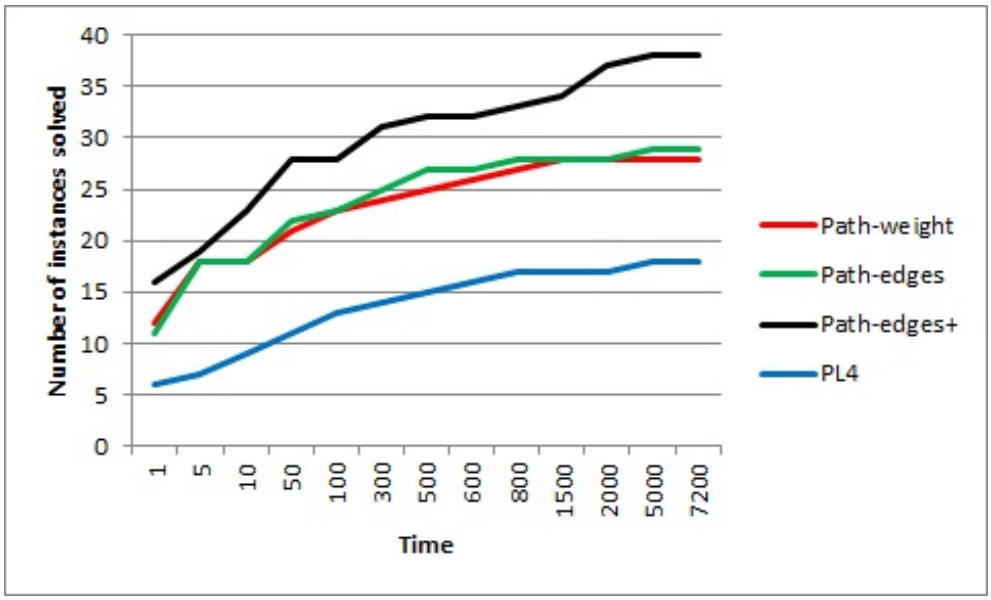


Figure 5: Performance profile.

Comparing, in Tables 1 and 2, the columns labelled with W of the Path-weight formulation and of the Path-edges formulation we notice that for some instances from the phylogenetics application the values obtained are different. This is due to the fact that the obtained trees are different. The tree we obtain when minimizing $\sum_{k\in V_t} \sum_{\substack{\ell\in V_t \\ \ell>k}} d_{k\ell} \sum_{i=2}^{n-1} 2^{-i} \cdot p^i_{k\ell}$ is such that, when assigning its edges weights the equality $\sum_{e\in P_{ij}} w_e = d_{ij}$ does not hold. In that case we obtain a

Table 4: Average and standard deviation values for the computational time.

n		Path-weight	Path-edges	Path-edges$^+$	PL4
5	average	0.25	0.28	0	0.42
	standard deviation	0.04	0.14	0.01	0.52
6	average	0.38	0.71	0.12	51.54
	standard deviation	0.15	0.26	0.29	45.58
7	average	1.87	2.13	0.72	940.63
	standard deviation	0.34	0.52	0.73	1607.35
8	average	48.46	34.56	6.51	-
	standard deviation	20.51	15.97	2.77	-
9	average	572.97	386.62	21.51	-
	standard deviation	316.93	240.81	7.14	-

tree that minimizes $\sum_{k\in V_t}\sum_{\substack{\ell\in V_t\\ \ell>k}} d_{k\ell}\sum_{i=2}^{n-1} 2^{-i}\cdot p^i_{k\ell}$ but does not minimizes $\sum_{i\in V_a}\sum_{\substack{j\in V\\ j>i}} w_{ij}$ (see column labelled DZ), as in this case it holds $\sum_{k\in V_t}\sum_{\substack{\ell\in V_t\\ \ell>k}} d_{k\ell}\sum_{i=2}^{n-1} 2^{-i}\cdot p^i_{k\ell} < \sum_{i\in V_a}\sum_{\substack{j\in V\\ j>i}} w_{ij}$.

When the optimal solution can not be found within the runtime limit imposed the Path-weight formulation obtains, on average, ten feasible solutions, the Path-edges$^+$ formulation obtains, on average, three feasible solutions and the PL4 formulation is unable to obtain feasible solutions within the runtime limit imposed, except for matrix M887 where for $n = 8$ this formulation obtained one feasible solution. The difference between the number of feasible solutions of the Path-weight formulation and of the Path-edges formulation is due to the fact that the only integer (binary) variables of the Path-weight formulation are those which identify the tree, whereas the Path-edges formulation only has integer (binary) variables.

For instances with the same number of terminal nodes, Tables 4, 5 and 6 display the average time, the average GAP and the average GAP_{LB}, respectively, and their corresponding standard deviation values. Table 4 presents the values for instances with $n \leq 9$, since for $n > 9$ only the Path-edges$^+$ obtains a feasible solution within the runtime limit imposed. Table 6 displays the values for the formulations Path-weight, Path-edges and Path-edge$^+$, since the formulation PL4 does not obtain feasible solutions for $n \geq 10$, and for instances with $n \geq 10$ because for $n < 10$ the GAP_{LB} is equal to zero (the optimal solution is achieved).

Now we compare the results between the Path-edges formulation and the Path-edges$^+$ formulation.The Path-edges$^+$ formulation is substantially faster, for example for $n = 9$ it is eighteen times faster. The average GAP of the Path-edges formulation varies from 25.93%, for $n = 5$, to 99.65%, for $n = 15$, while the average GAP of the Path-edges$^+$ formulation does not exceed 9%. Finally, the average GAP_{LB} of the Path-edges formulation is over 90% while for the Path-edges$^+$ formulation it does not exceed 9%. We may conclude that the equalities and inequalities included in the Path-edges formulation considerably improve the formulation.

4 Conclusion

In this paper, we introduce two formulations for the minimum weighted tree reconstruction (MWTR) problem: the Path-weight formulation and the Path-edges formulation. Some

Table 5: Average and standard deviation values for the GAP.

n		Path-edges	Path-edges$^+$	PL4
5	average	25.93	0.31	0
	standard deviation	12.88	0.77	0
6	average	57.03	0.27	0.08
	standard deviation	3.8	0.51	0.17
7	average	70.42	1.31	0.6
	standard deviation	2.71	1.33	1.3
8	average	81.36	1.81	-
	standard deviation	1.52	1.31	-
9	average	88.3	2.63	-
	standard deviation	1.52	0.54	-
10	average	92.96	2.42	-
	standard deviation	1.08	0.82	-
11	average	96.16	2.23	-
	standard deviation	0.45	0.63	-
12	average	97.99	2.52	-
	standard deviation	0.6	1.2	-
13	average	98.79	4.95	-
	standard deviation	0.21	3.11	-
14	average	99.37	8.52	-
	standard deviation	0.09	6.63	-
15	average	99.65	5.85	-
	standard deviation	0.02	6.59	-

Table 6: Average and standard deviation values for the GAP_{LB}.

n		Path-weight	Path-edges	Path-edges$^+$
10	average	40.04	92.96	0
	standard deviation	21.15	1.08	0
11	average	73.47	96.16	0
	standard deviation	9.3	0.45	0
12	average	96.62	97.99	2.52
	standard deviation	1.85	0.6	1.2
13	average	-	98.79	4.95
	standard deviation	-	0.21	3.11
14	average	-	99.37	8.52
	standard deviation	-	0.09	6.63
15	average	-	99.65	5.85
	standard deviation	-	0.02	6.59

valid equalities and inequalities are also introduced to strengthen the Path-edges formulation. Computational experiments performed on data sets coming from phylogenetics and networking applications show that the strengthened Path-edges formulation outperforms the previous formulations proposed in the literature. Several strategies could be applied to both formulations to improve their performance. Among them is the strategy proposed in [8]. Generally the application of these strategies is not straightforward. We leave this approach for future work.

Acknowledgments

The research of Cristina Requejo and Olga Oliveira has been partially supported by Portuguese funds through the *Center for Research and Development in Mathematics and Applications (CIDMA)* and the FCT, the Portuguese Foundation for Science and Technology, within project UID/MAT/04106/2013.

The research of Olga Oliveira has been supported by a research fellowship (grant SFRH/BD/-76268/2011) through FCT, the Portuguese Foundation for Science and Technology, within the project POPH-QREN.

The research of Bernard Fortz and Olga Oliveira has been funded by the Interuniversity Attraction Poles Programme P7/36 ≪COMEX≫ initiated by the Belgian Science Policy Office.

References

[1] H. Abdi. Additive-tree representations. In A. Dress and A. von Haeseler, editors, *Trees and Hierarchical Structures*, volume 84 of *Lecture Notes in Biomathematics*, pages 43–59. Springer Berlin Heidelberg, 1990.

[2] S. Bhamidi, R. Rajagopal, and S. Roch. Network delay inference from additive metrics. *Random Structures Algorithms*, 37(2):176–203, 2010.

[3] P. Buneman. A note on the metric properties of trees. *Journal of Combinatorial Theory*, 17:48–50, 1974.

[4] S.-S. Byun and C. Yoo. Reducing delivery delay in HRM tree. In M. Gavrilova, O. Gervasi, V. Kumar, C. Tan, D. Taniar, A. LaganÃ¡, Y. Mun, and H. Choo, editors, *Computational Science and Its Applications - ICCSA 2006*, volume 3981 of *Lecture Notes in Computer Science*, pages 1189–1198. Springer Berlin Heidelberg, 2006.

[5] D. Catanzaro. The minimum evolution problem: Overview and classification. *Networks*, 53(2):112–125, 2009.

[6] D. Catanzaro, R. Aringhieri, M. Di Summa, and R. Pesenti. A branch-price-and-cut algorithm for the minimum evolution problem. *European Journal of Operational Research*, 244(3):753 – 765, 2015.

[7] D. Catanzaro, M. Labbé, R. Pesenti, and J.-J. Salazar-Gonzaléz. Mathematical models to reconstruct phylogenetic trees under the minimum evolution criterion. *Networks*, 53(2):126–140, 2009.

[8] D. Catanzaro, M. Labbé, R. Pesenti, and J.-J. Salazar-Gonzaléz. The balanced minimum evolution problem. *INFORMS Journal on Computing*, 24(2):276–294, 2012.

[9] L. Cavalli-Sforza and A. Edwards. Phylogenetic analysis. Models and estimation procedures. *American Journal of Human Genetics*, 19:233–257, 1967.

[10] F. Chung, M. Garrett, R. Graham, and D. Shallcross. Distance realization problems with applications to internet tomography. *Journal of Computer and System Sciences*, 63(3):432–448, 2001.

[11] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang. Maximum likelihood network topology identification from edge-based unicast measurements. In *ACM SIGMETRICS Performance Evaluation Review*, volume 30, pages 11–20, 2002.

[12] M. Coates, M. Rabbat, and R. Nowak. Merging logical topologies using end-to-end measurements. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, IMC '03, pages 192–203, 2003.

[13] J. E. Corter and A. Tversky. Extended similarity trees. *Psychometrika*, 51(3):429–451, 1986.

[14] J. P. Cunningham. Trees as memory representations for simple visual patterns. *Memory & Cognition*, 8(6):593–605, 1980.

[15] W. H. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bulletin of Mathematical Biology*, 49(4):461–467, 1987.

[16] Z. Dias, A. Rocha, and S. Goldenstein. Image phylogeny by minimal spanning trees. *IEEE Transactions on Information Forensics and Security*, 7(2):774–788, 2012.

[17] B. Donnet. Internet topology discovery. In *Data Traffic Monitoring and Analysis: From Measurement, Classification, and Anomaly Detection to Quality of Experience*, volume 7754 of *Lecture Notes in Computer Science*, pages 44–81. Springer, 2013.

[18] B. Donnet and T. Friedman. Internet topology discovery: A survey. *IEEE Communications Surveys*, 9(4):2–14, 2007.

[19] B. Donnet, P. Raoult, T. Friedman, and M. Crovella. Deployment of an algorithm for large-scale topology discovery. *IEEE Journal on Selected Areas in Communications*, 24(12):2210–2220, 2006.

[20] A. Dress. Trees, tight extensions of metric spaces, and the cohomological dimension of certain groups: A note on combinatorial properties of metric spaces. *Advances in Mathematics*, 53:321–402, 1984.

[21] N. Duffield, J. Horowitz, F. L. Presti, and D. Towsley. Multicast topology inference from measured end-to-end loss. *IEEE Transactions on Information Theory*, 48(1):26–45, 2002.

[22] N. Duffield, J. Horowitz, and F. L. Prestis. Adaptive multicast topology inference. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE.*, volume 3, pages 1636–1645, 2001.

[23] M. Edelberg, M. Garey, and R. Graham. On the distance matrix of a tree. *Discrete Mathematics*, 14:23–39, 1976.

[24] B. Eriksson, G. Dasarathy, P. Barford, and R. Nowak. Toward the practical use of network tomography for internet topology discovery. In *INFOCOM 2010. Proceedings IEEE.*, pages 1–9, 2010.

[25] M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13(1):155–179, 1995.

[26] J. Felsenstein. Distance matrix methods. In *Inferring Phylogenies*, chapter 11, pages 147–175. Sinauer Associates, Inc., Sunderland, Massachusetts, 2004.

[27] *FICO Xpress Optimization Suite*. www.fico.com/xpress.

[28] S. Fiorini and G. Joret. Approximating the balanced minimum evolution problem. *Operations Research Letters*, 40(1):31–35, 2012.

[29] B. Gaschen, J. Taylor, K. Yusim, B. Foley, F. Gao, D. Lang, V. Novitsky, B. Haynes, B. Hahn, T. Bhattacharya, and B. Korber. Diversity considerations in HIV-1 vaccine selection. *Science*, 296(5577):2354–2360, 2002.

[30] G. Gilmore, H. Hersh, A. Caramazza, and J. Griffin. Multidimensional letter similarity derived from recognition errors. *Perception & Psychophysics*, 25(5):425–431, 1979.

[31] L. Gouveia. Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints. *Computers and Operations Research*, 22(9):959–970, 1995.

[32] L. Gouveia. Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research*, 95:178–190, 1996.

[33] R. Govindan and H. Tangmunarunkit. Heuristic for internet map discovery. In *Proceeding of the* 19^{th} *annual joint conference of IEEE Computer and Communications Societies*, volume 3, pages 1371–1380, 2000.

[34] H. Haddadi, M. Rio, G. Iannaccone, A. Moore, and R. Mortier. Network topologies: Inference, modelling and generation. *IEEE Communications Surveys*, 10(2):48–69, 2008.

[35] S. L. Hakimi and A. Patrinos. The distance matrix of a graph and its tree realization. *Quarterly of Applied Mathematics*, 30:255–269, 1972–73.

[36] S. L. Hakimi and S. S. Yau. Distance matrix of a graph and its realizability. *Quarterly of Applied Mathematics*, 22:305–317, 1965.

[37] D. Huson and C. Scornavacca. A survey of combinatorial methods for phylogenetic networks. *Genome Biology and Evolution*, 3:23–35, 2011.

[38] A. Isaev. *Introduction to Mathematical Methods in Bioinformatics*. Springer, 2006.

[39] A. L. J. Koolen and V. Moulton. Optimal realizations of generic five-point metric. *European Journal of Combinatorics*, 30:1164–1171, 2009.

[40] B. H. Junker and F. Schreiber. *Analysis of Biological Networks*. John Wiley & Sons, 2008.

[41] T. L. Magnanti and L. A. Wolsey. Optimal trees. In M. Ball, T. L. Magnanti, C. Monma, and G. L. Nemhauser, editors, *Network Models*, Handbooks in Operations Research and Management Science, Vol. 7, pages 503–615. Elsevier Science Publishers, North-Holland, 1995.

[42] C. Miller, A. Tucker, and R. Zemlin. Integer programming formulations and travelling salesman problems. *Journal of the Association for Computing Machinery*, 7(4):326–329, 1960.

[43] D. Mount. *Bioinformatics. Sequence and genome analysis*. Cold Spring Harbor Laboratory Press, 2001.

[44] M. Nei and S. Kumar. *Molecular evolution and phylogenetics*. Oxford University Press, 2000.

[45] *Network Simulator NS-3*. www.nsnam.org.

[46] J. Ni and S. Tatikonda. Network tomography based on additive metrics. *IEEE Transactions on Information Theory*, 57(12):7798–7809, 2011.

[47] J. Ni, H. Xie, S. Tatikonda, and Y. R. Yang. Network routing topology inference from end-to-end measurements. In *INFOCOM 2008, the 27th IEEE Conference on Computer Communications*, 2008.

[48] D. Parker and P. Ram. The construction of Huffman codes is a submodular ("convex") optimization problem over a lattice of binary trees. *SIAM Journal on Computing*, 28(5):1875–1905, 1996.

[49] Y. Pauplin. Direct calculation of a tree length using a distance matrix. *Journal of Molecular Evolution*, 51:41–47, 2000.

[50] S. Sattath and A. Tversky. Additive similarity trees. *Psichometrica*, 42(3):319–345, 1977.

[51] M.-F. Shih and A. Hero. Hierarchical inference of unicast network topologies based on end-to-end measurements. *IEEE Transactions on Signal Processing*, 55(5):1708–1718, 2007.

[52] J. Simões-Pereira. A note on distance matrices with unicyclic graph realizations. *Discrete Mathematics*, 65:277–287, 1987.

[53] J. Simões-Pereira. An algorithm and its role in the study of optimal graph realizations of distance matrices. *Discrete Mathematics*, 79:299–312, 1990.

[54] J. Simões-Pereira and C. Zamfirescu. Submatrices of non-tree-realizable distance matrices. *Linear Algebra and its Applications*, 44:1–17, 1982.

[55] G. D. Soete. A least squares algorithm for fitting additive trees to proximity data. *Psychometrika*, 48(4):621–626, 1983.

[56] S. C. Varone. Trees related to realizations of distance matrices. *Discrete Mathematics*, 192:337–346, 1998.

[57] S. C. Varone. A constructive algorithm for realizing a distance matrix. *European Journal of Operational Research*, 174(1):102–111, 2006.