MIP model-based heuristics for the minimum weighted tree reconstruction problem*

Olga Fajarda[1] and Cristina Requejo[2]

[1] *Institute of Electronics and Informatics Engineering of Aveiro (IEETA), University of Aveiro, 3810-193 Aveiro, Portugal*
[1,2] *University of Aveiro, 3810-193 Aveiro, Portugal*
[1] *olga.oliveira@ua.pt [O. Fajarda]*
[2] *crequejo@ua.pt [C. Requejo]*

Revised August 2020; Online October 2020; Issue Date July 2022

Abstract

We consider the Minimum Weighted Tree Reconstruction (MWTR) problem and two matheuristic methods to obtain optimal or near-optimal solutions: the Feasibility Pump heuristic and the Local Branching heuristic. These matheuristics are based on a Mixed Integer Programming (MIP) model used to find feasible solutions. We discuss the applicability and effectiveness of the matheuristics to obtain solutions to the MWTR problem. The purpose of the MWTR problem is to find a minimum weighted tree connecting a set of leaves in such a way that the length of the path between each pair of leaves is greater than or equal to a given distance between the considered pair of leaves. The Feasibility Pump matheuristic starts with the Linear Programming solution, iteratively fixes the values of some variables and solves the corresponding problem until a feasible solution is achieved. The Local Branching matheuristic, in its turn, improves a feasible solution by using a local search. Computational results using two different sets of instances, one from the phylogenetic area and another from the telecommunications area, show that these matheuristics are quite effective in finding feasible solutions and present small gap values. Each matheuristic can be used independently; however, the best results are obtained when used together. For instances of the problem having up to 17 leaves, the feasible solution obtained by the Feasibility Pump heuristic is improved by the Local Branching heuristic. Noticeably, when comparing with existing based models processes that solve instances having up to 15 leaves, this achievement of the matheuristic increases the size of solved instances.

Keywords: feasibility pump; local branching; mixed integer linear programming; matheuristics; tree realization; topology discovery; routing topology inference; minimum evolution problem; balanced minimum evolution problem.

1 Introduction

The Minimum Weighted Tree Reconstruction (MWTR) Problem is a combinatorial optimization problem that consists in reconstructing a weighted tree $T = (V, E)$. The tree reconstruction is obtained by knowing only the pairwise distances d_{ij} between all nodes i, j from a set V_t, subset of the set of nodes V of a graph. More precisely, given a $n \times n$ symmetric distance matrix $D = [d_{ij}]$ and a set V_t of n leaves, terminal nodes, the goal of the problem is to simultaneously (i) find an unrooted tree $T = (V, E)$ spanning $V = V_t \cup V_a$, where V_a is a subset of V with additional nodes which are internal nodes, and which wlog we can assume to be of degree three, and $V_t \cap V_a = \emptyset$, (ii) associate edge weights $w_e,\ e \in E$, such that the lenght of the unique path P_{ij} between any

*Cite: Fajarda, O., Requejo, C. MIP model-based heuristics for the minimum weighted tree reconstruction problem. Oper Res Int J 22, 2305–2342 (2022). https://doi.org/10.1007/s12351-020-00608-z

two leaves i and j from V_t is at least d_{ij}, i.e. $\sum_{e \in P_{ij}} w_e \geq d_{ij}$, and (iii) such that the total sum of the edge weights $\sum_{\{i,j\} \in E} w_{ij}$ is minimized.

The combinatorial problem associated with the MWTR problem is a tree realization problem for a distance matrix as it aims to reconstruct a tree by knowing only pairwise distances, stored in the input distance matrix. The tree realization problem is a specific version of the distance realization problem, which, in turn, is a graph realization problem [11]. Versions of this combinatorial problem were proved to be NP-complete [11, 15, 20]. Fiorini and Joret [22] and Catanzaro et al. [7] discussed the NP-hardness of two related optimization problems, namely the Balanced Minimum Evolution Problem (BMEP) [9, 22] and the Minimum Evolution Problem (MEP) [7], which are well-known distance realization problems from the computational biology area.

The MWTR arises in several areas: in telecommunications (namely, in network tomography) to discover the routing topology of a network [3, 11, 19, 27] as well as the logical underlying network [11, 12, 19, 30]; in psychology [13, 14, 16, 26, 34] to represent cognitive processes or proximity and similarity relations and in information security for the detection and recognition of documents duplications [18, 26]. However, possibly the most well-known application of the MWTR is, in computational biology, the reconstruction of phylogenetic trees [6, 21, 28].

Mixed Integer Programming (MIP) models for the MWTR appeared in [7, 8, 9, 25]. First, in [8] the authors introduced MIP models to solve the MEP and studied possible cuts and lower bounds for the optimal value of the problem. In [9] the authors presented a MIP model to exactly solve instances of the BMEP and developed branching rules and families of valid inequalities to further strengthen the model. In [7] the authors developed an exact solution approach for the MEP based on a nontrivial combination of a parallel branch-and-price-and-cut scheme and a non-isomorphic enumeration of all possible solutions to the problem. In the works [7, 8, 9], computational experiments were performed on phylogenetic datasets. In [25] the authors presented two compact MIP models to solve the problem, without requiring the development of specialized algorithms. Computational experiments in [25] were performed on two different datasets, one from the phylogenetic area already used in [7, 8, 9], and another from the telecommunications area.

Finding a feasible solution to a MIP problem can be very hard and may involve large computational effort. To cope with this situation, several heuristics are designed to efficiently produce feasible solutions of good quality. Several very efficient heuristics have been proposed in the literature [29, 33] for distance tree realization problems.

Among the heuristic approaches used to obtain feasible solutions to MIP problems, one can find the matheuristics that use MIP models. Fischetti et al. [23] proposed a very successful matheuristic, that they called the *Feasibility Pump* (FP) intending to find feasible solutions (if any exists) for generic MIP problems. The authors focused essentially on Mixed Binary Programming (MBP) problems. The FP heuristic is improved, mostly for generic MIP problems, by Achterberg and Berthold [1] and Bertacco et al. [2]. Another successful matheuristic is the *Local Branching* (LB), proposed by Fischetti and Lodi [24], which is in the spirit of local search heuristics. The neighborhood of a feasible solution is obtained by adding, to the original problem model, constraints which Fischetti and Lodi [24] designated as *Local Branching* cuts. To the best of our knowledge, matheuristics have not been used for the MWTR problem. Our contribution, with this paper, is to evaluate the performance of these widely used matheuristics when applied to the MWTR problem. Also, we discuss several details on the application of the matheuristics.

For the first matheuristic, following the proposal of Fischetti et al. [23], a Feasibility Pump scheme is used. A mixed integer model for the MWTR problem together with a MIP solver is used to obtain (fractional) linear relaxation solutions. These fractional solutions are rounded to find a feasible solution (if any exists). For the second matheuristic, we consider the Local Branching scheme proposed by Fischetti and Lodi [24] to solve MIP problems. This enumerative scheme constructs a sequence of feasible solutions to the MWTR problem with improving (decreasing) value of costs. This approach is considered a very effective improving method for large scale problems. Again, we use a mixed integer model for the MWTR problem together with a MIP solver to explore reduced feasible regions. These schemes are based on the MIP model Path-edges$^+$ formulation presented in [25].

The article is organized as follows. In Section 2 we describe the MWTR problem and a MIP model presented in [25] that will be used in the matheuristic schemes. In Section 3 we present a Feasibility Pump heuristic for the MWTR problem. In Section 4 we present a Local Branching heuristic for the MWTR problem. In Section 5 we present computational results for both approaches. Finally, Section 6 concludes the article.

2 The Minimum Weighted Tree Reconstruction Problem

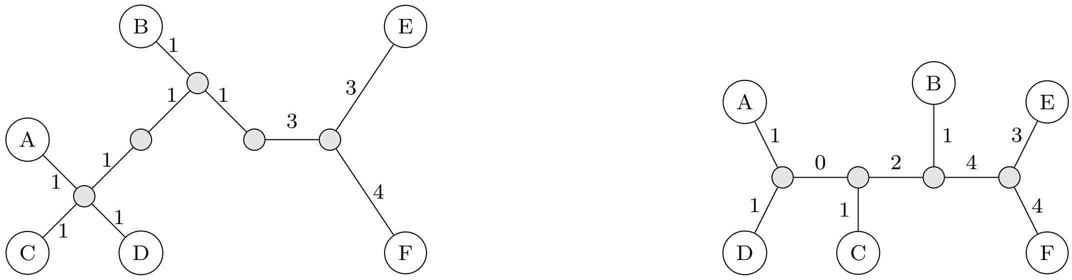
As already said, given a distance matrix, $D = [d_{ij}]$, that stores the pairwise distances d_{ij} between the leaves $i, j \in V_t$, our aim is to obtain a tree $T = (V, E)$ spanning the set of nodes $V = V_a \cup V_t$. The set V_t is the set of leaves, terminal or external nodes, V_a is the set of internal nodes and each tree edge has an associated weight. The unique path P_{ij} between leaves i and j has i and j as end nodes and, in between, it has a sequence of internal nodes from the set V_a. The distance between any two leaves i and $j \in V_t$ is given by the distance (obtained by summing the weights of the edges) of the unique path P_{ij} and is at least d_{ij}. In this work the tree structure and the weights associated to each edge are unknown and the objective is, by only knowing the distance matrix D, to determine the tree topology connecting the leaves and associate a weight to each edge, such that the obtained unique path satisfies the distance stored in the distance matrix. That is, the tree topology must be found and appropriate weights must be associated with the tree edges so as to satisfy the unique path distance constraint between leaves that provides a weak realization of the distance matrix D. More precisely, this weak realization is a weighted connected graph such that the obtained distance d^w_{ij} associated to every two leaves i and j is greater than or equal to the given distance d_{ij}. Any tree topology can be considered. However, to reduce the number of symmetric combinatorial solutions, the tree topologies to be considered are such that the internal nodes of the tree, nodes in set V_a, all have degree three [10]. This topology can be used because any tree can be transformed into a tree where every internal node may have degree three by adding "dummy" nodes and edges, as described in [5]. We observe that this is a stronger demand than the demand for being a binary tree (in which each node has at most two children), but at the same time, it reduces the number of symmetric combinatorial solutions. When $|V_t| = n$, to construct such an unrooted tree we must use $n - 2$ internal nodes with degree three and $2n - 3$ edges. Without loss of generality, we consider V_a such that $|V_a| = n - 2$, let $V_a = \{1, \dots, n - 2\}$ be the set of internal nodes, and renumber the n nodes in V_t such that $V_t = \{n - 1, \dots, 2n - 2\}$ be the set of leaves.

In Figure 1 we present in (a) a distance matrix D and in (b) and (c) two tree realizations of D of total weight sum of 17. The tree realization on the right-hand side is the optimal tree realization obtained by the model we present. All its internal nodes have degree three.

The MIP model-based procedure describing the MWTR problem must have the ability to both obtain a tree T and associate weights to the edges in the tree that provide a weak realization of the distance matrix D [11]. To achieve this goal two models are used. Both models were proposed in [25] and are described below for completeness as they are a fundamental part of the proposed matheuristics. The first model is used to obtain a tree T that is a weak realization of the distance matrix D and having the described properties. Next, the second model is used to associate weights to the edges of the obtained tree. To derive the two models, the following two sets of variables are a natural choice. For the first model let x_{ij}, $i \in V_a$, $j \in V$, $i < j$, denote a binary variable equal to 1 if edge $\{i, j\}$ belongs to the solution. For the second model, let $w_{ij} \geq 0$ denote a non-negative continuous variable representing the weight associated to edge $\{i, j\}$. In order to derive an extended first model, we use two more sets of variables. In particular, the binary variables p^{ℓ}_{ij}, for all $i, j \in V_t$, $i < j$ and $\ell \in \{2, 3, ..., (n - 1)\}$, specify the number of edges of a path P_{ij} between leaves i and j. These variables p^{ℓ}_{ij} indicate whether the path P_{ij} connecting leave i to leave j has (exactly) ℓ edges. Moreover, the binary flow variables $f^{k\ell}_{ij}$, 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 $\{i, j\}$

	A	B	C	D	E	F
A	0	4	2	2	10	11
B	4	0	4	4	8	9
C	2	4	0	2	10	11
D	2	4	2	0	10	11
E	10	8	10	10	0	7
F	11	9	11	11	7	0

(a) Distance matrix D.



(b) A tree realization of D. (c) A tree realization of D with all internal nodes with degree three.

Figure 1: A distance matrix D and two tree realizations.

belonging to the path connecting leave k to leave ℓ in the direction from node i to node j.

The following model has been presented in [25]. It reconstructs an unrooted tree and specifies both the edges and the number of edges of the path between every pair of leaves for the MWTR problem.

Path-edges$^+$ formulation

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

subject to

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

$$\sum_{i \in V_a} x_{ij} = 1 \qquad \forall j \in V_t \tag{3}$$

$$\sum_{\substack{j \in V \\ j > i}} x_{ij} + \sum_{\substack{j \in V_a \\ j < i}} x_{ji} = 3 \qquad \forall i \in V_a \tag{4}$$

$$x_{i,i+1} = 1 \qquad \forall i \in V_a, \ i = 1, ..., (\lceil n/2 \rceil - 1) \tag{5}$$

$$\sum_{j \in V_t} x_{1j} = 2 \tag{6}$$

$$\sum_{j \in V_t} x_{(n-2)j} = 2 \tag{7}$$

$$\sum_{j \in V_t} x_{ij} \leq 2 \qquad \forall i \in V_a \tag{8}$$

$$\sum_{i \in V_a} f_{ki}^{k\ell} = 1 \qquad \forall k, \ell \in V_t, k < \ell \tag{9}$$

$$\sum_{j \in \{\ell\} \cup V_a \setminus \{i\}} f_{ij}^{k\ell} - \sum_{j \in \{k\} \cup V_a \setminus \{i\}} f_{ji}^{k\ell} = 0 \qquad \forall i \in V_a, k, \ell \in V_t, k < \ell \tag{10}$$

$$\sum_{i \in V_a} f_{i\ell}^{k\ell} = 1 \qquad \forall k, \ell \in V_t, k < \ell \tag{11}$$

$$\sum_{h \in \{\ell\} \cup V_a \setminus \{i\}} f_{jh}^{k\ell} - f_{ij}^{k\ell} \geq 0 \qquad \forall i \in V_a \cup \{k\}, j \in V_a, k, \ell \in V_t, k < \ell \tag{12}$$

$$f_{ij}^{k\ell} + f_{ji}^{k\ell} \leq x_{ij} \qquad \forall i,j \in V, \forall k,\ell \in V_t, i<j, k<\ell \tag{13}$$

$$\sum_{\ell=2}^{n-1} p_{ij}^{\ell} = 1 \qquad \forall i,j \in V_t, i<j \tag{14}$$

$$2 + \sum_{i \in V_a} \sum_{\substack{j \in V_a \\ j \neq i}} f_{ij}^{k\ell} = \sum_{i=2}^{n-1} i \cdot p_{k\ell}^{i} \qquad \forall k,\ell \in V_t, k<\ell \tag{15}$$

$$2 \sum_{i \in V_t} \sum_{\substack{j \in V_t \\ j>i}} \sum_{\ell=2}^{n-1} 2^{-\ell}\, \ell\, p_{ij}^{\ell} = 2n-3 \tag{16}$$

$$\sum_{\ell=2}^{n-1} \sum_{\substack{j \in V_t \\ j>i}} 2^{-\ell} p_{ij}^{\ell} + \sum_{\ell=2}^{n-1} \sum_{\substack{j \in V_t \\ j<i}} 2^{-\ell} p_{ji}^{\ell} = \frac{1}{2} \qquad \forall i \in V_t \tag{17}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in V_a, \forall j \in V, i<j \tag{18}$$

$$p_{ij}^{\ell} \in \{0,1\} \qquad \forall \ell \in \{2,3,...,n-1\}, \forall i,j \in V_t, i<j \tag{19}$$

$$f_{ij}^{k\ell} \in \{0,1\} \qquad \forall i,j \in V_a \cup \{k,\ell\} \forall k,\ell \in V_t, i \neq j, k<\ell \tag{20}$$

The objective function (1) uses Pauplin's method [32] to calculate the sum of all weights of a minimum weighted tree realization of a distance matrix $D = [d_{ij}]$. Using the variables p_{ij}^{ℓ} and the strategy developed by Pauplin [32] it is possible to directly calculate the sum of all weights of a tree (the weight of the tree) by using the expression in the objective function and without having to explicitly assign its edge-weights w_{ij}. The edge-weights w_{ij} of a tree realization satisfy the following relation [7, see Proposition 9]

$$2 \sum_{i \in V_t} \sum_{\substack{j \in V_t \\ j>i}} d_{ij} \sum_{\ell=2}^{n-1} 2^{-\ell}\, p_{ij}^{\ell} \leq \sum_{i \in V_a} \sum_{\substack{j \in V \\ j>i}} w_{ij}$$

and are not used in the first model. However, they are *a posteriori* obtained by solving the simple optimization problem of the second model (24). Pauplin's method that avoids the explicit determination of the edge-weights has been used with success in [7, 17, 25].

The description of the constraints (2)–(20) can be found in [25]. The two valid equalities (16) and (17), presented in [7], are included to strengthen the model (improve the linear programming relaxation) and because their inclusion improves the performance of the model.

In coding theory, equality (17) is known as Kraft's equality [31]. In this context, the path-length value between two nodes i and j can be compared to the value of the distance d_{ij} from the tree realization problem. Notice that a sequence of n path-lengths represents a binary tree with n leaves where each leaf represents a symbol in a Huffman code, which is an optimal path-length sequence whose corresponding rooted binary tree determines a code. These path-length sequences may be characterized (see [31]) through Kraft's equality and in particular, they enforce that these binary trees obey the property established by Kraft's equality, which is a special case of Kraft's inequality [31]. Hence in the MWTR problem, we can use the property of the path-length sequences in a rooted binary tree characterized by Kraft's equality and the equality (17) can be established to reinforce the path-length between two leaves by using the path variables.

The following valid inequalities presented in [9] can also be included:

$$\sum_{\substack{j \in V_t \\ i<j}} p_{ij}^{n-1} + \sum_{\substack{j \in V_t \\ j<i}} p_{ji}^{n-1} \leq 2 \sum_{\substack{j \in V_t \\ i<j}} p_{ij}^{\ell} + 2 \sum_{\substack{j \in V_t \\ j<i}} p_{ji}^{\ell} \qquad \forall i \in V_t, \forall \ell \in \{2,3,...,n-2\} \tag{21}$$

$$\sum_{i \in V_t} \sum_{\substack{j \in V_t \\ j>i}} p_{ij}^{n-1} \leq 4 \tag{22}$$

$$\sum_{\substack{j \in V_t \\ i<j}} \sum_{q=2}^{\ell} 2^{\ell-q} p_{ij}^{q} \leq 2^{\ell-1} - 1 \qquad \forall i \in V_t, \forall \ell \in \{2,3,..., \lfloor \frac{n}{2} \rfloor\}, n > 2^{\ell-1}+1. \tag{23}$$

Inequalities (21) 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 (22) indicates that a tree has at most four paths of length $n-1$. Inequalities (23) are a consequence of Kraft's equality (17).

Feasible solutions of the previous MIP model correspond to unrooted trees together with a path between each pair of leaves. The weights of the edges are not determined and still have to be assigned to the tree edges. However, there is the guarantee that the distances will be satisfied and the obtained optimal tree corresponds to the one having the minimum weighted tree. To assign the weights to the tree edges the following simple linear program has to be solved.

$$\min \quad \sum_{i \in V_a} \sum_{\substack{j \in V \\ j > i}} w_{ij} \tag{24}$$

$$\textit{subject to}$$

$$\sum_{i \in V_a} \sum_{\substack{j \in V \\ i < j}} w_{ij}(f_{ij}^{k\ell} + f_{ji}^{k\ell}) \geq d_{k\ell} \qquad \forall k, \ell \in V_t, k < \ell$$

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

With the flow variables $f_{ij}^{k\ell}$, used here as constants, the path between each pair of leaves is exactly identified. This information is used to associate weights to the edges such that the total sum of the weights of the edges is minimized and the path-length between every pair of leaves dominates (is greater than) the corresponding distance from the distance matrix D.

To facilitate the description of the matheuristics in the next two sections, let $y = (x, f, p)$, with $x = (x_{ij}), f = (f_{ij}^{k\ell}), p = (p_{ij}^{\ell})$ with the appropriate index sets, denote the incidence vector of the feasible solutions and $\mathcal{P}$ denote the set of feasible solutions of the MWTR problem described by expressions (2)–(23). Therefore, compactly, Path-edges$^+$ formulation is

$$\begin{aligned} \min \quad & f(y) \\ \textit{subject to} \quad & \\ & y \in \mathcal{P} \end{aligned}$$

with $f(y)$ denoting the objective function, described by expression (1). Let $\mathcal{I}$ be the index set for vector y, hence $\mathcal{P} \subseteq \{0,1\}^{\mathcal{I}}$. Let $\mathcal{P}_L \subseteq [0,1]^{\mathcal{I}}$ denote the LP relaxation solution set of set $\mathcal{P}$ described by expressions (2)-(17) and (21)-(23).

3 The Feasibility Pump heuristic for the MWTR problem

In this section, following the scheme proposed by Fischetti et al. [23], we describe the Feasibility Pump heuristic (FP) adapted for the MWTR problem. As, in practice, it may be very time consuming to achieve a feasible integer solution to the MWTR problem, we impose a time limit and a maximum number of iterations. The basic idea of this matheuristic is to construct sequentially two sets of points, through a relax-and-fix approach, until finding a feasible solution to the problem. In the first set, the algorithm obtains points by solving a linear programming (LP) relaxation, thus these points satisfy the linear constraints but may not satisfy the integrality constraints. In the second set, the algorithm obtains points by rounding an LP feasible solution, therefore these points satisfy the integrality constraints but may not satisfy the linear constraints.

The Feasibility Pump heuristic adapted for the MWTR problem is described in Algorithm 1. We use a solver to obtain the LP solutions in line 1, in line 3, and iteratively in line 11.

In the first step, we obtain an LP solution to initialize the value of the variables (in line 4). For that, we solve twice the LP relaxation D-MWTR (25) as follows. First, we obtain the optimal LP solution. After, we fix to zero the topology variables x_{ij} with a value close to zero and then we solve the corresponding LP relaxation. This initialization of variables is specific to the MWTR problem.

Algorithm 1 Feasibility Pump heuristic for the MWTR problem (FP)

Require: problem data: sets $V = V_a \cup V_t$; distance matrix D; number of leaves $n = |V_t|$; procedure parameters: maxtime; maxiter.

solve the LP relaxation D-MWTR
fix to zero the variables x_{ij} such that $x_{ij} < 0.1$
solve (again) the LP relaxation D-MWTR (now with some variables fixed to zero)
let $\hat{y}$ be the optimal solution of the problem solved in the previous step
if $\hat{y}$ is integer **then**
    return $\hat{y}$ is a feasible integer solution of the MWTR problem
else
    $t \leftarrow 0$
    $\tilde{y}^t \leftarrow round(\hat{y})$
    while time $<$ maxtime and $t <$ maxiter **do**
        get $\hat{y}$, optimal solution of LP relaxation D-MWTR with $\Delta(y, \tilde{y}^t)$ as objective function
        let $\Delta(\hat{y}, \tilde{y}^t)$ be its optimal value
        if $\Delta(\hat{y}, \tilde{y}^t) = 0$ **then**
            return $\hat{y}$ is an integer feasible solution for the MWTR problem
        else
            $t \leftarrow t + 1$
            $\tilde{y}^t \leftarrow round(\hat{y})$
            if $\tilde{y}^t = \tilde{y}^{t-1}$ **then**
                fix to zero the components x_{ij} of $\tilde{y}^t$ such that $\hat{x}_{ij} < 0.1$, or $0.5 < \hat{x}_{ij} < 0.9$
                fix to one the remaining components x_{ij} of $\tilde{y}^t$
            end if
            for all $\ell = 0, \ldots, t-2$ **do**
                if $\tilde{y}^t = \tilde{y}^\ell$ **or** $\Delta(y, \tilde{y}^t) > 0.9 * \Delta(y, \tilde{y}^{t-2})$ **then**
                    for all $i \in \mathcal{I}$ **do**
                        $\rho_i = random(-0.3, 0.7)$
                        if $|\hat{y}_i - \tilde{y}_i^t| + \max\{\rho_i, 0\} > 0.5$ **then**
                            flip variables f and p components of $\tilde{y}^t$
                        end if
                    end for
                end if
            end for
        end if
    end while
end if
return the solution $\hat{y}$ and the integer solution $\tilde{y}^t$

The second step is the rounding and fixing variables value step in line 9. The obtained rounding vector $\tilde{y}$ is integer and, in general, $\tilde{y}$ is not a feasible solution. We fix the variables value, temporarily, to this integer value.

In the third step, the aim is to obtain the closest feasible solution to vector $\tilde{y}$. It is the so-called pumping-cycle that runs from line 10 to line 33 until either a feasible integer solution is found or the time limit, maxtime, or the maximum number of iterations, maxiter, is exceeded. In these two last cases, we do not obtain an integer solution.

To obtain the closest feasible solution to vector $\tilde{y}$, consider the distance function $\Delta(y, \tilde{y}) := \sum_{i \in S}(1 - y_i) + \sum_{i \in \overline{S}} y_i$. Hence, given an integer $\tilde{y}$, the closest vector $y \in \mathcal{P}_L$ can be determined by minimizing the value of the distance function $\Delta(y, \tilde{y})$ as follows

$$\text{(D-MWTR):} \qquad \begin{array}{rl} \min & \Delta(y, \tilde{y}) \\ \text{s.t.} & y \in \mathcal{P}_L. \end{array} \tag{25}$$

Let $\hat{y}$ be the optimal solution of the LP problem D-MWTR (25), and let $\Delta(\hat{y}, \tilde{y})$ be its optimal value. The vector $\hat{y}$ is the closest solution to the integer $\tilde{y}$ and two cases may occur, either $\Delta(\hat{y}, \tilde{y}) = 0$ or $\Delta(\hat{y}, \tilde{y}) > 0$. If $\Delta(\hat{y}, \tilde{y}) = 0$, then $\hat{y} = \tilde{y}$ is an integer feasible solution for the MWTR problem. If $\Delta(\hat{y}, \tilde{y}) > 0$, then we obtain a new integer solution, $\tilde{\hat{y}}$, by rounding $\hat{y}$, and two cases may occur, either $\tilde{\hat{y}} \neq \tilde{y}$ or $\tilde{\hat{y}} = \tilde{y}$. The FP heuristic may experience stalling and cycle problems and, in the specific case of the MWTR, we apply a perturbation mechanism [23].

When we obtain the same solution, $\tilde{\hat{y}} = \tilde{y}$, a stalling problem occurs. To solve this problem we apply a perturbation mechanism in lines 19 and 20, which switches some rounded to zero to one and some rounded to one to zero. Here only components x are switched and the remaining components f and p are not changed.

When we obtain a different vector, $\tilde{\hat{y}} \neq \tilde{y}$, two other cases may occur. Either $\tilde{\hat{y}}$ is different from any previously obtained vector, this is the desirable case, or $\tilde{\hat{y}}$ is equal to some previously obtained vector thus a cycle occurs. When a cycle is detected the same sequence of LP solutions and rounded vectors are obtained. In the first case, the iterative process continues by obtaining a new solution closest to the new integer vector $\tilde{\hat{y}}$ by solving the D-MWTR problem again. To avoid cycling, we apply a perturbation mechanism in line 27 to the integer solution $\tilde{y}$ that consists of modifying some randomly chosen components of the current integer solution $\tilde{y}$. That is, for a given parameter $\delta > 0$, for all $i \in \mathcal{I}$, we modify component $\tilde{y}_i$ of $\tilde{y}$ when $|y_i - \tilde{y}_i| + max\{\rho_i, 0\} > \delta$ with ρ_i randomly selected in $[-0.3, 0.7]$. Notice that this perturbation mechanism gives the possibility to modify the variables such that $|y_i - \tilde{y}_i| = 0$. In the line 27 only components f and p are switched, the components x are not changed.

Additionally, the perturbation mechanism can also be performed when the value $\Delta(\hat{y}, \tilde{y}^t)$ does not decrease. This perturbation mechanism is performed in line 27 when the value $\Delta(\hat{y}, \tilde{y}^t)$ does not decrease for at least 10% in the last three iterations.

We remark that different sets of variables are selected to change in the perturbation mechanisms applied to the MWTR problem. While in lines 19 and 20 only components x are changed, in line 27 only components f and p are changed.

Iteratively, we perform the previous steps and update the pair $(\hat{y}, \tilde{y})$. The FP heuristic algorithm constructs two trajectories of solutions, hopefully convergent. One is formed by a sequence of solutions satisfying constraints (2)-(17) and (21)-(23), solutions $\hat{y}$, thus the returned solution is not necessarily integer. The other sequence is formed by integer solutions $\tilde{y}^t$ that may not satisfy constraints (2)-(17) and (21)-(23).

4 The Local Branching for the MWTR problem

The Local Branching algorithm is an exact method that searches for better solutions in the neighborhood of a feasible solution [24]. It turns into a matheuristic when we set a criterion to stop it running before exhaustively searching in all the neighborhoods.

Algorithm 2 Local Branching scheme for the MWTR problem (LB)

Require: problem data: sets $V = V_a \cup V_t$; distance matrix D; number of leaves $n = |V_t|$; procedure parameters: k; maxtime; maxiter.

1: $P^0 \leftarrow$ MWTR problem
2: get a feasible solution $\tilde{y}$, the reference solution, to problem P^0
3: $t \leftarrow 1$
4: $\tilde{y}^t \leftarrow \tilde{y}$
5: **while** ((time $<$ maxtime) and ($t <$ maxiter)) **do**
6:     assign the value of k or adjust (apply intensification mechanism)
7:     $P^t \leftarrow$ introduce constraint $\Lambda(y, \tilde{y}^t) \leq k$ in problem P^{t-1}
8:     solve problem P^t and let $\tilde{y}$ be its optimal solution in neighborhood $\mathcal{N}(\tilde{y}^t, k)$
9:     **if** $f(\tilde{y}) < f(\tilde{y}^t)$ **then**
10:         $t \leftarrow t + 1$
11:         $\tilde{y}^t \leftarrow \tilde{y}$
12:     **else**
13:         $P^t \leftarrow$ introduce constraint $\Lambda(y, \tilde{y}^t) \geq k + 1$ in problem P^{t-1}
14:         solve problem P^t and let $\tilde{y}$ be its optimal solution in neighborhood $\mathcal{N}_+(\tilde{y}^t, k)$
15:         **if** $f(\tilde{y}) < f(\tilde{y}^t)$ **then**
16:             $t \leftarrow t + 1$
17:             $\tilde{y}^t \leftarrow \tilde{y}$
18:         **else**
19:             adjust the value of k (apply diversification mechanism)
20:         **end if**
21:     **end if**
22: **end while**
23: **return** integer solution $\tilde{y}^t$

Algorithm 2 displays a brief description of the Local Branching scheme applied to the MWTR problem. We use a solver in lines 8 and 14 to obtain a sequence of solutions $\tilde{y}^t$ in reduced solution spaces and with a decreasing sequence of costs.

In line 2 of Algorithm 2 we obtain the first feasible integer solution $\tilde{y}$ ($\equiv \tilde{y}^1$), which is taken as a reference solution. We can use a feasible solution $\tilde{y}$ to the MWTR problem previously obtained. Thus we can use the solution obtained by the FP heuristic.

An integer feasible solution of the MWTR problem corresponds to a spanning tree $T_{\tilde{y}}$ with cost $f(\tilde{y})$ and leaves V_t. Define two sets, set $S = \{i \in \mathcal{I} : \tilde{y}_i = 1\}$ and its complement set $\overline{S} = \{i \in \mathcal{I} : \tilde{y}_i = 0\}$. For a given positive integer parameter k', the *neighborhood* of $\tilde{y}$ is the set of feasible solutions of the MWTR problem satisfying the additional Local Branching constraint $\sum_{i \in S}(1 - y_i) + \sum_{i \in \overline{S}} y_i \leq k'$.This linear constraint limits to k' the total number of binary variables flipping their value with respect to the solution $\tilde{y}$, either from 1 to 0 or from 0 to 1. For every feasible solution to the MWTR problem, the cardinality of the set S is constant and equal to the number of edges of the corresponding feasible tree $T_{\tilde{y}}$. Further, the number of variables changing from 1 to 0 must be equal to the number of variables changing from 0 to 1. Thus the local branching constraint may assume the asymmetric form:

$$\Lambda(y, \tilde{y}) = \sum_{i \in S}(1 - y_i) \leq k \tag{26}$$

with $k = \frac{k'}{2}$. Define the *neighborhood* $\mathcal{N}(\tilde{y}, k)$ of $\tilde{y}$ as the set of feasible solutions of the MWTR problem satisfying the additional Local Branching constraint $\Lambda(y, \tilde{y}) \leq k$, and the *neighborhood* $\mathcal{N}_+(\tilde{y}, k)$ of $\tilde{y}$ as the set of feasible solutions of the MWTR problem satisfying the additional Local Branching constraint $\Lambda(y, \tilde{y}) \geq k + 1$. The choice of the size of the neighborhoods given by the parameter k is a problem that depends on the size and structure of the instances used. On one hand, the k must be large enough so that the neighborhood $\mathcal{N}(\tilde{y}, k)$ contains better-valued solutions than $\tilde{y}$ and, on the other hand, the k should be small enough to ensure that the neighborhood $\mathcal{N}(\tilde{y}, k)$ is quickly explored. Note that neighborhood $\mathcal{N}(\tilde{y}, k)$ of $\tilde{y}$ has solutions similar to $\tilde{y}$ and neighborhood $\mathcal{N}_{+}(\tilde{y}, k)$ contains solutions that differ from $\tilde{y}$ in more than $2\times(k+1)$ variables. When using as a heuristic, we explore the neighborhood $\mathcal{N}_{+}(\tilde{y}, k)$ only when a feasible solution better valued than $\tilde{y}$ is not found in the neighborhood $\mathcal{N}(\tilde{y}, k)$.

Depending on the size of the neighborhood that is being explored, finding the exact solution to the problem within that neighborhood can be very time-consuming. In case the time limit is exceeded, the obtained solution $\tilde{y}$ may not be the optimal solution, since the neighborhood is not fully explored, hence that neighborhood cannot be excluded. In that case, and depending on the solution obtained so far, the size of the neighborhood to be explored can be modified to either reduce or enlarge the region where the solution is sought. We observe that both the initial value of the parameter k and the mechanisms for its successive changes are very important issues for the success of the search. Also, they are specific to the problem and therefore require careful tunning. The mechanisms [24] that modify the size of the neighborhood are used in lines 6 and 19 of Algorithm 2 and are described next.

Intensification mechanism. The intensification mechanism aims to reduce the size of the neighborhood in an attempt to speed-up its exploration. The right hand side of the constraint $\Lambda(y, \tilde{y}) \leq k$ is reduced to $\lfloor \frac{k}{2} \rfloor$. This mechanism is applied to accelerate the neighborhood exploration when no improved feasible solution is found.

Diversification mechanism. The diversification mechanism aims to enlarge the size of the neighborhood. However, the exploration time is, consequently, also increased. First, a "soft" diversification mechanism is applied, in which the right hand side of the constraint $\Lambda(y, \tilde{y}) \leq k$ is increased by $\lceil \frac{k}{2} \rceil$, i.e., the constraint $\Lambda(y, \tilde{y}) \leq k + \lceil \frac{k}{2} \rceil$ is introduced. In case an improved solution is not found, a "strong" diversification mechanism is applied, in which the right hand side of the constraint $\Lambda(y, \tilde{y}) \leq k$ is increased with $2 \times \lceil \frac{k}{2} \rceil$, i.e., the constraint $\Lambda(y, \tilde{y}) \leq k + 2\lceil \frac{k}{2} \rceil$ is introduced. This exploration should be aborted as soon as the first solution is found.

When a time limit is exceeded and the obtained solution $\tilde{y}$ is not the optimal solution the following cases may occur. (i) The obtained solution $\tilde{y}$ has an improved value. In this case, the reference solution is updated, but the value of the parameter k is not modified. (ii) The obtained solution $\tilde{y}$ does not have an improved value, $f(\tilde{y}) > f(\tilde{y}^t)$. In this case, the intensification mechanism is applied to reduce the neighborhood. If again an improved solution is not found, apply a "weak" diversification mechanism. (iii) The obtained solution $\tilde{y}$ is infeasible. In this case, the "strong" diversification mechanism is applied to enlarge the neighborhood.

5 Computational Experiments

In this section, we report on the results of the computational tests of the FP and of the LB heuristics when applied to instances of the MWTR problem, whose size (in terms of leaves) varies between 8 and 20, for a total of 245 instances.

The computational tests have been performed on an Intel(R) Core(TM) i7-3770 CPU 3.40 GHz processor and 16Gb of RAM. The matheuristics using the Path-edges$^+$ formulation have been implemented by using the Mosel language and solved with FICO Xpress 7.8 [1] (Xpress-IVE 1.24.06 64 bit, Xpress-Optimizer 27.01.02 and Xpress-Mosel 3.8.0) with its default parameters, namely multi-threads. We have compared the performance of the matheuristics with the results presented in [25] and obtained by the solver with the Path-edges$^+$ formulation.

Two sets of data instances have been used, one set coming from a phylogenetics application, and the other one from a networking application. The first set of instances [9] is available from https://pubsonline.informs.org/doi/suppl/10.1287/ijoc.1110.0455. From this set we have used three phylogenetic distance matrices, matrices M391, Primate and M887, with $t = 17$, $t = 12$ and $t = 18$ taxa, respectively, and for each we have varied the number of leaves (taxa) between 8 and t, thus obtaining 26 instances. The data for the second set of instances were generated using the network-level simulator NS-3 (Network Simulator NS-3, http://www.nsnam.org/). We have performed four simulations named S15, S20, SS15, SS20 with $t = 15$, $t = 20$, $t = 12$ and $t = 12$ leaves, varying the number of leaves between 8 and t, thus obtaining 31 instances. We have, also, generated matrices with random numbers, using the values of the matrices: M391, Primate, M887, S15, and S20. For each matrix, $D = (d_{ij})$, we have generated ten random values belonging to $[d_{ij}, d_{ij} + a \times d_{ij}]$, where $a \in \{0.1; 0.15; 0.2; 1\}$. Then we have used the mean of the ten numbers to construct a new matrix. The names of the new matrices start with A10, A15, A20, and A100 when the value used for a is 0.1, 0.15, 0.20, and 1, respectively. Thus, we have obtained 188 more instances, making a total of 245 instances.

The proposed matheuristics use the Path-edges$^+$ formulation to obtain feasible solutions to the MWTR problem. In [25] it is reported that Path-edges$^+$ formulation performs better (obtains good feasible solutions in less time) than other models presented in the literature. To obtain a high-quality feasible solution in less time, several tests were performed by applying the matheuristics to the model without the valid equalities and inequalities (21)-(23) or by using only some combinations of the equalities and inequalities (21)-(23). On one hand, comparing the results obtained with the FP heuristic by several combinations, we concluded that by using some of these combinations we obtained a feasible solution in less time than the one that uses the Path-edges$^+$ formulation, but the quality of the obtained feasible solution was quite poor. Since the performance of the LB heuristic depends highly on the first feasible solution used, the Path-edges$^+$ formulation is used in the FP heuristic to obtain the computational results. The time gained when using some of these combinations did not compensate for the poor quality of the solution obtained, since the time obtained in the FP heuristic would be spent executing the LB heuristic procedure to improve the feasible solution obtained. To have an idea of the size of the instances Table 1 displays the number of variables $\frac{1}{2}n^4 + \frac{1}{2}n^3 - \frac{11}{2}n^2 + \frac{15}{2}n - 3$ and constraints $\frac{3}{2}n^4 - 5n^3 + 7n^2 + \frac{1}{2}n - 1 + \lceil \frac{n}{2} \rceil$ for each value n of leaves of the instances considered.

Table 1: Number of variables and constraints of formulation Path-edges$^+$ for each value of n.

n	8	9	10	11	12	13	14	15	16	17	18	19	20
number of variables	2009	3264	5022	7400	10527	14544	19604	25872	33525	42752	53754	66744	81947
number of contraints	4039	6772	10709	16164	23483	33052	45289	60652	79631	102756	130589	163732	202819

To quickly explore the feasible region, the appropriate size of the neighborhoods defined by inequality (26) must be carefully chosen. After several computational experiments, we concluded that the choice of the value k for the right-hand side of the Local Branching constraint depends on the value of n, the size of the instance. When the value of n is small, the value of k can also be small and the LB heuristic is still capable of finding an improved feasible solution within the considered neighborhood. However, for larger values of n, to be able to find an improved feasible solution in the considered neighborhood, the value of k must be larger. Nevertheless, since the time spent to find a feasible solution increases substantially for large values of n and a time limit is imposed, the value of k must be reduced in these cases. Thus, the values used for k were $k = 3$ for $n < 12$, $k = 5$ for $12 \leq n \leq 15$ and $k = 4$ for $n \geq 16$.

Figure 2 shows the average time (in seconds) and Figure 3 shows the average GAP (%) for instances with the same number of leaves for the FP, the LB heuristic and the Path-edges$^+$ formulation. Noticeably, we observe that the LB heuristic significantly improves the feasible solution obtained by the FP heuristic for all the instances.

The computational results are summarized in Tables 2–5 in which the first column, labeled **M**, refers to the name of the matrix instance used and the second column, labeled $|V_t|$, shows the size of the instance. The third, the forth and the fifth columns concern the results relative to the FP heuristic; the sixth to the ninth columns refer to the results relative to the LB heuristic, and additionally, the tenth, the eleventh and the twelfth columns present the results relative to the Path-edges$^+$ formulation. The columns labeled **T** show the execution time, in seconds, used to solve the instance and having a maximum runtime of 1200 seconds for the FP heuristic, a maximum runtime of 7000 seconds for the LB heuristic and the Path-edges$^+$ formulation a maximum

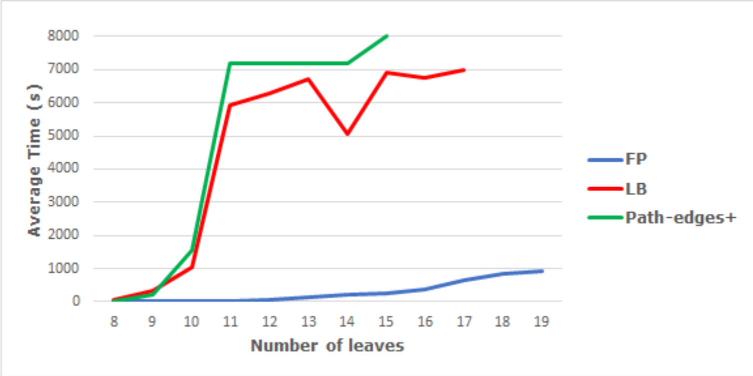


Figure 2: Average time (s) of the Path-edges$^+$ formulation, the FP and the LB heuristic.

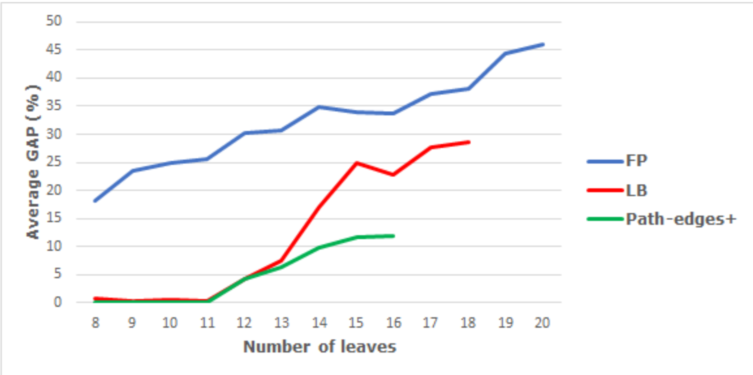


Figure 3: Average Gap (%) of the Path-edges$^+$ formulation, the FP and the LB heuristic.

runtime of 7200 when $n \leq 15$ and a maximum runtime of 8000 when $n \geq 16$. Additionally, a time limit was imposed for solving each subproblem of the LB heuristic. It is used a time limit of 1000 seconds when $n < 14$ and a time limit of 1400 seconds when $n \geq 14$. The columns labeled **W** present the optimum or the best value obtained for $\sum_{i \in V_a} \sum_{\substack{j \in V \\ j > i}} w_{ij}$ within the runtime limit and for a maximum of 1000 iterations. We only report values when within the maximum runtime imposed, the FP heuristic obtains a feasible solution, the LB heuristic obtains a better solution than the one obtained by the FP heuristic and the Path-edges$^+$ formulation obtains a feasible solution. The column labeled **TT** shows the total execution time, that is, the sum of the execution time of the FP heuristic plus the execution time of the LB heuristic. The columns labeled **GAP** present the gap between the value W obtained by the corresponding matheuristic or model and the best lower bound value known:

$$GAP = \frac{W - BLB}{W} \times 100$$

where W represents the best value obtained by the matheuristic or model within the runtime imposed and BLB represents the best known lower bound value obtained with the solver using the Path-edges$^+$ formulation.

Tables 2 and 3 refer to results concerning data from the phylogenetics application reporting the computational results obtained for 130 instances and Tables 4 and 5 refer to results concerning data from the networking application reporting the computational results obtained for 115 networking instances.

For the phylogenetic data, the computational time that the FP heuristic uses to obtain a feasible solution is less than 200 seconds in 100 out of 130 instances, this is, in approximately 76.9% of the instances. The FP heuristic finds a feasible solution within the runtime imposed for all instances, except for matrix A15M391 with $n = 17$. Therefore, we did not run the LB heuristic for matrix A15M91 with $n = 17$.

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

M	$\|V_t\|$	FP heuristic T (s)	FP heuristic W	FP heuristic GAP (%)	LB heuristic T (s)	LB heuristic TT (s)	LB heuristic W	LB heuristic GAP (%)	Path-edges$^+$ T (s)	Path-edges$^+$ W	Path-edges$^+$ GAP (%)
M391	8	0.41	0.0765	8.8	13.46	13.87	0.0705	1.0	**6.60**	**0.0698**	**0.0**
	9	0.86	0.1014	6.2	56.93	57.79	0.0951	0.0	**20.33**	**0.0951**	**0.0**
	10	12.04	0.1233	13.4	**262.47**	**274.51**	**0.1068**	**0.0**	735.92	0.1068	0.0
	11	10.06	0.1533	12.8	**1073.05**	**1083.11**	**0.1337**	**0.0**	2690.60	0.1337	0.0
	12	22.17	0.1619	18.3	**6908.38**	**6930.55**	**0.1378**	**4.1**	7200.00	0.1378	4.1
	13	193.66	0.1809	21.5	7000.00	7193.66	0.1557	8.8	**7200.00**	**0.1523**	**6.8**
	14	77.45	0.1901	22.7	7000.00	7077.45	0.1684	12.8	**7200.00**	**0.1562**	**6.0**
	15	104.04	0.1926	22.0	3000.06	3104.10	0.1926	22.0	**7200.00**	**0.1647**	**8.8**
	16	403.84	0.2121	27.6	**7000.00**	**7403.84**	**0.1863**	**17.6**	8000.00	–	–
	17	453.53	0.2237	30.2	**7000.00**	**7453.53**	**0.1994**	**21.7**	8000.00	–	–
Primate	8	0.36	0.1667	23.2	9.44	9.80	0.128	0.0	**2.39**	**0.128**	**0.0**
	9	0.77	0.1970	34.2	76.60	77.37	0.1348	3.9	**17.96**	**0.1296**	**0.0**
	10	2.42	0.2154	39.7	362.14	364.56	0.1326	2.1	**126.75**	**0.1298**	**0.0**
	11	26.24	0.2903	41.4	**713.78**	**740.02**	**0.1700**	**0.0**	1391.36	0.1700	0.0
	12	12.87	0.3143	37.7	**6202.03**	**6214.90**	**0.2058**	**4.8**	7200.00	0.2058	4.8
M887	8	0.31	0.214	17.6	20.06	20.37	0.1763	0.0	**6.07**	**0.1763**	**0.0**
	9	1.12	0.2182	14.8	86.97	88.09	0.1860	0.0	**24.27**	**0.1860**	**0.0**
	10	3.98	0.2327	11.5	266.12	270.10	0.2078	0.9	**239.37**	**0.2059**	**0.0**
	11	7.55	0.2567	16.8	1493.73	1501.28	0.2147	0.6	**1975.92**	**0.2135**	**0.0**
	12	20.17	0.2754	28.5	**6000.17**	**6020.34**	**0.2179**	**9.7**	7200.00	0.2179	9.7
	13	17.21	0.2842	24.0	**7000.00**	**7017.21**	**0.2376**	**9.1**	7200.00	0.2380	9.2
	14	125.41	0.3151	27.5	7000.00	7125.41	0.2784	18.0	**7200.00**	**0.2548**	**10.2**
	15	235.01	0.3247	16.4	4501.68	4736.69	0.3050	11.0	**7200.00**	**0.3040**	**10.7**
	16	137.14	0.3445	31.7	**7000.00**	**7137.14**	**0.2891**	**18.6**	8000.00	–	–
	17	673.42	0.3432	30.6	**7000.00**	**7.673.42**	**0.3102**	**23.2**	8000.00	–	–
	18	746.71	0.3728	34.7	**7000.00**	**7746.71**	**0.3440**	**29.2**	8000.00	–	–
A10M391	8	0.61	0.0829	11.5	17.11	17.72	0.0735	0.1	**14.87**	**0.0734**	**0.0**
	9	1.15	0.1111	10.2	55.43	56.58	0.0998	0.0	**9.89**	**0.0998**	**0.0**
	10	2.81	0.1265	11.1	288.58	291.39	0.1124	0.0	**197.67**	**0.1124**	**0.0**
	11	6.93	0.1628	13.4	1368.09	1375.02	0.1410	0.0	**1197.99**	**0.1410**	**0.0**
	12	9.22	0.1579	12.7	**5445.05**	**5454.27**	**0.1454**	**5.2**	7200.00	0.1461	5.6
	13	27.05	0.1826	18.2	**6414.28**	**6441.33**	**0.1586**	**5.8**	7200.00	0.1595	6.3
	14	72.17	0.2008	21.8	7000.00	7072.17	0.1808	13.1	**7200.00**	**0.1670**	**5.9**
	15	464.77	0.2023	20.3	**6002.45**	**6467.22**	**0.1786**	**9.7**	7200.00	0.1838	12.2
	16	359.96	0.2212	27.0	**7000.00**	**7359.96**	**0.1980**	**18.4**	8000.00	–	–
	17	426.72	0.2272	27.9	**7000.00**	**7426.72**	**0.2060**	**20.5**	8000.00	–	–
A10Pri	8	0.25	0.1541	12.3	22.14	22.39	0.1351	0.0	**1.89**	**0.1351**	**0.0**
	9	0.83	0.1958	30.3	78.36	79.19	0.1397	2.4	**12.93**	**0.1364**	**0.0**
	10	2.68	0.2482	45.2	280.02	282.7	0.1404	3.1	**138.33**	**0.1361**	**0.0**
	11	3.87	0.2741	34.5	1163.34	1167.21	0.1794	0.0	**574.74**	**0.1794**	**0.0**
	12	7.44	0.2632	21.7	**4751.00**	**4758.44**	**0.2162**	**4.6**	7200.00	0.2162	4.6
A10M887	8	0.33	0.2078	10.6	24.12	24.45	0.1858	0.0	**9.45**	**0.1858**	**0.0**
	9	3.25	0.2270	13.1	62.79	66.04	0.1972	0.0	**9.23**	**0.1972**	**0.0**
	10	9.02	0.2361	8.7	**224.59**	**233.61**	**0.2156**	**0.0**	276.40	0.2156	0.0
	11	6.55	0.2834	20.9	**1352.80**	**1359.35**	**0.2243**	**0.0**	2253.32	0.2243	0.0
	12	10.72	0.2631	21.4	**4893.51**	**4904.23**	**0.2243**	**7.8**	7200.00	0.2290	9.7
	13	62.03	0.3005	26.7	3999.60	4061.63	0.2537	13.1	**7200.00**	**0.2454**	**10.2**
	14	318.93	0.3345	32.9	7000.00	7318.93	0.2755	18.5	**7200.00**	**0.2668**	**15.9**
	15	273.53	0.3510	27.2	6001.81	6275.34	0.3208	20.4	**7200.00**	**0.2857**	**10.6**
	16	442.87	0.3842	35.7	7000.00	7442.87	0.3304	25.2	**8000.00**	**0.2872**	**13.9**
	17	429.41	0.4107	39.1	**7000.00**	**7429.41**	**0.3374**	**25.9**	8000.00	–	–
	18	**326.20**	**0.3265**	**21.8**	7000.00	7326.20	–	–	8000.00	–	–
A15M391	8	0.37	0.0851	11.6	15.71	16.08	0.0755	0.4	**16.43**	**0.0752**	**0.0**
	9	1.23	0.1213	15.8	69.97	71.20	0.1021	0.0	**13.37**	**0.1021**	**0.0**
	10	2.34	0.1305	11.6	448.89	451.23	0.1153	0.0	**211.55**	**0.1153**	**0.0**
	11	4.79	0.1645	12.9	1134.28	1139.07	0.1447	1.0	**2032.65**	**0.1433**	**0.0**
	12	34.12	0.1736	17.7	**6973.20**	**7007.32**	**0.1481**	**3.5**	7200.00	0.1492	4.2
	13	28.16	0.2013	23.9	6314.64	6342.80	0.1647	7.0	**7200.00**	**0.1642**	**6.8**
	14	82.48	0.1999	22.1	4200.82	4283.30	0.1999	22.1	**7200.00**	**0.1713**	**9.1**
	15	112.52	0.2102	23.0	2999.48	3112.00	0.2102	23.0	**7200.00**	**0.1799**	**10.0**
	16	401.19	0.2282	27.5	**6136.33**	**6537.52**	**0.2101**	**21.2**	8000.00	–	–
	17	1200.00	–	–	–	–	–	–	8000.00	–	–
A15Pri	8	0.33	0.1775	21.9	21.67	220.00	0.1386	0.0	**2.21**	**0.1386**	**0.0**
	9	0.91	0.2095	33.3	139.35	140.26	0.1397	0.0	**12.68**	**0.1397**	**0.0**
	10	2.62	0.2302	39.5	288.90	291.52	0.1439	3.2	**185.83**	**0.1393**	**0.0**
	11	4.77	0.2763	33.4	921.07	925.84	0.1841	0.0	**322.87**	**0.1841**	**0.0**
	12	10.01	0.3348	36.9	**6557.32**	**6567.33**	**0.2214**	**4.6**	7200.00	0.2214	4.6
A15M887	8	0.30	0.2081	8.5	15.15	15.45	0.1951	2.4	**11.61**	**0.1905**	**0.0**
	9	0.95	0.2257	10.5	35.96	36.91	0.2035	0.7	**9.70**	**0.2020**	**0.0**
	10	4.01	0.2637	16.4	368.27	372.28	0.2237	1.4	**128.84**	**0.2205**	**0.0**
	11	5.85	0.2840	19.1	**796.55**	**802.40**	**0.2297**	**0.0**	1881.22	0.2297	0.0
	12	15.69	0.2966	28.7	**6332.15**	**6347.84**	**0.2345**	**9.8**	7200.00	–	–
	13	24.52	0.3199	30.1	7000.00	7024.52	0.2584	13.5	**7200.00**	**0.2527**	**11.5**
	14	68.05	0.3249	27.3	7000.00	7068.05	0.3021	21.8	**7200.00**	**0.2714**	**12.9**
	15	68.58	0.3509	29.7	3000.63	3069.21	0.3509	29.7	**7200.00**	**0.3050**	**19.1**
	16	248.62	0.3741	32.3	**7000.00**	**7248.62**	**0.3156**	**19.8**	8000.00	–	–
	17	507.87	0.4128	38.0	**7000.00**	**7507.87**	**0.3625**	**29.4**	8000.00	–	–
	18	1200.00	0.4166	37.2	**7000.00**	**8000.00**	**0.3518**	**25.7**	8000.00	–	–

The LB heuristic obtains the same value as Path-edges$^+$ formulation in 46 out of the 129 instances, corresponding to approximately 35.7%. Contrarily, for the instances not completely solved by the Path-edges$^+$ formulation, using the LB heuristic we obtain better values in 35 out of the 67 instances, corresponding to approximately 52.2%. These better values were obtained

Table 3: Computational results for data from the phylogenetics application (continuation).

		FP heuristic			LB heuristic				Path-edges$^+$		
M	$\lvert V_t \rvert$	**T** (s)	**W**	**GAP** (%)	**T** (s)	**TT** (s)	**W**	**GAP** (%)	**T** (s)	**W**	**GAP** (%)
A20M391	8	0.55	0.0908	15.2	23.06	23.61	0.0770	0.0	**11.59**	**0.0770**	**0.0**
	9	1.09	0.1193	12.5	139.81	140.90	0.1048	0.4	**10.39**	**0.1044**	**0.0**
	10	5.05	0.1387	14.9	313.92	318.97	0.1181	0.0	**274.01**	**0.1181**	**0.0**
	11	7.80	0.1667	12.0	**1672.18**	**1679.98**	**0.1467**	**0.0**	1793.47	0.1467	0.0
	12	13.85	0.1735	16.6	**7000.00**	**7013.85**	**0.1514**	**4.4**	7200.00	0.1515	4.4
	13	63.21	0.2013	22.1	6113.15	6176.36	0.1715	8.6	**7200.00**	**0.1686**	**7.0**
	14	131.56	0.2072	19.7	7000.00	7131.56	0.1914	13.1	**7200.00**	**0.1716**	**7.2**
	15	159.84	0.2233	25.8	3000.88	3160.72	0.2233	25.8	**7200.00**	**0.1883**	**12.0**
	16	249.02	0.2369	28.5	**7000.00**	**7249.02**	**0.2080**	**28.5**	8000.00	–	–
	17	485.08	0.2487	31.1	**7000.00**	**7485.08**	**0.2299**	**25.5**	8000.00	–	–
A20Pri	8	0.33	0.2119	32.9	22.53	22.86	0.1461	2.7	**0.92**	**0.1422**	**0.0**
	9	2.79	0.1905	24.9	118.05	120.84	0.1431	0.0	**9.08**	**0.1431**	**0.0**
	10	5.63	0.2126	32.9	279.21	284.84	0.1474	3.2	**119.22**	**0.1427**	**0.0**
	11	8.08	0.2824	33.1	793.28	801.36	0.1889	0.0	**569.43**	**0.1889**	**0.0**
	12	8.44	0.3469	37.6	**6753.36**	**6761.80**	**0.2265**	**4.5**	7200.00	0.2265	4.5
A20M887	8	0.53	0.2284	14.5	16.66	17.19	0.1998	2.3	**11.89**	**0.1953**	**0.0**
	9	1.34	0.2493	17.1	34.54	35.88	0.2067	0.0	**9.02**	**0.2067**	**0.0**
	10	6.13	0.2705	16.7	176.44	182.57	0.2253	0.0	**120.14**	**0.2253**	**0.0**
	11	14.74	0.2815	16.4	1371.93	1386.67	0.2352	0.0	**2079.15**	**0.2352**	**0.0**
	12	23.90	0.3000	27.8	**3501.27**	**3525.17**	**0.2400**	**9.8**	7200.00	–	–
	13	35.87	0.3124	26.4	**4999.56**	**5035.43**	**0.2593**	**11.3**	7200.00	0.2639	12.9
	14	57.53	0.3363	27.7	5601.28	5658.81	0.2955	17.7	**7200.00**	**0.2728**	**10.9**
	15	64.55	0.3628	28.9	**4500.37**	**4564.92**	**0.3107**	**17.0**	7200.00	0.3110	17.0
	16	241.58	0.3954	34.5	**7000.00**	**7241.58**	**0.3390**	**23.6**	8000.00	–	–
	17	255.33	0.4375	40.2	**7000.00**	**7255.33**	**0.3756**	**30.4**	8000.00	–	–
	18	884.15	0.4255	37.1	**7000.00**	**7884.15**	**0.3799**	**29.6**	8000.00	–	–
A100M391	8	0.41	0.1230	15.1	10.22	10.63	0.1044	0.0	**9.64**	**0.1044**	**0.0**
	9	1.09	0.1721	16.9	53.45	54.54	0.1430	0.0	**13.60**	**0.1430**	**0.0**
	10	5.84	0.1850	11.4	432.61	438.45	0.1640	0.0	**279.43**	**0.1640**	**0.0**
	11	7.07	0.2254	9.0	891.78	898.85	0.2058	0.3	**3594.00**	**0.2052**	**0.0**
	12	12.29	0.2456	19.1	7000.00	7012.29	0.2106	5.7	**7200.00**	**0.2097**	**5.3**
	13	42.28	0.2792	23.4	**7000.00**	**7042.28**	**0.2304**	**7.1**	7200.00	0.2358	9.2
	14	151.32	0.2743	20.5	7000.00	7151.32	0.2550	14.4	**7200.00**	**0.2369**	**7.9**
	15	526.22	0.3075	26.2	2999.68	3525.90	0.3075	25.2	**7200.00**	**0.2616**	**13.2**
	16	284.37	0.3263	29.4	**7000.00**	**7284.37**	**0.2818**	**18.3**	8000.00	–	–
	17	399.88	0.3227	28.3	**4034.53**	**4434.41**	**0.2747**	**15.8**	8000.00	–	–
A100Pri	8	0.33	0.2907	31.5	11.9	12.23	0.2048	2.8	**1.93**	**0.1991**	**0.0**
	9	1.51	0.3454	41.1	94.24	95.75	0.2053	0.9	**9.77**	**0.2035**	**0.0**
	10	2.82	0.3642	46.0	347.08	349.90	0.2034	3.3	**94.63**	**0.1966**	**0.0**
	11	4.23	0.3741	29.0	1435.08	1439.31	0.2655	0.0	**1008.31**	**0.2655**	**0.0**
	12	18.24	0.4084	27.6	**4991.84**	**5010.08**	**0.3166**	**6.6**	7200.00	0.3166	6.6
A100M887	8	0.59	0.3190	14.9	11.56	12.15	0.2714	0.0	**2.75**	**0.2714**	**0.0**
	9	1.15	0.3262	13.2	35.52	36.67	0.2833	0.0	**8.08**	**0.2833**	**0.0**
	10	2.53	0.3611	15.1	**138.28**	**140.81**	**0.3064**	**0.0**	144.63	0.3064	0.0
	11	5.66	0.4070	21.9	**760.88**	**766.54**	**0.3177**	**0.0**	3532.92	0.3177	0.0
	12	18.02	0.4312	31.5	**6893.39**	**6911.41**	**0.3265**	**9.6**	7200.00	0.3265	9.6
	13	71.48	0.4541	31.4	**7000.00**	**7071.48**	**0.3579**	**12.9**	7200.00	0.3604	13.5
	14	224.58	0.4650	28.6	7000.00	7224.58	0.4028	17.6	**7200.00**	**0.3759**	**11.7**
	15	484.61	0.5172	30.8	**4500.50**	**4985.11**	**0.4451**	**19.6**	7200.00	0.4460	19.8
	16	136.14	0.4917	28.4	**7000.00**	**7136.14**	**0.4235**	**16.9**	8000.00	–	–
	17	304.20	0.5796	39.0	**7000.00**	**7304.20**	**0.4747**	**25.5**	8000.00	–	–
	18	**462.54**	**0.5615**	**35.3**	7000.00	7462.54	–	–	8000.00	–	–

for some of the instances with $12 \leq n \leq 15$, the ones with high size, using less than 7200 seconds and for which the Path-edges$^+$ formulation is not able to complete the search within the time limit of 7200 seconds and for instances with $16 \leq n \leq 18$, using less than 8000 seconds, and for which the Path-edges$^+$ formulation is only able to find a feasible solution within the time limit of 8000 for the matrix A10M887 with $n = 16$. Therefore we remark that the LB heuristic makes it possible to significantly improve the values for instances with high size. Also, in 52 out of 130 instances, corresponding to 40% of the instances, the FP heuristic followed by the LB heuristic took less time to obtain the same or a better solution than the Path-edges$^+$ formulation. The instances A10M887 with $n = 18$ and A100M887 with $n = 18$ are the only instances where the LB heuristic does not obtain an improved solution within the time limit imposed.

Table 4: Computational results for data from the networking application.

M	$\lvert V_t \rvert$	FP heuristic T (s)	FP heuristic W	FP heuristic GAP (%)	LB heuristic T (s)	LB heuristic TT (s)	LB heuristic W	LB heuristic GAP (%)	Path-edges$^+$ T (s)	Path-edges$^+$ W	Path-edges$^+$ GAP (%)
S15	8	0.87	0.5103	26.0	12.51	13.38	0.3778	0.0	**9.78**	**0.3778**	**0.0**
	9	0.94	0.6896	43.0	45.40	46.34	0.3932	0.0	**13.51**	**0.3932**	**0.0**
	10	2.21	0.7145	40.0	333.79	336.00	0.4286	0.0	**173.63**	**0.4286**	**0.0**
	11	16.64	0.9018	30.6	**700.35**	**716.99**	**0.6260**	**0.0**	1658.17	0.6260	0.0
	12	34.89	1.2111	40.4	**4287.08**	**4321.97**	**0.7220**	**0.1**	7200.00	0.7220	0.1
	13	33.65	1.4488	49.3	**5999.47**	**6033.12**	**0.7474**	**1.8**	7200.00	0.7474	1.8
	14	324.75	1.6359	55.4	**7000.00**	**7324.75**	**0.8239**	**11.4**	7200.00	0.8742	16.5
	15	122.24	1.6105	45.7	7000.00	7122.24	1.1941	26.7	**7200.00**	**1.0420**	**16.0**
S20	8	0.44	0.5275	13.5	16.72	17.16	0.4564	0.0	**8.21**	**0.4564**	**0.0**
	9	0.78	0.6188	16.4	78.05	78.83	0.5174	0.0	**32.42**	**0.5174**	**0.0**
	10	4.98	0.7564	24.2	**246.53**	**251.51**	**0.5736**	**0.0**	453.74	0.5736	0.0
	11	5.12	0.9646	36.9	1194.12	1199.24	0.6193	1.7	**1668.14**	**0.6090**	**0.0**
	12	15.20	0.9903	37.3	4745.40	4760.60	0.6347	2.2	**7200.00**	**0.6244**	**0.6**
	13	15.05	0.8735	17.1	3999.40	4014.45	0.7515	3.6	**7200.00**	**0.7301**	**0.8**
	14	531.09	1.2439	37.5	7000.00	7531.09	0.8781	11.5	**7200.00**	**0.7967**	**2.5**
	15	199.66	1.3002	32.5	7000.00	7199.66	1.1071	20.5	**7200.00**	**0.9140**	**3.7**
	16	121.24	1.3148	32.9	6419.19	6540.43	1.1574	23.8	**8000.00**	**1.0011**	**11.9**
	17	258.48	1.6955	42.8	**7000.00**	**7258.48**	**1.4280**	**32.0**	8000.00	–	–
	18	1167.63	1.6959	40.0	**7000.00**	**8000.00**	**1.4539**	**30.1**	8000.00	–	–
	19	**1035.78**	**1.9243**	**46.4**	7000.00	8000.00	–	–	8000.00	–	–
	20	**1127.90**	**1.9348**	**45.1**	7000.00	8000.00	–	–	8000.00	–	–
A10S15	8	0.52	0.5282	24.5	29.77	30.29	0.4189	4.8	**5.74**	**0.3987**	**0.0**
	9	1.03	0.6521	36.4	56.61	57.64	0.4150	0.0	**9.86**	**0.4150**	**0.0**
	10	3.15	0.6368	28.9	205.55	208.70	0.4528	0.0	**170.62**	**0.4528**	**0.0**
	11	5.96	0.9739	32.1	1238.49	1244.45	0.6612	0.0	**895.27**	**0.6612**	**0.0**
	12	17.35	1.2981	41.6	**5530.32**	**5547.67**	**0.7634**	**0.6**	7200.00	0.7634	0.6
	13	21.64	1.3191	41.5	6400.75	6422.39	0.8002	3.5	**7200.00**	**0.7891**	**2.1**
	14	98.69	1.5418	48.2	5600.16	5698.85	1.2005	33.5	**7200.00**	**0.8276**	**3.5**
	15	116.70	1.7335	46.5	2800.59	2917.29	1.5422	39.9	**7200.00**	**1.1003**	**15.7**
A10S20	8	0.31	0.4934	2.6	22.57	22.88	0.4814	0.1	**11.54**	**0.4807**	**0.0**
	9	0.73	0.6208	12.1	49.83	50.56	0.5460	0.1	**12.90**	**0.5454**	**0.0**
	10	4.12	0.7748	21.9	736.18	740.30	0.6048	0.0	**206.08**	**0.6048**	**0.0**
	11	6.18	0.9102	29.4	1054.94	1061.12	0.6435	0.1	**3055.13**	**0.6429**	**0.0**
	12	42.26	1.0354	37.9	6938.02	6980.28	0.6603	2.7	**7200.00**	**0.6596**	**2.6**
	13	30.20	1.1687	35.8	**7000.00**	**7030.20**	**0.7811**	**4.0**	7200.00	0.7821	4.1
	14	86.55	1.3491	39.6	7000.00	7086.55	0.9182	11.3	**7200.00**	**0.8830**	**7.7**
	15	97.83	1.5686	41.7	6000.47	6098.30	1.2649	27.7	**7200.00**	**0.9413**	**2.8**
	16	397.65	1.7692	47.7	**7000.00**	**7397.65**	**1.3286**	**30.3**	8000.00	–	–
	17	174.72	1.6816	39.5	**7000.00**	**7174.72**	**1.5008**	**32.2**	8000.00	–	–
	18	**400.47**	**1.9230**	**44.5**	7000.00	7400.47	–	–	8000.00	–	–
	19	1200.00	–	–	–	–	–	–	8000.00	–	–
	20	1200.00	–	–	–	–	–	–	8000.00	–	–
A15S15	8	0.55	0.6086	32.8	29.64	30.19	0.4102	0.2	**3.79**	**0.4092**	**0.0**
	9	1.54	0.6710	36.5	69.70	71.24	0.4260	0.0	**10.37**	**0.4260**	**0.0**
	10	2.20	0.6635	30.0	384.48	386.68	0.4646	0.0	**119.43**	**0.4646**	**0.0**
	11	5.13	0.9557	29.0	**587.64**	**592.77**	**0.6788**	**0.0**	902.20	0.6788	0.0
	12	18.38	1.0718	27.5	**4821.25**	**4839.63**	**0.7841**	**0.9**	7200.00	0.7841	0.9
	13	35.65	1.3749	42.9	7000.00	7035.65	0.8528	8.0	**7200.00**	**0.8105**	**3.2**
	14	57.74	1.5761	48.7	7000.00	7057.74	1.0515	23.1	**7200.00**	**0.9138**	**11.5**
	15	154.92	1.8676	50.0	6000.55	6155.47	1.5214	38.6	**7200.00**	**1.0738**	**13.0**
A15S20	8	0.41	0.5697	13.5	9.95	10.36	0.4928	0.0	**7.96**	**0.4928**	**0.0**
	9	1.39	0.7036	20.5	67.17	68.56	0.5593	0.0	**11.75**	**0.5593**	**0.0**
	10	6.37	0.8080	23.2	217.65	224.02	0.6215	0.2	**179.48**	**0.6204**	**0.0**
	11	7.21	1.0445	36.8	738.29	745.50	0.6608	0.2	**1804.07**	**0.6598**	**0.0**
	12	28.14	1.0764	37.6	**6794.65**	**6822.79**	**0.6771**	**0.8**	7200.00	0.6771	0.8
	13	31.33	1.1518	32.9	5999.35	6030.68	0.8122	4.8	**7200.00**	**0.7917**	**2.4**
	14	42.68	1.1830	29.9	7000.00	7042.68	0.9105	8.9	**7200.00**	**0.8884**	**6.7**
	15	417.66	1.6028	38.9	6001.44	6419.10	1.4244	31.2	**7200.00**	**1.0100**	**3.0**
	16	145.24	1.4633	35.2	**7000.00**	**7145.24**	**1.3178**	**28.1**	8000.00	–	–
	17	520.93	1.8673	44.3	**6749.68**	**7270.61**	**1.5765**	**34.0**	8000.00	–	–
	18	1200.00	–	–	–	–	–	–	8000.00	–	–
	19	1200.00	–	–	–	–	–	–	8000.00	–	–
	20	1200.00	–	–	–	–	–	–	8000.00	–	–

For the networking application data, the computational times used by the FP heuristic are very small and most of the instances use less than 200 seconds. Only 24 out of 115 instances use more than 200 seconds to obtain a feasible solution. The FP heuristic finds a feasible solution

Table 5: Computational results for data from the networking application (continuation).

		FP heuristic			LB heuristic				Path-edges$^+$		
M	$\lvert V_t \rvert$	**T (s)**	**W**	**GAP** (%)	**T (s)**	**TT (s)**	**W**	**GAP** (%)	**T (s)**	**W**	**GAP** (%)
A20S15	8	0.53	0.6021	30.3	25.76	26.29	0.4197	0.0	**7.71**	**0.4197**	**0.0**
	9	0.62	0.7094	38.4	52.90	53.52	0.4369	0.0	**8.52**	**0.4369**	**0.0**
	10	2.82	0.6806	30.0	305.56	308.38	0.4766	0.0	**158.62**	**0.4766**	**0.0**
	11	2.61	0.8724	20.2	1594.18	1596.79	0.6964	0.0	**909.73**	**0.6964**	**0.0**
	12	22.85	1.3066	39.9	**6245.47**	**6268.32**	**0.8048**	**2.5**	7200.00	0.8048	2.5
	13	12.87	1.1160	28.4	7000.00	7012.87	0.8315	3.8	**7200.00**	**0.8307**	**3.7**
	14	201.43	1.6262	51.0	7000.00	7201.43	0.9661	17.5	**7200.00**	**1.0304**	**22.6**
	15	77.09	2.0582	53.3	7000.00	7077.09	1.5668	38.7	**7200.00**	**1.2272**	**21.7**
A20S20	8	0.47	0.5883	14.2	23.28	23.75	0.5049	0.0	**13.00**	**0.5049**	**0.0**
	9	1.44	0.7646	25.0	66.44	67.88	0.5733	0.0	**11.45**	**0.5733**	**0.0**
	10	21.89	0.8726	27.1	603.35	625.24	0.6359	0.0	**178.03**	**0.6359**	**0.0**
	11	9.52	0.9222	26.6	968.82	978.34	0.6871	1.5	**944.07**	**0.6768**	**0.0**
	12	7.89	1.0314	33.3	**6672.18**	**6680.07**	**0.6947**	**1.0**	7200.00	0.6947	1.0
	13	28.89	1.1722	32.8	4999.51	5028.40	0.8350	5.7	**7200.00**	**0.8120**	**3.0**
	14	62.65	1.3819	38.6	7000.00	7062.65	0.9934	14.5	**7200.00**	**0.8989**	**5.5**
	15	126.83	1.4517	31.6	6002.27	6129.10	1.2953	23.3	**7200.00**	**0.9988**	**0.5**
	16	165.80	1.7468	44.5	**7000.00**	**7165.80**	**1.5153**	**36.0**	8000.00	–	–
	17	253.39	1.8681	43.0	**7000.00**	**7253.39**	**1.6227**	**34.4**	8000.00	–	–
	18	**685.26**	**2.1629**	**48.4**	7000.00	7685.26	–	–	8000.00	–	–
	19	**915.00**	**1.9522**	**42.0**	7000.00	7915.00	–	–	8000.00	–	–
	20	**820.69**	**2.1985**	**46.8**	7000.00	7820.69	–	–	8000.00	–	–
A100S15	8	0.37	0.7342	20.0	19.72	20.09	0.5937	1.1	**8.56**	**0.5873**	**0.0**
	9	1.00	0.9982	38.7	68.25	69.25	0.6116	0.0	**9.63**	**0.6116**	**0.0**
	10	4.13	1.0909	38.7	405.85	409.98	0.6685	0.0	**142.88**	**0.6685**	**0.0**
	11	5.16	1.5028	34.9	1037.53	1042.69	1.0214	4.2	**700.94**	**0.9782**	**0.0**
	12	81.42	1.8071	39.7	**4423.61**	**4505.03**	**1.1363**	**4.1**	7200.00	1.1363	4.1
	13	72.93	2.0986	47.4	7000.00	7072.93	1.2298	10.3	**7200.00**	**1.1860**	**7.0**
	14	142.18	2.5720	56.8	7000.00	7142.18	1.4485	23.2	**7200.00**	**1.3554**	**18.0**
	15	89.61	2.5570	46.0	6566.83	6656.44	2.1839	36.8	**7200.00**	**1.6641**	**17.0**
A100S20	8	0.47	0.9234	24.3	12.28	12.75	0.7062	1.0	**8.53**	**0.6990**	**0.0**
	9	0.94	1.0606	24.8	51.17	52.11	0.7972	0.0	**12.36**	**0.7972**	**0.0**
	10	2.92	1.2049	26.5	287.18	290.10	0.8927	0.8	**137.65**	**0.8853**	**0.0**
	11	7.25	1.3079	27.5	**585.28**	**592.53**	**0.9477**	**0.0**	1042.36	0.9477	0.0
	12	23.56	1.3210	29.6	**7000.00**	**7023.56**	**0.9756**	**4.7**	7200.00	0.9756	4.7
	13	25.37	1.6664	35.9	7000.00	7025.37	1.1514	7.3	**7200.00**	**1.1311**	**5.8**
	14	46.40	1.9935	41.7	7000.00	7046.40	1.4268	18.5	**7200.00**	**1.2801**	**9.2**
	15	146.13	2.2779	42.3	6002.14	6148.27	1.4971	12.2	**7200.00**	**1.4284**	**8.0**
	16	354.18	2.2987	42.8	**7000.00**	**7354.18**	**1.7614**	**25.3**	8000.00	–	–
	17	259.15	2.6641	46.0	**7000.00**	**7259.15**	**2.2377**	**35.7**	8000.00	–	–
	18	**410.94**	**2.6754**	**43.8**	7000.00	7410.94	–	–	8000.00	–	–
	19	**654.56**	**2.7688**	**44.7**	7000.00	7654.56	–	–	8000.00	–	–
	20	1200.00	–	–	–	–	–	–	8000.00	–	–
SS15	8	0.47	0.5617	28.8	33.34	33.81	0.3997	0.0	**4.51**	**0.3997**	**0.0**
	9	1.67	0.6899	39.8	45.30	46.99	0.4154	0.0	**8.50**	**0.4154**	**0.0**
	10	2.08	0.6367	29.1	238.66	240.74	0.4517	0.0	**103.65**	**0.4517**	**0.0**
	11	14.74	0.9298	29.9	1031.54	1046.28	0.6518	0.0	**762.98**	**0.6518**	**0.0**
	12	12.62	1.2374	39.8	**5610.45**	**5623.07**	**0.7478**	**0.4**	7200.00	0.7478	0.4
SS20	8	0.70	0.5223	11.7	46.74	47.44	0.4614	0.1	**7.67**	**0.4610**	**0.0**
	9	0.73	0.5625	7.2	41.39	42.12	0.5227	0.1	**12.21**	**0.5220**	**0.0**
	10	4.07	0.7328	21.1	455.65	459.72	0.5784	0.0	**280.8**	**0.5784**	**0.0**
	11	16.60	0.9079	32.4	**563.50**	**580.10**	**0.6136**	**0.0**	1108.13	0.6136	0.0
	12	34.94	0.8805	28.9	**6445.89**	**6480.83**	**0.6287**	**0.4**	7200.00	0.6287	0.4

within the runtime imposed for all instances, except for matrix A10S20 with $19 \leq n \leq 20$, matrix A15S20 with $18 \leq n \leq 20$ and matrix A100S20 with $n = 20$, and, therefore, we did not run the LB heuristic for those matrices.

The LB heuristic obtains the same solution value as the one obtained by the Path-edges$^+$ formulation in 44 out of the 109 instances, corresponding to approximately 40.4% of the instances. It is worth noting that the LB heuristic obtains a better solution value for 13 out of the 53 instances that were not completely solved by the Path-edges$^+$ formulation, for approximately 24.5% instances. In 28 out of 115 instances, corresponding to 24.3%, the FP heuristic followed by the LB heuristic took less time to obtain the same or a better solution than the Path-edges$^+$ formulation. The LB heuristic obtains a better feasible solution than the one obtained by the FP heuristic, for all instances with $n \leq 17$ and for matrix S20 with $n = 18$. Comparatively, the Path-edges$^+$ formulation, only, obtains feasible solutions for instances with $n \leq 15$.

We noticed that if we increase the runtime limit to 2500 seconds, the FP heuristic obtains a feasible solution for all the instances. Thus, we run the FP heuristic with the runtime limit of 7200 seconds for several instances with $n \geq 20$ and verified that this matheuristic obtains a feasible solution for instances up to $n = 23$.

The next two tables, Tables 6 and 7, display the average computational time in seconds (in Table 6) and the average GAP in % (in Table 7), respectively, for the instances with the same number of leaves as well as their corresponding standard deviation values, for the FP heuristic, the LB heuristic and the Path-edges$^+$ formulation.

Table 6: Average and Standard Deviation (SD) values for the computational time (in seconds) of the FP heuristic, LB heuristic and Path-edges$^+$ formulation.

n		FP heuristic	LB heuristic	Path-edges$^+$
8	Average	0.45	19.89	7.68
	SD	0.14	8.43	4.21
9	Average	1.22	67.78	12.73
	SD	0.59	28.09	5.41
10	Average	4.85	329.55	206.57
	SD	4.10	128.06	130.81
11	Average	8.38	1045.80	1568.49
	SD	5.20	320.01	855.68
12	Average	21.36	5915.44	7200.00
	SD	15.15	1032.85	0.00
13	Average	43.65	6261.99	7200.00
	SD	39.60	1002.03	0.00
14	Average	145.18	6720.11	7200.00
	SD	123.38	732.14	0.00
15	Average	202.12	5044.09	7200.00
	SD	149.83	1573.13	0.00
16	Average	272.59	6903.70	8000.00
	SD	114.15	259.69	0.00
17	Average	385.87	6770.30	–
	SD	139.26	790.23	–
18	Average	657.36	7000.00	–
	SD	266.77	0.00	–
1	Average	856.52	–	–
	SD	179.99	–	–
18	Average	910.35	–	–
	SD	126.79	–	–

The average computational time of the FP heuristic ranges from 0.45 to 910.35 seconds and

Table 7: Average and Standard Deviation (SD) values for the GAP (%) of the FP heuristic, LB heuristic and Path-edges$^+$ formulation.

n		FP heuristic	LB heuristic	Path-edges$^+$
8	Average	18.23	0.70	0.00
	SD	8.31	1.23	0.00
9	Average	23.44	0.31	0.00
	SD	11.87	0.87	0.00
10	Average	24.99	0.67	0.00
	SD	11.35	1.19	0.00
11	Average	25.66	0.36	0.00
	SD	9.12	0.90	0.00
12	Average	30.27	4.26	4.22
	SD	8.62	3.08	3.23
13	Average	30.59	7.50	6.37
	SD	9.28	3.46	3.78
14	Average	34.91	17.13	9.87
	SD	12.08	5.73	5.31
15	Average	33.94	24.95	11.74
	SD	10.80	9.03	6.01
16	Average	33.71	22.79	11.93
	SD	6.57	5.49	3.36
17	Average	37.14	27.59	–
	SD	6.25	5.94	–
18	Average	38.10	28.64	–
	SD	7.67	2.00	–
19	Average	44.39	–	–
	SD	2.23	–	–
20	Average	45.95	–	–
	SD	1.21	–	–

the average GAP ranges from 18.23 to 45.95%. The average time of the LB heuristic ranges from 19.89 to 7000 seconds and the average GAP ranges from 0.31 to 28.64%. It is worth noting that for $n < 12$ the average GAP of the LB heuristic is less than 1%. The FP heuristic finds a feasible solution very quickly and the LB heuristic finds good solutions.

Finally, we remark that, besides the FP heuristic discussed, we also implemented and tested the Objective Feasibility Pump (OFP) and the Reweighted OFP (ROFP) presented in [4]. Among the two, the ROFP obtained better results. However, when the number of leaves of the instances increase, both were unable to find an integer feasible solution within the imposed time limit for more instances than our implementation of the FP.

6 Conclusions

We have described matheuristic procedures to find feasible solutions to the MWTR problem, namely, the FP (Feasibility Pump) and the LB (Local Branching) heuristics. The FP heuristic is a constructive matheuristic that relies upon a relax-and-fix strategy [23] to obtain a good feasible solution to the problem. The LB heuristic uses a local branch strategy [24] to improve a feasible solution. Our computational results show that the FP heuristic is fast in obtaining feasible solutions for the MWTR problem, and the LB heuristic can be used to improve the obtained feasible solution. Although both matheuristics can be used independently, the best strategy is to use them together, starting from the FP heuristic and using the LB heuristic after.

We report computational results for instances with the size n of leaves varying between 5 and 20. The best approach to obtain solutions to the problem is the MILP model Path-edges$^+$ formulation. However, we can use it to obtain solutions to the problem instances having a maximum of $n = 15$ leaves. The matheuristics are simple heuristics based on a MIP model and can significantly improve a previously obtained solution. Although they are time-consuming, they are both able to improve several feasible solutions and able to obtain feasible solutions to instances unsolved with the MIP model. In the literature and by using a MIP model-based procedure one can find feasible solutions to the problem instances having, at most, $n = 15$ leaves. By using the proposed matheuristics we can obtain feasible solutions to the problem instances having until $n = 17$ leaves, which is an increase in the size of the instances that can be solved using MIP model-based procedures.

When using MIP model-based procedures, the FP and the LB heuristics are a good choice in obtaining feasible solutions for the MWTR problem and can be used together for better quality solutions and with very competitive computational times.

Acknowledgments

Olga Fajarda was funded by the FCT under Grant SFRH/BD/76268/2011. This work is supported by The Center for Research and Development in Mathematics and Applications (CIDMA) through the Portuguese Foundation for Science and Technology (FCT - Fundação para a Ciência e a Tecnologia), references UIDB/04106/2020 and UIDP/04106/2020.

References

[1] T. Achterberg and T. Berthold. Improving the feasibility pump. *Discrete Optimization*, 4(1):77–86, 2007.

[2] L. Bertacco, M. Fischetti, and A. Lodi. A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization*, 4(1):63–76, 2007.

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

[4] N. L. Boland, A. C. Eberhard, F. Engineer, and A. Tsoukalas. A new approach to the feasibility pump in mixed integer programming. *SIAM Journal on Optimization*, 22(3):831–861, 2012.

[5] S.-S. Byun and C. Yoo. Reducing delivery delay in HRM tree. In M. Gavrilova, O. Gervasi, V. Kumar, C. Tan, D. Taniar, A. Laganà, 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.

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

[7] D. Catanzaro, R. Aringhieri, M. D. 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.

[8] 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.

[9] 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.

[10] L. L. Cavalli-Sforza and A. W. Edwards. Phylogenetic analysis. Models and estimation procedures. *Evolution*, 21(3):550–570, 1967.

[11] 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.

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

[17] R. Desper and O. Gascuel. Theoretical Foundation of the Balanced Minimum Evolution Method of Phylogenetic Inference and Its Relationship to Weighted Least-Squares Tree Fitting. *Molecular Biology and Evolution*, 21(3):587–598, 03 2004.

[18] 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.

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

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

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

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

[23] M. Fischetti, F. Glover, and A. Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2005.

[24] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1-3):23–47, 2003.

[25] B. Fortz, O. Oliveira, and C. Requejo. Compact mixed integer linear programming models to the minimum weighted tree reconstruction problem. *European Journal of Operational Research*, 256(1):242–251, 2017.

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

[27] 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.

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

[29] S. Kumar. A stepwise algorithm for finding minimum evolution trees. *Molecular biology and evolution*, 13(4):584–593, 1996.

[30] J. Ni, H. Xie, S. Tatikonda, and Y. R. Yang. Network routing topology inference from end-to-end measurements. In *Proceedings of the IEEE INFOCOM 2008, the 27th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 36–40, 2008.

[31] 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.

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

[33] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425, 1987.

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