mathematics

MDPI

Article

An Efficient Mixed Integer Linear Programming Model for the Minimum Spanning Tree Problem

Tamer F. Abdelmaguid [1,2]

[1] Department of Mechanical Engineering, School of Sciences and Engineering, American University in Cairo, AUC Avenue, P.O. Box 74, New Cairo 11835, Egypt; tabdelmaguid@eng.cu.edu.eg
[2] Department of Mechanical Design and Production, Faculty of Engineering, Cairo University, Giza 12613, Egypt

Received: 3 August 2018; Accepted: 27 September 2018; Published: 29 September 2018

Abstract: Finding a minimum spanning tree in a given network is a famous combinatorial optimization problem that appears in different engineering applications. Even though this problem is solvable in polynomial time, having efficient mathematical programming models is important as they can provide insights for formulating larger models that integrate other decisions in more complex applications. In the literature, there are ten different integer and mixed integer linear programming (MILP) models for this problem. They are variants of set packing, cuts, network flow and node level formulations. In addition, this paper introduces an efficient node level MILP model. Comparisons for the eleven models are provided. First, the models are compared in terms of the number of decision variables and the number of constraints. Then, computational comparisons using a commercial MILP solver on sets of randomly generated instances of different sizes are conducted. Results provide evidence that the proposed MILP model is competitive in terms of the computational time needed for proving optimality of generated solutions for instances with up to 50 nodes. Meanwhile, the LP relaxation of a multi-commodity flow MILP model which has integer polyhedron provides stable computational times despite its larger size.

Keywords: minimum spanning tree; combinatorial optimization; mathematical programming

MSC: 90C05; 90C11; 90C27

1. Introduction

For a given undirected, connected network $\mathcal{W} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes, $\mathcal{A}$ is the set of arcs and l_a is the length of arc $a \in \mathcal{A}$, the minimum spanning tree (MST) is a subset of arcs $\mathcal{M} \subset \mathcal{A}$ that connect all nodes in $\mathcal{N}$ such that $\sum_{a \in \mathcal{M}} l_a$ is minimum. To illustrate, Figure 1 shows a sample undirected, connected network in which $\mathcal{N} = \{1, 2, 3, 4, 5, 6\}$, and $\mathcal{A} = \{\{i, j\} : i, j \in \mathcal{N} \text{ and } i \neq j\}$. Each arc $a \in \mathcal{A}$ is represented in Figure 1 using a dashed line, and its associated length (l_a) is written on top of the dashed line. Figure 2 shows four different selected subsets of arcs which are represented by solid lines. In Figure 2a, the selected arcs do not form a spanning tree as they do not connect all nodes in the network. In Figure 2b, the selected arcs connect all nodes but they form the cycle 1-2-4-5-1. In cycles, an arc can be unselected to reduce the summation of the selected arcs' lengths while the involved nodes remain connected. Therefore, this set of arcs does not represent a minimum spanning tree. In Figure 2c, the selected arcs span all nodes in the network without forming a cycle, yet the summation of the selected arcs' lengths is not minimum. Meanwhile, Figure 2d shows selected arcs that form a spanning tree having the minimum summation of the arcs' lengths. Therefore, $\mathcal{M} = \{\{1, 2\}, \{1, 5\}, \{1, 6\}, \{3, 4\}, \{4, 5\}\}$.

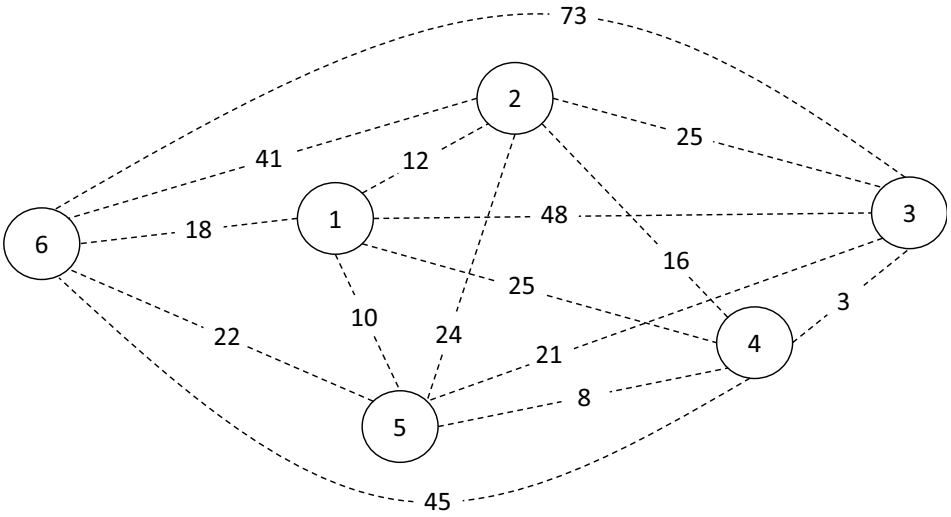

Figure 1. A sample undirected, connected network.

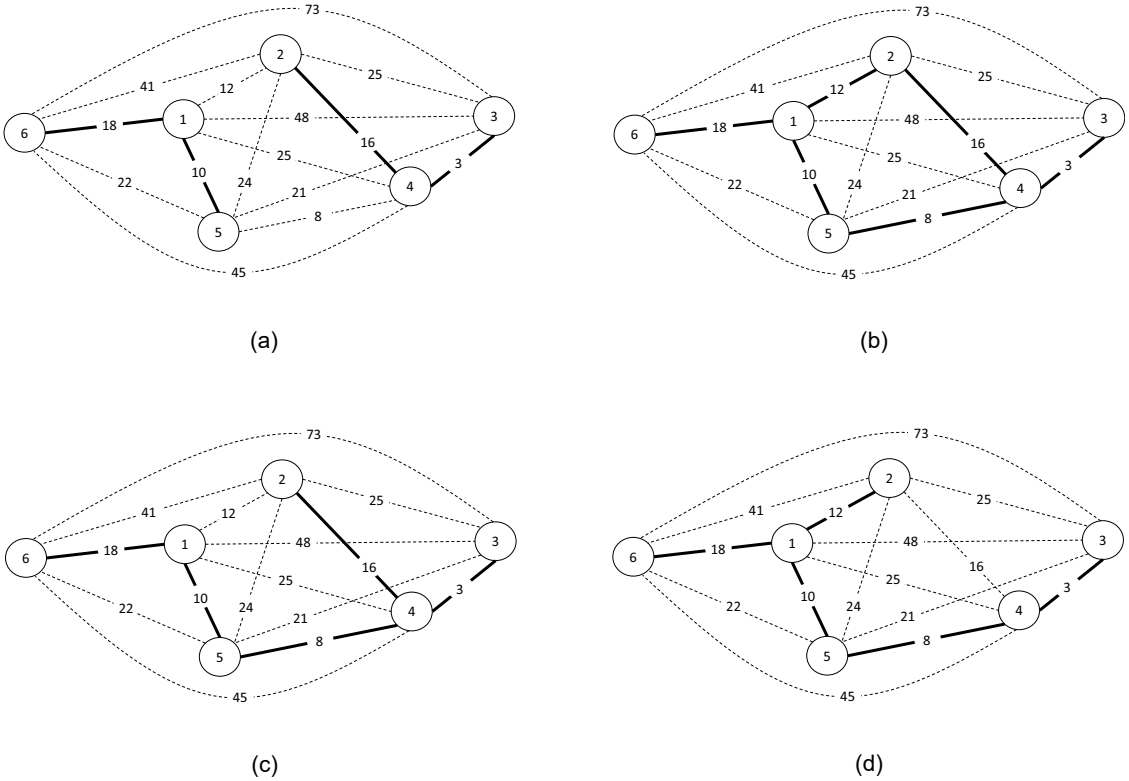

Figure 2. Subsets of selected arcs connecting nodes in the sample network of Figure 1: (**a**) selected arcs are not sufficient to form a spanning tree that connects all nodes; (**b**) selected arcs are spanning all nodes but they form a cycle; (**c**) selected arcs form a spanning tree that does not have the minimum total arc length; and (**d**) selected arcs form a minimum spanning tree.

The problem of finding $\mathcal{M}$ is a famous combinatorial optimization problem for which polynomial time algorithms exist [1–4]. As explained in [5], this problem appears in different engineering and service applications, particularly in designing computer, telecommunications, transportation and water supply networks. In addition, it has a number of computational applications such as clustering of data points in a plane [6], handwriting recognition [7] and providing approximate solutions for the traveling salesman problem [8]. Some recent applications include cell nuclei segmentation [9], Alzheimer's classification [10], water looped network equilibrium [11] and characterizing local urban patterns [12].

As indicated by Magnanti and Wolsey [5], having alternate linear programming models for the problem of finding MST can offer theoretical and algorithmic insights into solving the problem. This can be extended to a larger application in which finding an MST is a subproblem. In the literature, there are ten different integer and mixed integer linear programming models for the MST problem. They are variants of set packing, cuts, network flow and node level formulations. The set packing formulation is based on defining constraints that restrict the cardinality of any selected subset of arcs not to exceed the number of connected nodes less than one. These constraints are used for eliminating cycles. In cuts formulations, a cut set is a subset of arcs that connect two subsets of nodes. Constraints that define lower bounds on the number of arcs in cut sets are used to ensure that all nodes are connected. The network flow formulations utilize artificial decision variables to represent quantities of commodity flows along the arcs. By maintaining consistent flow starting at a source node and ending at a destination node along the network, a tree is constructed. In the node level formulations, a level decision variable is associated with each node. Constraints that restrict the inclusion of arcs connecting a lower level node to a higher level one are used for eliminating cycles.

Some of the models are notorious for their combinatorial explosion as a result of their exponential number of constraints with respect to the number of nodes. Other models aim to have polynomial number of constraints while maintaining integer polyhedron for their linear programming relaxations. Such a property reduces the computational effort by directly using polynomial-time linear programming algorithms without the need for the non-polynomial MILP solution algorithms. However, this does not necessarily guarantee finding an optimal solution in reduced computational time. Therefore, any comparison between the models has to take into consideration two aspects: the first one is the size of the model represented by the number of decision variables and the number of constraints; and the second one being the computational time needed to prove optimality of a generated solution.

This paper presents a node level MILP model and compares it with existing models in terms of the number of decision variables and constraints. Numerical experiments are conducted using randomly generated instances of various sizes to provide computational comparisons between a selected subset of the models that have polynomial number of constraints in the size of the network represented by the number of nodes. IBM ILOG CPLEX version 12.8 [13] (IBM, Armonk, NY, USA) is used to generate solutions to these instances based on the compared MILP models. The models are entered to CPLEX using the optimization programming language (OPL) [14]. The computational time needed for each model to prove optimality of a generated solution is used for comparison.

The rest of this paper is organized as follows. A review is provided in Section 2 for the existing integer and mixed integer linear programming models of the problem of finding MST. The proposed model is presented along with a proof for its validity in Section 3. In Section 4, a comparison is provided for the two aspects mentioned above; followed by the conclusion in Section 5.

2. Review of Existing Models

This section reviews the different integer and mixed integer linear programming models developed in the literature for the problem of finding $\mathcal{M}$. The first nine models are provided in [5], and the tenth model is presented in [15]. They are presented here with different exposition and insights, and for the sake of the integrality of this paper. To simplify comparisons between models, we consider a network in which $|\mathcal{A}| = n(n-1)/2$ where $n = |\mathcal{N}|$, i.e., for every pair of nodes i and $j \in \mathcal{N}$, there is an arc $\{i, j\} \in \mathcal{A}$.

2.1. Cycle Elimination Model

The first model is a direct intuitive mathematical expression of the definition of the MST. A binary decision variable x_a is used to represent the decision of whether arc $a \in \mathcal{A}$ will be included in the tree ($x_a = 1$) or not ($x_a = 0$). Accordingly, the first model is formulated as follows.

$$\text{Minimize} \quad \sum_{a \in \mathcal{A}} l_a x_a \tag{1}$$

Subject to:

$$x_a \in \{0, 1\} \qquad \forall a \in \mathcal{A} \tag{2}$$

$$\sum_{a \in \mathcal{A}} x_a = n - 1 \tag{3}$$

$$\sum_{a \in A(S)} x_a \leqslant |S| - 1 \qquad \forall S \subset \mathcal{N} \text{ where } |S| > 1 \tag{4}$$

The set of constraints $\mathcal{T}_1 = \{(2)–(4)\}$ defines the domain of the decision variables of the first model. Constraint (3) represents the condition that a tree in $\mathcal{W}$ has exactly $n - 1$ arcs. Constraints (4) take the form of set packing constraints, and they are referred to as cycle elimination constraints. Eliminating cycles is achieved by limiting the number of arcs included in the sub-tree connecting the nodes in every subset S not to exceed $|S| - 1$. Here, $A(S) \subset \mathcal{A}$ is the subset of all arcs connecting the nodes in S. Logically, all the aforementioned constraints ensure that all nodes in the network will be connected. The linear programming (LP) relaxation of this model has integer polyhedron. Even though this model has exponential number of cycle elimination constraints ($2^n - n - 2$), it has interesting theoretical properties that can be used to prove the optimality of the greedy algorithm in [2], as illustrated in [5].

2.2. Cut Sets Models

The second model is based on defining cut sets. For a non-empty subset $S \subset \mathcal{N}$, a cut set is a subset of arcs, denoted $C(S)$, that whenever removed from $\mathcal{W}$ will disconnect the nodes in S from the nodes in $\mathcal{N} - S$. This model has definitions of the decision variables and the objective function that are similar to the first model. The set of constraints $\mathcal{T}_2 = \{(2), (3), (5)\}$ defines the domain of the decision variables of the second model.

$$\sum_{a \in C(S)} x_a \geqslant 1 \quad \forall S \subset \mathcal{N} \text{ where } |S| \geq 1 \tag{5}$$

In $\mathcal{T}_2$, the cycle elimination constraints are replaced by connectivity constraints which ensure that for all subsets $S \subset \mathcal{N}$, the size of the cut set is at least one. The latter set of constraints is also exponential in size which equals $2^{n-1} - 1$ after excluding duplicate cut sets. For $n > 3$, the number of Constraints (5) is less than the number of Constraints (4), yet it is proven in [5] that the LP relaxation of $\mathcal{T}_2$ does not have integer polyhedron.

The third model is based on defining multicuts. For some K where $1 < K < n$, let $S_k \subset \mathcal{N}$ where $k = 1, \ldots, K$, such that $S_k \neq \emptyset$, $\bigcup_{k=1,\ldots,K} S_k = \mathcal{N}$ and $S_k \cap S_{k'} = \emptyset$ for any $k \neq k'$. The multicut model utilizes the same decision variables and objective function as the first two models. Let $C(S_1, S_2, \ldots, S_K) \subset \mathcal{A}$ be the arcs that whenever removed from $\mathcal{W}$ will result in K separate cliques. The set of Constraints $\mathcal{T}_3 = \{(2), (3), (6)\}$ defines the domain of the decision variables.

$$\sum_{a \in C(S_1, S_2, \ldots, S_K)} x_a \geqslant K - 1 \qquad \forall S_1, S_2, \ldots, S_K \subset \mathcal{N} \tag{6}$$

Constraints (6) maintain connectivity in the MST for all node partitions satisfying the aforementioned properties. When $K = 2$, Constraints (6) reduce to Constraints (5). It is proven in [5] that, for $K > 2$, the LP relaxation of $\mathcal{T}_3$ has integer polyhedron which makes it advantageous

over the single cut domain $\mathcal{T}_2$. However, no clue is provided for the upper limit of K at which feasible trees can be obtained. The following proposition states an important property of this model.

Proposition 1. *The set of constraints $\mathcal{T}_3$ does not prevent cycles when $K = n - 1$.*

Proof. Observe that, for an arbitrary non-empty subset of nodes $\sigma \subset \mathcal{N}$, the set of constraints $\mathcal{T}_3$, when $K = n - 1$, does not prevent the following condition:

$$\sum_{i \in \sigma} \sum_{j \in \mathcal{N} - \sigma} x_{\{i,j\}} = 0$$

Therefore, cycles can be formed. □

Regarding the size of the third model, the number of Constraints (6) is the Stirling number of the second kind which is denoted $\{{n \atop K}\}$ [16]. It is known that $\{{n \atop n-1}\} = n(n-1)/2$, which is polynomial, but there is no guarantee for feasible trees as indicated by Proposition 1. Meanwhile, $\{{n \atop K}\}$ is exponential in n when $2 \leq K < n - 1$.

2.3. Digraph Models

The MST is equivalent to a set of shortest paths that start at an arbitrary root node and connect it with every other node in the network. A model that finds such shortest paths with the minimum total length for all paths finds an MST as well. Based on this concept, the fourth model is defined. A digraph $\mathcal{G} = (\mathcal{N}, \mathcal{D})$ is constructed by replacing every arc $a = \{i, j\} \in \mathcal{A}$ in $\mathcal{W} = (\mathcal{N}, \mathcal{A})$ by directed arcs (i, j) and $(j, i) \in \mathcal{D}$, both having a length of l_a. For every pair of arcs (i, j) and $(j, i) \in \mathcal{D}$, variables $z_{(i,j)}$ and $z_{(j,i)}$ are defined, respectively. An arc $a \in \mathcal{A}$ is included in $\mathcal{M}$ whenever either $z_{(i,j)}$ or $z_{(j,i)}$ equals 1. Constraints (8) define this relationship, and Constraints (9) and (10) define, respectively, the limit on the number of included arcs in $\mathcal{M}$ and the cycle elimination constraints.

$$z_d \geq 0 \qquad \forall d \in \mathcal{D} \tag{7}$$

$$x_a = z_{(i,j)} + z_{(j,i)} \qquad \forall a = \{i, j\} \in \mathcal{A} \tag{8}$$

$$\sum_{d \in \mathcal{D}} z_d = n - 1 \tag{9}$$

$$\sum_{d \in D(S)} z_d \leqslant |S| - 1 \qquad \forall S \subset \mathcal{N} \text{ where } |S| > 1 \tag{10}$$

where $D(S) \subset \mathcal{D}$ is the subset of all directed arcs connecting the nodes in S. Let $\alpha \in \mathcal{N}$ denote the arbitrarily selected root node, and $\mathcal{N}^- = \mathcal{N} - \{\alpha\}$. To construct the set of shortest paths, it should be exactly one arc coming into every node in $\mathcal{G}$ except the root node. Constraints (11) represent this condition.

$$\sum_{(i,v) \in \mathcal{D}} z_{(i,v)} = 1 \qquad \forall v \in \mathcal{N}^- \tag{11}$$

Meanwhile, the root node should not have any incoming arc from any other node. This is implied by Constraints (9) and (11). The fourth model, which is represented by the Objective Function (1) and the set of Constraints $\mathcal{T}_4 = \{(2), (7)–(11)\}$, is a directed version of the first model. It does not reduce the number of constraints in compensation of the increase in the number of decision variables. However, it may provide some insights for algorithmic approaches.

The fifth model is a directed version of the second model that is based on the digraph $\mathcal{G} = (\mathcal{N}, \mathcal{D})$, arbitrarily selected root node α and the decision variables $z_{(i,j)}$ and $z_{(j,i)}$. The set of constraints $\mathcal{T}_5 = \{(2), (7)–(9), (12)\}$ defines the domain of the decision variables in the fifth model.

$$\sum_{d \in \widehat{D}(S)} z_d \geqslant 1 \qquad \forall S \subset \mathcal{N}^- \text{ where } |S| \geq 1 \tag{12}$$

where $\widehat{D}(S) \subset \mathcal{D}$ is the subset of all directed arcs that originate from the nodes in the subset $S \subset \mathcal{N}^-$ and end at nodes in the subset $\mathcal{N} - S$ in the digraph $\mathcal{G}$. It is proven in [5] that the LP relaxation of the fifth model has equivalent polyhedron to the LP relaxation of the fourth model, and both are equivalent to the first model.

2.4. Single Commodity Flow Model

The sixth model is based on defining fictitious flow variables through the arcs of the network. It is presented in [5] as a single commodity flow model in which one arbitrary node is selected as the source node and the other nodes are defined as demand nodes. Let α denote the source node and $\mathcal{N}^- = \mathcal{N} - \{\alpha\}$. For each arc $a = \{i, j\} \in \mathcal{A}$ connecting nodes i and j, two non-negative decision variables, $y_{i,j}$ and $y_{j,i}$, are defined to represent quantity of commodity flow from node i to node j and vice versa. At every node $i \in \mathcal{N}^{-1}$, the demand is exactly one unit. Therefore, the difference between inflows and outflows must be exactly one. Meanwhile, the supply quantity at the source node α has to be exactly $n - 1$. These flow conservation constraints are formulated by Equations (13) and (14).

$$\sum_{\{i,j\} \in \mathcal{A}} y_{j,i} - \sum_{\{i,j\} \in \mathcal{A}} y_{i,j} = 1 \qquad \forall i \in \mathcal{N}^- \tag{13}$$

$$\sum_{\{\alpha,j\} \in \mathcal{A}} y_{\alpha,j} - \sum_{\{\alpha,j\} \in \mathcal{A}} y_{j,\alpha} = n - 1 \tag{14}$$

$$y_{i,j} \text{ and } y_{j,i} \geq 0 \qquad \forall \{i, j\} \in \mathcal{A} \tag{15}$$

In the sixth model, The MST solution is captured by the binary decision variables $\{x_a : a \in \mathcal{A}\}$ just like the first three models. For arc $a = \{i, j\}$, $y_{i,j} = 0$ and $y_{j,i} = 0$ whenever $x_a = 0$. Constraints (16) and (17) represent this relationship.

$$y_{i,j} \leq (n-1)x_a \qquad \forall a = \{i, j\} \in \mathcal{A} \tag{16}$$

$$y_{j,i} \leq (n-1)x_a \qquad \forall a = \{i, j\} \in \mathcal{A} \tag{17}$$

where $(n-1)$ is the upper bound on the amount of flow on any arc throughout the network. Whenever $x_a = 1$, Constraints (13) and (14) force $y_{i,j}$ or $y_{j,i}$ to take a value greater than zero as needed. The model defined by Objective Function (1) and the set of constraints $\mathcal{T}_6 = \{(2), (3), (13)–(17)\}$ is based on a common approach used in formulating routing problems in which tying flow variables to binary arc inclusion-exclusion variables, along with flow conservation constraints, is used to replace the cycle elimination constraints. As shown in [5], the LP relaxation of this model results in non-integer solutions. The main advantage of this model compared to the previous ones is that it has polynomial number of constraints.

2.5. Multicommodity Flow Models

Based on the idea of the equivalence of the MST and the set of shortest paths from an arbitrary source node, the seventh model is defined. On the digraph $\mathcal{G} = (\mathcal{N}, \mathcal{D})$, defined in Section 2.3, starting at the root node α, a single unit of commodity k is to be sent to node $k \in \mathcal{N}^-$. The decision variable f_d^k

is used to represent the flow quantity along the directed arc $d \in \mathcal{D}$ of commodity k. The following flow conservation equations define part of the constraints on these variables.

$$\sum_{(i,k)\in\mathcal{D}} f^k_{(i,k)} - \sum_{(k,j)\in\mathcal{D}} f^k_{(k,j)} = 1 \qquad \forall k \in \mathcal{N}^- \tag{18}$$

$$\sum_{(\alpha,j)\in\mathcal{D}} f^k_{(\alpha,j)} - \sum_{(i,\alpha)\in\mathcal{D}} f^k_{(i,\alpha)} = 1 \qquad \forall k \in \mathcal{N}^- \tag{19}$$

$$\sum_{(i,v)\in\mathcal{D}} f^k_{(i,v)} - \sum_{(v,j)\in\mathcal{D}} f^k_{(v,j)} = 0 \qquad \forall v \in \mathcal{N}^- - \{k\} \quad \forall k \in \mathcal{N}^- \tag{20}$$

$$f^k_d \geq 0 \qquad \forall k \in \mathcal{N}^- \quad \forall d \in \mathcal{D} \tag{21}$$

By Constraints (18), every node $k \in \mathcal{N}^-$ receives exactly one unit of commodity k which originates at the source node by Constraints (19), and flows through other nodes without being consumed by Constraints (20). Similar to the single commodity flow model, the flow decision variables are tied to the binary arc inclusion–exclusion decision variables $z_{(i,j)}$ by Constraints (22).

$$f^k_d \leq z_d \qquad \forall d \in \mathcal{D} \quad \forall k \in \mathcal{N}^- \tag{22}$$

$$z_d \quad \text{integer} \qquad \forall d \in \mathcal{D} \tag{23}$$

Accordingly, the seventh model is defined by Objective Function (1) and the set of constraints $\mathcal{T}_7 = \{(2), (7)–(9), (18)–(23)\}$. It is shown in [5] that, based on the maximal flow-minimal cut theorem [17], the LP relaxation of the seventh model has an integer polyhedron identical to the fifth model.

The eighth model is an undirected version of the seventh model defined over the network $\mathcal{W} = (\mathcal{N}, \mathcal{A})$. Similar to the single commodity flow model, the decision variables $f^k_{(i,j)}$ and $f^k_{(j,i)}$ are defined on arc $\{i,j\} \in \mathcal{A}$. The integer decision variables z_d of the seventh model are not used in the eighth model. The flow decision variables $f^k_{(i,j)}$ and $f^k_{(j,i)}$ are tied to the binary variables x_a as represented by Constraints (24).

$$f^k_{(i,j)} + f^{k'}_{(j,i)} \leq x_a \qquad \forall a = \{i,j\} \in \mathcal{A} \quad \forall k \text{ and } k' \in \mathcal{N}^- \tag{24}$$

The eighth model is constituted by Objective Function (1) and the set of constraints $\mathcal{T}_8 = \{(2), (3), (18)–(21), (24)\}$. This reduces the number of decision variables compared to the seventh model without affecting the polyhedron of the LP relaxation as shown in [5].

Based on the eighth model, Magnanti and Wolsey [5] suggested a ninth model in which Constraints (24) are replaced by weaker Constraints (25).

$$f^k_{(i,j)} \leq x_a \qquad \forall a = \{i,j\} \in \mathcal{A} \quad \forall k \in \mathcal{N}^- \tag{25}$$

However, Constraints (25) are found to be insufficient for generating feasible solutions as Constraints (26) have to be considered as well.

$$f^k_{(j,i)} \leq x_a \qquad \forall a = \{i,j\} \in \mathcal{A} \quad \forall k \in \mathcal{N}^- \tag{26}$$

Therefore, the ninth model is defined by the set of Constraints $\mathcal{T}_9 = \{(2), (3), (18)–(21), (25), (26)\}$. Based on the maximal flow-minimal cut theorem, it is shown in [5] that the polyhedron of the LP relaxation of the ninth model is equivalent to the polyhedron of the LP relaxation of the second model.

2.6. Node Level Model

The node level formulation is based on the idea of assigning level decision variables for the nodes and using them to eliminate cycles. This is done by introducing level difference constraints that are tied to the arc inclusion/exclusion binary decision variables. This approach was originally

introduced by Miller, Tucker and Zemlin for the traveling salesman problem [18]. The tenth model presented here was introduced in [15] using LINGO mathematical programming modeling language. It utilizes the digraph $\mathcal{G} = (\mathcal{N}, \mathcal{D})$ and the binary decision variables z_d for $d \in \mathcal{D}$ defined in Section 2.3. Additional decision variables are defined to represent the node level in the tree, which are denoted V_i for $i \in \mathcal{N}$. An arbitrary root node α is selected and assigned a zero level. For other nodes $k \in \mathcal{N} - \{\alpha\}$, if $z_{(i,k)} = 1$, the level of node k is higher than the level of node i by 1. The following constraints represent these relationships.

$$V_\alpha = 0 \tag{27}$$

$$V_k \geq V_i + z_{(i,k)} - (n-2)(1 - z_{(i,k)}) + (n-3)z_{(k,i)} \qquad \forall \{i,k\} \in \mathcal{A} \quad \forall k \in \mathcal{N} - \{\alpha\} \tag{28}$$

In addition, two domain constraints are defined for the level decision variables as follows.

$$V_k \geq 1 \qquad \forall k \in \mathcal{N} - \{\alpha\} \tag{29}$$

$$V_k \leq n - 1 - (n-2)z_{(\alpha,k)} \qquad \forall k \in \mathcal{N} - \{\alpha\} \tag{30}$$

To make sure that all nodes are connected while maintaining their level relationships, every node $k \in \mathcal{N} - \{\alpha\}$ must have exactly one arc coming into it, while the root node α must have at least one arc going out of it.

$$z_{(i,j)} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{D} \tag{31}$$

$$\sum_{(i,k)\in\mathcal{D}} z_{(i,k)} = 1 \qquad \forall k \in \mathcal{N} - \{\alpha\} \tag{32}$$

$$\sum_{(\alpha,k)\in\mathcal{D}} z_{(\alpha,k)} \geq 1 \tag{33}$$

Accordingly, the tenth model is represented by the set of Constraints $\mathcal{T}_{10} = \{(8), (27)\text{–}(33)\}$. No proof is provided in [15] for the validity of this model. However, this can be deduced since Constraints (32) and (33) ensure that all nodes are connected, and the level Constraints (27) to (30) eliminate cycles. It is important to note here that the structure of Constraints (28) along with Constraints (31)–(33) allow only one of the arcs (i,j) and (j,i) to be included in the solution but not both.

3. The Proposed MILP Model

The proposed eleventh model is another variant of the node level formulation. It utilizes the digraph $\mathcal{G} = (\mathcal{N}, \mathcal{D})$ and the binary decision variables z_d for all $d \in \mathcal{D}$ as defined in Section 2.3. The main idea behind the proposed model is based on the analogy between rivers which have multiple sources and a single sink and trees which have multiple branches and a single trunk. Neither rivers nor trees not have cycles. In the proposed model, an arbitrary node $\omega \in \mathcal{N}$ is labeled as a sink node. Other nodes represent either sources of water or intermediate intersection points. In both cases, water only flows from a higher level node to a lower level one. We define decision variables V_i for all $i \in \mathcal{N}$ to represent the water level at each node. The sink node has the lowest level, which is set to zero, while the other nodes have levels that must be greater than zero. Some constraints are introduced to link the z_d variables with the level variables. We start with the following formulation.

$$\text{Minimize} \quad \sum_{d \in \mathcal{D}} l_d z_d \tag{34}$$

Subject to:

$$\sum_{(i,j) \in \mathcal{D}} z_{(i,j)} = 1 \qquad \forall i \in \mathcal{N} - \{\omega\} \tag{35}$$

$$\sum_{(\omega,j) \in \mathcal{D}} z_{(\omega,j)} = 0 \tag{36}$$

$$V_i \geq V_j + z_{(i,j)} - n(1 - z_{(i,j)}) \qquad \forall (i,j) \in \mathcal{D} \;\; \forall i \in \mathcal{N} - \{\omega\} \tag{37}$$

$$V_\omega = 0 \tag{38}$$

$$z_d \in \{0,1\} \qquad \forall \;\; d \in \mathcal{D} \tag{39}$$

The Objective Function (34) is the same as (1), defined using the decision variables z_d, where l_d for both $d = (i,j)$ and $d = (j,i)$ equals l_a for $a = \{i,j\}$. Constraints (35) represent the condition that there is exactly one arc going out of every node $i \in \mathcal{N} - \{\omega\}$. Meanwhile, Constraint (36) ensures that the sink node does not have any arcs going out of it. These constraints imitate directions of water flow in rivers, and they ensure that the number of arcs included in the solutions is $n - 1$, which is equivalent to Constraint (9).

Constraints (37) define the relationship between the binary decision variables $z_{(i,j)}$ and the corresponding level variables V_i and V_j. It is formulated to make sure that if $z_{(i,j)} = 1$, the level at node i is greater than the level at node j by one; otherwise, this constraint will be redundant. Constraints (38) sets the level of the sink node to zero. Constraints (39) are the domain constraints for the variables z_d. The following theorem is necessary to proof the validity of the proposed model.

Theorem 1. *Constraint* (37) *eliminates cycles.*

Proof. (by contradiction) Suppose without loss of generality that, for the subset of nodes $S = \{\alpha, \alpha + 1, \ldots, \nu\} \subseteq \mathcal{N} - \{\omega\}$, where $|S| \geq 2$, a cycle is formed by having $z_{(\nu,\alpha)} = 1$ and $z_{(i,i+1)} = 1 \;\; \forall i, \; i+1 \in S$. Therefore, according to Constraints (37), $V_\nu \geq V_\alpha + 1$ and $V_i \geq V_{i+1} + 1 \; \forall i, \; i+1 \in S$. The latter leads to $V_\alpha > V_\nu$ which contradicts the former. □

Corollary 1. *Constraints* (35)–(39) *result in valid trees.*

Proof. As a result of Theorem 1 and since Constraints (35), (36) and (39) ensure that there is exactly one arc going out from every node $i \in \mathcal{N} - \{\omega\}$, there has to be at least one arc coming into node ω. Therefore, a valid acyclic tree in which all nodes are connected is obtained by Constraints (35)–(39). □

Therefore, the MILP model presented by Equations (34)–(39) is sufficient to find a minimum spanning tree. However, additional bounding constraints for the decision variables help in reducing the number of iterations needed by a solution algorithm. Therefore, we define Constraints (40) and (41) to provide lower and upper bounds for the level decision variables.

$$V_i \leq n - 1 \qquad \forall i \in \mathcal{N} - \{\omega\} \tag{40}$$

$$V_i \geq 1 \qquad \forall i \in \mathcal{N} - \{\omega\} \tag{41}$$

In addition, valid inequalities [19] can be defined to strengthen the proposed formulation. Logically, since there must be at most one arc connecting every pair of nodes, the following constraints are added.

$$z_{(i,j)} + z_{(j,i)} \leq 1 \qquad \forall i,j \in \mathcal{N} - \{\omega\} \tag{42}$$

Proposition 2. *Constraints* (42) *are valid inequalities for Formulations* (34)–(39).

Proof. Since Formulations (34)–(39) results in solutions that do not contain cycles, as stated in Theorem 1 and Corollary 1, for every pair of nodes i and $j \in \mathcal{N} - \{\omega\}$ either $z_{(i,j)} = 1$ or $z_{(j,i)} = 1$ but not both. Since Inequality (42) are mathematical representations of these conditions, they are valid inequalities for Formulations (34)–(39). □

In addition, there must be at least one arc coming into the sink node ω. Since this is implied by Constraints (35) and (36), the following constraint is another valid inequality for Formulations (34)–(39).

$$\sum_{(i,\omega)\in\mathcal{D}} z_{(i,\omega)} \geq 1 \tag{43}$$

Therefore, the set of constraints $\mathcal{T}_{11} = \{(35)\text{–}(43)\}$ defines the domain of the decision variables of the proposed model.

4. Models Comparison

The first aspect to consider in comparing the eleven models presented herein is the memory size needed by an MILP solution algorithm, which is represented by the number of decision variables and the number of constraints. This aspect not only defines the memory requirements, but also reflects the computational time needed in each iteration of the solution algorithm. The second aspect is the computational time needed to prove optimality. Both aspects are not necessarily correlated. The following sections discuss these two aspects.

4.1. Model Size

A comparison of the characteristics of the eleven models and their sizes is presented in Table 1. For each model, an abbreviated name is assigned in Table 1 which is used in the following analysis, tables and figures to provide a meaningful reference. The size of each model is represented in Table 1 by the total number of variables, the number of integer variables and the number of constraints. The number of arcs in the network, $A = n(n-1)/2$, is used to simplify the formulas used for the calculated numbers. Following the convention of the MILP solvers, the non-negativity, integer and binary constraints for all models are not counted.

As shown in Table 1, the first to fifth models suffer from the combinatorial explosion as a result of their exponential number of constraints. For the remaining models, the number of decision variables and constraints are polynomial in terms of the number of nodes. With respect to the number of decision variables, the tenth (D-level) and the proposed models have the lowest number of decision variables among the models with polynomial number of constraints. In terms of the number of integer variables, the LP relaxations of the seventh (D-m-flow) and eighth (m-flow-1) models have obvious advantage over the other models since they are proven to provide integer solutions without the need to explicitly define integer variables. This allows for solving them using polynomial time linear programming algorithms instead of the non-polynomial branch-and-bound or branch-and-cut algorithm. However, this does not necessarily mean that the computational time will be less as it is also affected by the model size. For the other models with polynomial number of constraints, the ninth (m-flow-2) model has the lowest number of integer variables. This may help in reducing the number of branch-and-bound iterations needed; however, again, it does not necessarily mean that the computational time is less.

Table 1. Model characteristic and size comparison.

Model No.	Model Name Abbreviation	LP Relax. Is Integer	Number of Variables	Number of Integer Variables	Number of Constraints
1st	Cycle	yes	A	A	$2^n - n - 1$
2nd	1-cut	no	A	A	2^{n-1}
3rd	m-cut	yes	A	A	$\{{n \atop K}\} + 1$
4th	D-cycle	yes	$3A$	A	$2^n + A - 2$
5th	D-1-cut	yes	$3A$	A	$2^{n-1} + A$
6th	D-1-flow	no	$3A$	A	$2A + n + 1$
7th	D-m-flow	yes	$A(2n + 1)$	$3A$	$2An + A + 1$
8th	m-flow-1	yes	$A(2n - 1)$	A	$(A + 1)(n - 1)^2 + n$
9th	m-flow-2	no	$A(2n - 1)$	A	$2An + 1$
10th	D-level	no	$2A + n$	$2A$	$2A + 2n$
11th	Proposed	no	$2A + n$	$2A$	$3A + n + 2$

The sixth (D-1-flow) model has the lowest number of constraints, followed by the tenth (D-level) and proposed models, respectively. The differences in the number of constraints between the D-1-flow, D-level and the proposed models are not significant, while the D-m-flow, m-flow-1 and m-flow-2 models have comparatively larger number of constraints. To demonstrate, Figure 3 presents a comparison between selected models in terms of their number of constraints at different values of the number of nodes. The difference between the number of constraints of the D-m-flow and the m-flow-2 models is A as given in Table 1, which is comparatively small. This explains the overlapping curves for these two models in Figure 3. Similarly, the difference between the number of constraints of D-1-flow and D-level models is $n - 2$ which is relatively small.

To further demonstrate the differences between the models with respect to their number of constraints, consider an instance with 500 nodes. The proposed model has a number of constraints that is 374,752 as opposed to 250,001 and 250,500 of D-1-flow and D-level models, respectively. Meanwhile, D-m-flow and m-flow-2 models are above 124 million constraints and their LP relaxations are almost 125 million constraints, since in the LP relaxations additional bounding constraints for binary variables are added. For m-flow-1 model, this number exceeds 3 billion.

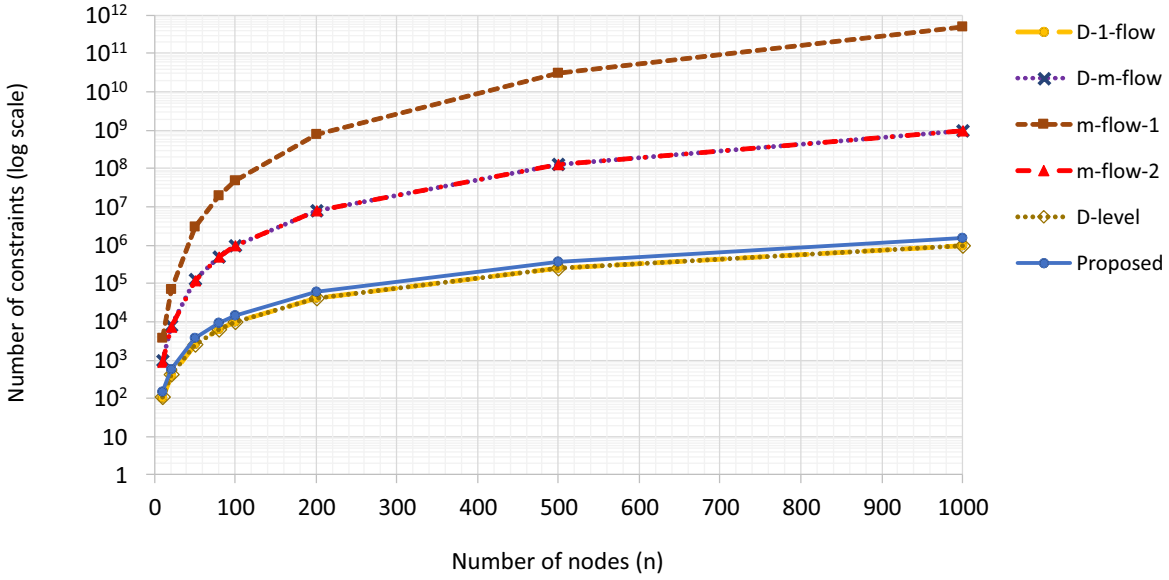

Figure 3. Comparison between selected models based on the number of constraints.

4.2. Computational Time

The computational time needed to prove optimality of a generated solution is an important criterion that needs to be considered in comparing mathematical programming models.

This comparison have to be done using a state-of-the-art software that implements efficient algorithmic techniques. IBM ILOG CPLEX version 12.8 fits this requirement as it implements concurrently a mix of primal/dual simplex and barrier methods for solving linear programming models. For MILP models, CPLEX implements branch-and-bound, branch-and-cut and dynamic search algorithms in addition to pre-processing algorithms for model reductions and heuristic methods for speeding up the search [13]. For linear programming models, an optimality check is performed at each step, and the iterations continue until a proof of optimality is reached. For MILP models, a proof of optimality is concluded when the lower bound equals the upper bound. All computational experiments in this paper were conducted on a laptop computer with Intel Core i7 processor running at 2.70 GHz with 2 cores, 4 logical processors and 8 GB physical memory, under Microsoft Windows 10.

Random instances are generated by assigning two random coordinates for each node. The arc length between any two nodes equals the Euclidean distance between them based on their coordinates. This approach maintains the triangular inequality property in the randomly generated instances, which is commonly found in most, if not all, applications. The randomly generated coordinates and the calculated distances are prepared in Microsoft Excel and loaded through the interface provided by IBM ILOG CPLEX Optimization Studio. The loading time for the data as well as the model loading time are taken into consideration in the recorded computational times.

Preliminary experiments with less than 20 nodes do not show any significant differences in the computational times needed to prove optimality of generated solutions based on the models presented in this paper. Since the number of constraints of the first to fifth models is quite large for instances with 20 or more nodes (exceeds half million constraints), they are not considered in the computational comparison. Since the LP relaxations of the seventh (D-m-flow) and eighth (m-flow-1) models, denoted LP (D-m-flow) and LP (m-flow-1), respectively, have integer polyhedrons, they are used in the computational comparisons instead of their original MILP models.

4.2.1. Results for 30-Node Instances

The first set of experiments is based on 30-node instances. The coordinates are randomly generated integers following a uniform distribution between 1 and 100. For 100 randomly generated instances, Table 2 presents some descriptive statistics for the total computational time in seconds, and Figure 4 provides a comparison between the Box and Whisker plots of the models. The logarithm to the base 10 of the computational times is presented in Figure 4 for a better clarity of the distribution of the points.

Based on the statistics provided in Table 2, it is evident that the proposed model provides the best performance in terms of the average computational time, followed by the LP relaxation of the D-m-flow model and the D-level model respectively, both with slight differences. However, the LP relaxation of the D-m-flow model has the lowest standard deviation which makes it more robust to changes in the input data. Despite the relatively large variability of the computational times of the D-level and the proposed models, they both have median computational times that are almost half the median computational time of the LP relaxation of the D-m-flow model. This indicates that, in most cases, both models need less computational times, which occurred in 87 and 86 instances for the D-level and the proposed models, respectively, as the detailed results reveal. Meanwhile, the proposed model achieved less computational times in 74 instances compared to the D-level model.

The statistics of the D-1-flow model indicate that it has both higher average and standard deviation values for the computational time when compared to the LP relaxation of the D-m-flow, the D-level and the proposed models. Even though it is capable in some cases to find an optimal solution faster than these three models, which occurred in 12 cases, its computational time in 12 cases was more than twice the computational time of any one of the other three models.

Table 2. Descriptive statistics of the computational times in seconds for the 30-node instances.

Model	Mean	Std. Dev.	Minimum	Median	Maximum
D-1-flow	7.59	14.3	1.76	3.22	75.8
LP (D-m-flow)	3.963	0.2642	3.482	3.959	5.013
LP (m-flow-1)	28.73	19.2	9.34	23.41	89.73
m-flow-2	319	1101	3	23	8117
D-level	4.087	6.958	1.664	2.113	56.926
Proposed	3.51	5.036	1.613	2.104	43.215

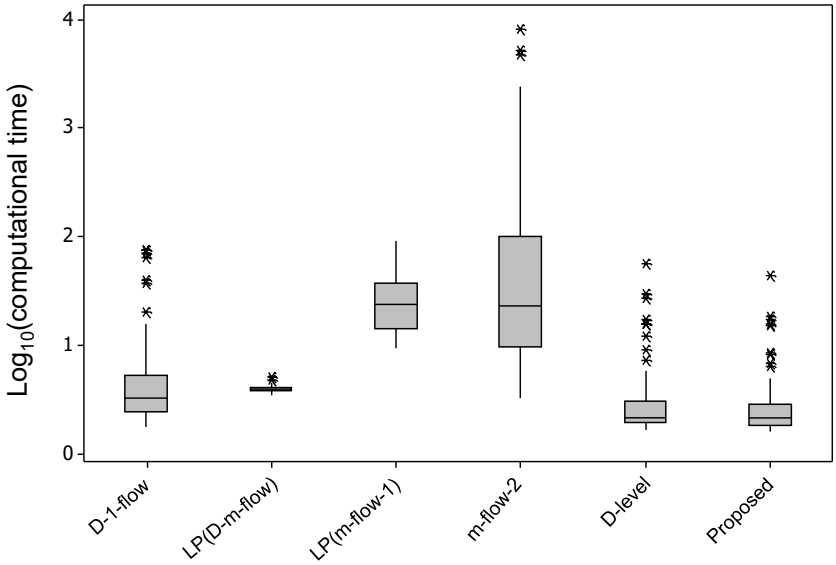

Figure 4. Box and Whisker plots for the distribution of computational times in seconds of selected models for the 30-node instances.

As shown in Figure 4, It is evident that the m-flow-2 model results in the worst computational time that went over two hours for one of the instances. Furthermore, its standard deviation of the computational time is the highest, which makes it unfavorable candidate when compared to the LP relaxation of the D-m-flow model which has slightly larger number of constraints. Similarly, even though the LP relaxation of the m-flow-1 model is a slight modification of the D-m-flow model, such a modification does not provide any benefit in terms of reducing the computational time.

4.2.2. Results for 40-Node Instances

The second set of experiments excludes both the LP relaxation of the m-flow-1 model and the m-flow-2 model from the comparison as based on the 30-node instances, they are not expected to provide competitive computational times. In the second set of experiments, 100 instances of 40 nodes are randomly generated. Similar to the previous set, node coordinates are random integers generated using uniform distribution between 1 and 100. Table 3 presents some descriptive statistics for the total computational time in seconds, and Figure 5 provides a comparison between the Box and Whisker plots of the models.

Table 3. Descriptive statistics of the computational times in seconds for the 40-node instances.

Model	Mean	Std. Dev.	Minimum	Median	Maximum
D-1-flow	210.2	664.3	2.3	12.6	4900.6
LP (D-m-flow)	10.603	1.253	7.904	10.527	13.708
D-level	12.62	27.95	2.18	4.93	254.53
Proposed	9.89	20.02	1.92	4.21	177.65

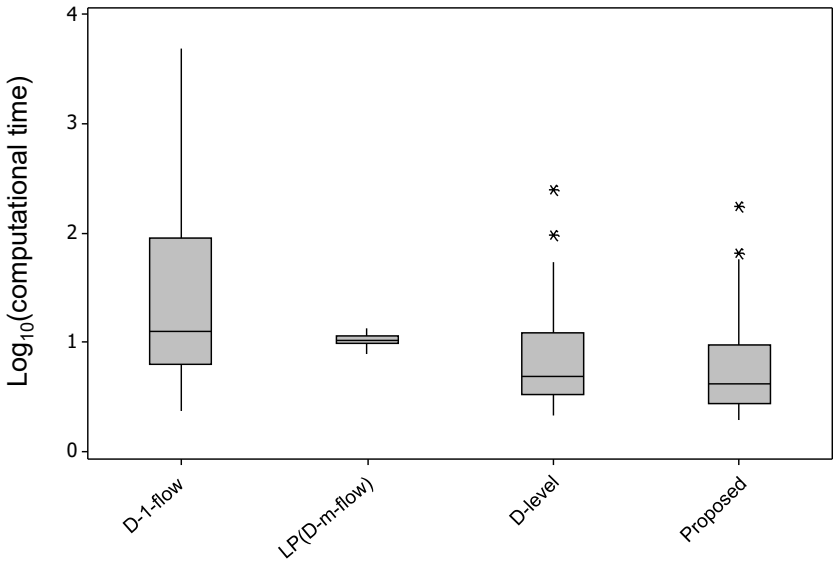

Figure 5. Box and Whisker plots for the distribution of computational times in seconds of selected models for the 40-node instances.

The results of the 40-node set is coherent with the previous 30-node set. As shown in Table 3, the proposed model provides the best average computational time followed by the LP relaxation of the D-m-flow model and the D-level model respectively. Meanwhile, the performance of the D-1-flow model is comparatively very poor on average, it has the highest median and standard deviation values, and its computational time exceeded 1.5 h in one instance.

As illustrated in Figure 5, the LP relaxation of the D-m-flow model has the lowest variability in computational time; nevertheless, its median is more than double the medians of the proposed and the D-level models, which conforms with the results of the 30-node instances. The detailed results reveal that the proposed model is capable of achieving less computational time than the LP relaxation of the D-m-flow model in 81 instances, and less than the D-level model in 78 instances.

4.2.3. Results for 50 and 60-Node Instances

The remaining computations in this paper are intended to compare the proposed model with the LP relaxation of the D-m-flow model. The advantage of the proposed model is attributed to its smaller size, yet the algorithmic techniques used for solving it are not polynomial in time. On the other hand, the latter model is easier to solve using polynomial time algorithms, but due to its larger size, the computational time needed can be larger than the proposed model. Therefore, the intention of conducting computational experiments with 50-node and more than 50-node instances is to investigate and compare the trends of the computational times for both models.

Starting with 50-node instances, the coordinates of the nodes are randomly generated integers uniformly distributed between 1 and 500. Increasing the scale for the generated coordinates compared to previous instances is done to reduce the chances of having duplicate nodes with the same coordinates. However, this does not have any effect on the computational time. Table 4 presents some descriptive statistics for the total computational time in seconds, and Figure 6 provides a comparison between the Box and Whisker plots of the two models for all problem sizes investigated so far.

The statistics provided in Table 4 show that the median of the computational times of the proposed model is a little bit higher than half the median of the LP relaxation of the D-m-flow model which is not significantly different from the results of smaller size instances. However, the proposed model results in an average computational time that is higher than that of the LP relaxation of the D-m-flow model which represents a shift from the results of smaller-size instances. However, in general, the relative distributions of the computational time results of the two models for the 50-node instances are not significantly different from those of smaller instances, as depicted in Figure 6.

Table 4. Descriptive statistics of the computational times in seconds for the 50-node instances.

Model	Mean	Std. Dev.	Minimum	Median	Maximum
LP (D-m-flow)	27.293	3.373	17.261	27.379	39.929
Proposed	29.82	54.52	2.36	14.29	421.84

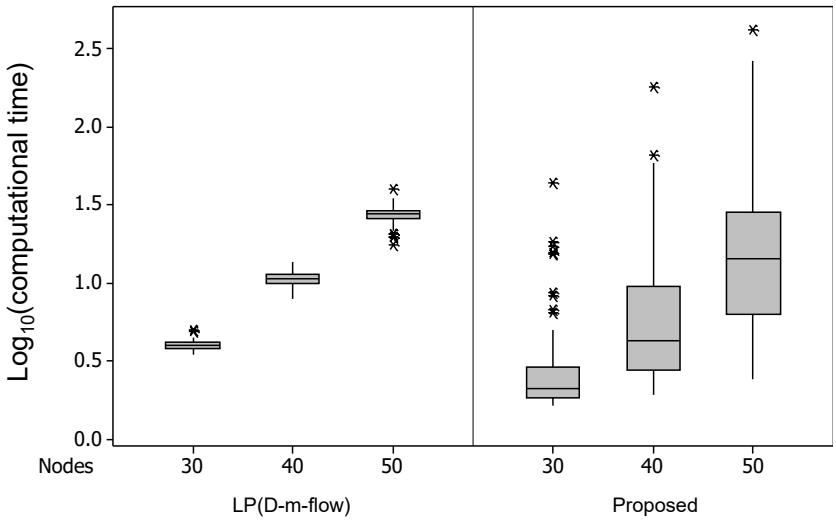

Figure 6. Box and Whisker plots for the distribution of computational times in seconds of the LP relaxation of the D-m-flow model and the proposed model for the 30-, 40- and 50-node instances.

For the 50-node instances, the proposed model achieved less computational times in 74 instances. Even though this number represents a relatively large proportion, it is less than the figures obtained for small size instances, which are 86 and 81 for the 30-node and 40-node instances, respectively. This signifies a downward trend for the proposed model.

To further investigate this trend, additional instances with 60 nodes are randomly generated. Similar to the 50-node instances, the coordinates of the nodes are randomly generated integers uniformly distributed between 1 and 500. only 30 randomly generated instances are sufficient to reveal the deterioration of the performance of the proposed model. It is capable of achieving less computational time in only 13 instances. The median of the computational time of the proposed model is found to be 108.2 s, while, for the LP relaxation of the D-m-flow model, it is found to be 71.49 s. Even though the proposed model can achieve relatively low computational times—less than 10 s in 10 cases—it takes more than 5 min in eight cases with slow convergence rate to a proven optimality. The LP relaxation of the D-m-flow model on the other hand is very stable in its computational time requirement with a mean of 73.42 s and relatively small standard deviation of 9.63 s.

5. Conclusions

Combinatorial optimization problems are characterized by logical relationships that can be represented mathematically using different models. This paper follows an assessment approach for such models that is based on two aspects. The first is the memory size needed; and the second is the computational time needed by a solution algorithm to prove optimality. This paper presents a node level MILP model for the famous combinatorial optimization problem of finding a minimum spanning tree in a given network. Despite the ten existing models in the literature and the polynomial time solution algorithms for this problem, introducing a model with a modified structure can provide more insights for modeling and solving larger problems in which the MST problem is a subproblem.

The proposed model is based on the analogy between rivers and trees, as both do not contain cycles. It is proven that the proposed model will result in valid minimum spanning trees. In terms of the model size, the proposed model has the third lowest number of constraints, which increases

polynomially with the number of nodes. Meanwhile, the number of decision variables of the proposed model is the second lowest, which also increases polynomially with the number of nodes.

To compare the proposed model with the other models in terms of the computational time needed to prove optimality of generated solutions, computational experiments were conducted using a state-of-the-art commercial MILP solver. The computational comparison focused only on models that have polynomial number of constraints. These computational experiments were done sequentially using increasing number of nodes in a graph to filter out inefficient models early. Starting with 30-node instances, two multi-commodity flow-based models were found to be inefficient and were excluded from the next set of 40-node instances. The computational results for the 40-node instances showed that the proposed model produces the best results and is competitive with a linear programming relaxation of a digraph multi-commodity flow model. These two models were investigated further using 50-node instances, for which the results showed that the proposed model achieved less computational time in 74 cases out of 100. However, since that figure is found to be decreasing when compared to the previous results of 30- and 40-node instances, additional experiments with 60-node instances were conducted. The latter results suggest that the proposed model may not be efficient with instances of more than 50 nodes. For larger instances, the computational results show slow convergence for the lower and upper bounds. On the other hand, the LP relaxation of the digraph multi-commodity flow model demonstrated stable computational time results; however, its size is relatively large.

Funding: This research received no external funding.

Acknowledgments: IBM ILOG is acknowledged for its academic initiative of providing a free-of-charge license to CPLEX version 12.8 which was used in the computations presented in this paper. The author is grateful to two anonymous reviewers whose comments and suggestions helped to improve an earlier version of this paper.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Borůvka, O. O jistém problému minimálním. *Práca Mor. Přirodověd. Spol.* **1926**, *3*, 37–58.
2. Kruskal, J.B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Am. Math. Soc.* **1956**, *7*, 48–50. [CrossRef]
3. Prim, R.C. Shortest connection networks and some generalizations. *Bell. Syst. Tech. J.* **1957**, *36*, 1389–1401. [CrossRef]
4. Gabow, H.N.; Galil, Z.; Spencer, T.; Tarjan, R.E. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica* **1986**, *6*, 109–122. [CrossRef]
5. Magnanti, T.L.; Wolsey, L. Optimal Trees. In *Handbooks in Operations Research and Management Science*; Ball, M.O., Magnanti, T.L., Monma, C., Nemhauser, G., Eds.; Elsevier Science: New York, NY, USA, 1995; Volume 7, Chapter 9, pp. 503–615.
6. Asano, T.; Bhattacharya, B.; Keil, M.; Yao, F. Clustering Algorithms Based on Minimum and Maximum Spanning Trees. In *Proceedings of the Fourth Annual Symposium on Computational Geometry, SCG '88*; ACM: New York, NY, USA, 1988; pp. 252–257. [CrossRef]
7. Yin, F.; Liu, C.L. Handwritten text line extraction based on minimum spanning tree clustering. In Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China, 2–4 November 2007; Volume 3, pp. 1123–1128.
8. Held, M.; Karp, R.M. The Traveling-Salesman Problem and Minimum Spanning Trees. *Oper. Res.* **1970**, *18*, 1138–1162. [CrossRef]
9. Abreu, A.; Frenois, F.X.; Valitutti, S.; Brousset, P.; Denefle, P.; Naegel, B.; Wemmert, C. Optimal cut in minimum spanning trees for 3-D cell nuclei segmentation. In Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis (ISPA), Ljubljana, Slovenia, 18–20 September 2017; pp. 195–199.
10. Guo, H.; Liu, L.; Chen, J.; Xu, Y.; Jie, X. Alzheimer Classification Using a Minimum Spanning Tree of High-Order Functional Network on fMRI Dataset. *Front. Neurosci.* **2017**, *11*, 639. [CrossRef] [PubMed]

11. Hafsi, Z.; Elaoud, S.; Mishra, M.; Akrout, M. Automated Framework for Water Looped Network Equilibrium. *Water Resour. Manag.* **2018**, *32*, 641–657. [CrossRef]
12. Wu, B.; Yu, B.; Wu, Q.; Chen, Z.; Yao, S.; Huang, Y.; Wu, J. An Extended Minimum Spanning Tree method for characterizing local urban patterns. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 450–475. [CrossRef]
13. CPLEX Optimizer. Available online: https://www.ibm.com/analytics/cplex-optimizer (accessed on 5 April 2018).
14. Van Hentenryck, P. *The OPL Optimization Programming Language*; MIT Press: Cambridge, MA, USA, 1999.
15. Schrage, L. *Optimization Modeling with LINGO*, 5th ed.; Lindo Systems Inc.: Chicago, IL, USA, 2002.
16. Graham, R.L.; Knuth, D.E.; Patashnik, O. *Concrete Mathematics*; Addison-Wesley: Reading, MA, USA, 1988; p. 244.
17. Bazaraa, M.S.; Jarvis, H.J.; Sherali, H.D. *Linear Programming and Network Flows*, 3rd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2005.
18. Miller, C.E.; Tucker, A.W.; Zemlin, R.A. Integer Programming Formulation of Traveling Salesman Problems. *J. ACM* **1960**, *7*, 326–329. [CrossRef]
19. Wolsey, L. *Integer Programming*; Wiley Series in Discrete Mathematics and Optimization; Wiley: Hoboken, NJ, USA, 1998.