Computers & Operations Research 37 (2010) 72 – 82

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

Min-degree constrained minimum spanning tree problem: New formulation via Miller–Tucker–Zemlin constraints

İbrahim Akgün*, Barbaros Ç. Tansel

Department of Industrial Engineering, Bilkent University, Bilkent 06800, Ankara, Turkey

ARTICLE INFO

Available online 24 March 2009

Keywords:
Mixed integer programming
Degree-enforcing constraints
Miller–Tucker–Zemlin constraints
Minimum spanning tree
Flow formulation
Rooted arborescence

ABSTRACT

Given an undirected network with positive edge costs and a positive integer $d>2$, the *minimum-degree constrained minimum spanning tree problem* is the problem of finding a spanning tree with minimum total cost such that each non-leaf node in the tree has a degree of at least d. This problem is new to the literature while the related problem with upper bound constraints on degrees is well studied. Mixed-integer programs proposed for either type of problem is composed, in general, of a *tree-defining* part and a *degree-enforcing* part. In our formulation of the minimum-degree constrained minimum spanning tree problem, the tree-defining part is based on the Miller–Tucker–Zemlin constraints while the only earlier paper available in the literature on this problem uses single and multi-commodity flow-based formulations that are well studied for the case of upper degree constraints. We propose a new set of constraints for the degree-enforcing part that lead to significantly better solution times than earlier approaches when used in conjunction with Miller–Tucker–Zemlin constraints.

1. Introduction

Minimum spanning tree (MST) problems arise quite naturally in transportation and communication network design when it is necessary to provide a minimum-cost connectivity among a number of geographically dispersed locations or system components. Various examples of minimum cost tree networks are given by Ahuja et al. [1] from network design in transportation, telecommunication, data storage, and cluster analysis. We consider in this paper a topology constrained version of the minimum spanning tree problem in which a minimum cost spanning tree is sought for while requiring that each node in the tree be either a leaf node or a central (non-leaf) node that is adjacent to at least d nodes.

The minimum-degree requirement for central nodes may arise in distribution networks when fixed charges associated with a facility may be large enough to suggest that at least a certain number of end-users be served by it to justify the opening and operation costs associated with it. Minimum-degree constraints in tree networks are also encountered in telecommunication networks in the process of designing local access networks that feed traffic between a main network and a large number of end-users (terminals) (e.g., Green [2]). Installation costs for network components (e.g., concentrators and computers) can be better justified when the number of nodes served by them is not less than an acceptable threshold. The problem is also of interest from a modeling and computational standpoint as approaches that are well studied for upper degree constrained problems do not necessarily work well for lower degree constrained problems.

To define the problem of interest, let $G=(V,E)$ be an undirected connected network with node set V, edge set E, and positive edge costs c_e $(e \in E)$. A spanning tree of G is a connected sub-graph of G that has no cycles and spans all nodes. Given a positive integer d, a spanning tree is a *minimum-degree constrained spanning tree* if the degree of each node relative to the tree is either 1 or at least d. Fig. 1 gives two such trees for $d=4$. From now on, we refer to minimum-degree constrained spanning trees as *feasible trees*. If $d \leqslant 2$, all spanning trees are feasible trees and the degree constraints can be ignored. The distinction between feasible and infeasible trees becomes important for $d>2$. Note that if $d \geqslant n \equiv |V|$, no feasible tree exists for G. We assume from this point on that $2<d<n$.

We refer to the problem of finding a minimum cost spanning tree of G as the *minimum spanning tree* problem and that of finding a minimum cost feasible tree as the *minimum-degree constrained minimum spanning tree (MDC-MST)* problem. While MST is solvable in low order polynomial time by the algorithms of Kruskal [3] and Prim [4], MDC-MST is proven to be NP-hard for $d \geqslant 4$

* Corresponding author. Fax: +90 312 2664054.
E-mail address: iakgun@bilkent.edu.tr (İ. Akgün).

0305-0548/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2009.03.006

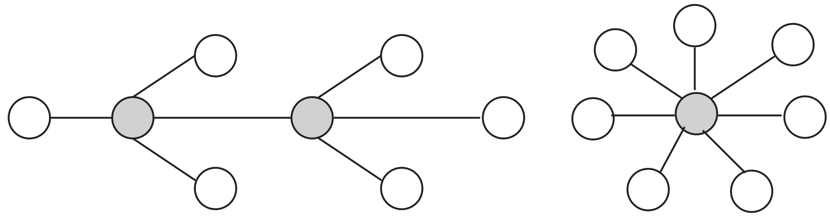

Fig. 1. Two feasible solutions for MDC-MST with $d = 4$.

by Almeida et al. [5]. The complexity status of this problem is open for $d = 3$.

The MDC-MST is new to the literature. To our knowledge, the first and the only study in the literature on MDC-MST is that of Almeida et al. [5] (referred to as AMS in the sequel). In their study, AMS introduced the problem, discussed its properties and complexity, presented some properties regarding the number of leaf and non-leaf nodes, gave single- and multi-commodity flow-based formulations for the problem, and present computational results for their formulations. Their reported results show that many of the test problems with up to 50 nodes can be solved within 3 h of CPU time, but there are also many test problems in the same test bed that remain unsolved within the 3 h limit.

We propose in this research new formulations that give substantially improved solution times for the same set of test problems. Additional improvement has been obtained by observing that the linear programming (LP) relaxations of the proposed formulations lead, in general, to tighter bounds if the root node for the tree is more judiciously selected. Based on this, we give a methodology to select a root node for the tree that significantly improves solution times for proposed and previous models.

MDC-MST is closely related to the degree-constrained minimum spanning problem (DC-MST) where the degree requirement for non-leaf nodes is an upper bound rather than a lower bound. DC-MST is proven to be NP-hard by Garey and Johnson [7]. Unlike MDC-MST, DC-MST is a well-studied problem. Some of the notable contributions on DC-MST are Deo and Hakimi [8], Savelsbergh and Volgenant [9], Zhou and Gen [10], Knowles and Corne [11], Caccetta and Hill [12], Ribeiro and Souza [13], Andrade et al. [14], and Krishnamoorthy et al. [15].

AMS compared DC-MST and MDC-MST by using flow-based formulations of both. They observed that flow-based formulations of MDC-MST show significantly poor performance with respect to LP bounds and solution times when compared to flow-based formulations of DC-MST. In this regard, MDC-MST appears to be quite elusive, at least when compared to DC-MST.

The remainder of this paper is organized as follows. Section 2 reviews flow-based models of AMS. Section 3 gives our proposed formulations. Section 4 gives our methodology to select the root node. Section 5 gives computational results and compares proposed and previous models. Section 6 concludes this paper.

2. Review of the models of AMS

Minimum spanning tree problems with additional restrictions on the structure of the tree (e.g., degree requirements) are formulated in general using two sets of constraints, one set ensuring that a spanning tree is obtained and the other set ensuring that the resulting tree satisfies the structural requirements. The structural requirement in the problem we study is the minimum-degree requirement on non-leaf nodes and will accordingly be referred to as *degree-enforcing* constraints. The remaining portion of minimum spanning tree formulations consists of constraints that ensure that the resulting set of arcs is a spanning tree. We refer to this portion of the formulations as *tree-defining* constraints. A number of different approaches are available for modeling spanning tree features including formulations based on packing, cut-sets, and flows (Magnanti and Wolsey [16]). Among different formulations, flow-based formulations seem to be a most preferred one because they are compact in the number of variables and that they, especially the multi-commodity versions, give a better representation of the spanning tree polyhedron [16]. Following this fact, AMS use directed single- and multi-commodity flow-based formulations to model spanning tree features.

Flow-based formulations are defined on a directed network $G' = (V, A)$ obtained from $G = (V, E)$ by replacing each undirected edge $\{i, j\} \in E$, where $i \neq j$, by two directed arcs (i, j) and (j, i) with symmetric costs $c_{ij} = c_{ji}$. A node r is selected as the *root* node and acts as a single source for the flow to be sent to the remaining $n-1$ nodes each of which acts as a sink node with a demand of one unit. In the single-commodity formulation, the root node has a supply of $n-1$ units of a commodity and sends them out into the network to satisfy the unit demand at each sink node. In the multi-commodity case, the $n-1$ demand nodes still have unit demands but each demand is for a different commodity and the root node has a supply of one unit of each commodity (e.g., Magnanti and Wolsey [16]). In either case, the set of arcs with positive flows in a feasible solution define an *arborescence* which is a directed tree such that every node other than the root node has exactly one incoming arc while the root node has no incoming arc.

In a feasible arborescence, if the root node is a leaf node, it has one outgoing arc but no entering arcs. Any demand node that is a leaf node has one incoming arc but no outgoing arcs. If the root node is a central node, it has d or more outgoing arcs but no incoming arcs while a demand node that is a central node has one incoming arc and at least $d-1$ outgoing arcs. Two example arborescences are shown in Fig. 2 for the case of $d = 4$.

AMS use three sets of decision variables to formulate MDC-MST: (1) binary design variables x_{ij} that take on the value of 1 if arc (i, j) is in the design and 0 otherwise, (2) binary node variables w_i that take on the value of 1 if node i is a central node and 0 if node i is a leaf node, and (3) non-negative flow variables y_{ij} specifying the amount of flow in arc (i, j) for the single-commodity flow formulation and the flow variables f_{ij}^k specifying the amount of flow sent from the root node r to demand node k passing through arc (i, j) for the multi-commodity flow formulation.

In the flow-based formulations that we give next, the degree enforcing constraints are constraints (2)–(5). We refer to constraints (2)–(5) as **DEF1** (the first set of degree-enforcing constraints). The remaining set of constraints, other than set restrictions and non-negativity, constitutes the tree-defining part of the formulation. The tree-defining part is different for single and multi-commodity formulations.

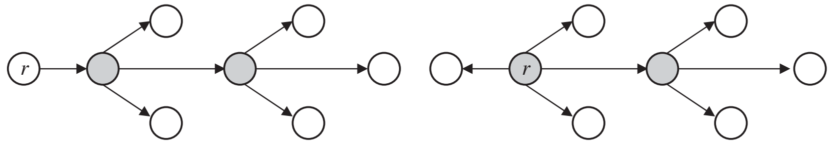

Fig. 2. Two feasible solutions on a directed graph with $d=4$.

SCF/DEF1: Single-commodity flow model with the first set of degree-enforcing constraints

$$z^* = \min_{x,w} \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_j x_{ij} \geqslant 1+(d-1)w_i, \quad i=r \tag{2}$$

$$\sum_j x_{ij} \leqslant 1+(n-2)w_i, \quad i=r \tag{3}$$

$$\sum_{j\neq r} x_{ij} \geqslant (d-1)w_i, \quad i\in(V-r) \tag{4}$$

$$\sum_{j\neq r} x_{ij} \leqslant (n-2)w_i, \quad i\in(V-r) \tag{5}$$

$$\sum_{i\neq j} x_{ij} = 1, \quad j\in(V-r) \tag{6}$$

$$\sum_i y_{ij} - \sum_{i\neq r} y_{ji} = 1, \quad j\in(V-r) \tag{7}$$

$$x_{ij} \leqslant y_{ij}, \quad i\in V,\ j\in(V-i-r) \tag{8}$$

$$y_{ij} \leqslant (n-1)x_{ij}, \quad i\in V,\ j\in(V-i-r) \tag{9}$$

$$x_{ij}\in\{0,1\}, \quad i\in V,\ j\in(V-i-r) \tag{10}$$

$$w_i\in\{0,1\}, \quad i\in V \tag{11}$$

$$y_{ij}\geqslant 0, \quad i\in V,\ j\in(V-i-r) \tag{12}$$

Objective function (1) minimizes the total cost of the arcs in the solution. DEF1 constraints are (2)–(5) and tree-defining constraints are (6)–(9). Constraints (2) and (3) and constraints (4) and (5) define lower and upper bounds on the number of outgoing arcs from the root node and non-root nodes, respectively. Constraints (6) require that the number of incoming arcs to any non-root node be equal to 1. Constraints (7) are flow-conservation constraints. Constraints (8) and (9) are coupling constraints requiring that any arc with a positive flow be in the design and that the amount of flow through an arc be bounded above by $n-1$. Even though this can be improved to $n-2$ for non-root nodes, we retain $n-1$ in (9) to be consistent with the form used by AMS. Constraints (10)–(12) give the appropriate set restrictions and non-negativity on the decision variables.

MCF/DEF1: Multi-commodity flow model with the first set of degree-enforcing constraints.

In addition to (1)–(6), (10), and (11),

$$\sum_{i\neq k} f_{ij}^k - \sum_{i\neq r} f_{ji}^k = 0, \quad j,k\in(V-r),\ j\neq k \tag{13}$$

$$\sum_i f_{ij}^j = 1, \quad j\in(V-r) \tag{14}$$

$$f_{ij}^k \leqslant x_{ij}, \quad i\in V,\ j,k\in(V-i-r) \tag{15}$$

$$f_{ij}^k \geqslant 0, \quad i\in V,\ j,k\in(V-i-r) \tag{16}$$

Constraints (13)–(15) together with constraints (6) are multi-commodity flow-based tree-defining constraints. Constraints (13) and (14) are flow-balance constraints and constraints (15) are coupling constraints. Note that constraints (13) and (14) and constraints (15) are commodity-distinguished versions of constraints (7) and constraints (8) and (9), respectively. Constraints (16) are non-negativity restrictions on flow variables.

AMS define two valid inequalities which are added to the models SCF/DEF1 and MCF/DEF1 to obtain a total of six different formulations (three for each). These valid inequalities are

$$x_{ij} \leqslant w_i, \quad i,j\in(V-r),\ i\neq j \tag{17}$$

$$\sum_{i\in V} w_i \leqslant \left\lfloor \frac{n-2}{d-1} \right\rfloor \tag{18}$$

Valid inequality (17) requires that a node be a central node if there is an outgoing arc from it while valid inequality (18) defines an upper bound on the number of central nodes in a solution. The validity of the upper bound is proven by AMS. We use these valid inequalities in our formulations as well. We refer to the version of DEF1 that includes the valid inequalities (17) and (18) as DEF1′.

As to the number of constraints and variables, SCF/DEF1′ has $3n^2-2n-1$ constraints, n^2-n binary variables, and n^2-n continuous variables while MCF/DEF1′ has n^3+n^2-n-1 constraints, n^2-n binary variables, and n^3-2n^2 continuous variables.

3. Proposed formulations for MDC-MST

In this section, we propose a new set of degree-enforcing constraints referred to as **DEF2**. We also propose to use Miller–Tucker–Zemlin (MTZ) [6] constraints for the tree-defining part as an alternative to single or multi-commodity flow constraints.

3.1. DEF2: the proposed set of degree-enforcing constraints

Let w_{ic} and w_{il} be a pair of binary variables associated with node i with $w_{ic}=1$ ($w_{il}=1$) if node i is a central (leaf) node and $w_{ic}=0$ ($w_{il}=0$) if not. DEF2 constraints are as follows:

$$w_{ic}+w_{il}=1, \quad i\in V \tag{19}$$

$$\sum_j x_{ij} \geqslant 1, \quad i=r \tag{20}$$

$$\sum_{j\neq r} x_{ij} \geqslant dw_{ic}+w_{il}, \quad i=r \tag{21}$$

$$\sum_{j\neq r} x_{ij} \leqslant 1+(n-2)w_{ic}, \quad i=r \tag{22}$$

$$\sum_{j\neq r} x_{ij} \geqslant (d-1)w_{ic}, \quad i\in(V-r) \tag{23}$$

$$\sum_j x_{ji} + \sum_{j\neq r} x_{ij} \geqslant d-(d-1)w_{il}, \quad i\in(V-r) \tag{24}$$

$$\sum_j x_{ji} + \sum_{j\neq r} x_{ij} \leqslant 1+(n-2)w_{ic}, \quad i\in(V-r) \tag{25}$$

$$x_{ij} \leqslant w_{ic}, \quad (i,j)\in A,\ i\neq r,\ j\neq r \tag{26}$$

$$x_{ij}+w_{il}+w_{jl} \leqslant 2, \quad (i,j)\in A,\ j\neq r \tag{27}$$

$$\sum_{i\in V} w_{ic} \leqslant \left\lfloor \frac{n-2}{d-1} \right\rfloor \tag{28}$$

$$x_{ij}\equiv 0, \quad (i,j)\in A,\ j=r \tag{29}$$

$$x_{ij}+x_{ji}\leqslant 1, \quad (i,j)\in A,\ i<j \tag{30}$$

$$\sum_{j\neq i} x_{ij} = n-1 \tag{31}$$

$$w_{ic}, w_{il}\in\{0,1\}, \quad i\in V \tag{32}$$

Constraints (19) require that each node be either a central node or a leaf node. Constraints (20)–(31) express some structural properties of a feasible solution. Constraints (20)–(22) define lower and upper bounds on the number of outgoing arcs from the root node. Constraint (20) establishes that the number of outgoing arcs at the root node r is at least 1. Constraints (21) and (22) require that the number of outgoing arcs at the root node be equal to 1 when r is a leaf node and be at least d and at most $n-1$ when r is a central node. Constraint (20) is actually redundant; however, it helps to improve the solution times. Constraints (21) and (22) are equivalent to constraints (2) and (3), respectively, in terms of the new node variables.

Constraints (23)–(25) set upper and lower limits on the degree of non-root nodes. Constraints (23) require, as constraints (4), that the number of outgoing arcs from a non-root node be at least $d-1$ if the node is a central node. Constraints (24) state that the total number of outward and inward arcs of each non-root node is at least d when a non-root node is a central node and at least 1 when a non-root node is a leaf node. Constraints (24) are similar to constraints (23), and hence to constraints (4), except that constraints (24) take into account both inward and outward arcs while constraints (23) take into account only outward arcs. Constraints (24) can be obtained by adding constraints (4) and (6) in terms of new variables, i.e., constraints (23) and (6). Because each non-root node is required to have exactly one incoming arc by constraints (6), constraints (24) are actually nothing more than adding the same terms to the left- and right-hand sides of constraints (4).

Constraints (25) restrict the number of inward and outward arcs of a non-root node to be at most 1 when the node is a leaf node and at most $n-1$ when the node is a central node. Constraints (25) can be obtained by adding constraints (5) and (6) in terms of new variables. Thus, the relationship between constraints (25) and (5) is similar to that between constraints (24) and (4).

Constraints (26) are exactly the valid inequalities (17) in terms of w_{ic}. They require that a non-root node be a central node if there is an outgoing arc from it. Note that constraints (5) can be obtained by summing both sides of constraints (26) over all nodes j adjacent to node i, i.e., constraints (26) are a disaggregated version of constraints (5).

Constraints (27) prevent arcs between pairs of leaf nodes. Constraints (28) are the valid inequalities (18) in terms of the new variables. Constraints (29) do not allow any arcs incoming to the root node. Constraints (30) state that a pair of arcs of opposite directions between a pair of nodes is not possible. This set of constraints is actually a set of valid inequalities. Constraints (31) require that the total number of arcs in the solution be equal to $n-1$, which is a known fact for a tree (e.g., [1]). Finally, constraints (32) give the zero/one restrictions on the decision variables w_{ic} and w_{il}.

Note that constraints (26) and (27), and (29)–(31) must be satisfied by any tree problem and are not particular, in this sense, to MDC-MST. They are not an essential part of degree-enforcing constraints, but we keep them there because their presence leads to better computational performance than their omission. Note that these constraints are not an essential part of tree-defining constraints, either.

New flow-based formulations for MDC-MST can easily be obtained by replacing DEF1 constraints (2)–(5) with DEF2 constraints (19)–(32). In fact, DEF2 (or any other set of degree-enforcing constraints) can be coupled with any set of tree-defining constraints to obtain a new formulation. For instance, MCF/DEF2, the multi-commodity flow-based model with the proposed set of constraints, is composed of the objective function (1) and constraints (6), (10), (13)–(16), and (19)–(32).

Proposition 1. *Let DEF1P and DEF2P be two different formulations of MDC-MST where DEF1 and DEF2 are used as degree-enforcing constraints in the two formulations, respectively, while all remaining constraints, including tree-defining constraints and integer restrictions, are common. Denoting by* $\mathrm{F}(\mathrm{P}_{\mathrm{LP}})$ *the set of feasible solutions of the LP relaxation of any integer linear programming problem P, we have* $\mathrm{F}(\mathrm{DEF2P}_{\mathrm{LP}}) \subseteq \mathrm{F}(\mathrm{DEF1P}_{\mathrm{LP}})$. *Accordingly, DEF2 dominates DEF1.*

Proof. Let $(x, y, w_c, w_l) \in \mathrm{F}(\mathrm{DEF2P}_{\mathrm{LP}})$ where x, y, w_c, and w_l are the vectors of variables x_{ij}, y_{ij} , w_{ic}, and w_{il}, respectively. Put $w = w_c$. We now prove $(x, y, w) \in \mathrm{F}(\mathrm{DEF1P}_{\mathrm{LP}})$. It suffices to show that (x, y, w) satisfies constraints (2)–(5) as the only constraints that are in $\mathrm{DEF1P}_{\mathrm{LP}}$ that are not included in $\mathrm{DEF2P}_{\mathrm{LP}}$ are these constraints. The feasibility of (x, y, w_c, w_l) to $\mathrm{DEF2P}_{\mathrm{LP}}$ implies that (x, y, w_c, w_l) satisfies the DEF2 constraints (19)–(32) as well as the tree-defining constraints (6)–(9). Constraint (2) is implied by (19), (21), and the fact that $w = w_c$. Constraint (3) is implied by (22) and $w = w_c$. Constraints (4) are implied by (23) and $w = w_c$. Constraints (5) are implied by (6), (25) and $w = w_c$. Hence, $(x, y, w) \in \mathrm{F}(\mathrm{DEF1P}_{\mathrm{LP}})$ and the proof is complete. □

We remark that the proof of the proposition is still valid if we change DEF1 to DEF1′ in the proposition. This follows from the fact that constraints (17) and (18) of DEF1′ are nothing but constraints (26) and (28) of DEF2, respectively, upon replacing w with w_c.

Proposition 2 is an immediate consequence of Proposition 1 and the foregoing remarks when the problem under consideration is taken to be a flow-based formulation, MCF or SCF.

Proposition 2.

(i) $\mathrm{F}(\mathrm{MCF/DEF2}_{\mathrm{LP}}) \subseteq \mathrm{F}(\mathrm{MCF/DEF1'}_{\mathrm{LP}}) \subseteq \mathrm{F}(\mathrm{MCF/DEF1}_{\mathrm{LP}})$.
(ii) $\mathrm{F}(\mathrm{SCF/DEF2}_{\mathrm{LP}}) \subseteq \mathrm{F}(\mathrm{SCF/DEF1'}_{\mathrm{LP}}) \subseteq \mathrm{F}(\mathrm{SCF/DEF1}_{\mathrm{LP}})$.

While it is possible to drop the variables w_{il} from DEF2 by replacing w_{ic} with w_i and w_{il} by $1 - w_i$, the presence of the variables w_{il} in DEF2 produces on the average better solution times than when they are absent. We attribute this to different branch-and-bound structures and cuts that may be generated by the solver when these variables are present than when they are absent.

Computational studies indicate that flow-based models with DEF2 show better performance with respect to both LP bounds and solution times than the ones with DEF1 and DEF1′. The solution times for problems solved to optimality are almost halved. In particular, MCF/DEF2 gives considerably better LP bounds than SCF/DEF1 or SCF/DEF1′ (discussed in Section 5). However, due to relatively high memory storage requirements of MCF/DEF2, it cannot solve most of the problems with 50 nodes within the 3-h limit of CPU time. SCF/DEF2 can solve more problems than MCF/DEF2; however, there still remain problems not solved within the allotted time. For these reasons, we use the Miller–Tucker–Zemlin constraints (Miller et al. [6]) as an alternative to flow-based formulations.

3.2. Formulations based on the Miller–Tucker–Zemlin sub-tour elimination constraints

Formulations based on MTZ constraints also create a rooted arborescence. In this regard, the directed network structure defined above is used.

To formulate MTZ constraints, in addition to the binary design variables x_{ij}, non-negative node-labeling variables u_i are used. These labels are assigned in such a way in any feasible solution that each directed arc included in the arborescence is directed from a node with a lower label into a node with a higher label. This ensures that the node labels form an increasing sequence on any directed path so that any node previously visited on a directed path cannot be re-visited, thereby preventing formation of sub-tours.

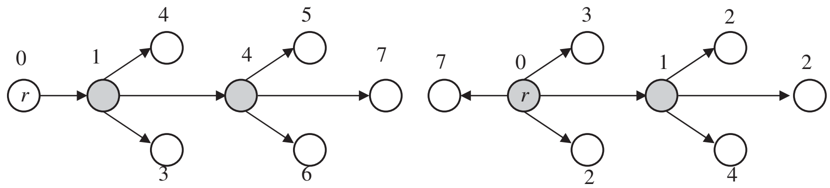

Fig. 3. Two feasible spanning trees with labels assigned by MTZ constraints.

The *basic* MTZ constraints [6] are given below. The term "basic" is used here because these constraints are changed later to obtain an improved version of these constraints.

BMTZ: Basic Miller–Tucker–Zemlin sub-tour elimination constraints

$$u_i - u_j + nx_{ij} \leqslant n-1, \quad (i,j) \in A,\ j \neq r \tag{33}$$

$$u_i \leqslant n-1, \quad i \in (V-r) \tag{34}$$

$$u_i \geqslant 1, \quad i \in (V-r) \tag{35}$$

$$u_i \equiv 0, \quad i = r \tag{36}$$

$$u_i \geqslant 0 \quad \forall i \tag{37}$$

MTZ constraints are originally defined for the traveling salesman problem (TSP) (Lawler et al. [17], Padberg and Sung [18], Nemhauser and Wolsey [19]). In the context of TSP, MTZ constraints eliminate all sub-tours that do not contain the base (root) node r by assigning *unique* labels u_i to nodes such that the label of a node represents the rank-order in which the node is visited in a traveling salesman tour. That is, base node r is assigned a label of 0 while the i-th node visited after node r is assigned a label of i. In our case, constraints (33) prevent sub-tours by ensuring that each arc included in the arborescence is directed from a lower labeled node to a higher labeled node. The uniqueness of node labels is not required. Constraint (36) assigns a label of 0 to the root node, while constraints (34) and (35) define upper and lower bounds on the labels that can be assigned to non-root nodes, respectively. In the original paper [6], the u_i variables are unrestricted. Bounds (34) and (35) are introduced later on.

Two new formulations of MDC-MST where the tree-defining part consists of MTZ constraints while the degree-enforcing part is either DEF1 or DEF2 are given below:

BMTZ/DEF1: Basic MTZ model with the first set of degree-enforcing constraints.
Objective function (1), constraints (2)–(6), (10) and (11), and (33)–(37).

BMTZ/DEF2: Basic MTZ model with the proposed set of degree-enforcing constraints.
Objective function (1), constraints (6), (10), and (19)–(37).

By specializing Proposition 1 to the Basic MTZ-based formulations, we have the following.

Proposition 3. $F(\text{BMTZ/DEF2}_{LP}) \subseteq F(\text{BMTZ/DEF1}'_{LP}) \subseteq F(\text{BMTZ/DEF1}_{LP})$.

In the context of TSP, $u_j = u_i + 1$ whenever $x_{ij} = 1$ given that $j \neq r$ and hence the whole range of label values is used. In our formulation of MDC-MST, the fact that the same label value may be assigned to more than one node results in not using the whole range of label values. This actually allows feasible solutions with different labeling structures. For example, a feasible solution where the same label is not assigned to all nodes at the same distance from the root is possible. Specifically, in the assignment of labels to nodes, there are three possible cases for an edge $\{i,j\}$: either $x_{ij} = 1$, or $x_{ji} = 1$, or both $x_{ij} = 0$ and $x_{ji} = 0$. If $x_{ij} = 1$, then $u_j \geqslant u_i + 1$. Similarly, if $x_{ji} = 1$, then $u_i \geqslant u_j + 1$. If both $x_{ij} = 0$ and $x_{ji} = 0$, then $u_i - u_j \leqslant n-1$ and $u_j - u_i \leqslant n-1$. In this respect, any assignment of labels satisfying the aforementioned conditions gives a feasible solution. Two example feasible solutions are given in Fig. 3.

The special structure of MDC-MST allows us to make some improvements in the MTZ constraints that improve both the LP bounds and the solution times. These improvements are obtained in two steps.

In the first step, constraints are added to allow feasible solutions with a certain labeling structure. In a feasible solution of MDC-MST, each node is either a central node or a leaf node. Because a non-root leaf node has one incoming arc whose origin is necessarily a central node, then a feasible solution can be obtained by requiring that the labels of all non-root leaf nodes be greater than the highest possible label of central nodes. This condition is easily fulfilled if we assign the label value $n-1$ to each non-root leaf node while permitting central nodes to take label values of at most $n-2$. If all nodes other than the root node are leaf nodes, then the root node receives the node label 0 and all other nodes receive node labels of $n-1$. If there is a non-root central node, then its label will be between 1 and $n-2$. Thus, in finding feasible solutions for MDC-MST, looking only for solutions in which the label values of non-root leaf nodes are restricted to $n-1$ and the label values of central nodes are restricted to be less than or equal to $n-2$ is sufficient.

In the second step, the range of labels is restricted to a certain interval so that the feasible region of the linear programming relaxation is further decreased. Recalling that the number of central nodes is bounded above by $\left\lfloor \frac{n-2}{d-1} \right\rfloor$, it is direct to conclude that the labels of non-root central nodes may be restricted to the interval from $n-1-\left\lfloor \frac{n-2}{d-1} \right\rfloor$ to $n-2$. The label of the root node is not included in this interval because it is not known a priori if the root node will be a central node or not. Thus, the label of the root node is set to $n-1-\left\lfloor \frac{n-2}{d-1} \right\rfloor - 1$. Two such feasible solutions with $d=4$ are given in Fig. 4. Note that non-root central node in the graph on the right can also take on the value of 6.

We now give **IMTZ**, the *Improved* MTZ constraints.

IMTZ: Improved Miller–Tucker–Zemlin sub-tour elimination constraints

In addition to (33) and (34), and (37),

$$u_i \geqslant (n-1)w_{il}, \quad i \in (V-r) \tag{38}$$

$$u_i \leqslant (n-1) - w_{ic}, \quad i \in (V-r) \tag{39}$$

$$u_i \equiv (n-1) - \left\lfloor \frac{n-2}{d-1} \right\rfloor - 1, \quad i = r \tag{40}$$

$$u_i \geqslant (n-1) - \left\lfloor \frac{n-2}{d-1} \right\rfloor, \quad i \in (V-r) \tag{41}$$

Constraints (38), together with constraints (34) that give an upper bound on the node labels u_i, establish that the labels of all non-root leaf nodes are equal to $n-1$. Constraints (39) restrict the labels of central nodes to be at most $n-2$. Constraint (40) sets the label of the root node to $n-1-\left\lfloor \frac{n-2}{d-1} \right\rfloor - 1$. Constraints (41) require that the label values of non-root nodes be at least $n-1-\left\lfloor \frac{n-2}{d-1} \right\rfloor$.

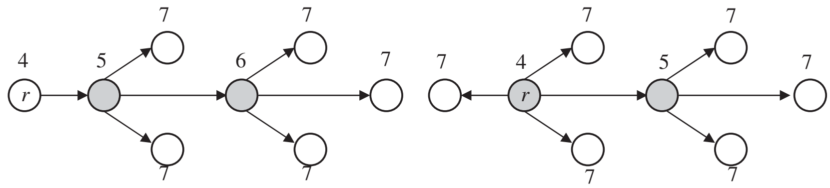

Fig. 4. Two feasible spanning trees with labels assigned by MTZ constraints with $d=4$.

Table 1
Characteristics of test problems.

Pr. ID	Pr. type	$\|V\|$	d	Instance	$r=m^*$
1	SYM	30	3	1	29
2	SYM	30	3	2	10
3	SYM	30	3	3	17
4	SYM	30	5	1	29
5	SYM	30	5	2	10
6	SYM	30	5	3	17
7	SYM	50	3	1	5
8	SYM	50	3	2	6
9	SYM	50	3	3	12
10	SYM	50	5	1	5
11	SYM	50	5	2	6
12	SYM	50	5	3	12
13	SYM	50	10	1	5
14	SYM	50	10	2	6
15	SYM	50	10	3	12
16	CRD	30	3	1	21
17	CRD	30	3	2	16
18	CRD	30	3	3	20
19	CRD	30	5	1	21
20	CRD	30	5	2	16
21	CRD	30	5	3	20
22	CRD	50	3	1	32
23	CRD	50	3	2	42
24	CRD	50	3	3	26
25	CRD	50	5	1	32
26	CRD	50	5	2	42
27	CRD	50	5	3	26
28	CRD	50	10	1	32
29	CRD	50	10	2	42
30	CRD	50	10	3	26

Proposition 4. *Let BMTZP and IMTZP be two different formulations for MDC-MST where BMTZ and IMTZ are used as tree-defining constraints, respectively, together with a set of degree-enforcing constraints, e.g., DEF1 or DEF2. Then,* $F(\text{IMTZP}_{LP}) \subseteq F(\text{BMTZP}_{LP})$, *i.e., IMTZP dominates BMTZP.*

Proof. We note first that all constraints of IMTZP and BMTZP are alike except that (35) and (36) in BMTZP are replaced by (38)–(41) in IMTZP. Consider now any feasible solution (x, u, w_c, w_l) to IMTZP_{LP} where u includes all node variables except u_r which is just a constant defined by (40). This constant is replaced by another constant defined by (36) in BMTZP. The solution (x, u, w_c, w_l) satisfies all constraints of BMTZP_{LP} since the range $[1, n-1]$ for node labels $u_i (i \neq r)$ in BMTZP_{LP} includes the range $\left[(n-1) - \left\lfloor \frac{n-2}{d-1} \right\rfloor, n-2\right]$ imposed on central nodes by constraints (39) and (41) in IMTZP_{LP} as well as the range $[n-1, n-1]$ imposed on leaf nodes by constraints (34) and (38) in IMTZP_{LP}. This implies $(x, u, w_c, w_l) \in F(\text{BMTZP}_{LP})$ and completes the proof. □

Due to Proposition 4, the LP polytope of IMTZP is a subset of the LP polytope of BMTZP. Computational studies in Section 5 (Table 1) verify this fact empirically.

We give below two new formulations of MDC-MST where the tree-defining part consists of IMTZ:

IMTZ/DEF1: Improved MTZ model with the first set of degree-enforcing constraints.
Objective function (1), constraints (2)–(6), (10) and (11), (33) and (34), and (37)–(41).

IMTZ/DEF2: Improved MTZ model with the proposed set of degree-enforcing constraints.
Objective function (1), constraints (6), (10), (19)–(32), (33) and (34), and (37)–(41).

As a corollary to Propositions 1 and 3, we can state the following proposition.

Proposition 5.

(i) $F(\text{IMTZ/DEF2}_{LP}) \subseteq F(\text{IMTZ/DEF1}'_{LP})$.
(ii) $F(\text{IMTZ/DEF1}'_{LP}) \subseteq F(\text{BMTZ/DEF1}'_{LP})$.
(iii) $F(\text{IMTZ/DEF2}_{LP}) \subseteq F(\text{BMTZ/DEF2}_{LP})$.

IMTZ/DEF2 has $3.5n^2 + 5.5n - 4.5$ constraints, $n^2 + n$ binary variables, and n continuous variables and is much more compact than MCF/DEF1 with respect to the number of variables and constraints. On the other hand, IMTZ/DEF2 has more constraints but fewer variables than SCF/DEF1.

A feasible solution requires that each non-root node has exactly one inward arc, which is provided by constraints (6). However, computational studies show that better solution times are obtained by using them in "$\geqslant$" form, i.e., $\sum_i x_{ij} \geqslant 1$. In this regard, all solution times for models using DEF2 are obtained by using this form. The inequality form of (6) is well justified by the presence of constraint (31) that limits the number of arcs in the solution to $n-1$. Without (31), a solution resulting from the inequality form of constraints (6) may violate the tree structure if the arc costs do not satisfy the triangle inequality and if the whole range of labels is not used, but this will not occur with (31).

MTZ constraints are attractive due to their compactness. However, they are well known for producing weak LP relaxation bounds. Orman and Williams [20] compared the strengths of several formulations of TSP by their LP relaxation bounds. They found that the LP relaxation polytope obtained by MTZ constraints contains some of the seven existing formulations. Specifically, the formulation with MTZ constraints gives weaker LP bounds than the ones based on single- and multi-commodity flow formulations. This has led to various studies that augment the MTZ constraints to strengthen the LP bounds (e.g., Desrochers and Laporte [21]; Gouveia [22]; Gouveia and Pires [23]; Sherali and Driscoll [24]). Although most studies focus on the TSP or TSP-related problems, the formulations or liftings in those studies can be adapted to other problems where sub-tours are not allowed. For instance, Gouveia [22] used MTZ constraints in the context of hop-constrained MST (HMST) where each path starting from the root is required to have at most a fixed number of hops (arcs) and offers liftings to constraints (33)–(35). We try those liftings and the ones by Desrochers and Laporte [21] in our study as well. The

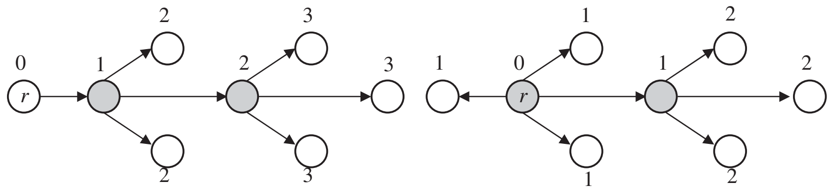

Fig. 5. Two feasible spanning trees with $u_j = u_i + 1$ whenever $x_{ij} = 1$.

liftings contribute to increase the LP relaxation bounds significantly for BMTZ/DEF1 and IMTZ/DEF1 but do not increase or slightly increase (under 1%) the LP bounds for BMTZ/DEF2 and IMTZ/DEF2. However, the liftings do not help to improve the solution times, which is in compliance with what Gouveia [22] has obtained for HMST. Because the contributions of the liftings to LP bounds and computation times are very marginal for DEF2, we decide not to use the liftings in our models to better assess the computational effects of our proposed formulations. Note also that by augmenting MTZ constraints (e.g., [21]), it is actually possible to have $u_j = u_i + 1$ when $x_{ij} = 1$. In this case, solutions such as the ones given in Fig. 5 are obtained. However, this prevents us from using constraints (38), i.e., the first improvement suggested above.

4. Root node selection

Clearly, the optimal solution values of MDC-MST instances do not change depending on the root node. However, the LP relaxation bounds or how the solver proceeds may change, affecting the solution times significantly. Of the models of AMS, only MCF/DEF1 is symmetric relative to the root and hence the LP relaxation bounds are the same for all roots [5]. All other models given in the paper are not symmetric and hence the LP bounds may change. AMS empirically showed that solution times may change significantly even when the formulation is symmetric and suggested that the selection of the root node may be of importance. However, they do not propose any methodology to select the root node. They test the performance of their models by selecting the first node as the root node. In our studies, we obtain results for two different root nodes, namely the first node and the node selected by a new methodology proposed herein.

The methodology consists of (1) finding the smallest three values in each row of the cost matrix, (2) finding the sum of the three smallest values in each row, and (3) selecting the node corresponding to the row with the smallest sum found in step (2). The methodology is based on the idea that arcs with lowest costs are likely to be in the solution. Empirical results show that the solution times are improved significantly for the models of AMS and IMTZ/DEF2 when the root node is selected with the proposed methodology. The root nodes m^* calculated by using the methodology are given in the last column in Table 1.

5. Computational studies

Computational studies are performed by using specially structured, hard CRD and SYM instances which are complete graphs with Euclidean costs set to integer units (e.g., [15]). CRD instances are 2-dimensional Euclidean problems where the points are generated randomly with a uniform distribution in a square. SYM instances are analogous to CRD instances but with points generated in higher dimensional Euclidean space. These problems have been widely used in the literature to test DC-MST (e.g., [9,25,26]).

Following AMS, CRD and SYM instances defined on networks with 30 and 50 nodes are used in our computational studies. For each network size, three different instances are tested for different values of d. Table 1 summarizes the characteristics of test problems.

Computational tests are performed on a PC with a 3.0 GHz Intel Core 2 duo processor and 3 GB of RAM by using ILOG CPLEX 9.0. The models are run until optimality is attained or for 3 h (10,800 CPU s) at maximum and by using default settings of CPLEX (e.g., moving the best bound strategy for branching is used, cuts are allowed, see [27]) except that file storage is set to 3, which allows tree file to be stored on the hard disk when it reaches the default limit in order not to run out of memory. To compare our results to those of AMS, we have modeled and solved the models of AMS on the same PC.

In the tables presenting computational studies, LP relaxation bounds, run times, optimal objective function values, and relative optimality gaps are given. Relative optimality gap is defined as $|BP - BF|/(1^{-10} + |BP|)$, where BP is the objective function value of the best integer solution and BF is the best remaining objective function value of any unexplored node (see [27]). Underlined values in the tables show that the problem is not solved to optimality within the allotted time of 10,800 s.

5.1. Comparison of MTZ-based models among themselves

Table 2 gives computational results for BMTZ/DEF1, BMTZ/DEF2, IMTZ/DEF1, and IMTZ/DEF2 for $r = 1$.

In terms of LP bounds, the results show that the weakest LP bounds are obtained for BMTZ/DEF1. The bounds for IMTZ/DEF1 are better than those for BMTZ/DEF1 implying that IMTZ constraints are stronger than BMTZ constraints. The difference between BMTZ and IMTZ when they are used with DEF1 is not observed when they are used with DEF2. Both BMTZ/DEF2 and IMTZ/DEF2 give the same LP bounds which are much better than those obtained from IMTZ/DEF1. The fact that both basic and improved versions of MTZ give the same bounds when used with DEF2 indicates that DEF2 dominates and overshadows any contributions that might have been coming from the improved structure of IMTZ over BMTZ. The fact that IMTZ/DEF2 (as well as BMTZ/DEF2) produces much better LP bounds than IMTZ/DEF1 indicates that the major contribution to the improvement in the LP bounds comes from DEF2. This shows that DEF2 is significantly stronger in producing LP bounds than DEF1.

In terms of solution times, IMTZ/DEF2 gives the best performance with respect to CPU times. Even though the performance of BMTZ/DEF2 is close to that of IMTZ/DEF2 for SYM instances, the superior performance of IMTZ/DEF2 in CPU time becomes more pronounced for CRD instances. In this regard, we take IMTZ/DEF2 as the main model to compare with the flow-based models.

5.2. Comparison of degree-enforcing constraints based on LP bounds

Table 3 gives LP relaxation bounds for MCF/DEF1′, MCF/DEF2, BMTZ/DEF1, and BMTZ/DEF2 (IMTZ/DEF2). The last two columns clearly indicate that, as explained in Section 5.1, DEF2 is strongly better than DEF1 when used with MTZ constraints. The columns for MCF/DEF1′ and MCF/DEF2 give further evidence for the dominance of DEF2 over DEF1. Its dominance becomes more apparent as the degree requirement d increases.

Table 2
Solution times, integrality gaps, and LP relaxation bounds for BMTZ/DEF1, BMTZ/DEF2, IMTZ/DEF1, and IMTZ/DEF2 with $r=1$.

Pr. ID	BMTZ/DEF1				IMTZ/DEF1			
	BP	Gap (%)	Time (s)	z_{LP}	BP	Gap (%)	Time (s)	z_{LP}
1	1197	0.00	2.02	752.21	1197	0.00	1.41	761.73
2	1435	0.00	0.42	1060.66	1435	0.00	0.16	1071.53
3	1408	0.00	0.09	1142.45	1408	0.00	0.11	1149.40
4	1765	0.00	1.38	752.21	1765	0.00	2.13	779.00
5	2090	0.00	1.42	1060.66	2090	0.00	1.47	1091.25
6	2008	0.00	1.39	1142.45	2008	0.00	3.14	1162.00
16	4026	0.00	9473.05	2915.17	4026	0.00	2794.86	2948.00
17	3796	6.93	10,800.00	2505.90	3793	6.26	10,800.00	2521.67
18	4293	0.00	256.09	3109.59	4293	0.00	90.19	3152.13
19	5026	0.00	1315.56	2915.17	5026	0.00	229.73	3007.50
20	4648	0.00	108.73	2505.90	4648	0.00	47.25	2550.25
21	5425	0.00	267.78	3109.59	5425	0.00	258.83	3229.25

Pr. ID	BMTZ/DEF2				IMTZ/DEF2			
	BP	Gap (%)	Time (s)	z_{LP}	BP	Gap (%)	Time (s)	z_{LP}
1	1197	0.00	0.22	1148.50	1197	0.00	0.11	1148.50
2	1435	0.00	0.05	1395.00	1435	0.00	0.05	1395.00
3	1408	0.00	0.06	1390.00	1408	0.00	0.06	1390.00
4	1765	0.00	1.20	1645.57	1765	0.00	1.33	1645.57
5	2090	0.00	1.38	1928.27	2090	0.00	1.14	1928.27
6	2008	0.00	0.16	1967.92	2008	0.00	0.13	1967.92
7	1278	0.00	21.64	1227.50	1278	0.00	27.73	1227.50
8	1178	0.00	0.77	1120.25	1178	0.00	0.92	1120.25
9	1615	0.00	16.91	1576.50	1615	0.00	2.20	1576.50
10	2054	0.00	7.08	1840.23	2054	0.00	10.06	1840.23
11	1760	0.00	3.53	1639.70	1760	0.00	3.73	1639.70
12	2525	0.00	24.14	2340.57	2525	0.00	26.66	2340.57
13	4121	0.00	8.61	3724.49	4121	0.00	7.47	3724.49
14	4166	0.00	10.81	3628.90	4166	0.00	9.66	3628.90
15	4979	0.00	30.27	4373.40	4979	0.00	21.20	4373.40
16	4026	0.00	5176.73	3582.50	4026	0.00	2009.00	3582.50
17	3848	6.57	10,800.00	3091.50	3796	3.45	10,800.00	3091.50
18	4293	0.00	25.44	3842.50	4293	0.00	53.83	3842.50
19	5026	0.00	38.83	4482.36	5026	0.00	35.47	4482.36
20	4648	0.00	18.42	4135.02	4648	0.00	12.36	4135.02
21	5425	0.00	34.00	4929.83	5425	0.00	13.50	4929.83
22	5522	4.30	10,800.00	4838.17	5525	4.15	10,800.00	4838.17
23	5813	1.33	10,800.00	5239.00	5814	1.19	10,800.00	5239.00
24	5590	0.00	2865.11	5130.67	5590	0.00	1891.31	5130.67
25	6971	3.63	10,800.00	6072.89	6915	1.88	10,800.00	6072.89
26	7204	0.00	1413.20	6646.51	7204	0.00	1030.52	6646.51
27	7279	1.58	10,800.00	6511.21	7279	1.36	10,800.00	6511.21
28	9633	0.00	25.52	8928.31	9633	0.00	20.11	8928.31
29	9743	0.00	22.53	9347.30	9743	0.00	15.94	9347.30
30	9855	0.00	34.17	9413.84	9855	0.00	20.95	9413.84

Underlined values show that the problem is not solved to optimality.

The dominance of DEF2 over DEF1 can be better assessed by comparing the columns for MCF/DEF2 and BMTZ/DEF2. Even though MCF constraints are known to give much stronger representation of the spanning tree polytope than that of the MTZ constraints (e.g., Orman and Williams [20]), BMTZ/DEF2 becomes competitive with MCF/DEF2 especially for SYM instances. For CRD instances, MCF/DEF2 gives better LP relaxation bounds than BMTZ/DEF2. In this regard, the dominance of MCF constraints over BMTZ is not compensated for by DEF2. However, DEF2 considerably decreases the gap between the LP relaxation bounds of MCF- and MTZ-based formulations implying its strength over DEF1.

5.3. Comparison of results for different root nodes

Table 4 gives LP relaxation bounds of MCF/DEF1′, MCF/DEF2, and IMTZ/DEF2 for $r=1$ and $r=m^*$, i.e., the methodology-selected root node. The table demonstrates that LP relaxation bounds obtained with $r=m^*$ are better in general than the ones obtained with $r=1$. That is, the methodology does not guarantee a node with the best LP bound; computational studies show that LP bounds are on the average better (sometimes the best) at least for the problems studied. However, Table 5 indicates that significant improvements in solution times of IMTZ/DEF2 are realized for $r=m^*$, for which more details are given in Section 5.4. As Table 6 demonstrates, the solution times of MCF/DEF1 are also significantly improved, implying that using $r=m^*$ improves the solution times of the flow-based models as well, especially for harder CRD instances. For example, the solution times of 734.23, 2480.11, and 5769.39 s for Pr. 16, 20, and 21, respectively, are improved to 88.72, 888.61 and 3097.06 s.

5.4. Comparison of IMTZ/DEF2 and flow-based models with respect to solution times

Table 5 gives computational results for IMTZ/DEF2 and the flow-based models of AMS. In reporting the computational results for the flow-based models, results are not given for each model separately.

Table 3
Comparison of degree-enforcing constraints by LP relaxation bounds with $r=1$.

Pr. ID	MCF		BMTZ	
	DEF1′	DEF2	DEF1	DEF2 (IMTZ/DEF2)
1	1112.70	1148.63	752.21	1148.50
2	1395.25	1395.25	1060.66	1395.00
3	1391.33	1393.67	1142.45	1390.00
4	1598.76	1647.62	752.21	1645.57
5	1935.14	1939.46	1060.66	1928.27
6	1930.10	1968.30	1142.45	1967.92
7	1223.50	1235.84	920.33	1227.50
8	1127.50	1127.50	893.88	1120.25
9	1589.50	1589.50	1251.71	1576.50
10	1748.77	1840.23	920.33	1840.23
11	1647.89	1662.87	893.88	1639.70
12	2325.89	2352.78	1251.71	2340.57
13	3500.07	3724.49	920.33	3724.49
14	3444.14	3647.60	893.88	3628.90
15	4258.41	4374.45	1251.71	4373.40
16	3761.65	3764.33	2915.17	3582.50
17	3601.50	3613.67	2505.90	3091.50
18	4124.50	4124.50	3109.59	3842.50
19	4626.44	4626.96	2915.17	4482.36
20	4294.35	4437.56	2505.90	4135.02
21	4922.35	5034.61	3109.59	4929.83
22	5202.50	5202.50	3786.33	4838.17
23	5365.43	5456.25	3999.45	5239.00
24	5286.08	5391.18	3369.61	5130.67
25	6300.55	6340.89	3786.33	6072.89
26	6692.43	6762.99	3999.45	6646.51
27	6568.32	6694.90	3369.61	6511.21
28	8682.90	9120.48	3786.33	8928.31
29	9202.11	9391.19	3999.45	9347.30
30	9301.64	9482.44	3369.61	9413.84

Table 4
LP relaxation bounds of different models for different root nodes.

Pr. ID	MCF/DEF1′		MCF/DEF2		IMTZ/DEF2	
	$(r=1)$	$(r=m^*)$	$(r=1)$	$(r=m^*)$	$(r=1)$	$(r=m^*)$
1	1112.70	1135.35	1148.63	1135.35	1148.50	1133.17
2	1395.25	1395.25	1395.25	1395.25	1395.00	1395.00
3	1391.33	1408.00	1393.67	1408.00	1390.00	1406.00
4	1598.76	1658.35	1647.62	1658.35	1645.57	1653.03
5	1935.14	1929.23	1939.46	1966.72	1928.27	1944.10
6	1930.10	1945.67	1968.30	1950.47	1967.92	1950.47
16	3761.65	3819.90	3764.33	3819.90	3582.50	3684.00
17	3601.50	3605.37	3613.67	3605.37	3091.50	3085.50
18	4124.50	4076.00	4124.50	4076.00	3842.50	3889.00
19	4626.44	4720.77	4626.96	4724.13	4482.36	4591.56
20	4294.35	4414.88	4437.56	4428.45	4135.02	4101.42
21	4922.35	4957.21	5034.61	5012.76	4929.83	4914.17

For $r=m^*$, see Table 1.

In all cases, the best objective function value (BP), the run time, and the optimality gap of the model with the best results (the smallest solution time for the problems solved to optimality and the smallest optimality gap for the problems not solved to optimality) are reported.

Computational results show that the flow-based models of AMS can optimally solve all 12 problem instances with 30 nodes by at least one of their formulations. The solution times change from 0.91 to 14.00 s for SYM instances and from 67.42 to 734.23 s for CRD instances. Regarding problems with 50 nodes, flow-based models of AMS cannot optimally solve 6 instances, all of which are CRD instances, out of 18 within the allotted time. For 12 solved problems (of which 9 are SYM and 3 are CRD instances), the solution times change from 113.88 to 1701.56 s for SYM instances and from 124.16 to 661.77 s for CRD instances. These results show that the solution times of SYM instances are much better than those of CRD instances.

Computational results for IMTZ/DEF2 show that, when the root node is the first node, IMTZ/DEF2 can solve all instances with 30 nodes except Pr. 17. For all solved problems except Pr. 16, the solution times are incomparably better than those of AMS. The solution times change from 0.05 to 1.14 s for SYM instances and from 12.36 to 53.83 s for CRD instances. As to the problems with 50 nodes, IMTZ/DEF2 can solve all instances solved by AMS with incomparably much better solution times. The solution times change from 0.92 to 27.73 s for SYM instances and from 15.94 to 20.95 s for CRD instances. IMTZ/DEF2 cannot solve four instances, which are also not

Table 5
Solution times and integrality gaps.

Pr. ID	SCF/MCF with DEF1 or DEF1′ ($r=1$)			IMTZ/DEF2 ($r=1$)			IMTZ/DEF2 ($r=m^*$)		
	BP	Gap (%)	Time (s)	BP	Gap (%)	Time (s)	BP	Gap (%)	Time (s)
1	1197	0.00	7.48	1197	0.00	0.11	1197	0.00	0.14
2	1435	0.00	7.91	1435	0.00	0.05	1435	0.00	0.05
3	1408	0.00	1.78	1408	0.00	0.06	1408	0.00	0.05
4	1765	0.00	14.00	1765	0.00	1.33	1765	0.00	1.53
5	2090	0.00	4.92	2090	0.00	1.14	2090	0.00	1.11
6	2008	0.00	0.91	2008	0.00	0.13	2008	0.00	0.63
7	1278	0.00	917.98	1278	0.00	27.73	1278	0.00	0.44
8	1178	0.00	532.05	1178	0.00	0.92	1178	0.00	0.53
9	1615	0.00	217.16	1615	0.00	2.20	1615	0.00	0.52
10	2054	0.00	824.25	2054	0.00	10.06	2054	0.00	10.08
11	1760	0.00	1031.64	1760	0.00	3.73	1760	0.00	2.22
12	2525	0.00	1701.56	2525	0.00	26.66	2525	0.00	41.03
13	4121	0.00	113.88	4121	0.00	7.47	4121	0.00	5.80
14	4166	0.00	226.84	4166	0.00	9.66	4166	0.00	8.70
15	4979	0.00	664.47	4979	0.00	21.20	4979	0.00	24.05
16	4026	0.00	734.23	4026	0.00	2009.00	4026	0.00	15.06
17	3793	0.00	508.25	3796	3.45	10,800.00	3793	0.00	945.16
18	4293	0.00	124.20	4293	0.00	53.83	4293	0.00	6.86
19	5026	0.00	237.50	5026	0.00	35.47	5026	0.00	4.03
20	4648	0.00	67.42	4648	0.00	12.36	4648	0.00	7.17
21	5425	0.00	91.84	5425	0.00	13.50	5425	0.00	6.91
22	5594	5.15	10,800.00	5525	4.15	10,800.00	5522	2.67	10,800.00
23	5826	8.79	10,800.00	5814	1.19	10,800.00	5813	0.68	10,800.00
24	5681	4.80	10,800.00	5590	0.00	1891.31	5590	0.00	400.53
25	6964	7.03	10,800.00	6915	1.88	10,800.00	6915	0.00	1980.20
26	7230	3.01	10,800.00	7204	0.00	1030.52	7204	0.00	549.17
27	7286	5.35	10,800.00	7279	1.36	10,800.00	7277	0.00	3421.89
28	9633	0.00	661.77	9633	0.00	20.11	9633	0.00	28.61
29	9743	0.00	216.11	9743	0.00	15.94	9743	0.00	15.28
30	9855	0.00	124.16	9855	0.00	20.95	9855	0.00	14.56

Underlined values show that the problem is not solved to optimality. For $r=m^*$, see Table 1.

Table 6
Solution times and integrality gaps of flow-based models for different root nodes.

Pr. ID	MCF/DEF1 ($r=1$)			MCF/DEF1 ($r=m^*$)		
	BP	Gap (%)	Time (s)	BP	Gap (%)	Time (s)
1	1197	0.00	44.84	1197	0.00	31.53
2	1435	0.00	12.50	1435	0.00	12.53
3	1408	0.00	13.81	1408	0.00	5.61
4	1765	0.00	225.27	1765	0.00	127.83
5	2090	0.00	257.36	2090	0.00	203.14
6	2008	0.00	88.69	2008	0.00	63.98
16	4026	0.00	734.23	4026	0.00	88.72
17	3793	0.00	502.63	3793	0.00	148.02
18	4293	0.00	56.94	4293	0.00	80.33
19	5026	0.00	3336.22	5026	0.00	2157.56
20	4648	0.00	2480.11	4648	0.00	888.61
21	5425	0.00	5769.39	5425	0.00	3097.06

For $r=m^*$, see Table 1.

solved by AMS. However, IMTZ/DEF2 can solve two CRD instances unsolved by AMS. The solution times for those problems are 1030.52 and 1891.31 s, which are also incomparably better.

Computational results for IMTZ/DEF2 show that, when the root node is selected by using the proposed methodology, the solution times obtained with the first node being root node are improved significantly. For example, the solution time of 2009.00 s for Pr. 16 is improved to 15.06 s and Pr. 17 not solved in 10,800 s is solved in 945.16 s. In this case, four problems out of the six that are not solved by AMS are now solved to optimality with solution times changing from 400.53 to 3421.89 s. For unsolved problems, IMTZ/DEF2 with $r=m^*$ has smaller optimality gaps than flow-based models and IMTZ/DEF2 with $r=1$. Moreover, it has better objective function values and lower bounds. Specifically, the lower bounds for Pr. 22 are 5306, 5295.76, and 5374.72 for flow-based model, IMTZ/DEF2 with $r=1$, and IMTZ/DEF2 with $r=m^*$, respectively. For Pr.23, the lower bounds are 5314.01, 5744.66, and 5773.27. In this regard, because increasing the lower bounds constitutes most of the solution time, it is highly likely that optimality will be reached in shorter times when $r=m^*$.

When the results are considered as a whole, it is observed that the solution times of IMTZ/DEF2 are incomparably better than those of the flow-based models in general. This combined with the fact that the test problems are specially structured leads us to conclude that IMTZ/DEF2 can solve larger instances with average difficulty in reasonable times.

Even though MCF/DEF2 is tighter than IMTZ/DEF2, this dominance is not reflected in the solution times. This is probably due to the large number of constraints and variables in MCF/DEF2. We think that IMTZ/DEF2 solves the test instances appropriately because it is much more compact than MCF/DEF2 and the highly developed solution procedures for linear integer programs in CPLEX facilitate its solution. In this regard, IMTZ/DEF2 may be more appropriate to use when there is a good optimization package to use.

6. Conclusions

This paper studies the MDC-MST which consists of finding a spanning tree with minimum total cost such that each node $i \in V$ either has a degree of at least d or is a leaf node. The paper proposes a new set of degree-enforcing constraints and proposes to use the Miller–Tucker–Zemlin sub-tour elimination constraints as an alternative to single or multi-commodity flow constraints for the tree-defining part of the formulation. Various formulations can be obtained by coupling a degree-enforcing set with a tree-defining set. Our computational tests indicate that the proposed degree-enforcing constraints are significantly stronger than the earlier degree-enforcing constraints in terms of LP bounds as well as in solution times. The best performance is obtained from a coupling of the proposed set of degree-enforcing constraints with an improved version of Miller–Tucker–Zemlin constraints. This last model gives incomparably better solution times than those proposed earlier in the literature. Additional improvements in computational times are obtained by a more judicious choice of the root node for which we also give a method of selection.

Acknowledgments

The authors are grateful to Pedro Martins, the coauthor of [5], for providing datasets used in computational studies and to an anonymous referee for providing constructive feedback that has helped improve in major ways the presentation of the material in this paper.

References

[1] Ahuja RK, Magnanti TL, Orlin JB. Network flows. New Jersey: Prentice-Hall; 1993.
[2] James HG. The Irwin handbook of telecommunications. 5th ed., New York: McGraw-Hill; 2006.
[3] Kruskal JB. On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematics Society 1956;7(1):48–50.
[4] Prim R. Shortest connection networks and some generalizations. Bell System Technical Journal 1957;36:1389–401.
[5] Almeida AM, Martins P, Souza MC. Min-degree constrained minimum spanning tree problem: complexity, proprieties, and formulations. Working paper, 2006 ⟨http://cio.fc.ul.pt/files/6.2006.pdf⟩.
[6] Miller C, Tucker A, Zemlin R. Integer programming formulation of traveling salesman problems. Journal of ACM 1960;7:326–9.
[7] Garey MR, Johnson DS. Computers and intractability: a guide to the theory of *NP*-completeness. San Francisco: W.H. Freeman; 1979.
[8] Deo N, Hakimi SL. The shortest generalized Hamiltonian tree. In: Proceedings of the 6th annual Alberton conference, 1968. p. 879–88.
[9] Savelsbergh M, Volgenant T. Edge exchanges in the degree-constrained minimum spanning tree problem. Computers and Operations Research 1985;12:341–8.
[10] Zhou G, Gen M. Approach to the degree-constrained minimum spanning tree problem using genetic algorithms. Engineering Design and Automation 1997;3(2):156–65.
[11] Knowles JD, Corne DW. A new evolutionary approach to the degree-constrained minimum spanning tree problem. IEEE Transactions on Evolutionary Computation 2000;4(2):125–34.
[12] Caccetta L, Hill SP. A branch and cut method for the degree-constrained minimum spanning tree problem. Networks 2001;37(2):74–83.
[13] Ribeiro CC, Souza MC. Variable neighborhood search for the degree-constrained minimum spanning tree problem. Discrete Applied Mathematics 2002;118(1–2):43–54.
[14] Andrade R, Lucena A, Maculan N. Using Lagrangian dual information to generate degree constrained spanning trees. Discrete Applied Mathematics 2006;154(5):703–17.
[15] Krishnamoorthy M, Ernst AT, Sharaiha YM. Comparison of algorithms for the degree constrained spanning tree. Journal of Heuristics 2001;7:587–611.
[16] Magnanti T, Wolsey L. Optimal trees. In: Ball MO, Magnanti TL, Monma CL, Nemhauser GL, editors. Network models, Handbooks in operations research and management science, vol. 7. Amsterdam: North-Holland; 1995. p. 503–615.
[17] Lawler E, Lenstra J, Kan A, Shmoys D. The traveling salesman problem. New York: Wiley; 1992.
[18] Padberg M, Sung T. An analytical comparison of different formulations of the traveling salesman problem. Mathematical Programming 1992;52:315–57.
[19] Nemhauser GL, Wolsey LA. Integer and combinatorial optimization. New York: Wiley; 1999.
[20] Orman AJ, Williams HP. A survey of different integer programming formulations of the traveling salesman problem. Preprint series # OR101, Faculty of Mathematical Studies, University of Southampton, Southampton, England, 1999.
[21] Desrochers M, Laporte G. Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints. Operations Research Letters 1991;10:27–36.
[22] Gouveia L. Using the Miller–Tucker–Zemlin constraints to formulate minimal spanning trees with hop constraints. Computers and Operations Research 1995;22:959–70.
[23] Gouveia L, Pires JM. The asymmetric travelling salesman problem and a reformulation of the Miller–Tucker–Zemlin constraints. European Journal of Operational Research 1999;112:134–46.
[24] Sherali HD, Driscoll PJ. On tightening the relaxations of Miller–Tucker–Zemlin formulations for asymmetric traveling salesman problems. Operations Research 2002;50:656–69.
[25] Narula SC, Ho CA. Degree constrained minimum spanning tree. Computers and Operations Research 1980;7:239–49.
[26] Volgenant T. A Lagrangian approach to the DCMST problem. European Journal of Operational Research 1989;39:325–31.
[27] ILOG/CPLEX 9.0, Reference manual, 1997.