TECHNISCHE UNIVERSITÄT WIEN
Institut für Computergraphik und Algorithmen

Spanning Trees with Variable Degree Bounds

Luis Gouveia, Pedro Moura, Mario Ruthmair, Amaro Sousa

Forschungsbericht / Technical Report

TR–186–1–14–02

May 21, 2014

Favoritenstraße 9-11 / E186, A-1040 Wien, Austria
Tel. +43 (1) 58801-18601, Fax +43 (1) 58801-18699
www.cg.tuwien.ac.at

Spanning Trees with variable degree bounds[☆]

L. Gouveia[a], P. Moura[a,*], M. Ruthmair[b], A. Sousa[c]

[a] *CIO-DEIO, Bloco C6-Piso4, Faculdade de Ciências da Universidade de Lisboa, Cidade Universitária, Campo Grande, 1749-016 Lisboa, Portugal*

[b] *Institute of Computer Graphics and Algorithms, Vienna University of Technology, Vienna, Austria*

[c] *Instituto de Telecomunicações, Universidade de Aveiro, 3810-193, Aveiro, Portugal*

Abstract

In this paper, we introduce and study a generalization of the degree constrained minimum spanning tree problem where we may install one of several available transmission systems (each with a different cost value) in each edge. The degree of the endnodes of each edge depends on the system installed on the edge. We also discuss a particular case that arises in the design of wireless mesh networks (in this variant the degree of the endnodes of each edge depend on the transmission system installed on it as well as on the length of the edge). We propose three classes of models using different sets of variables and compare from a theoretical perspective as well as from a computational point of view, the models and the corresponding linear programming relaxations. The computational results show that some of the proposed models are able to solve to optimality instances with 100 nodes and different scenarios.

Keywords: OR in telecommunications networks, spanning tree, degree constraints, wireless mesh networks

1. Introduction

The Degree Constrained Minimum Spanning Tree Problem (DCMSTP) is a well known variant of the classical Minimum Spanning Tree problem. The DCMSTP contains additional constraints imposing a maximum value on the degree of the nodes (see for example [4, 6, 13]). Another variant imposing a minimum degree in all nodes except the leaves has been proposed in [1].

In this paper, we introduce and study a generalization of the DCMSTP where we may install one of several available transmission systems (each with a different cost value) in each edge. The degree of the endnodes of each edge depends

[☆] This work is supported by National Funding from FCT - Fundação para a Ciência e a Tecnologia, under the project: PEst-OE/MAT/UI0152 and by the Vienna Science and Technology Fund (WWTF), under the project ICT10-027.

* Corresponding author

Email addresses: legouveia@fc.ul.pt (L. Gouveia), pmmoura@fc.ul.pt (P. Moura), ruthmair@ads.tuwien.ac.at (M. Ruthmair), asou@ua.pt (A. Sousa)

on the system installed on the edge. We also discuss a particular case of the problem introduced here that arises in the design of wireless mesh networks. In this variant the degree of the endnodes of each edge depend on the transmission system installed on it as well as on the length of the edge (shorter edges and/or with a better transmission system allow higher degrees on its endpoints).

The paper is organized as follows. Section 2 describes the new problem and subsection 2.1 describes the variant arising in the context of the design of wireless mesh networks. In section 3, we describe three classes of models for the problems and compare them from a theoretical perspective. In section 4 we present computational results for the general variant as well as for the wireless based variant to compare the models in terms of the Linear Programming gaps and running times to obtain the optimal solutions. Finally, section 5 summarizes the main conclusions of this work.

2. Description and motivation of the problem

Consider an undirected graph $G = (X, E)$ where $X = \{1, \ldots, n\}$ represents the set of network nodes and $E \subseteq X^2$ is the set of edges $\{i, j\}$, representing possible network links (we denote by $E(i)$ the set of edges incident in node i). We assume that S is the set of available types of transmission systems that may be used in the network design solution. For each link $\{i, j\}$ and each transmission system $s \in S$, we associate a cost C_{ij}^s and a maximum degree D_{ij}^s of its endnodes i and j (typically, $D_{ij}^{s+1} > D_{ij}^s$ and $C_{ij}^{s+1} > C_{ij}^s$). Note that for a given transmission system s, the values D_{ij}^s and C_{ij}^s may differ for different pairs of nodes i and j. Also, it may happen that for some pairs, i and j, a given system s is not available and, in fact, E is the set of pairs of nodes such that at least one of the available transmission systems can be used.

We aim to find a "minimum" cost tree that satisfies the required degree constraints. Note that, for each edge such that more than one transmission system can be used, we have the option of installing a more expensive transmission system, allowing both endnodes to have higher degrees, or alternatively the option of installing a lower cost transmission system constraining the degree of the endnodes to be lower.

Note that, when we only have one transmission system, say $s = s^*$, and $D_{ij}^{s^*}$ is the same for all links $\{i, j\}$, we obtain the DCMSTP mentioned in the previous section. Thus, the problem as described is *NP*-Hard (see [9]).

Next, we explain that this problem is closely related to the design of point-to-point wireless networks when C_{ij}^s is constant for all $\{i, j\}$ such that the transmission system $s \in S$ can be used.

2.1. Wireless networks variant

In the network design of *point-to-point* wireless mesh networks, each link is implemented through a point-to-point wireless system composed by a pair of transmitter/receiver antennas and signal processing units (one at each endnode of the link) working on a frequency channel, chosen from a possible set of channels. A wireless system has always an associated distance range (*i.e.*, maximum distance between antennas), defined in its technical specification, which is roughly the same for all wireless channels and, in general, systems with higher distance ranges are more expensive. Therefore, given a set S of available wireless systems, the ones that can be installed on link $\{i, j\}$ are the ones whose distance range is not lower than the line of sight (*i.e.*, with no wireless obstacles between them) distance between i and j.

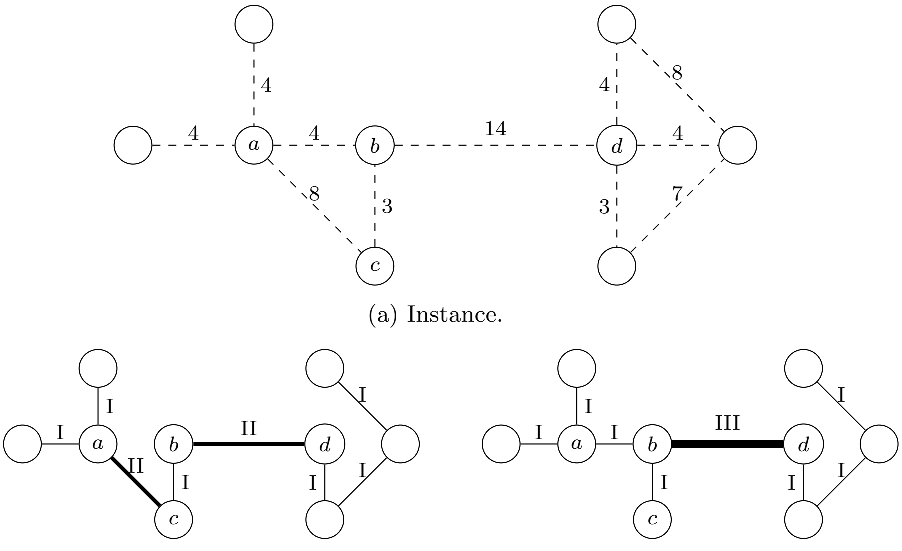
The distance range of a wireless system, though, always assumes no wireless interference from other sources. In current wireless technologies, due to the scarcity of the spectrum, there is a limited set of available frequency channels and many of them are partially overlapped between each other. For example, in IEEE 802.11 WiFi wireless mesh technologies, there is a total of 13 frequency channels, numbered from 1 to 13, but the maximum number of non-overlapped channels is three (for example, channels 1, 6 and 11) [12]. If the network is configured with only non-overlapping channels, the maximum degree of the solution is quite constrained, which might not be a problem if graph G is dense but might be unfeasible when graph G is sparse.

In this paper, we address the variant with a possible overlapping set of frequency channels on each node. The usage of overlapping channels let the number of channels used by the links starting/ending on the same node to be higher (and, therefore, the node degree can be higher on a feasible solution) but the adjacent channel interference must be taken into consideration (see, for example, [14]). A node with wireless links for different neighbor nodes uses different frequency channels. In a node using partially overlapped adjacent channels to different neighbor nodes, part of the transmitted signal on one channel is added as interference to the received signal on the other channels. The effect of this interference is that the distance range of the wireless systems is shortened. Note that if more wireless links are set on the edges adjacent to a node, the frequency channels must be closer between each other and more adjacent interference is added to each wireless system.

The maximum amount of interference of the other channels can be used to determine the resulting reduced distance range for each wireless system belonging to S for each possible node degree value (see [19, 5] for methods and models to estimate adjacent-channel interference). Then, for each pair of network nodes i and j, we can determine the wireless systems that can still be installed based on the distance between i and j and for each possible node degree value.

For example, a given system link s, with cost f_s and with a distance range of 15 when there is no interference, may have the distance range reduced to 5 if one (or both) of the endnodes of the link where we want to install the system has a degree of 3 or 4 or might not work at all if one of the endnodes has a degree larger than 4. In this case, if we want to install a system of type s on the link between a given pair of nodes i and j, whose distance is 10, for example, then we have $C_{ij}^s = f_s$ and $D_{ij}^s = 2$. On the other hand, if the distance between i and j is 4, for example, then we have $C_{ij}^s = f_s$ and $D_{ij}^s = 4$. Finally, the system s cannot be used if the distance between i and j is 20, for example.

Consider the example in Figure 1 where the values associated with each link on Figure 1a represent the link distances and the maximum degree (given by the number of overlapping frequencies that the operator may use) is 3. In this example, there are three available system types costing 5 (type I), 9 (type II) and 12 (type III). In links with a distance value less than 5, all system types can be used with the maximum degree in their endnodes. In links with a distance value between 5 and 10, systems of type I require a maximum degree of 2. Finally, in links with distance value between 10 and 15, systems of type I cannot be used (their distance range is lower) and systems of type II require a maximum degree of 2.



(a) Instance.

(b) Solution with a cost = 48.

(c) Solution with a cost = 47.

Figure 1: Solutions example.

If we try to use a system of type I in the link $\{b, d\}$ there is no solution. If we use a system of type II in this link, the best solution has a cost of 48 (see the solution in Figure 1b). If we upgrade the system installed on this link, using a system of type III, we obtain a solution with a lower cost of 47 (see solution in Figure 1c) by replacing link $\{a, c\}$ (with a system of type II installed), by the link $\{a, b\}$ (with a system of type I installed). The upgrade of the system on the link $\{b, d\}$ allowed the degree of node b to be increased from 2 to 3.

In this problem, we assume that interference is critical only between channels used on the same node. In fact, a channel used on a wireless link between node i and j might also produce co-channel interference on other nodes whose wireless links (not involving i and j) are set on the same frequency channel. We consider, though, that such interference is negligible since, in general, the directionality of the antennas concentrates the wireless signal power in the direction towards the receiver antenna and attenuates strongly the signal towards other directions.

Note that the complete design of wireless mesh networks involves a larger set of issues like node location or channel assignment (see [3] in applying mathematical optimization models in the design of WLANs). In this paper, we assume that node location is already decided. Although the variant addressed in this paper may be viewed as a simpler version of the problem since no channel assignment is performed, it still contains node degree constraints on the pair of network nodes that can be connected by a wireless system, that have not been considered before in the combinatorial optimization/network design area.

3. Formulations

In this section we describe three classes of integer linear formulations for the problem. The formulations described in this section contain a set of variables and inequalities that is common to all of them. These constraints are described in section 3.1 and contain the binary variables $x_{\{i,j\}}$ indicating whether edge $\{i,j\} \in E$ is included in the solution and binary variables y_i^d indicating whether node $i \in X$ has degree equal to $d \in \{1,\ldots,D\}$ in the solution. These variables and similar constraints have already been used in the models introduced and described in [7, 10, 11] for problems with non-linear costs associated to the node degrees.

The first class of formulations (described in section 3.2) uses additional binary variables $v^m_{\{i,j\}}$ indicating whether the edge $\{i,j\} \in E$ is selected and the maximum degree of nodes i and j is m (with $m = 2,\ldots,D$). These variables are sufficient to describe the objective function of the problem since the extra index permits us to define the cost of the system to be installed between these two nodes. In fact, in our models we use the C^s_{ij} and D^s_{ij} parameters (see section 2) to model the cost structure in the following way: for each link $\{i,j\}$, the value $c^m_{\{i,j\}}$, gives the cost of the cheapest cost transmission system that can be used, provided that m is the maximum between the degrees of nodes i and j. On the other hand, as will be noted in section 3.2, constraints linking the variables $v^m_{\{i,j\}}$ with the variables y_i^d are not as straightforward as the similar corresponding linking constraints for the models that use other sets of variables with more information (as in sections 3.3 and 3.4). We can consider the variables $v^m_{\{i,j\}}$ as *symmetric* in the sense that we have no information on which node the maximum degree is attained. In fact, it is this lack of information that leads to "clumsy" inequalities relating the two sets of variables.

The second class of formulations (described in section 3.3) uses "asymmetric" binary variables $t^{mk}_{\{i,j\}}$ where indexes i, j and m have the same meaning as in the definition of the $v^m_{\{i,j\}}$ variables, and the extra index k indicates the node (i or j) where the maximum degree is obtained. In some sense, the new variables provide more information than the v variables as an "arc" variable provides more information than a corresponding "edge" variable. As we shall see in the computational results on section 4, the additional information leads to drastic improvements in the value of the linear programming bounds.

The third class of formulations (described in section 3.4) can be seen, in a certain way, as a natural follow up of previous works by the authors (see in [7, 10, 11]). The formulations use *edge-degree* variables that provide information on the degree of the two nodes, that is, binary variables $z^{pq}_{\{i,j\}}$ indicating whether the edge $\{i,j\} \in E$ is selected and the degrees on node i and j are p and q, respectively. Although leading to models with more variables, the information on the degree of both nodes permits us to write equality constraints linking the new variables and the degree variables y_i^d, leading to a valid model with much fewer constraints and which may be preferable from a computational point of view to obtain the optimal integer solutions. We will also show that the information attached to the new variables permits us to derive a set of straightforward valid inequalities that lead to a model which dominates the linear programming relaxation of the model with the strongest linear programming bound from the second class. However, our results will also show that the linear programming bounds provided by these models are better, but in certain cases only slightly better, than the ones provided by the best model of the second class. Thus it is not clear that they should be preferred over the ones of the second class.

3.1. Common part

Consider the binary variables $x_{\{i,j\}}$ indicating whether edge $\{i,j\} \in E$ is included in the solution and binary variables y_i^d indicating whether node $i \in X$ has degree equal to $d \in \{1,\ldots,D\}$ in the solution. The three formulations studied in this paper include the set of constraints (1.1) - (1.5).

Constraints (1.2) and (1.3) define the degree variables y_i^d and guarantee that $y_i^d = 1$ *iff* the number of edges adjacent to node i is equal to d. Constraints (1.4) and (1.5) define the domain of the variables. Constraints (1.1), stating that the solution defined in the $x_{\{i,j\}}$ variables is a spanning tree, are still written in a generic form and can be modeled in several ways (see, for instance, [16]).

$$\big\{\{i,j\} \in E : x_{\{i,j\}} = 1\big\} \text{ is a spanning tree} \tag{1.1}$$

$$\sum_{d=1}^{D} d \cdot y_i^d = \sum_{\{i,j\}\in E(i)} x_{\{i,j\}} \qquad i \in X \tag{1.2}$$

$$\sum_{d=1}^{D} y_i^d = 1 \qquad i \in X \tag{1.3}$$

$$x_{\{i,j\}} \in \{0,1\} \qquad \{i,j\} \in E \tag{1.4}$$

$$y_i^d \in \{0,1\} \qquad i \in X; 1 \leq d \leq D \tag{1.5}$$

For our computational experiment we have modelled the set of constraints (1.1) using the following well known directed model (see [16]):

$$\sum_{(i,j)\in A} a_{ij} = 1 \qquad j = 2,\ldots,n \tag{1.6}$$

$$\sum_{\substack{(i,j)\in A\\ i\notin U, j\in U}} a_{ij} \geq 1 \qquad U \subseteq X\backslash\{1\} \tag{1.7}$$

$$a_{ij} + a_{ji} = x_{\{i,j\}} \qquad \{i,j\} \in E \tag{1.8}$$

$$a_{ij} \in \{0,1\} \qquad (i,j) \in A \tag{1.9}$$

In this model, a_{ij} are binary *arc* variables indicating whether or not arc (i,j) is in the directed tree rooted at a given node, *e.g.*, node 1. The set A denotes the set of arcs that are defined by directing every edge both ways (for edges incident to node 1 we only consider the arc leaving node 1),

$$A = \{(i,j),(j,i) : \{i,j\} \in E \text{ and } i,j \neq 1\} \cup \{(1,j) : \{1,j\} \in E\}$$

Directed inequalities (1.7) are added to the model within a cutting plane algorithm. Violated cuts are detected in the usual way by computing maximum flows from the root node to all the other nodes. If a maximum flow is below 1 we add the corresponding cut inequality to the model. If multiple minimum cuts are found for one particular source-target pair we consider both the inequalities corresponding to the cut nearest to the source and nearest to the target. Additionally, we search for the minimum cut with the smallest number of arcs by adding $\epsilon = 1e^{-6}$ to all arc capacities (see [15] for further details).

As noted before, the three classes of models differ on the set of *edge-degree* variables that characterize the type of transmission system to be installed.

3.2. V *models*

Consider the binary variables $v^m_{\{i,j\}}$ indicating whether the edge $\{i,j\} \in E$ is in the solution and the maximum degree between nodes i and j is m (with $2 \leq m \leq D$). These variables are not defined for $m = 1$, since edges where the degree of both endpoints is equal to one exist only in graphs with two nodes.

The problem can then be formulated as follows:

$$(V) \quad \min \sum_{\{i,j\} \in E} \sum_{m=2}^{D} c^m_{\{i,j\}} \cdot v^m_{\{i,j\}} \tag{2.1}$$

$$(1.2) - (1.9)$$

$$x_{\{i,j\}} = \sum_{m=2}^{D} v^m_{\{i,j\}} \qquad \{i,j\} \in E \tag{2.2}$$

$$v^m_{\{i,j\}} \leq y^m_i + y^m_j \qquad \{i,j\} \in E; 2 \leq m \leq D \tag{2.3}$$

$$v^m_{\{i,j\}} \leq \sum_{d=1}^{m} y^d_i \qquad i \in X; \{i,j\} \in E(i); 2 \leq m \leq D-1 \tag{2.4}$$

$$v^m_{\{i,j\}} \in \{0,1\} \qquad \{i,j\} \in E; 2 \leq m \leq D \tag{2.5}$$

The objective cost function is straightforward. Constraints (2.2) link the two sets of edge variables, $x_{\{i,j\}}$ and $v^m_{\{i,j\}}$. Constraints (2.3) and (2.4) link the *node-degree* variables y^d_i with the *edge-degree* variables $v^m_{\{i,j\}}$. For a given edge $\{i,j\}$, constraints (2.3) guarantee that one of the nodes i or j must have a degree equal to m, if $v^m_{\{i,j\}} = 1$ and constraint (2.4) guarantee that neither one of these nodes has a degree greater than m.

We discuss next several model enhancements. First we observe that constraints (2.4) can be lifted into

$$\sum_{d=2}^{m} v^d_{\{i,j\}} \leq \sum_{d=1}^{m} y^d_i \qquad i \in X; \{i,j\} \in E(i); 2 \leq m \leq D-1 \tag{2.4*}$$

At first sight, it may appear that these inequalities are not valid since they appear to allow non-feasible situations such as, $v^a_{\{i,j\}} = 1$ and $y^b_i = 1$ with $2 \leq a < b \leq m'$ for a given $m = m'$. This situation, however, cannot occur since, $v^a_{\{i,j\}} = 1$ and the constraint (2.4*) for $m = a$ and the same edge and node i, implies that $1 \leq \sum_{d=1}^{a} y^d_i$, contradicting $y^b_i = 1$ $(b > a)$ due to constraints (1.3).

Our computational results indicate that for almost all instances tested, the effect of the lifted constraints (2.4*) over constraints (2.4) is null. Thus, for simplicity, we keep the same designation - V - for the model with constraints (2.4*) replacing constraints (2.4). We observe that this non-straightforward lifted version of constraints (2.4), has been found when comparing the linear programming relaxation of model V with the linear programming relaxation of the model presented in section 3.4.

For a given $m \in \{2, \ldots, D\}$, the relation between the variables $v^m_{\{i,j\}}$ and y^d_i can be further strengthened by considering the following set of valid inequalities,

$$m \cdot y^m_i \leq \sum_{\{i,j\} \in E(i)} \sum_{d=m}^{D} v^d_{\{i,j\}} \qquad i \in X; 3 \leq m \leq D$$

These valid inequalities state that, if a given node i has a degree equal to m, then there must exist m edges incident to node i with a maximum degree greater than or equal to m. Note that, we only need to write these inequalities for $m \geq 3$ since for $m = 2$, the inequality is dominated by constraint (1.2) for that same node i and constraints (2.2) for the edges in $E(i)$. These inequalities can be further lifted to

$$\sum_{d=m}^{D} d \cdot y^d_i \leq \sum_{\{i,j\} \in E(i)} \sum_{d=m}^{D} v^d_{\{i,j\}} \qquad i \in X; 3 \leq m \leq D \tag{2.6}$$

The validity of the inequalities (2.6) and the lifted inequalities (2.4*) can be indirectly established by the proof of Proposition 3.3 (see section 3.4) where we show that a set of inequalities already proven valid implies two sets of inequalities that are stronger versions of the inequalities (2.4*) and (2.6), respectively.

We denote by $V+$ the model V with the addition of the valid inequalities (2.6). Computational results given in section 4 show that these inequalities are effective to improve the linear programming bound given by the original model.

We conclude this section by pointing out that the inequalities (1.3) and (2.2), permit us to rewrite the inequalities (2.4*) in several different equivalent ways. Similarly, inequalities (2.6) can also be rewritten in equivalent forms by using inequalities (1.2) and (2.2).

3.3. T *models*

Consider the binary variables $t^{mk}_{\{i,j\}}$ where the indexes i, j and m have the same meaning as in the definition of the $v^m_{\{i,j\}}$ variables. The extra index k indicates the node (i or j) where the maximum degree is obtained. Since we need to distinguish the case where the maximum degree is obtained in both nodes i and j, we also consider variables $t^m_{\{i,j\}}$ for this situation and in the definition of variables $t^{mk}_{\{i,j\}}$ the maximum degree is obtained exactly in one of the endpoints, $k \in \{i, j\}$. The first model of this section - model T - is described below,

$$(T) \quad \min \sum_{\{i,j\}\in E} \sum_{m=2}^{D} c^m_{\{i,j\}} \cdot \left(t^{mi}_{\{i,j\}} + t^{mj}_{\{i,j\}} + t^m_{\{i,j\}} \right) \tag{3.1}$$

(1.2) – (1.9)

$$x_{\{i,j\}} = \sum_{m=2}^{D} (t^{mi}_{\{i,j\}} + t^{mj}_{\{i,j\}} + t^m_{\{i,j\}}) \qquad \{i,j\} \in E \tag{3.2}$$

$$t^{mi}_{\{i,j\}} + t^m_{\{i,j\}} \leq y^m_i \qquad i \in X; \{i,j\} \in E(i); 2 \leq m \leq D \tag{3.3}$$

$$t^{mi}_{\{i,j\}} \leq \sum_{d=1}^{m-1} y^d_j \qquad i \in X; \{i,j\} \in E(i); 2 \leq m \leq D \tag{3.4}$$

$$t^m_{\{i,j\}}, t^{mi}_{\{i,j\}}, t^{mj}_{\{i,j\}} \in \{0,1\} \qquad \{i,j\} \in E; 2 \leq m \leq D \tag{3.5}$$

The objective function is straightforward. Constraints (3.2) relate the new *edge-degree* variables with the *edge* variables $x_{\{i,j\}}$. For a given edge $\{i,j\}$ constraints (3.3) guarantee that, if the maximum degree is obtained (uniquely or not) on one of its endnodes, say node i, then its degree must be equal to that maximum value; if the maximum degree is obtained uniquely in node i, then constraint (3.4) for that same edge and maximum degree guarantees that the degree of the other endnode, node j, must be strictly less than the maximum degree. Constraints (3.5) are the domain constraints for the new variables.

The *edge-degree* variables in models V and T are related by the following equalities

$$v^m_{\{i,j\}} = t^{mi}_{\{i,j\}} + t^{mj}_{\{i,j\}} + t^m_{\{i,j\}} \qquad \{i,j\} \in E; 2 \leq m \leq D \tag{3.6}$$

which permit us to evaluate the effect of adding information on which node attains the maximum degree.

Note that, in model V we had one constraint (2.3) for each edge and m and now, in model T, we have two constraints (3.3) for the same edge and m, one for each of its endnodes. By comparing constraints (3.3) and (3.4) with constraints (2.3) and (2.4) from the models of the first class, we illustrate what we said before in the beginning of section 3, that the variables $t^{mk}_{\{i,j\}}$ are an asymmetric version of the $v^m_{\{i,j\}}$ variables. In fact, for a fixed edge $\{i,j\}$ and a fixed value of $m \geq 2$, by adding constraints (3.3) for both nodes i and j and using the linking equalities (3.6), we obtain

$$v^m_{\{i,j\}} + t^m_{\{i,j\}} = t^{mi}_{\{i,j\}} + t^{mj}_{\{i,j\}} + 2 \cdot t^m_{\{i,j\}} \leq y^m_i + y^m_j$$

which is a stronger version of constraints (2.3).

With respect to constraints (2.4), consider a given edge $\{i,j\}$ and a maximum degree $m \geq 2$. If we add constraint (3.3) for node i (node j) with constraint (3.4) for node j (node i) and use the linking equalities (3.6) we obtain constraint (2.4) for node i (node j) for the given edge and degree.

Similar arguments are used to show that "directed" linking constraints using *arc variables* imply "undirected" linking constraints using *edge variables*.

Constraints (3.4) can be lifted in the same way as constraints (2.4) in model V were lifted and thus we obtain the following lifted constraints:

$$\sum_{d=2}^{m} t_{\{i,j\}}^{di} + \sum_{d=2}^{m-1} \left(t_{\{i,j\}}^{dj} + t_{\{i,j\}}^{d} \right) \leq \sum_{d=1}^{m-1} y_j^d \qquad i \in X; \{i,j\} \in E(i); 2 \leq m \leq D \quad (3.4^*)$$

An indirect formal proof that constraints (3.4*) are valid is left for Proposition 3.3. Like in the previous class of models, for almost all instances tested, the effect of the lifted constraints (3.4*) over the constraints (3.4) is null and for simplicity we keep the designation T for the lifted model.

The lifted constraints (2.4*) of the model V, for a given edge edge $\{i,j\}$ and $m \geq 2$, can be obtained in a similar way as before, from the T model, by adding constraints (3.3) (for one of the nodes) and constraints (3.4*) for the other node and then using the linking equalities (3.6).

Adding the linking equalities (3.6) to model T does not alter its linear programming relaxation since these equalities only define the v variables in terms of the t variables. For simplicity, we still denote by T, the model T with the linking equalities, (3.6). The arguments used above to relate constraints (3.3) and (3.4*) with constraints (2.3) and (2.4*), respectively, permit us to conclude (we omit the details for the remainder of the proof) that

Proposition 3.1. *The projection of the set of feasible solutions of the linear programming relaxation of* T *on the subspace defined by the variables x, y and v is contained in the set of feasible solutions of the linear programming relaxation of the model* V.

For this class of models, we can also derive a set of valid inequalities that are similar to the valid inequalities (2.6) presented before for model V:

$$\sum_{d=m}^{D} d \cdot y_i^d \leq \sum_{\{i,j\} \in E(i)} \left(\sum_{d=m}^{D} \left(t_{\{i,j\}}^{di} + t_{\{i,j\}}^{d} \right) + \sum_{d=m+1}^{D} t_{\{i,j\}}^{dj} \right) \qquad i \in X; 2 \leq m \leq D \quad (3.7)$$

Note that, for $m = D$, the summation on $t_{\{i,j\}}^{dj}$ is null). Note also that unlike the inequalities (2.6) in model V, the inequalities (3.7) for $m = 2$ are not implied by constraints (1.2) and (3.2).

We denote by $T+$ the model T enhanced with the set of valid inequalities (3.7). Our computational results will show that the inclusion of these valid inequalities improve the lower bounds obtained with the linear programming relaxation of model T.

Inequalities (2.6) and (3.7) have a similar interpretation. However, in the case of inequalities (3.7) we observe that if a given node i has degree equal to m, then there must exist m edges $\{i,j\}$ incident to node i such that, for each one there are only two possible situations: either the maximum degree is equal to m (obtained exclusively in node i or in both nodes i and j) or the maximum degree is greater than m (obtained exclusively in node j). Because of this fact, it is quite easy to see that by using the equalities (3.6) the inequalities (3.7) are equivalent to stronger versions of the inequalities (2.6). This allows us to conclude that (again, we still denote by $T+$ the model $T+$ augmented with the definitional equalities (3.6)),

Proposition 3.2. *The projection of the set of feasible solutions of the linear programming relaxation of* T+ *on the subspace defined by the variables x, y and v is contained in the set of feasible solutions of the linear programming relaxation of the model* V+.

Similarly to some of the inequalities in the V model, by using inequalities (1.2), (1.3) and (3.2), inequalities (3.4*) and (3.7) can be rewritten in different equivalent ways.

3.4. Z *models*

Consider the binary variables $z_{\{i,j\}}^{pq}$, indicating whether the edge $\{i,j\} \in E$ is selected and the degrees on node i and j are p and q, respectively. For the same reason explained before for variables $v_{\{i,j\}}^{1}$, the $z_{\{i,j\}}^{11}$ variables are not defined. The new model is denoted as Z. Again, the objective function is straightforward. Constraints (4.2) relate the two types of *edge* variables in this model whereas constraints (4.3) state that, if the degree of node i is equal to p then, in the solution, exactly p edges are incident in that node, whatever the degree of node j is. Note that, a model involving the new variables z is much simpler to write than any of the models of previous two classes due to the information on the degree of the two nodes.

$$(Z) \quad \min \sum_{\{i,j\} \in E} \sum_{m=2}^{D} c_{\{i,j\}}^{m} \cdot \left(\sum_{q=1}^{m} z_{\{i,j\}}^{mq} + \sum_{p=1}^{m-1} z_{\{i,j\}}^{pm} \right) \tag{4.1}$$

$$(1.2) - (1.9)$$

$$x_{\{i,j\}} = \sum_{p=1}^{D} \sum_{q=1}^{D} z_{\{i,j\}}^{pq} \qquad \{i,j\} \in E \tag{4.2}$$

$$d \cdot y_i^d = \sum_{\{i,j\} \in E(i)} \sum_{q=1}^{D} z_{\{i,j\}}^{dq} \qquad i \in X; 1 \leq d \leq D \tag{4.3}$$

$$z_{\{i,j\}}^{pq} \in \{0,1\} \qquad \{i,j\} \in E; 1 \leq p,q \leq D \tag{4.4}$$

We also note that by adding constraints (4.3) for all $p = 1, \ldots, D$ and a given node i, and then using constraints (4.2) for the edges incident on node i, we obtain constraints (1.2) for the same node i. Thus, constraints (4.3) are a disaggregation of (1.2) and the latter can be omitted from the model.

Computational results obtained with some instances permit us to state that there is no relationship between the linear programming bound given by model Z and the linear programming bounds given by the best, $T+$, and the worst, V, of the models described in the two previous sections. Thus, this non-dominance relation applies to Z and all the models presented before.

However, as we have stated before, one of the reasons for using the new set of variables is that it permits us to use the additional information on the degree of the two nodes in order to derive (hopefully) stronger valid inequalities. One such example is given by the following inequalities

$$\sum_{q=1}^{D} z_{\{i,j\}}^{pq} \leq y_i^p \qquad i \in X; \{i,j\} \in E(i); 2 \leq p \leq D \tag{4.5}$$

These inequalities state that if edge $\{i,j\}$ is in the solution and node i has degree equal to p, whatever the degree on node j is, then the corresponding y variable associated to node i and degree p must be equal to 1. We do not need to consider inequalities (4.5) for $p = 1$ since they are implied by constraints (4.3) for node i and $d = 1$.

The valid inequalities (4.5) are similar to the inequalities arising in the so-called *strong location models* and also to the valid inequalities included in the complete description of a small polytope introduced in [11]. Consider the following small polytope for a given node i and a degree d:

$$d \cdot y_i^d = \sum_{\{i,j\} \in E(i)} \sum_{q=1}^{D} z_{\{i,j\}}^{dq} \tag{4.6}$$

$$0 \leq z_{\{i,j\}}^{dq} \leq 1 \qquad \{i,j\} \in E(i) \tag{4.7}$$

$$0 \leq y_i^d \leq 1 \tag{4.8}$$

For the given node and degree, including the corresponding set of valid inequalities (4.5) for every edge $\{i,j\} \in E(i)$ in the polytope defined by (4.6)-(4.8), gives a complete description of the convex hull defined by the integer solutions of the polytope. The proof is similar to the one given in [11]. Thus, in a certain sense, there are no more valid inequalities, relating the $z_{\{i,j\}}^{pq}$ and y_i^p variables that may improve the linear programming bound of model Z, for a given node i and a degree p.

The variables of the two classes of models, T and Z, are related by the following equalities:

$$t_{\{i,j\}}^{mi} = \sum_{q=1}^{m-1} z_{\{i,j\}}^{mq} \qquad \{i,j\} \in E; 2 \leq m \leq D \tag{4.9a}$$

$$t_{\{i,j\}}^{mj} = \sum_{p=1}^{m-1} z_{\{i,j\}}^{pm} \qquad \{i,j\} \in E; 2 \leq m \leq D \tag{4.9b}$$

$$t_{\{i,j\}}^{m} = z_{\{i,j\}}^{mm} \qquad \{i,j\} \in E; 2 \leq m \leq D \tag{4.9c}$$

We denote by $Z+$ the model Z augmented with inequalities (4.5). We also let $Z+$ denote the same model augmented with the definitional equalities (4.9). In terms of linear programming relaxation, we can state the following result between this stronger model $Z+$ and the best of the previous models,

Proposition 3.3. *The projection of the set of feasible solutions of the linear programming relaxation of* Z+ *on the subspace defined by the variables x, y and* *t is contained in the set of feasible solutions of the linear programming relaxation of the model* T+.

Proof. For a given edge $\{i,j\}$, constraints (3.2) can be obtained from constraints (4.2) after some index rearrangement and then using the equalities (4.9),

$$\begin{aligned} x_{\{i,j\}} &= \sum_{p=1}^{D}\sum_{q=1}^{D} z_{\{i,j\}}^{pq} = \sum_{p=2}^{D} z_{\{i,j\}}^{pp} + \sum_{q=2}^{D}\sum_{p=1}^{q-1} z_{\{i,j\}}^{pq} + \sum_{p=2}^{D}\sum_{q=1}^{p-1} z_{\{i,j\}}^{pq} = \\ &= \sum_{p=2}^{D} t_{\{i,j\}}^{p} + \sum_{q=2}^{D} t_{\{i,j\}}^{qj} + \sum_{p=2}^{D} t_{\{i,j\}}^{pi} = \sum_{m=2}^{D} \left(t_{\{i,j\}}^{m} + t_{\{i,j\}}^{mj} + t_{\{i,j\}}^{mi} \right) \end{aligned}$$

Now, consider a given node i, an edge $\{i,j\} \in E(i)$ and a maximum degree equal to m. To obtain the constraint (3.3), we use equalities (4.9a) and (4.9c) together with the valid inequality (4.5) for the given node, edge and degree (in the last inequality),

$$t_{\{i,j\}}^{mi} + t_{\{i,j\}}^{m} = \sum_{q=1}^{m-1} z_{\{i,j\}}^{mq} + z_{\{i,j\}}^{mm} \leq \sum_{q=1}^{D} z_{\{i,j\}}^{mq} \leq y_i^m$$

The lifted constraints (3.4*) for a given node i, an edge $\{i,j\} \in E(i)$ and $m \geq 2$ can be obtained using equalities (4.9) and the respective valid inequalities (4.5) for node j and $p \leq m-1$:

$$\begin{aligned} \sum_{d=2}^{m} t_{\{i,j\}}^{dj} + \sum_{d=2}^{m-1} \left(t_{\{i,j\}}^{di} + t_{\{i,j\}}^{d} \right) &\underset{(4.9)}{=} \sum_{d=2}^{m}\sum_{p=1}^{d-1} z_{\{i,j\}}^{pd} + \sum_{d=2}^{m-1}\sum_{q=1}^{d} z_{\{i,j\}}^{dq} = \\ &= \sum_{p=1}^{m-1}\sum_{d=p+1}^{m} z_{\{i,j\}}^{pd} + \sum_{d=2}^{m-1}\sum_{q=1}^{d} z_{\{i,j\}}^{dq} \end{aligned}$$

By renaming d as q and p as d in the first term on the last expression, we obtain

$$\begin{aligned} \sum_{d=2}^{m} t_{\{i,j\}}^{dj} + \sum_{d=2}^{m-1} \left(t_{\{i,j\}}^{di} + t_{\{i,j\}}^{d} \right) &= \sum_{d=1}^{m-1}\sum_{q=d+1}^{m} z_{\{i,j\}}^{dq} + \sum_{d=2}^{m-1}\sum_{q=1}^{d} z_{\{i,j\}}^{dq} \underset{(d,\,q)\neq(1,\,1)}{=} \\ &= \sum_{d=1}^{m-1}\sum_{q=1}^{m} z_{\{i,j\}}^{dq} \leq \sum_{d=1}^{m-1}\sum_{q=1}^{D} z_{\{i,j\}}^{dq} \underset{(4.5)}{\leq} \sum_{d=1}^{m-1} y_i^d \end{aligned}$$

To obtain the valid inequality (3.7), first consider a given node i and a maximum degree $m \geq 2$. For every edge $\{i,j\} \in E(i)$ we observe that

$$\begin{aligned} \sum_{d=m}^{D}\sum_{q=1}^{D} z_{\{i,j\}}^{dq} &\leq \sum_{d=m}^{D}\sum_{q=1}^{D} z_{\{i,j\}}^{dq} + \sum_{p=1}^{m-1}\sum_{d=m+1}^{D} z_{\{i,j\}}^{pd} = \sum_{d=m}^{D}\sum_{q=1}^{d} z_{\{i,j\}}^{dq} + \sum_{d=m}^{D-1}\sum_{q=d+1}^{D} z_{\{i,j\}}^{dq} + \\ &+ \sum_{d=m+1}^{D}\sum_{p=1}^{m-1} z_{\{i,j\}}^{pd} = \sum_{d=m}^{D}\sum_{q=1}^{d} z_{\{i,j\}}^{dq} + \sum_{q=m+1}^{D}\sum_{d=m}^{q-1} z_{\{i,j\}}^{dq} + \sum_{d=m+1}^{D}\sum_{p=1}^{m-1} z_{\{i,j\}}^{pd} \end{aligned}$$

By renaming d as p and q as d in the second term of the last expression, we obtain

$$\begin{aligned}\sum_{d=m}^{D}\sum_{q=1}^{D} z_{\{i,j\}}^{dq} &\leq \sum_{d=m}^{D}\sum_{q=1}^{d} z_{\{i,j\}}^{dq} + \sum_{d=m+1}^{D}\sum_{p=m}^{d-1} z_{\{i,j\}}^{pd} + \sum_{d=m+1}^{D}\sum_{p=1}^{m-1} z_{\{i,j\}}^{pd} = \\ &= \sum_{d=m}^{D}\left(\sum_{q=1}^{d-1} z_{\{i,j\}}^{dq} + z_{\{i,j\}}^{dd}\right) + \sum_{d=m+1}^{D}\sum_{p=1}^{d-1} z_{\{i,j\}}^{pd} = \\ &= \sum_{d=m}^{D}\left(t_{\{i,j\}}^{di} + t_{\{i,j\}}^{d}\right) + \sum_{d=m+1}^{D} t_{\{i,j\}}^{dj}\end{aligned}$$

Finally, by adding constraints (4.3) for $d = m, \ldots, D$ we obtain the valid inequalities (3.7) from model $T+$ for the given node and degree,

$$\sum_{d=m}^{D} d \cdot y_i^d = \sum_{\{i,j\}\in E(i)} \sum_{d=m}^{D}\sum_{q=1}^{D} z_{\{i,j\}}^{dq} \leq \sum_{\{i,j\}\in E(i)} \left\{\sum_{d=m}^{D}\left(t_{\{i,j\}}^{di} + t_{\{i,j\}}^{d}\right) + \sum_{d=m+1}^{D} t_{\{i,j\}}^{dj}\right\}$$

The domain constraints, $0 \leq t_{\{i,j\}}^{m} \leq 1$, $\{i,j\} \in E, 2 \leq m \leq D$, are obvious by using the linking equality (4.9c). As for the domain constraints, $0 \leq t_{\{i,j\}}^{mk} \leq 1$, $\{i,j\} \in E, 2 \leq m \leq D, k = i, j$, they are easy to obtain by using the equalities (4.9a) for $k = i$ (or equalities (4.9b) for $k = j$) together with the constraints (4.2) and the relaxed domain constraints on the $x_{\{i,j\}}$ variables. □

As a corollary to the last proposition we have

Corollary 3.4. *The projection of the set of feasible solutions of the linear programming relaxation of* Z+ *on the subspace defined by the variables x, y and t is contained in the set of feasible solutions of the linear programming relaxation of the model* T.

Note that, in the proof of Proposition 3.3, we only made use of constraints (4.3) of model *Z+* to obtain the valid inequalities (3.7) of the lifted model *T+*. It is also not difficult to observe that in the presence of the valid inequalities (4.5) on model *Z+*, we still obtain a valid model for the problem by using only the weaker constraints (1.2) instead of constraints (4.3). Thus, Corollary (3.4) also holds if we use this weaker model instead of the stronger model *Z+*. Although reducing the dimension of the model *Z+* by removing the constraints (4.3), a few computational results showed us that, when solving the problem, the CPU times obtained with this weaker model are usually higher than the CPU times obtained with the stronger model and we omit it from the results analysis in the next section.

We end this section with a figure (see Figure 2) summarizing the relation between all the proposed models, in terms of linear programming relaxations.

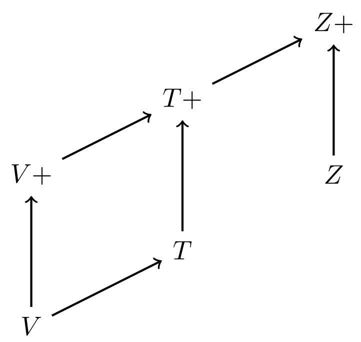


Figure 2: Relation between the different models (arrow direction: from the weaker model to the stronger model).

4. Computational Results

In this section we anaylse computational results obtained to compare the two versions of the classes of models V, T and Z. The maximum CPU time to obtain the optimal solutions for the integer models and the respective linear programming relaxations was set to 7200 seconds. Each test run was performed on a single core of an Intel Xeon E5540 or E5649 machine both with 2.53 GHz. Preliminary tests showed that both machines have nearly the same performance with respect to our type of experiments. The memory limit per test run was set to 6 GB. We used IBM ILOG CPLEX 12.5 as the LP solver and branch-and-cut framework. All CPLEX parameters were left at their default settings except the MIP Emphasis parameter which was set to OPTIMALITY. In previous studies in similar problems, we have obtained slightly better results (lower CPU times) when this parameter was set to OPTIMALITY instead of the default setting BALANCED (feasibility and optimality). For the current paper, we did the same testing on some "easy" and "hard" instances and the conclusion was nearly the same. In the problem under study, we find that it is better that CPLEX spends time on proving optimality, rather than on proving optimality as well as searching for more feasible solutions. In most of the cases, the optimal solution is found earlier but it is difficult to prove its optimality, so the pure OPTIMALITY setting works (slightly) better

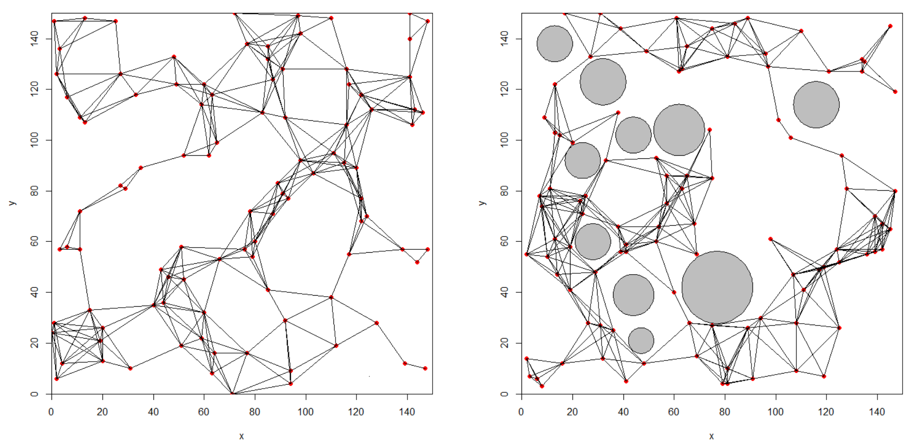
4.1. Data Generation

4.1.1. The Instances

Wireless mesh networks based on WiFi have been considered a cost-effective solution mainly for rural and remote areas where the deployment of wired networks is too expensive both from a revenue and a technical point of view (see [2, 8, 18, 17]). In these references, the deployment scenarios are characterized by a few tens of nodes at most. In this paper, we have used a data set with sparse instances with 100 nodes. We have generated sparse graphs since, in realistic wireless based situations (again, see see [2, 8, 18, 17]), for many pairs of nodes, either there exists an obstacle between them or the distance between them is greater than the distance range of any available type of wireless link system.

In order to generate instances with realistic properties, the nodes generation basically consists in scattering the nodes in a square grid, avoiding the areas where circle shaped obstacles have been randomly placed. The radius of each circle is a random integer between two fixed given values.

We tested instances with 0, 5, 10 and 15 obstacles. Instances with no obstacles correspond to "free areas" (see Figure 3a) where the only limitation is the distance between the two points. In conclusion, for each instance, only the edges that correspond to a distance no greater than the maximum distance range for any link to work (see the next subsection) and that do not "cross" any obstacle, are considered. This leads to instances with different quantities of edges (therefore, densities) in the complete set of instances for our experiment.



(a) A "free area" instance.

(b) An instance with 10 obstacles.

Figure 3: Instances examples.

Although these topologies were created based on wireless scenarios, where C_{ij}^s are the same for every link $\{i, j\}$, we have used them also in the more general cases where C_{ij}^s vary with the distance for each $\{i, j\}$ (see the next section).

4.1.2. The Links

We have considered three types of systems (I, II and III, ordered in cost increasing order) that differ in terms of the respective cost and distance range. We have also considered a distance range (when there is no interference) of 15 for type I and 25 for types II and III.

Three *costs configurations* (see Table 1, configurations α_1, α_2 and α_3) have been considered for the wireless-based configurations. Note that, the *upgrading cost* from one system type to the next ($f_{\text{II}} - f_{\text{I}}$ and $f_{\text{III}} - f_{\text{II}}$) increases from one configuration to the next.

Consider δ_{ij} as the euclidean distance between nodes i and j. In order to further examine the behavior and strength of our models, we also have considered a different type of costs configuration (see Table 1, configurations α_4 and α_5) where the costs are given as follows: the cost of the cheapest type of system is exactly the euclidean distance between the endnodes, given by δ_{ij} for each pair of node i and j, whereas the other two system costs are obtained by adding a concave upgrading cost to the previous one. We call these "euclidean-based" configurations.

Cost configurations		System types		
		type I	type II	type III
Wireless-based	α_1	5	7	8
	α_2	5	8	10
	α_3	5	9	12
Euclidean-based	α_4	δ_{ij}	$\delta_{ij}+20$	$\delta_{ij}+30$
	α_5	δ_{ij}	$\delta_{ij}+40$	$\delta_{ij}+70$

Table 1: Cost Configurations $(f_{\rm I}, f_{\rm II}, f_{\rm III})$.

As explained in section 2.1, the distance range is reduced if the number of links on one (or both) of the endnodes of the link (where the system is to be installed) is above a certain value. The maximum degree, D, is given by the number of overlapping frequencies that the operator wants to use. Then, we consider two generic degree parameters D_1 and D_2, such that $D_1 < D_2 < D$, to define two different maximum degrees of each type of system on each pair of nodes i and j (corresponding to two different reduced distance ranges). Based on these parameters, we define :

- Systems of type I require a maximum degree of D_2 if $\delta_{ij} \leq 5$ and a maximum degree of D_1 if $5 < \delta_{ij} \leq 15$;
- Systems of type II require a maximum degree of D if $\delta_{ij} \leq 5$, a maximum degree of D_2 if $5 < \delta_{ij} \leq 15$ and a maximum degree of D_1 if $15 < \delta_{ij} \leq 25$;
- Systems of type III require a maximum degree D if $\delta_{ij} \leq 15$ and a maximum degree D_2 if $15 < \delta_{ij} \leq 25$.

We tested three degree configurations (D_1, D_2, D) as it is depicted in Table 2.

Degree Configuration	D_1	D_2	D
β_1	2	3	4
β_2	2	3	5
β_3	2	4	5

Table 2: Degree Configurations (D_1, D_2, D).

In the next sections, the gap between the value of a generic integer model (Int) and its linear programming relaxation $(LPRel)$, is determined as

$$\frac{V(Int) - \lceil V(LPRel) \rceil}{V(Int)}$$

where $\lceil \cdot \rceil$ is the usual "integer rounding up" operator.

4.2. Analysis of the wireless-based cost configurations.

In this section we analyse the results obtained for the wireless-based cost configurations α_1, α_2 and α_3. In the next section we analyse the results obtained for the euclidean-based cost configurations α_4 and α_5.

Although we have generated instances with densities from 6% to 18% (the density is determined as the number of edges in the instance, over the number of edges in a complete graph with the same number of nodes), for the present analysis we only report on instances with *low-densities*, between 6% and 9% (corresponding to 0, 5 or 10 obstacles). The reason why we do not present the results of the high-density instances is because the linear programming gaps are close to zero for all models, even for the models producing worse CPU times.

Table 3 presents the results for the linear programming relaxation of all the models whereas Table 4 presents the results about obtaining the optimal integer solution. The first three columns of both tables identify the *cost configuration*, *degree configuration* and *number of obstacles*. In Table 3, the next six columns present the average gaps (in percentage) and the last six columns present the median CPU times (in seconds) of all models. In Table 4, the columns 4 to 8 present the median CPU times (in percentage) for all models and the next eight columns give information about the average percentage of nodes for each degree value and the average percentage of system links in the optimal solution obtained with model *Z+*. We use the median value instead of the average to evaluate the CPU times since a single abnormal running time does not affect the median value whereas it can significantly affect the average value.

4.2.1. Wireless-based cost configurations: Linear Programming results.

The median CPU times in Table 3 can be considered negligible since the median times to obtain the linear programming bounds are at most 2 seconds for all models. We observe that, as the upgrading costs increase (from configuration α_1 to α_3) the gaps increase for all models. Also, we observe bigger gaps for instances with more obstacles. This fact becomes more obvious when we compare the results for the "free area" instances (0 obstacles) against the results for instances with obstacles. Note that the number of obstacles is related with the density of the instance - more obstacles, lower density. Apparently, the *degree configurations* does not seem to have a relevant effect on the gap variation.

The results also show that the gaps are small (less than 5%) for all models. Comparing model classes, the two variants V and T have the same gap which is always worse than the gap obtained with the weakest model in class Z (recall that there is no theoretical relationship between the linear programming bound given by model Z and the linear programming bounds given by models V and T). Clearly, the stronger model in each class (the one with valid inequalities added) always produces a smaller gap (by at least 1%) than the other variant. An interesting situation is that the model *T+* produces almost always the same gap as model *Z+* being outperformed only in the case where the degree configuration is β_3. In some sense this indicates that the information given by the lower degree node (in variables $z^{pq}_{\{i,j\}}$) is in general not relevant. However, as we have pointed out before, this extra information leads to models such that the constraints linking the two sets of variables, y^d_i and $z^{pq}_{\{i,j\}}$, can be written as

			Average LP gaps						Median CPU times					
cost	*deg.*	*obst.*	V	$V+$	T	$T+$	Z	$Z+$	V	$V+$	T	$T+$	Z	$Z+$
α_1	β_1	0	1.5	0.6	1.5	0.2	1.0	0.2	1	1	1	1	0	1
		5	1.8	0.8	1.8	0.3	1.3	0.3	0	1	1	1	0	1
		10	1.8	0.7	1.8	0.3	1.2	0.3	1	1	1	1	0	1
	β_2	0	1.5	0.6	1.5	0.2	1.0	0.2	1	1	1	1	1	1
		5	1.9	0.8	1.9	0.3	1.3	0.3	1	1	1	1	0	1
		10	1.9	0.7	1.9	0.3	1.2	0.3	1	1	1	1	1	1
	β_3	0	1.5	0.7	1.5	0.4	1.2	0.3	1	1	1	1	1	1
		5	1.8	0.8	1.8	0.5	1.4	0.4	1	1	1	1	1	1
		10	1.9	0.8	1.9	0.5	1.4	0.4	1	1	1	1	1	1
α_2	β_1	0	2.3	1.0	2.3	0.3	1.6	0.3	1	0	1	1	0	1
		5	2.7	1.2	2.7	0.3	1.9	0.3	1	0	1	1	0	1
		10	2.8	1.2	2.8	0.4	1.9	0.4	1	1	1	1	0	1
	β_2	0	2.4	1.0	2.4	0.3	1.6	0.3	1	1	1	1	1	1
		5	2.8	1.2	2.8	0.3	1.9	0.3	1	1	1	1	0	1
		10	2.9	1.2	2.9	0.4	1.9	0.4	1	1	1	1	1	2
	β_3	0	2.4	1.0	2.4	0.6	1.8	0.4	1	1	1	2	1	1
		5	2.8	1.2	2.8	0.8	2.2	0.5	1	1	1	1	0	1
		10	3.0	1.3	3.0	0.9	2.2	0.6	1	1	1	1	1	1
α_3	β_1	0	2.9	1.3	2.9	0.5	2.1	0.5	1	1	1	1	0	1
		5	3.6	1.6	3.6	0.5	2.6	0.5	1	0	1	1	0	1
		10	3.7	1.6	3.7	0.6	2.5	0.6	1	0	1	1	0	1
	β_2	0	3.1	1.3	3.1	0.5	2.1	0.5	1	1	1	1	1	1
		5	3.7	1.6	3.7	0.5	2.6	0.5	1	1	1	1	1	1
		10	3.9	1.6	3.9	0.6	2.5	0.6	1	0	1	1	1	1
	β_3	0	3.1	1.4	3.1	0.8	2.3	0.6	1	1	1	1	1	1
		5	3.6	1.7	3.6	1.1	2.9	0.7	1	1	1	1	0	1
		10	4.0	1.8	4.0	1.1	2.9	0.8	0	1	1	1	1	1

Table 3: Linear programming relaxation results for the wireless-based cost configurations.

equalities and are fewer than in the models of the other classes. As we see in the next subsection this may explain the observed CPU times for obtaining the optimal integer solution.

4.2.2. Wireless-based configurations: Integer Programming results.

The median CPU times in Table 4 are in general bigger for each model when using the degree configuration β_3. The instances tested with this degree configuration are in general harder to solve and this might be related with the fact that β_3 was the only degree configuration where the linear programming gaps of model *Z+* were strictly better than the gaps of model *T+*.

Clearly, for each class, the model with the best linear programming bound was the fastest. These models have more constraints than the other model in its class but, the lower CPU times may be explained by the fact that they provide better linear programming bounds among the models of the class.

Although the difference is small,the model *Z+* takes less CPU time than model *T+* to obtain the optimal integer solutions (but still, there were instances where model *T+* performed faster than model *Z+*). In every solution there is no node with degree equal to 5; the majority of the nodes (approximately 94%) has a degree value of 2, whereas approximately 4% are leaf nodes. Only for the degree configuration β_3 we find nodes with degree equal to 4 in the solution. In

Config.			Median CPU time						Degrees					Links		
cost	*deg.*	*obst.*	V	$V+$	T	$T+$	Z	$Z+$	1	2	3	4	5	I	II	III
α_1	β_1	0	363	6	306	6	19	7	4.1	93.8	2.1	0.0	-	79.3	19.2	1.5
		5	250	4	701	9	12	6	4.2	93.6	2.2	0.0	-	78.7	19.8	1.5
		10	78	3	135	3	11	2	4.5	92.9	2.5	0.0	-	79.8	17.9	2.3
	β_2	0	262	5	424	8	24	6	3.8	94.3	1.8	0.0	0.0	79.3	19.2	1.5
		5	579	11	637	5	15	5	4.0	94.0	2.0	0.0	0.0	78.7	19.8	1.5
		10	148	3	339	7	13	2	4.6	92.7	2.6	0.0	0.0	79.7	18.2	2.1
	β_3	0	1107	42	1664	53	87	23	4.0	94.2	1.7	0.2	0.0	79.4	18.9	1.7
		5	814	34	1785	53	109	33	4.4	93.4	2.0	0.2	0.0	79.1	18.9	2.0
		10	371	21	983	26	120	14	4.5	93.1	2.2	0.2	0.0	79.9	17.7	2.4
α_2	β_1	0	306	8	537	13	25	3	3.8	94.3	1.8	0.0	-	79.0	20.2	0.8
		5	250	4	429	10	12	4	4.2	93.6	2.2	0.0	-	78.4	20.7	0.9
		10	105	5	284	4	15	2	4.2	93.6	2.2	0.0	-	79.6	18.5	1.9
	β_2	0	488	6	696	9	18	5	3.9	94.2	1.9	0.0	0.0	79.0	20.2	0.8
		5	609	4	785	5	16	4	4.1	93.8	2.1	0.0	0.0	78.4	20.7	0.9
		10	127	2	853	5	31	3	4.4	93.3	2.4	0.0	0.0	79.6	18.5	1.9
	β_3	0	1326	38	2252	53	106	19	3.7	94.8	1.3	0.2	0.0	79.0	20.2	0.8
		5	1338	22	2363	48	89	25	4.2	93.9	1.6	0.3	0.0	78.7	20.1	1.2
		10	823	24	1000	50	101	23	4.6	93.1	1.9	0.4	0.0	79.7	18.3	2.0
α_3	β_1	0	588	8	330	9	20	8	3.8	94.5	1.8	0.0	-	79.0	20.2	0.8
		5	491	7	1021	5	20	4	4.0	94.0	2.0	0.0	-	78.4	20.7	0.9
		10	333	3	220	4	20	6	4.1	93.8	2.1	0.0	-	79.6	18.5	1.9
	β_2	0	487	6	956	6	24	6	3.8	94.3	1.8	0.0	0.0	79.0	20.2	0.8
		5	440	6	1701	5	14	2	4.2	93.6	2.2	0.0	0.0	78.4	20.7	0.9
		10	216	3	546	4	25	4	4.5	93.1	2.5	0.0	0.0	79.6	18.5	1.9
	β_3	0	1348	25	1943	61	96	27	3.8	94.7	1.4	0.2	0.0	79.0	20.2	0.8
		5	796	34	2779	40	95	16	4.1	94.0	1.7	0.2	0.0	78.5	20.6	0.9
		10	695	26	828	42	77	13	4.5	93.2	2.0	0.3	0.0	79.7	18.3	2.0

Table 4: Integer programming results for the wireless-based cost configurations.

terms of system links, there is a majority of approximately 79% of type I system links. Although, in average, the percentage is very similar among the different scenarios, we observe that, as the number of obstacles increases (the instances get denser) the number of type III system links increases and the number of type II system links decreases.

4.3. Analysis of the euclidean-based cost configurations.

In this section we analyse the more general cost function corresponding to cost configurations α_4 and α_5. For this case we use the complete set of generated instances, that is, instances with *low-densities*, between 6% and 9% (corresponding to 0, 5 or 10 obstacles) used in the previous results, as well as instances with *high-densities*, between 10% and 18% (corresponding to 5, 10 or 15 obstacles).

Tables 5 and 6 are organized in a similar way as Tables 3 and 4, except for the first column which identifies the density (L for low density instances and H for high density instances).

4.3.1. Euclidean-based cost configurations: Linear Programming results.

The median CPU times in Table 5 can once again be considered negligible (less than 3 seconds). However, for this class, the linear programming gaps increase considerably. This can be explained by the fact that the links now are more

				Average LP gap						Median CPU time					
dens.	*cost*	*deg.*	*obst.*	*V*	*V+*	*T*	*T+*	*Z*	*Z+*	*V*	*V+*	*T*	*T+*	*Z*	*Z+*
			0	8.2	3.1	8.2	0.8	5.4	0.8	0	0	1	1	0	1
		β_1	5	9.7	4.0	9.7	1.3	6.5	1.3	0	0	0	1	0	1
			10	10.2	3.8	10.2	1.1	6.5	1.1	0	0	1	1	0	1
			0	9.0	3.1	9.0	0.8	5.5	0.8	0	0	1	1	0	1
	α_4	β_2	5	10.5	4.0	10.5	1.3	6.6	1.3	0	0	1	1	0	1
			10	11.0	3.8	11.0	1.1	6.7	1.1	0	0	1	1	0	1
			0	9.1	3.6	9.1	2.0	6.8	1.4	0	0	1	1	0	1
		β_3	5	10.6	4.5	10.6	2.7	7.9	2.1	0	0	1	1	0	1
			10	11.2	4.6	11.2	2.7	8.5	1.9	0	0	1	1	0	1
L			0	11.0	4.3	11.0	1.4	7.6	1.4	0	0	1	1	0	1
		β_1	5	13.3	5.6	13.3	1.8	9.3	1.8	0	0	1	1	0	1
			10	13.9	5.6	13.9	1.7	9.2	1.7	0	0	0	1	0	1
			0	11.9	4.3	11.9	1.4	7.6	1.4	0	0	1	1	0	1
	α_5	β_2	5	14.1	5.6	14.1	1.8	9.3	1.8	0	0	1	1	0	1
			10	14.8	5.6	14.8	1.7	9.2	1.7	0	0	1	1	0	1
			0	12.0	4.8	12.0	2.7	9.1	1.9	0	0	1	1	0	1
		β_3	5	14.2	6.3	14.2	3.9	11.0	2.9	0	0	1	1	0	1
			10	15.1	6.3	15.1	3.8	11.5	2.7	0	0	1	1	0	1
			5	6.2	3.9	6.2	0.7	6.0	0.7	1	1	1	1	1	1
		β_1	10	6.3	3.6	6.3	0.7	5.8	0.7	1	1	1	1	1	1
			15	7.2	4.3	7.2	0.6	6.8	0.6	0	0	1	1	0	1
			5	6.6	3.9	6.6	0.7	6.0	0.7	1	1	1	2	1	2
	α_4	β_2	10	6.6	3.6	6.6	0.7	5.8	0.7	1	1	1	1	1	1
			15	7.6	4.3	7.6	0.6	6.8	0.6	1	1	1	1	1	1
			5	6.9	4.0	6.9	1.2	6.8	0.8	1	1	1	2	1	2
		β_3	10	6.9	3.7	6.9	1.2	6.6	0.7	1	1	1	2	1	1
			15	7.7	4.4	7.7	1.5	7.8	0.7	0	1	1	1	0	1
H			0	6.5	4.2	6.5	0.8	6.2	0.8	1	1	1	1	1	2
		β_1	5	6.3	3.6	6.3	0.8	5.7	0.8	1	1	1	1	1	1
			10	8.8	5.4	8.8	1.1	8.1	1.1	0	0	1	1	0	1
			0	6.8	4.2	6.8	0.8	6.2	0.8	1	1	2	2	1	2
	α_5	β_2	5	6.7	3.6	6.7	0.8	5.7	0.8	1	1	1	1	1	1
			10	9.1	5.4	9.1	1.1	8.1	1.1	0	1	1	1	1	1
			0	7.2	4.2	7.2	1.2	7.0	0.9	1	1	2	2	1	2
		β_3	5	6.9	3.6	6.9	1.0	6.4	0.7	1	1	1	1	1	1
			10	9.0	5.4	9.0	1.6	8.8	0.9	1	1	1	1	0	1

Table 5: Linear programming relaxation results for the euclidean-based cost configurations.

expensive. Notice that in the wireless case, the gaps increase when the upgrading costs increase. Also, as in the previous case, the gaps increase with the number of obstacles in the instance (with more obstacles in the instances the densities get lower). This can be observed when we compare the gaps obtained with the low density instances against high density instances: the gaps of the latter are lower than the gaps of the former. The upgrading costs in the two cost configurations, α_4 and α_5, seem to have a similar effect as in the wireless-based cost configurations: gaps increase as upgrading costs increase.

Although the gaps are now higher than in the wireless case, models *T+* and *Z+* obtain the best gaps as before (again model *Z+* outperformed model *T+* only in the β_3 degree configuration).

4.3.2. Euclidean-based cost configurations: Integer Programming results.

In this case, the CPU times in Table 6 are much higher than in the wireless-based case, but the main conclusion is similar to the previous case. The degree configuration β_3 instances are more difficult to solve to optimality than any of the other degree configurations. For this degree configuration there were even some instances (low density with 10 obstacles) where for the weaker models (models V and T) an optimally proven integer solution could not be found within the time limit of 7200 seconds. The high density instances were usually solved faster than the low density instances.

Config.				Median CPU time (seconds)						Degrees					Links		
dens.	*cost*	*deg.*	*obst.*	V	$V+$	T	$T+$	Z	$Z+$	1	2	3	4	5	I	II	III
			0	194	7	347	3	18	2	4.2	93.7	2.2	0.0	-	79.1	19.2	1.7
		β_1	5	357	18	372	16	18	6	5.1	91.8	3.1	0.0	-	79.0	18.1	2.9
			10	270	6	542	5	17	2	5.3	91.5	3.3	0.0	-	79.6	17.5	2.8
			0	530	6	437	7	19	2	4.2	93.7	2.2	0.0	0.0	79.0	19.4	1.6
	α_4	β_2	5	749	19	742	15	21	6	5.1	91.8	3.1	0.0	0.0	79.0	18.1	2.9
			10	387	6	1188	3	16	2	5.3	91.5	3.3	0.0	0.0	79.6	17.5	2.8
			0	2198	55	4769	78	80	32	4.4	93.4	1.9	0.2	0.0	79.0	19.3	1.7
		β_3	5	5781	66	4194	76	114	49	5.3	91.8	2.5	0.4	0.0	79.1	18.0	2.9
			10	4524	36	2119	45	137	26	5.3	91.7	2.7	0.3	0.0	79.7	17.5	2.8
L			0	357	3	300	8	15	4	4.0	94.0	2.0	0.0	-	79.0	20.2	0.8
		β_1	5	603	10	647	8	17	7	4.3	93.4	2.3	0.0	-	78.3	20.9	0.8
			10	382	5	515	3	12	2	4.6	92.7	2.6	0.0	-	79.4	18.8	1.7
			0	876	5	643	5	17	3	4.0	94.0	2.0	0.0	0.0	79.0	20.2	0.8
	α_5	β_2	5	1146	11	1278	13	17	9	4.3	93.4	2.3	0.0	0.0	78.3	20.9	0.8
			10	2262	3	440	3	18	2	4.6	92.7	2.6	0.0	0.0	79.4	18.8	1.7
			0	3955	45	2692	54	56	16	4.0	94.1	1.8	0.1	0.0	79.0	20.2	0.8
		β_3	5	6246	46	5102	61	65	37	4.3	93.5	2.1	0.1	0.0	78.4	20.8	0.8
			10	6447	47	4427	52	91	14	4.7	92.8	2.2	0.3	0.0	79.6	18.5	1.9
			5	399	4	504	5	27	5	6.1	89.8	4.1	0.0	-	98.1	1.8	0.1
		β_1	10	127	2	239	3	12	3	7.1	87.8	5.1	0.0	-	96.6	3.3	0.1
			15	189	1	87	2	9	2	6.4	89.1	4.4	0.0	-	95.8	3.7	0.4
			5	1002	6	562	10	19	9	6.2	89.6	4.2	0.0	0.0	98.1	1.8	0.1
	α_4	β_2	10	485	2	949	3	7	2	7.0	88.0	5.0	0.0	0.0	96.6	3.3	0.1
			15	125	1	263	2	11	2	6.4	89.1	4.4	0.0	0.0	95.8	3.7	0.4
			5	1985	39	5155	35	276	7	6.2	90.0	3.3	0.4	0.0	98.1	1.8	0.1
		β_3	10	863	7	1590	13	61	3	7.2	88.3	3.7	0.8	0.0	96.7	3.1	0.1
			15	282	7	749	8	36	2	6.2	89.8	3.8	0.2	0.0	95.7	4.0	0.2
H			5	362	3	381	6	15	5	5.9	90.2	3.9	0.0	-	98.2	1.8	0.0
		β_1	10	138	2	253	4	11	2	7.1	87.8	5.1	0.0	-	96.9	3.1	0.0
			15	22	1	90	2	6	2	6.4	89.1	4.4	0.0	-	96.1	3.6	0.3
			5	341	4	1064	6	20	5	6.0	90.0	4.0	0.0	0.0	98.2	1.8	0.0
	α_5	β_2	10	308	2	566	4	7	2	7.0	88.0	5.0	0.0	0.0	96.9	3.1	0.0
			15	175	1	445	2	6	2	6.6	88.9	4.6	0.0	0.0	96.1	3.6	0.3
			5	2837	17	1381	15	86	7	6.1	90.1	3.4	0.3	0.0	98.2	1.8	0.0
		β_3	10	576	3	860	4	51	3	7.3	87.9	4.2	0.6	0.0	96.9	3.1	0.0
			15	143	2	630	2	33	2	6.8	88.7	4.3	0.2	0.0	96.1	3.7	0.2

Table 6: Integer programming results for the euclidean-based cost configurations.

Again, the strongest model (in terms of linear programming bounds) in each class is the fastest and model $Z+$ takes less CPU time than model $T+$ to obtain the optimal integer solutions (although there were instances when model $T+$performed faster than model $Z+$). The optimal solutions obtained for the low density instances have a similar analysis as the wireless-case instances. As for the solutions obtained for the high density instances again there is no node with degree equal to 5 but now the majority of the nodes (approximately 89%) has a degree value of 2, whereas approximately 6% are leaf nodes. Only for the degree configuration β_3 we find nodes with degree equal to 4 in the solution. In terms of system types, the number of type I system links in high density instances is bigger than in the previous case (low density cases), approximately 97%, which is natural since high density instances have more potential links to choose.

In terms of system links, the number of type I links is bigger than in the previous case, approximately 97%, which is natural since these instances have less potential links to choose. As the number of obstacles increase so do the number of type II and type III system links.

As a final remark, these euclidean-based instances prove to be more difficult to solve than the wireless-based ones justifying, in a certain sense, their inclusion in the paper for studying the robustness of the models.

5. Conclusions

In this paper we have studied a problem that generalizes the Degree Constrained Minimum Spanning Tree Problem by considering constraints on the degree of the nodes that vary with the "type" of edges adjacent to the node in the solution. This problem is motivated by the network design of *point-to-point* wireless mesh networks where different types of transmission systems may be installed in each link. We have proposed three classes of linear programming models that differ on the set of variables that identify the type of transmission system to be installed. For each of these classes we considered a basic model and enhanced models obtained by lifting constraints and adding valid inequalities. These models and the corresponding linear programming relaxations were compared theoretically and through a computational point of view using instances with 100 nodes and different scenarios. The test results showed that (i) the model with valid inequalities of the third class should be chosen when the main objective is to obtain the optimal integer solution and (ii) both the best models in the second and third class can be used to obtain a good linear programming relaxations bound when the main objective is, for example, to evaluate the quality of a known feasible solution.

Acknowdgements: We thank the two reviewers whose comments have helped us to improve the paper.

References

[1] A.M. Almeida, P. Martins, and M.C. Souza, *Min-degree constrained minimum spanning tree problem: Complexity, properties and formulations*, International Transactions in Operational Research **19** (2012), no. 3, 323–352.

[2] G. Bernardi, P. Buneman, and M. K. Marina, *Tegola tiered mesh network testbed in rural scotland*, proc. of ACM workshop on Wireless networks and systems for developing regions (2008), 9–16.

[3] S. Bosio, A. Eisenblatter, H.-F. Geerdes, I. Siomina, and D. Yuan, *Graphs and algorithms in communication networks*, ch. Mathematical Optimization Models for WLAN Planning, pp. 283–309, (A. Koster and X. Munoz, eds.), 2010.

[4] L. Caccetta and S. P. Hill, *A branch and cut method for the degree-constrained minimum spanning tree problem*, Networks **37** (2001), no. 2, 74–83.

[5] F. Chen, B. Hugel, and F. Dressler, *Realistic simulation and experimental validation of adjacent-channel interference in planning of industrial wireless networks*, 8th ACM Symposium on Performance Evaluation of Wireless Ad Hoc Sensor and Ubiquitous Networks, 2011, pp. 97–104.

[6] A. Cunha and A. Lucena, *Lower and upper bounds for the degree-constrained minimum spanning tree problem*, Networks **50** (2007), no. 1, 55–66.

[7] C. Duhamel, L. Gouveia, P. Moura, and M. Souza, *Models and heuristics for the k-degree constrained minimum spanning tree problem with node-degree costs*, Networks **60** (2012), no. 1, 1–18.

[8] P. Dutta, S. Jaiswal, and R. Rastogi, *Villagenet: a low-cost, ieee 802.11-based mesh network for connecting rural areas*, Bell Labs Technical Journal **12** (2007), no. 2, 119–131.

[9] M.R. Garey and D.S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman & Co., New York, USA, 1979.

[10] L. Gouveia and P. Moura, *Spanning trees with node degree dependent costs and knapsack reformulations*, Electronic Notes in Discrete Mathematics **36** (2010), 985–992.

[11] L. Gouveia, P. Moura, and A. Sousa, *Prize collecting steiner trees with node degree dependent costs*, Computers & Operations Research **38** (2010), no. 1, 234–245.

[12] IEEE 802.11 Working Group, *IEEE 802.11g-2003: Further higher data rate extension in the 2.4 ghz band*, 2003.

[13] J. Knowles, D. Corne, and M. Oates, *A new evolutionary approach to the degree constrained minimum spanning tree problem*, IEEE Transactions on Evolutionary Computation **4** (2000), 125–134.

[14] Y. Liu, R. Venkatesan, and L. Cheng, *Channel assignment exploiting partially overlapping channels for wireless mesh networks*, IEEE GLOBECOM Conference, 2009.

[15] I. Ljubic, R. Weiskircher, U. Pferschy, G. W. Klau, P. Mutzel, and M. Fischetti, *An algorithmic framework for the exact solution of the prize-collecting steiner tree problem*, Mathematical Programming **105** (2006), no. 2, 427–449.

[16] T. L. Magnanti and L. A. Wolsey, *Optimal trees*, Network Models (M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, eds.), Handbooks in Operations Research and Management Science, vol. 7, Elsevier, 1995, pp. 503 – 615.

[17] V. Pejovic, D. Johnson, M. Zheleva, E. Belding, L. Parks, and G. Stam, *The bandwidth divide: obstacles to efficient broadband adoption in rural sub-saharan africa*, International Journal of Communication **6** (2012), 2467–2491.

[18] B. Raman and K. Chebrolu, *Experiences in using wifi for rural internet in india*, IEEE Communications Magazine **45** (2007), no. 1, 104–110.

[19] E. Villegas, E. Lpez-Aguilera, R. Vidal, and J. Paradells, *Effect of adjacent-channel interference in IEEE 802.11 WLANs*, 2nd Int. Conf. on Cognitive Radio Oriented Wireless Networks and Communications, 2007, pp. 118–125.