Computers & Operations Research 41 (2014) 44–52

Contents lists available at ScienceDirect

Computers & Operations Research

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

Looking for edge-equitable spanning trees

Mercedes Landete [a,*], Alfredo Marín [b]

[a] Departamento de Estadística, Matemáticas e Informática, Centro de Investigación Operativa, Universidad Miguel Hernández de Elche, Spain
[b] Departamento de Estadística e Investigación Operativa, Universidad de Murcia, Spain

ARTICLE INFO

Available online 6 August 2013

Keywords:
Integer programming
Spanning tree
Ordered median function

ABSTRACT

This paper studies the Ordered Spanning Tree Problem, where the weights of the edges of the tree are sorted and then linearly combined using a previously given coefficients vector. Depending on the coefficients, several objectives can be incorporated to the problem. We pay special attention to the search of spanning trees with balanced weights, i.e., where the differences among the weights are, in some sense, minimized. To solve the problem, we propose two Integer Programming formulations, one based on flow and the other one on the Miller–Tucker–Zemlin constraints. We analyze several potential improvements for both the formulations whose behaviors are checked by means of a computational experiment. Finally, we show how both the formulations can be adapted to incorporate to the objective non-linear functions of the weights.

1. Introduction

Given a connected network (undirected connected graph (V, E) plus weights on the edges) with $n+1$ nodes[1] and m edges, a spanning tree (ST) is defined by a connected subgraph containing $n+1$ nodes and n edges. That of finding the spanning tree with minimum sum of weights is the very well-known *Minimum Spanning Tree Problem* (MSTP). Although the number of spanning trees in a given graph can be huge ($(n+1)^{n-1}$ for a labeled complete graph), finding an ST of minimum weight is a simple task. The first algorithm for finding a minimum ST was developed in 1926 [1,2]. Kruskal [11] and Prim [21] designed the most widely used algorithms, which can be implemented in $O(m \log n)$ and $O(m+n \log n)$ times, respectively.

Although the simplicity and efficiency of the solution algorithms prevent against using Integer Programming models to solve the MSTP, extensions of the problem which are not so simple do benefit of good formulations. Examples of these extensions are the different Steiner tree problems [9], the so-called Generalized MSTP [16], the degree-constrained MSTP [17], the Optimum Communication Spanning Tree Problem [3] and the Tree of Hubs Location Problem [4], among many others. More than a simple formulation, Edmonds [6] introduced a polyhedral description of the MSTP, that is to say, a polytope whose vertexes are the incidence vectors of spanning trees of the graph:

$$\left\{ y \in [0,1]^m : \sum_{(i,j) \in E} y_{ij} = n, \ \sum_{\substack{(i,j) \in E \\ i,j \in S}} y_{ij} \le |S|-1 \quad \emptyset \ne S \subset V \right\}.$$

The exponential number of inequalities in the above description has led to the development of other IP formulations which, not being descriptions of the corresponding polyhedra (the respective convex hulls of their sets of integer solutions), have more tractable sizes. The keys in these formulations are either (i) guaranteeing connectivity, or (ii) avoiding the formation of cycles, since this leads to connectivity of the graph when the number of edges is fixed to n. We highlight two of the different approaches that have been conceived. The first one (see e.g. [12]) is to consider the MSTP as a network design problem where some flow between the nodes of the network is sent; here edge variables y_e indicate whether or not the edge e is available to carry any flow and additional flow variables distinguish the two directions of the edge. The second approach is to incorporate the Miller–Tucker–Zemlin (MTZ) [15] constraints (and corresponding variables) to the formulation. These constraints impose an order to the nodes of the graph in such a way that, forcing each node to point a previous one in the order, cycles cannot be closed.

Generally speaking, spanning tree optimization models in the literature deal with the minimization of the sum of edge weights. Establishing a parallelism with the Location Analysis field, they focus on the *median* objective, but do not pay great attention to *equitable* solutions, like those got using the *center* and *range* objectives (for an introduction to equity measures in the field of Location see [14]). Using, for instance, the range objective, i.e., minimizing the difference between the heaviest and the lightest

* Corresponding author. Tel.: +34 626 961780.
E-mail addresses: landete@umh.es (M. Landete), amarin@um.es (A. Marín).

[1] Although the natural number of nodes is n, we have fixed it to $n+1$ for the sake of simplicity through the rest of the paper.

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.07.023

edges in the tree, balanced[2] trees will be obtained, which can be useful in certain applications where the weight represents the capacity of a link. As we will see shortly, different objectives result in completely different optimum spanning trees. But, instead of limiting our paper to a concrete model, we have extended the parallelism with location models to flexible objectives. These have been introduced in the study of location problems by means of the *ordered median objective* (a monograph on the theme is [18]). The idea, once adapted to the context of spanning trees, is to optimize a linear combination of the sorted (we assume in increasing order) weights of the edges in the ST. By changing the coefficients of the linear combination, several models can be unified, in particular many different equitable objectives like the range and the absolute deviation (sum of the absolute values of the differences between all pairs of weights). The interest of this problem was previously observed in [7], where it was presented as an example inside the general framework of *ordered discrete optimization*.

In Table 1 we present a few examples of objectives and their corresponding coefficients (note that the non-zero coefficients in the interquartile range must be fixed in the correct positions). For instance, to get the coefficients for the absolute deviation objective, it should be noticed that the i-th weight in the tree will appear in the sum (i) $i-1$ times with positive sign, and (ii) $n-i$ times with negative sign. Thus its coefficient will be $(2i-1)-n$.

A critical difference with respect to the location models is that some of the objectives will have null interest when generating trees. Indeed, all the vectors without negative coefficients will have the MST as optimal solution, as we will prove in the next section. Therefore, center and centrum objectives have no special interest. On the other hand, optimal solutions obtained using vectors which include at least one negative coefficient can be completely different. As a sample, we show in Fig. 1 (top–bottom, left–right) the optimal solutions which correspond with the coefficients given in Table 2. For the graph with weights in Fig. 1 we have identified which are the spanning trees which minimize the objective functions in Table 2 and we have highlighted them with bold lines: the first tree in bold lines is the Minimum Spanning Tree and it has cost $20=(1,1,1,1,1,1,1)\cdot(1,1,2,2,2,5,7)$; the second tree is the one with smallest range and it has cost $5=(-1,0,0,0,0,0,1)\cdot(2,2,2,3,4,7,7)$; the following one is the spanning tree which minimizes the "maximum positive deviation respect the median" and it has cost $3=(0,0,0,-1,0,0,1)\cdot(1,2,3,5,6,7,7)$. Apart from the sixth and the seventh cases, any other pair of trees are different.

Our goal then is to produce and analyze a good Integer Programming formulation for the problem, named *Ordered Spanning Tree Problem* (OSTP). This formulation must combine two main ingredients, namely the construction of a tree and the order of the weights. In this first approach we are going to compare the two formulations based on both flow and MTZ constraints (regarding the tree construction). The ordering part is shared by both the formulations, and is based on a covering point of view which gave good results when applied to ordered location problems.

Section 2 gives technical details needed to formulate the problem. Sections 3 and 4 give the flow and MTZ formulations, respectively. Section 5 introduces valid inequalities for both the formulations. Later on, Section 6 generalizes both the flow and the MTZ formulations of the OSTP to the case in which the objective function is not linear. Finally, Section 7 summarizes our computational results when comparing both formulations and Section 8 closes the paper with some conclusions.

[2] We will not use the word *balanced* through the rest of the paper to avoid confusion with the data structure used in computer science.

Table 1
Equitable objectives obtained by means of different coefficients.

Objective	Coefficients
Median (MSTP)	$(1,\dots,1)$
Range	$(-1,0,\dots,0,1)$
Center	$(0,\dots,0,1)$
3-centrum	$(0,\dots,0,1,1,1)$
Median deviation (n odd)	$(-1,\dots,-1,0,1,\dots,1)$
Interquartile range	$(0,\dots,0,-1,0,\dots,0,1,0,\dots,0)$
Absolute deviation	$(1-n,3-n,\dots,n-3,n-1)$

2. The ordered spanning tree problem

2.1. Technical details

Let $G=(V,E)$ be a connected, undirected graph with set of vertices $V=\{1,\dots,n+1\}$, set of edges $E\subseteq\{(i,j)\in V^2 : i<j\}$ and positive weights associated to the edges $c_{ij}\in\mathbb{R}^+\ \forall(i,j)\in E$. Let $\lambda=(\lambda_i)\in\mathbb{R}^n$ be an n-dimensional vector of real coefficients. Let $\mathcal{T}$ be the set of spanning trees in G. Given any tree $T=(V,E_T)\in\mathcal{T}$, we sort the (not necessarily different) weights $(c_{ij} : (i,j)\in E_T)$ in

$$c_{(1)}(T)\le c_{(2)}(T)\le\cdots\le c_{(n)}(T).$$

Then, the Ordered Spanning Tree Problem is defined as

$$\text{(OSTP)}\quad \min\left\{\sum_{i=1}^{n}\lambda_i c_{(i)}(T) : T\in\mathcal{T}\right\}.$$

With $v($ OSTP$)$ we represent the optimal value of (OSTP) and let $N:=\{1,\dots,n\}$. We also define the neighborhood of $i\in V$ as

$$N(i):=\{j\in V : (i,j)\in E \text{ or } (j,i)\in E\}.$$

2.2. The case of non-negative coefficients

Although the following result, as well as Proposition 2, can be considered like particular applications of Theorem 6.1 in [10], in order to avoid the introduction of a large number of concepts regarding polyhedral techniques in Combinatorial Optimization, we present demonstrations oriented to our concrete problem. The interested reader can find another approach for the proof of Proposition 1 based on matroid bases in Theorem 5.3 of [24].

Proposition 1. *Let $T^*\in\mathcal{T}$ be a minimum spanning tree of G. Then T^* is optimal of (OSTP) for any $\lambda\in\mathbb{R}^n_+$.*

Proof. Let T' be an optimal solution to (OSTP) for a given non-negative coefficient vector $\lambda\neq\mathbf{1}$. It suffices to prove that an optimal solution to the MSTP obtained by means of Kruskal's algorithm gives the same objective value as T' for (OSTP), i.e.,

$$\sum_{i=1}^{n}\lambda_i c_{(i)}(T')=\sum_{i=1}^{n}\lambda_i c_{(i)}(T^*).$$

Let then T^* be such an optimal solution of MSTP.

Let $a_1\le a_2\le\cdots\le a_n$ be the weights associated to the edges $e_1,e_2,\dots,e_n$ of E_{T^*} sorted in the order of the algorithm, i.e., e_1 is a minimum weight edge and e_i, $i=2,\dots,n$, has the minimum weight among those edges such that the subgraph of G induced by $\{e_1,\dots,e_i\}$ is acyclic. It is evident that the first i edges $e_1,\dots,e_i$ form a forest with minimum weight among all the spanning forests in G with i edges, $i\in N$. Since the weights a_i are obtained in increasing order, T^* gives an objective value of $\sum_{i=1}^{n}\lambda_i a_i$ in problem (OSTP).