Computers and Operations Research 98 (2018) 69–83

Contents lists available at ScienceDirect

Computers and Operations Research

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

A new approach for the multiobjective minimum spanning tree

J.L. Santos[a,*], Luigi Di Puglia Pugliese[b], Francesca Guerriero[b]

[a] CMUC, Department of Mathematics, University of Coimbra, Apartado 3008, EC Santa Cruz, Coimbra 3001 - 501, Portugal
[b] Department of Mechanical, Energy and Management Engineering, University of Calabria, Italy

ARTICLE INFO

Article history:
Received 28 July 2017
Revised 5 May 2018
Accepted 7 May 2018

Keywords:
Multicriteria optimization
Minimum spanning tree

ABSTRACT

This paper presents a new algorithm for the multiobjective minimum spanning tree problem that can be used with any number of criteria. It is based on a labelling algorithm for the multiobjective shortest path problem in a transformed network. Some restrictions are added to the paths (minimal paths) in order to obtain a one-to-one correspondence between trees in the original network and minimal paths in the transformed one. The correctness of the algorithm is proved as well as the presentation of a short example. Finally, some computational experiments were reported showing the proposed method outperforms the others in the literature. A deep study is also done about the number of nondominated solutions and a statistical model is presented to predict its variation in the number of nodes and criteria. All the test instances used are available through the web page http://www.mat.uc.pt/~zeluis/INVESTIG/MOMST/momst.htm.

1. Introduction

The minimum spanning tree (MST) problem is a classical combinatorial optimization problem studied since the beginning of the last century. It was proposed by Borůvka (1926) where he describes an algorithm which runs in $O(m \ln n)$ time to solve a problem with n nodes and m edges. Other methods were proposed like Prim's algorithm, which was first developed by Jarník (1930) and later rediscovered by Prim (1957) and Dijkstra (1959). It has time complexity of $O(m \ln n)$, but it can be improved to $O(m + n \ln n)$ if the Fibonacci heap is used as data-structure (Fredman and Tarjan, 1987). Kruskal (1956) proposed two methods: the well-known Kruskal's algorithm which runs in $O(m \ln n)$ time and the reverse-delete algorithm with time complexity of $O(m \, \ln n (\ln \ln n)^3)$, using the Thorup algorithm (Thorup, 2000), working with the edges in the reverse cost order. A more detailed development of the beginnings of the MST problem, its subsequent evolution and other methods can be found in Graham and Hell (1985); Gupta (2015); Karger et al. (1995); Nešetřil and Nešetřilová (2012) and Pettie and Ramachandran (2002).

The MST was motivated by a practical application in order to minimize the cost of the construction of a power line network in Southern Moravia after World War I (Graham and Hell, 1985). Despite the practical motivation in the origin of the MST problem, some interesting theoretical properties can be derived from this problem (Dasgupta et al., 2006) as well as other kind of applications in different fields of science, such as finance (Brookfield et al., 2013), ecotoxicology (Devillers and Dore, 1989), clustering analysis (Duran and Odell, 2013), weather forecasting (Gombos et al., 2007), image registration and segmentation (Meyer and Najman, 2013). The MST is also related to other combinatorial optimization problems like the Euclidian traveling salesman problem (Christofides, 1976), the multi-terminal minimum cut problem (Dahlhaus et al., 1994), matroids (Edmonds, 1971), the minimum-cost weighted perfect matching (Supowit et al., 1980) and the Steiner tree problem (Wei et al., 2015). To obtain more details about MST properties, applications or related problems, the reader is referred to Bazlamaçç and Hindi (2001); Graham and Hell (1985), Wikipedia (2017), Wu and Chao (2004) and Zsak (2006).

There are several variants of the MST problem in the literature like the Euclidean MST (Agarwal et al., 1991), the degree constrained MST (Torkestani, 2013), the k-smallest spanning trees (Gabow, 1977), the arborescence (Georgiadis, 2003), the hop-constrained and the diameter-constrained minimum spanning tree (Gouveia et al., 2011), the minimum labelling spanning tree (Krumke and Wirth, 1998), the maximum spanning tree (McDonald et al., 2005), the capacitated MST (Öncan, 2007), kth MST (Ravi et al., 1996), the delay MST (Salama et al., 1997) and the dynamic MST (Spira and Pan, 1975). Other variants of the MST problem are presented in Wikipedia (2017) and Wu and Chao (2004).

* Corresponding author.
E-mail addresses: zeluis@mat.uc.pt (J.L. Santos), luigi.dipugliapugliese@unical.it (L. Di Puglia Pugliese), francesca.guerriero@unical.it (F. Guerriero).

https://doi.org/10.1016/j.cor.2018.05.007
0305-0548/

In this work, we are focused on the multiobjective MST (MOMST) problem, which is a straightforward generalization of the classical MST problem where several weights are associated with each edge. In this problem, we aim to find the Pareto front, that is, a set of solutions for which it is not possible to improve one of the criteria without worsening any of the others. Regarding complexity analysis, it is NP-complete (Camerini et al., 1983) and intractable (Hamacher and Ruhe, 1994). The first algorithm proposed to the MOMST problem is due to Corley in 1985 (Corley, 1985) which is based on Prim's algorithm, but it can lead us to a solution set with non-efficient spanning trees (Hamacher and Ruhe, 1994). Hamacher and Ruhe (1994) proved that the efficient set is not connected when two spanning trees are considered adjacent (i.e., they have $n-2$ edges in common). He also provides an exact algorithm to compute the supported efficient spanning trees in the biobjective MST problem. The majority of the papers in the literature about this issue covers only the biobjective case (Clímaco and Pascoal, 2012; Ehrgott, 2005). Additionally, they deal mainly with approximate techniques, such as evolutionary algorithms (Knowles and Corne, 2001; Moradkhan and Browne, 2006), genetic algorithm (Han and Wang, 2005; Sanger and Agrawal, 2010; Zhou and Gen, 1999), local search (Andersen et al., 1996) and ant colony (Cardoso, 2006; Li et al., 2013).

As far as we know, only 3 types of exact algorithms are proposed to find the full Pareto front: the exhaustive search, dynamic programming (Di Puglia Pugliese et al., 2015) and two-phase methods (Ramos et al., 1998; Steiner and Radzik, 2008). The exhaustive search can only be used in a very small graphs because the number of spanning trees increases quickly with the number of the nodes; for instance, in a complete graph with n nodes, there are n^{n-2} spanning trees (Cayley, 1889). In the two-phase method, firstly the set of supported efficient spanning trees is computed. Then, in the second phase, the unsupported efficient spanning trees are searched inside "viable regions" formed by two adjacent supported efficient solutions. Steiner and Radzik (2008) showed that their approach, using a ranking procedure to search the unsupported solutions, outperforms the branch-and-bound technique applied in Ramos et al. (1998). However, the two-phase method is designed for the biobjective problems and the generalization for more than two criteria is hard to accomplish, (Ehrgott and Gandibleux, 2000). An extended review of the existing methods and properties for the MOMST problem can be found in Cardoso (2006); Ehrgott and Gandibleux (2000, 2002) and Ruzika and Hamacher (2009).

In this paper, we propose a new method based on a network transformation which produces a one-to-one correspondence between spanning trees in the original network and paths in the transformed one. Consequently, a labelling algorithm can be used taking advantage of its efficiency, easy implementation and generalization for more than 2 criteria. Applying this method to the transformed network, we find the Pareto solutions in the original network.

Regarding computational experiments, only one (Di Puglia Pugliese et al., 2015) of the papers, which consider the computation of the entire Pareto front, reports results for more than two objectives. The remainder focus exclusively on the bicriteria case. Furthermore, the lack of information about the tested problems prevents the replication of the computational experiments by other authors and the comparison of the results among different papers. In fact, in the papers analysed, the size of the networks varies from a dozen to hundreds of nodes, but there is no reference to the number of Pareto solutions. However, it is known the higher the number of solutions in the Pareto front, the larger the computational cost. According to the type of network considered, it is possible to build a small-size network with a large number of nondominated solutions and a large-size network with the opposite situation. This paper contributes to reducing the lack of information on this topic by providing a model to estimate the variation of the number of Pareto solutions with the number of nodes and the number of criteria. In addition, a public database with benchmark instances for this problem was created, which is available through the web page http://www.mat.uc.pt/~zeluis/INVESTIG/MOMST/momst.htm.

The paper is organized as follows. In Section 1, an introduction to the MST and the MOMST is presented, while Section 2 gives us the formal definition and notation used throughout the paper. Section 3 describes the new algorithm and the proof of its correctness. The computational results and the model to predict the number of Pareto solutions are reported in Section 4. Finally, Section 5 summarizes the paper and conclusions.

2. Definitions and notation

A graph (or an undirected graph) is an ordered pair $G = (N, E)$ where:

- $N = \{1, \ldots, n\}$ is the set of nodes;
- $E = \{e_1, \ldots, e_m\}$ is the set of edges, where each edge e_k ($k = 1, \ldots, m$) is a 2-element subset of N. Thus, e_k will be represented by an unordered pair $[i, j]$ with $i, j \in N$ and consequently $[i, j] = [j, i]$. Let E_i be the subset of E containing the edges linked to node i, that is, $E_i = \{e_\ell \in E : e_\ell = [i, j],$ for some $j \in N\}$. Multi-edges occur when there are different edges defined between the same pair of nodes.

A subgraph of G is a graph whose set of nodes is a subset of N and whose set of edges is a subset of E.

A path $p = \langle v_0, e_1, v_1, \ldots, e_\ell, v_\ell \rangle$ between s and t is a sequence of nodes and edges such that

- $v_i \in N, \ \forall i \in \{0, \ldots, \ell\}$;
- $s = v_0$ and $t = v_\ell$;
- $e_i = [v_{i-1}, v_i] \in E, \ \forall i \in \{1, \ldots, \ell\}$.

If $p = \langle v_0, e_1, v_1, \ldots, e_\ell, v_\ell \rangle$ and $q = \langle w_0, f_1, w_1, \ldots, f_k, w_k \rangle$ represent two paths in G where $v_\ell = w_0$, then $p \Diamond q = \langle v_0, e_1, v_1, \ldots, e_\ell, v_\ell = w_0, f_1, w_1, \ldots, f_k, w_k \rangle$ represents a path in G between v_0 and w_k.

A path can be represented only by the sequence of edges. If there are no multiple edges, it can also be represented only by the sequence of nodes. We denote by $\mathcal{P}_{s,t}$ the set of paths from node s to node t.

A graph G is connected if there is a path between each pair of nodes in N. A tree T is a connected subgraph of G with $n = |N|$ nodes and $n-1$ edges. We denote by $\mathcal{T}_X$ the set of trees whose set of nodes is $X \subseteq N$.

A graph is directed if each edge has an orientation. In this case, e_k will be represented by an ordered pair (i, j) and it will be called *arc*. Note that in this case $(i, j) \neq (j, i)$. In a directed graph, multiple-arcs are defined as arcs between the same pair of nodes with the same direction. A path in a directed graph has to preserve the arc orientation. An undirected graph can be transformed into a directed graph $G^d = (N, A)$, where each edge $e = [i, j] \in E$ defines two arcs in A, (i, j) and (j, i), between i and j with opposite directions.

A network is a triplet (N, E, c) where:

- (N, E) is a graph;
- c is a vectorial cost function that assigns the vector cost $c([i, j]) \in \mathbb{R}^k$ to the edge $[i, j]$,

where k is the number of objectives or criteria. We will assume that all the components of the vector cost are nonnegatives.

Fig. 1 shows a network example that will be used throughout the paper. The cost of a path p, $c(p)$, is the sum of the cost for