Paper

A note on the computation of ordered supported non-dominated solutions in the bi-criteria minimum spanning tree problems

Carlos Gomes da Silva and João C. N. Clímaco

Abstract—This paper presents a new procedure for computing the set of supported non-dominated solutions of bi-criteria minimum spanning tree problems in ordered manner. The procedure is based on the systematic detection of edges which must be replaced in one efficient solution to obtain the adjacent one, in the criteria space. This new approach avoids solving unnecessary problems and makes use of previous computations.

Keywords— minimum spanning tree, supported non-dominated solutions, combinatorial problems.

1. Introduction

The minimum spanning tree problem (MST) is a well-known combinatorial problem that consists of identifying a spanning tree in a weighted connected graph with the smallest sum of costs (weights). The MST has several important practical applications such as physical systems design, reducing data storage, cluster analysis [2]. It is also important because it arises as a subproblem of other complex problems. For instance, in telecommunication multipath models the area of multicast routing (associated with point-multipoint problems) has attracted increasing attention both in terms of quality of service (QoS) routing models, and in terms of explicit consideration of multicriteria in the future. These models are increasingly important as a result of the emergence of multimedia applications such as audio, video services and video-conferencing, specially in the Internet. A typical formulation of multicast routing models in a QoS routing context involves Steiner trees [5]. These problems are in general very difficult to solve. So, it is valuable to point out that in some particular approaches the multicriteria minimal spanning tree models can be useful, and this is a major motivation for our focus on this problem.

A connected graph with n nodes has a maximum of n^{n-2} spanning trees, thus the *brute force* method is useless to solve the MST. The greedy Prim's, Kruskal's and Sollin's algorithm solve efficiently the MST [2].

A natural extension of the MST consists of taking into account more than one criterion function to evaluate feasible solutions. This new problem is known to be NP-complete [3]. The multiple criteria version of the MST is a challenge both from theoretical and practical perspectives, as can be seen in the literature survey by Ehrgott and Gandibleux [6].

A common approach when considering several criteria functions is the computation of the entire set of efficient solutions, also known as Pareto solutions (an efficient solution is a feasible solution such there is no other feasible one that can improve one of the criteria functions without degrading the value of at least one of the others). Some efficient solutions can be found by optimizing weighted-sums of the criteria (*supported efficient solutions*) while others cannot be obtained by this manner (*non-supported efficient solutions*). The supported efficient solutions can also be *extreme* (solutions that correspond to an extreme efficient solution in the *convex-hull* of the feasible region) or *non-extreme*. The image of an efficient solution, by using the criteria functions is called a *non-dominated* solution.

The computation of extreme efficient solutions is much easier when an efficient algorithm exists to optimize the single criterion version of the problem. Due to this fact, the computation of the two types of solutions is frequently made separately. In the MST this approach can be found in [1, 7, 8], for instance. The weighted-sum method is usually used to compute the extreme efficient solutions. It is an iterative procedure which computes, for a given pair of consecutive extreme efficient solutions x', x'' another extreme efficient solution x''' (if there exists any) between the images of x' and x''. This is done by optimizing a weighted-sum function parallel to the line that links the images of x' and x''. The method requires the optimization of several single criterion functions, some of them without producing any new efficient solution.

In this paper we propose an alternative procedure for the bi-criteria minimum spanning tree problems (BMST), which is based on the extension of ideas behind the Kruskal's algorithm. The main features of this procedure are the use of computations made in previous iterations and the fact that it avoids the repetitive resolution of single criterion problems. This is due to the fact that the edges that must leave an efficient MST and the edges that must enter that MST are identified. By removing and inserting these edges, the adjacent non-dominated MST is obtained. In summary, the paper gives a transition rule

from non-dominated solution to the adjacent one. The work by Tarjan [9], related to sensitivity analysis in minimum spanning trees, is useful in the present research.
The paper is organized as follows. In Section 2 it is presented a procedure to find the set of extreme non-dominated solutions in order, in Section 3 it is shown the interactive potential of the proposed approach, and Section 4 is devoted to the main conclusions of this work.

2. Finding supported non-dominated solutions in order

Let $G = (V_n, E)$ be an undirected graph with V_n being the set of n vertices and E being the set of edges of G. Here it is considered the existence of two criteria, hence each edge e_j has associated two costs $c_j^i (i = 1, 2)$. The criteria, to be minimized, are as follows:

$$z_t(T) = \sum_{j: e_j \in T} c_j^t,$$

where T is a spanning tree on G.

Supported non-dominated solutions optimize weighted sum functions. In the bi-criteria case, these functions are $f_\lambda(T) = \lambda z_1(T) + (1-\lambda) z_2(T)$, with $0 \le \lambda \le 1$. When the entire interval $[0,1]$ for λ was analyzed, then all the non-dominated supported solutions were found. Let G^λ be the graph G such that each edge e_j has the cost $p_j^\lambda = \lambda c_j^1 + (1-\lambda) c_j^2$.
We are interested in the computation of such a solutions but in an ordered manner, which is a new result considering the available procedures in the literature. The process of obtaining the ordered generation of supported non-dominated solutions is explained below.
A spanning tree is composed of $n-1$ edges, and following the Kruskal's algorithm one has to consider the edges according to non-decreasing costs and select the $n-1$ edges that draw a spanning tree on the given graph.
In the bi-criteria case, the cost of each edge e_j is p_j^λ, which depends on the value of λ. Thus, the minimum spanning tree of $f_\lambda(T)$ also depends on the value of λ. As λ changes in the interval $[0,1]$ some edges also change their relative position, which may lead to a different minimum spanning tree.
According to the path optimality condition [2] a spanning tree T is a minimum spanning tree if and only if every non-tree edge e_j has a cost greater or equal than the cost of any edge in the unique path of T that links the nodes concerning e_j.
Suppose that for a given λ, λ^k, the corresponding minimum spanning tree is T^{λ^k} and the associated supported non-dominated solution is $\left(z_1(T^{\lambda^k}), z_2(T^{\lambda^k})\right)$. This tree remains optimum for every λ which observes the path optimality condition.
Let $P^{\lambda^k}(e_j)$ be the path in T^{λ^k} which links the nodes concerning the non-tree edge e$_j$. For the non-tree edge e_j the maximum value of λ is given by the solution of the linear problem:

$$\begin{aligned} & \max\ \lambda \\ & s.t.: \\ & p_j^\lambda \ge p_i^\lambda, \forall e_i \in P^{\lambda^k}(e_j) \\ & 0 \le \lambda \le 1, \lambda > \lambda^k. \end{aligned} \tag{1}$$

The optimal solution of Eq. (1), $\lambda_{\max}^{e_j}$, is given below (if the problem is impossible, which occurs when the cost of two edges never gets equal or when it requires a weight outside the constraints $0 \le \lambda \le 1, \lambda > \lambda^k$, it is assumed, for computational reasons, that $\lambda_{\max}^{e_j}$ takes the value $+\infty$):

$$\lambda_{\max}^{e_j} = \min_{e_i \in P^{\lambda^k}(e_j)} \left\{ \frac{c_i^2 - c_j^2}{\left(c_j^1 - c_j^2\right) - \left(c_i^1 - c_i^2\right)} : \left(c_j^1 - c_j^2\right) - \left(c_i^1 - c_i^2\right) < 0; +\infty \right\}. \tag{2}$$

In order to preserve the optimality of the spanning tree, the path optimality condition must be observed by every non-tree edge. Thus the overall maximum value is given by $\lambda_{\max}^k = \min_{e_j \notin T^{\lambda^k}} \left\{\lambda_{\max}^{e_j}\right\}$.

Let $N_{\lambda^k} = \left\{e_j \notin T^{\lambda^k} : \lambda_{\max}^{e_j} = \lambda_{\max}^k\right\}$, representing candidates to entering the tree. Introducing e_j in the tree leads to a cycle. Thus, the candidate edges to be removed are the ones belonging to the set

$$C_{\lambda^k}(e_j) = \left\{ e_i \in P^{\lambda^k}(e_j) : \frac{c_i^2 - c_j^2}{\left(c_j^1 - c_j^2\right) - \left(c_i^1 - c_i^2\right)} = \lambda_{\max}^k, e_j \in N_{\lambda^k} \right\}.$$

If $\lambda = \lambda_{\max}^k$ the current tree is still the optimum tree, but there exists at least another tree with the same weighted cost. Finding all the supported non-dominated solutions with the same weighted cost as T^{λ^k}, requires replacing in the tree every combinations of possible pairs of edges $\left(e_j \in N_{\lambda^k}, e_i \in C_{\lambda^k}(e_j)\right)$. Let us note these combinations by $N_{\lambda^k} \otimes C_{\lambda^k}$.

If $\lambda = \lambda_{\max}^k + \varepsilon$ (ε is a small value) the current tree is not the optimum tree of the problem $\min\left\{f_\lambda(T) : T \text{ is a spanning tree on } G\right\}$, since at least one non-tree edge has a smaller weight compared with at least one edge of the associated path in the tree. An optimum tree in this case is the one which has the minimum value for criterion $z_1(T)$ (note that λ is increasing towards the value 1), among all the efficient MST obtained with $\lambda = \lambda_{\max}^k$.

The procedure that generates the supported non-dominated solutions of a bi-criteria MST problem is as follows.

Procedure *SBMST*

Begin

$k \leftarrow 1$;// iterations counter

Compute the first MST, T^{λ^k}, with the cost edges calculated with $\lambda^k = \varepsilon$ (very small positive value);

$H \leftarrow Z(T^{\lambda^k}) = \left(z_1(T^{\lambda^k}), z_2(T^{\lambda^k})\right)$ //set of supported non-dominated spanning trees

Compute $\lambda^k_{\max}$ using expression (2) in T^{λ^k} and define N_{λ^k};

While $\left(\lambda^k_{\max} < 1\right)$ **Do**

Begin

Compute $C_{\lambda^k}(e_j)$ for all $e_j \in N_{\lambda^k}$;

Consider all the possible combinations of edges $N_{\lambda^k} \otimes C_{\lambda^k}$;

Let $T_1^{\lambda^k}, \ldots, T_h^{\lambda^k}$ be the trees obtained from T^{λ^k} by inserting/removing edges considering individually each of the previous combinations of edges;

$H \leftarrow H \bigcup\limits_{i=1}^{h} Z(T_i^{\lambda^k})$ //set of supported non-dominated spanning trees

$k \leftarrow k+1$;

$T^{\lambda^k} \leftarrow \arg\min\{z_1(T) : T \in H\}$;// the tree with the lowest value in criterion z_1;

Compute $\lambda^k_{\max}$ using expression (2) in T^{λ^k} and define N_{λ^k};

End

End

In summary, the procedure starts with the computation of an initial MST, which optimizes criterion z_2 (chosen arbitrarily). The maximum value of λ which maintains the MST is computed as well as the candidates edges to enter the tree, are computed. If the maximum value of λ is greater than or equal to 1 the procedure stops. Otherwise, the leaving edges are identified for each entering candidate. Supported non-dominated trees generated by removing and inserting identified edges are used to update the list of solutions. The tree corresponding to the lowest value of criterion z_1 is the new reference tree and the above steps are repeated.

In order to illustrate the above procedure, let as consider the following example.

Example 1. Let G be the network presented in Fig. 1, where the cost of the edges according to the two criteria are also presented. The purpose is to obtain all the extreme non-dominated minimum spanning trees.

In Fig. 2 the functions $p_j^{\lambda}, j = 1, \ldots, |E|$ are represented. The vertical lines correspond to a change in the orders of the costs of the edges.

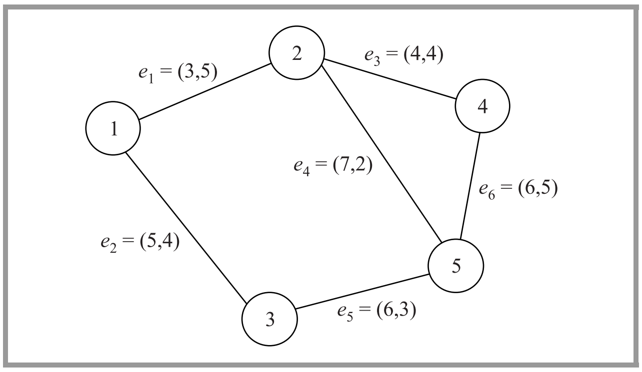

Fig. 1. The starting graph.

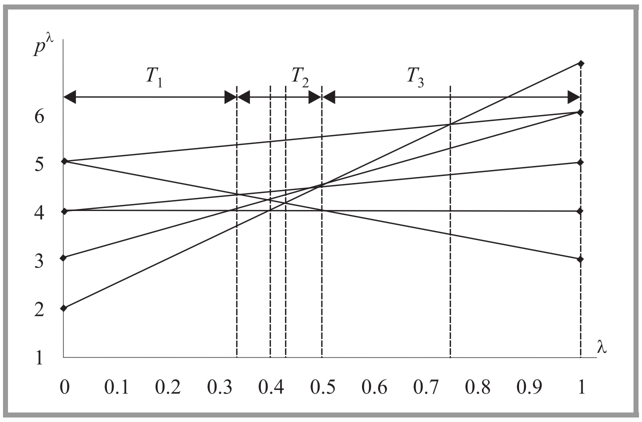

Fig. 2. Functions p_j^{λ}.

Using the procedure SBMST proposed above, the interval [0,1] for λ is partitioned into 3 relevant sub-intervals, each of them corresponding to a different non-dominated extreme MST.

$Z(T^{\lambda^1}) = (22,13)$; $\lambda^1_{\max} = \frac{1}{3}$; $N_{\lambda^1} = \{e_1\}$; $C_{\lambda^1}(e_1) = \{e_2\}$;

$Z(T^{\lambda^2}) = (20,14)$; $\lambda^2_{\max} = \frac{1}{2}$; $N_{\lambda^2} = \{e_2\}$;

$C_{\lambda^2}(e_2) = \{e_4, e_5\}$.

$N_{\lambda^2} \otimes C_{\lambda^2} = \left\{\left(e_2, e_4\right), \left(e_2, e_5\right)\right\}$; $Z(T_1^{\lambda^2}) = (18,16)$;

$Z(T_2^{\lambda^2}) = (19,15)$; $T^{\lambda^3} = T_1^{\lambda^2}$; $\lambda^3_{\max} > 1$.

Figures 3, 4 and 5 show three extreme efficient MST of the initial problem. Figure 6 presents the images in the criteria

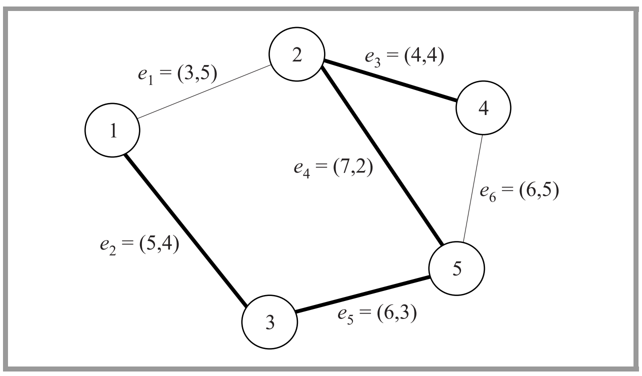

Fig. 3. First extreme eff.MST.

space of each solution (the points are connected for graphic visualization of the Pareto frontier).

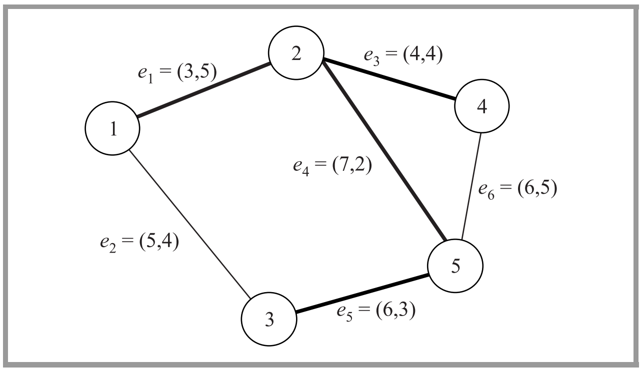

Fig. 4. Second extreme eff.MST.

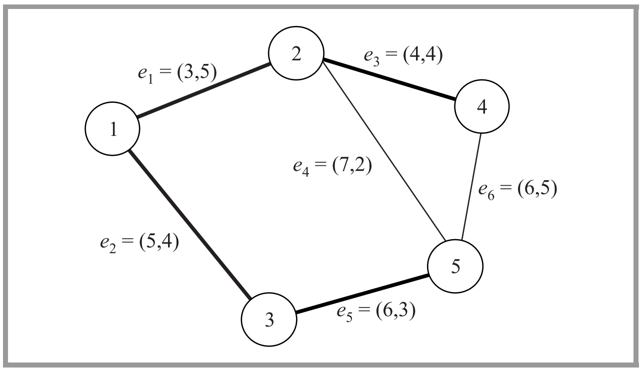

Fig. 5. Third extreme eff.MST.

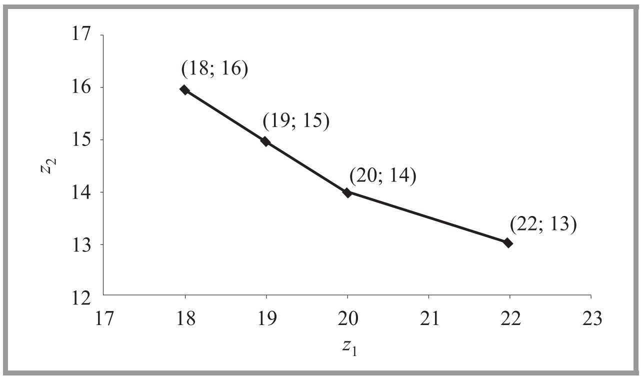

Fig. 6. Pareto front.

Remark. From the previous presentation it is easy to see that if an instance of a BMST observing the two following conditions: 1) $p_i^\lambda < p_j^\lambda, i = 1,...,n-1;\ j = n,...,|E|$ for all $0 \leq \lambda \leq 1$; 2) then $\{e_1, e_2, e_3, ..., e_{n-1}\}$ are the edges of the single non-dominated spanning tree of the problem.

Instances observing the conditions above are exceptions. In general, the BMST problem has several supported non-dominated solutions.

3. Interactive usefulness

The results presented in the previous section can be useful for building an interactive procedure dedicated to a progressive and selective calculation of the supported non-dominated spanning trees, according to the preferences of the decision maker elicited during the dialogue phase of the interactive process.

A very simple extension of the procedure SBMST, enables the obtainment of the sub-interval of values of λ leading to a same extreme efficient solution. The upper-bound of the sub-interval is obtained as in Eq. (2) and the lower bound, $\lambda_{\min}^k$, is obtained by solving problem Eq. (1) replacing its objective function by min λ and the constraint $\lambda > \lambda^k$ by $\lambda > \lambda_{\max}^{k-1}$. The optimal solution of the latter problem, is given by

$$\lambda_{\min}^{e_j} = \max_{e_i \in P^{\lambda^k}(e_j)} \left\{ \frac{c_i^2 - c_j^2}{\left(c_j^1 - c_j^2\right) - \left(c_i^1 - c_i^2\right)} : \left(c_j^1 - c_j^2\right) - \left(c_i^1 - c_i^2\right) > 0; +\infty \right\}. \tag{3}$$

Thus, $\lambda_{\min}^k = \max\limits_{e_j \notin T^{\lambda^k}} \left\{ \lambda_{\max}^{e_j} \right\}$.

The progressive focus in part, or parts of the original interval of λ, i.e., [0,1], can be achieved by an ad hoc procedure that consists of eliminating the sub-intervals corresponding to efficient solutions already calculated and, possibly, other sub-intervals specified indirectly, for instance, by constraints on the objective function values introduced by the decision maker. This may happen when those constraints intersect edges of the convex hull connecting adjacent extreme non-dominated solutions already calculated.

Alternatively, the progressive focus in part, or parts, of the interval of λ, i.e., [0, 1], can also be achieved using a NISE-like approach (see [4]). Note that, in this case, the combination of the NISE-procedure steps with the progressive calculation of the sub-intervals of λ, corresponding to the extreme non-dominated solution calculated following the NISE approach, enables a faster reduction of the unexploited sub-intervals thereby accelerating the convergence of the process.

Example 2. Let us consider the data from Example 1 and suppose that the decision maker specifies $\lambda = 0.8$. By optimizing the weighted-sum function $f_\lambda(T) = 0.8 z_1(T) + (1-0.8) z_2(T)$, the extreme non-dominated solution $z = (22, 13)$ is obtained. This solution is associated with the sub-interval $[0.5, 1]$ thus, in the following iteration, the decision maker is asked to select a weight in the interval $[0,1] \setminus [0.5, 1]$.

Extension to the multicriteria case. When there are more than two criterion functions, the above results can be adapted. Given a supported efficient solution of the multicriteria problem, i.e., an optimal solution of the problem $\{\min \lambda_1^0 z_1(T) + \lambda_2^0 z_2(T) + ... + \lambda_q^0 z_q(T) : T$ is a spanning tree on G, where $\lambda_1^0+\lambda_2^0+...+\lambda_q^0=1$ and $\lambda_j^0>0$ $(j=1,...,q)$ it still optimizes the weighted-sum functions: $\{\min\lambda_1 z_1(T)+\lambda_2 z_2(T)+...+\lambda_q z_q(T) : T$ is a spanning tree on $G\}$, such that $\lambda_1 c_j^1+\lambda_2 c_j^2+...+\lambda_q c_j^q \geq \lambda_1 c_i^1 + \lambda_2 c_i^2+...+\lambda_q c_i^q, \forall e_i \in P^{\lambda^0}(e_j), \forall e_j \notin T^{\lambda^0}$.

Thus the feasible region for the weights can be presented to the DM in order to avoid redundant specifications of new weights for the objective functions.

4. Conclusions

A constructive procedure was proposed to compute the entire set of supported non-dominated solutions of the BMST. The procedure relies on the weighted-sum functions of the edges and on the ideas behind Kruskal's algorithm and sensitivity analysis in minimum spanning trees. With this procedure we can identify the edges which must be inserted/deleted from a supported efficient MST, to obtain an adjacent efficient MST. The procedure for finding the weight λ which conducts to the same extreme non-dominated solution was also used to support an interactive framework. The integration of other well-known (potentially interesting) interactive tools was just outlined. The method can also be useful in the multicriteria case.
The exploration of the procedure for finding the non-supported non-dominated solutions is a future line of research.

References

[1] K. Andersen, K. Jörnsten, and M. Lind, "On bicriterion minimal spanning trees: an approximation", *Comput. Oper. Res.*, vol. 23, pp. 1171–1182, 1996.

[2] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows – Theory, Algorithms And Applications*, New Yersey: Prentice Hall, 1993.

[3] P. Camerini, G. Galbiati, and F Maffioli, "The complexity of multi-constrained spanning tree problems", in *Theory of Algorithms, Colloquium Pecs 1984*, L. Lovász, Ed. Amsterdam: North-Holland, 1984, pp. 53–101.

[4] J. Cohon, *Multiobjective Programming and Planning*. New York: Academic Press, 1978.

[5] J. Clímaco, J. Craveirinha, and M. Pascoal, "Multicriteria routing models in telecommunication networks – overview and a case study", in *Advances in Multiple Criteria Decision Making and Human Systems Management: Knowledge and Wisdom*, Y. Shi, D. Olson, and A. Stam, Eds. Amsterdam: IOS Press, 2007, pp. 17–46.

[6] M. Ehrgott and X. Gandibleux, "A survey and annoted bibliography of multiobjective combinatorial optimazation", *OR Spektrum*, vol. 22, pp. 425–460, 2000.

[7] H. W. Hamacher and G. Ruhe, "On spanning tree problems with multiple objectives", *Ann. Oper. Res.*, vol. 52, pp. 209–230, 1994.

[8] R. M. Ramos, S. Alonso, J. Sicilia, and C. González, "The problem of the optimal biobjective spanning tree", *Eur. J. Oper. Res.*, vol. 111, pp. 617–628, 1998.

[9] R. Tarjan, "Sensitivity analysis of minimum spanning trees and shortest path trees", *Inform. Proces. Lett.*, vol. 14, pp. 30–33, 1982.

Carlos Gomes da Silva received his Ph.D. degree in operational research at University of Coimbra, Portual, in June 2005. He is Professor at Polytechnic of Leiria and a Researcher at INESC-Coimbra. His main research interests are multicriteria combinatorial problems and meta-heuristic resolution procedures.

e-mail: cgsilva@estg.ipleiria.pt
Escola Superior de Tecnologia e Gestão de Leiria
Morro do Lena, Alto Vieiro
2401-951 Leiria, Portugal

INESC-Coimbra
Rua Antero de Quental, 199
3000-033 Coimbra, Portugal

João C. N. Clímaco – for biography, see this issue, p. 10.