A Bi-Criteria Algorithm for Multipoint-to-Multipoint Virtual Connections in Transport Networks

Lúcia Martins
Dep. of Electrical and Computer Eng.
University of Coimbra
INESC Coimbra
Rua Antero de Quental, 199
3000–033 Coimbra Portugal
Email: lucia@deec.uc.pt

Nuno Ferreira
INESC Coimbra
Rua Antero de Quental, 199
3000–033 Coimbra Portugal
Email: nfgferreira@hotmail.co.uk

José Craveirinha
Dep. of Electrical and Computer Eng.
University of Coimbra
INESC Coimbra
Email: jcrav@deec.uc.pt

***Abstract*—For a Carrier Ethernet or MPLS-TP service provider the determination of a multipoint-to-multipoint (mp2mp) virtual connection must be done in an efficient way in order to maximise network performance. This problem can be formulated as a Steiner tree problem which is a very complex combinatorial problem, being NP-complete. In general, any transport network ideal management system should seek to balance the load of the links while, at the same time, seeking to minimize the resources involved in each connection. In this work it is presented a bi-criteria formulation for the Steiner tree problem which includes these objectives. In order to solve this problem a heuristic was developed based on a known bi-criteria spanning tree algorithm that finds the set of efficient supported solutions in a given sub-graph. This new heuristic allows to obtain 'good' compromise Steiner trees in an efficient manner. The obtained solution set contains some times the optimal solution for each objective included in the problem formulation. It is important to note that the analysis of the whole set of solutions may be important in some management transport network scenarios.**

I. Introduction

In telecommunication networks, bandwidth is a very important resource and it must be efficiently managed at the transport network level in order that the right amount of bandwidth be available in each part of the network according to some previous agreement between the transport network service provider and each transport network user, which can be also a telecommunication service provider. At the transport network level each connection usually is active for a significant amount of time.

In the new packet-based transport technologies such as Carrier Ethernet or MPLS–TP (Multiprotocol Label Switching–Transport Profile) the services that can be delivered are point-to-point (p2p), point-to-multipoint (p2mp) and multipoint-to-multipoint (mp2mp) virtual connections, each one characterized by a parameter set known as SLA (Service Level Agreement) [1], [2].

From the perspective of the transport network bandwidth management, each virtual connection must be created using the minimum amount of network resources while network load must be distributed in a balanced manner over the links. These two objectives can be integrated in a bi-criteria optimization problem in which, in general, a set of Pareto optimal solutions can be defined where a 'good' compromise solution can be chosen. A Pareto optimal solution (or non dominated solution) is a feasible solution such that there is no other feasible solution which improves one objective function value without worsening the value of any other objective. These solutions can be supported or unsupported. Supported non dominated solutions are non dominated solutions located on the boundary of the convex-hull of the feasible solution set while unsupported solutions are located in the interior. An in depth analysis of the potential advantages of using multi-criteria approaches in communication network routing and an overview of works in this area, including multicast routing models, is in [3], [4]. Furthermore, there are network particularities which impose restrictions that are not easily integrated in the algorithms for virtual connections calculation and so, having a set of non dominated solutions instead of a single one, can be an advantage.

In a previous work [5] a bi-criteria approach to find p2p virtual connections in a transport network, was proposed. In that work the network overall performance was evaluated in several aspects and it was shown that bi-criteria optimization can be a very good approach for bandwidth management. In this paper a heuristic to find bi-criteria mp2mp virtual connections in transport networks is presented and its performance is evaluated through a comparison between the obtained results and some reference values from a library of Steiner tree problems [6].

This paper is organized as follows. In section II the problem of finding 'optimal' mp2mp virtual connections in transport

Work financially supported by programme COMPETE of the EC Community Support Framework III and cosponsored by the EC fund FEDER and national funds (FCT).

networks is formulated as a bi-criteria Steiner tree problem. In section III a heuristic to find 'good' compromise bi-criteria Steiner trees is presented and in section IV the performance evaluation of the proposed heuristic is presented. Finally the conclusions are outlined in the last section.

II. A Bi-Criteria Steiner Tree Problem

A mp2mp virtual connection between a subset of nodes in a network consists of a sequence of nodes and links connected in such a way that there is only a single path between each pair of nodes of that subset. This graph structure corresponds to a tree where it is not possible the existence of cycles [7]. In general there are not enough direct links between the nodes which have to be connected so that some other nodes need to be added for establishing that connection. The problem of finding a minimum amount of network resources, according to a given metric, for the establishment of a mp2mp virtual connection corresponds to a Steiner tree problem [8], [7]. This is a combinatorial problem for which several heuristics such as [9], [10], [11], [12] and meta-heuristics such as [13], [14], [15] have been proposed.

Formally, given an undirected graph $G(N,A)$ where N is a set of $1,2,\ldots,|N|$ nodes (or vertices) and A is the set of links (or arcs) (i,j) connecting node i to node j with a cost $C_{i,j}$ and given a subset $S \subset N$ of nodes, the problem of finding the minimum cost tree T_S spanning all nodes in S and possibly some optional nodes in $N \setminus S$ is known as a Steiner tree problem (STP). The nodes in S are designated as terminal nodes and the optional nodes in $N \setminus S$ are the Steiner nodes. The cost of the tree is the sum of its arc costs.

Finding the minimum tree spanning all the nodes in the graph is known as the minimum spanning tree (MST) problem which can be solved by known polynomial-time algorithms such as Kruskal or Prim's algorithms [7].

In a bi-criteria MST approach there are two, usually conflicting, objective functions associated with two costs $\left(C^1_{i,j}, C^2_{i,j}\right)$ in every arc of the graph. The problem consists of finding the set of efficient trees $\{T\} \subset \mathcal{T}$, where $\mathcal{T}$ is the set of all the trees $T=(N,A(T))$ in G, which corresponds to the set of non-dominated points in the objective functions space. The cost of the tree $C_T=(C^1_T, C^2_T)$ is represented as a vector-valued function, such that

$$C^l_T = \sum_{i,j:(i,j)\in T} C^l_{i,j}, \quad l=1,2 \tag{1}$$

The set of efficient trees $\{T\} \subset \mathcal{T}$ is such that, for every $T^{'} \in \mathcal{T}$, $C^l_T \le C^l_{T'}$, with $l=1,2$, $T \ne T^{'}$ and $C^l_T < C^l_{T'}$ for, at least, one value of l. The number of efficient spanning trees in a graph is exponential, with a maximum equal to $|N|^{|N|-2}$ and so the bi-criteria MST problem is intractable and NP-hard [16]. Nevertheless the number of supported efficient spanning trees is polynomially bounded by $|A|^2$ for the bi-criteria case and can be efficiently computed using a weighted sum method such as in [17]. Note that each MST of $\mathcal{T}$ has always $N-1$ arcs. For an overview of multiple objective MST problems see [16]. In [18], for instance, all the efficient solutions of the bi-criteria MST are computed by recurring to two approaches. The one that presents better performance is based on a k–minimum spanning tree algorithm such as the one in [19].

Concerning the bi-criteria Steiner tree problem there are a few heuristics and meta-heuristic proposals such as [20], [21] but as the complexity of the problem is very high there is no single methodological approach that must be followed. A bi-criteria STP is formulated in this paper in order to find 'good' compromise solutions, in terms of bandwidth management, for mp2mp virtual connections in transport networks . The metrics that are going to be used are hop count and load cost, defined from the piecewise linear function of the occupied bandwidth in [22], in order to find solutions that use as less network resources as possible and to distribute as much as possible in a balanced form the traffic load all over the network. These metrics are additive and the tree cost is given, as before, by the cost vector $C_{T_S} = (C^1_{T_S}, C^2_{T_S})$. Note that in the case of STP, for the same number of terminal nodes, each optimal tree may have a different number of arcs in different networks with the same size, i. e. with the same number of arcs and nodes, and that's why we use hop count as a metric in this problem. Another aspect to be considered is that, in a heuristic, each obtained solution that dominates all the calculated solutions can be in fact dominated by some other solution which was not found by the heuristic because the solution space was not completely explored.

III. A Bi-Criteria Steiner Tree Heuristic

The heuristic approach for the bi-criteria STP is based on the Kou et al. [9] heuristic for the single criterion problem. This procedure has the following steps:

- Step 1: Construct the complete undirected graph $G_1(S,A_1)$ such that A_1 is a set of arcs between each pair of nodes in S and so $|A_1| = |S|(|S|-1)/2$. The arc $(i,j) \in A_1$ corresponds to the shortest path $r_{i,j}$ in the graph G calculated with $C_{i,j}$ (the cost of arc (i,j) in graph G) using the Dijkstra algorithm and denoting by $\mathcal{C}_{i,j}$ the cost of arc (i,j) in graph G_1, this is given by $\mathcal{C}_{i,j} = \sum_{(k,l)\in r_{i,j}} C_{k,l}$;
- Step 2: Find the MST T_1 of G_1, for instance with Kruskal's algorithm;
- Step 3: Construct the sub-graph $G_{S'}$ of G by replacing each arc in T_1 by its corresponding shortest path in G;
- Step 4: Find the new MST $T_{S'}$ by removing the cycles in $G_{S'}$;
- Step 5: Construct T_S from $T_{S'}$ by removing unnecessary arcs in order that all leaves in the tree are terminal nodes.

The first attempt to develop a heuristic to obtain bi-criteria Steiner trees was based on the previous one by replacing the MST computation in Step 2 by the Hamacher et al. algorithm which finds all supported efficient spanning trees in graph G_1 [17]. For that purpose a cost pair $(\mathcal{C}^1_{i,j}, \mathcal{C}^2_{i,j})$ is associated with every arc in G_1in which the second cost considered, $\mathcal{C}^2_{i,j}$, is the hop count. This cost can be easily obtained by computing the hop count of each path $r_{i,j}$ in G.

The results obtained with this strategy were very encouraging because it is possible with this heuristic to find, in some networks, solutions which dominate the solution obtained by the original Kou heuristic, using the load cost metric. This means that with this new approach it is possible, in some cases, to find a solution that has lower load cost and lower hop count than the solution obtained by Kou et al. heuristic [9]. This fact was a strong motive to improve this first bi-criteria STP heuristic.

The basic key idea to improve this heuristic was the fact that the optimal solution for the (single criterion) STP is often composed of paths of low cost order in the cases where the shortest path for some node pairs is not the best option. This fact leads to changing the Step 1 of the previous heuristic replacing G_1 by $G_2(S, A_2)$ with $|A_2| = k|S|(|S|-1)/2$ where k represents the number of shortest paths between each pair of nodes calculated by the k–shortest path algorithm MPS [23]. This means that we obtain k parallel arcs between each pair of nodes in G_2 where each arc corresponds to one of the k shortest paths.

The use of parallel arcs can be easily tackled by our implementation of the Hamacher et al. algorithm [17] which is used in Step 2. Additionally the second metric used in the bi criteria minimum spanning tree algorithm was replaced by a new metric which results from counting the number of times that each arc in G belongs to each arc in A_2. This new metric in Hamacher et al. algorithm tends to lead to better solutions because it tends to consider arcs in G_2 which have in common a greater number of arcs in G hence promoting the consideration of lower cost Steiner trees (in terms of load cost and hop count).

Let us consider that r_{od} is the path in G between $o, d \in S$ which corresponds to $(o,d)^{r_{od}}$ in G_2 and that R_{od} is the set of all the paths r_{od} between o and d. Let $R = \cup_{s,t \in S} R_{st}$ be the union of all the possible sets R_{st} and N_{ij}^{st} be the number of times that the arc $(i,j) \in r_{od}$ appears in each one of the sets R_{st}. Then the new second cost of each arc in G_2, $\mathcal{C}^2_{(o,d)^{r_{od}}}$ is given by:

$$\mathcal{C}^2_{(o,d)^{r_{od}}} = -\frac{\sum_{(i,j)\in r_{od}} \sum_{R_{st} \subset R \setminus R_{od}} \left(N_{ij}^{st}/|R_{st}|\right)}{\text{hop count}(r_{od})} \qquad (2)$$

The Bi-Criteria Steiner Tree Problem (BCSTP) heuristic can now be summarized as follows:

1) Construct the complete undirected graph $G_2(S, A_2)$ with k parallel arcs between each pair of terminal nodes, as previously described. Three cases were considered for the construction of graph G_2: i) with load cost in MPS algorithm on graph G; ii) with hop count in MPS algorithm on graph G; iii) with a linear combination of load cost and hop count in MPS algorithm on graph G. Note that this last case corresponds to the consideration of a bi-criteria shortest path formulation [24], [5].
2) Compute the second cost of each arc in G_2 according to equation 2;
3) Find the set of efficient supported MST, $\{T_2\}$, of G_2 with the Hamacher et al. algorithm [17];
4) Construct the set of sub-graphs $\{G_{S'}\}$ of G by replacing each arc of each T_2 by its corresponding shortest path in G;
5) Find the set $\{T_{S'}\}$ of new MST by removing the cycles in each $G_{S'}$ through the elimination of the arc with the higher load cost;
6) Construct the set $\{T_S\}$ from $\{T_{S'}\}$ by removing unnecessary arcs in order that all leaves in the tree are terminal nodes and store it.
7) Remove all dominated solutions from the set of all stored $\{T_S\}$ according to the hop count as well as the load cost values of each tree.

IV. Analysis of Results

The proposed BCSTP heuristic was evaluated by using benchmark graphs from the the Steinlib Testdata Library [6]. For this purpose, load cost was replaced by the arcs costs in the SteinLib.

One parameter that has to be tuned first is the number of parallel arcs that must be considered in the heuristic. As a first approximation k should be a small number such as 2 or 3 but in Steinlib there are some networks which have a lot of alternative paths with the same cost between each pair of terminal nodes. In order to deal with this situation, the number of parallel arcs considered in the construction of graph G_2 was divided into two parameters: one is the path cost order number which was set to 2 or 3, as mentioned before; the other one is the total number of paths which was set to 10.

In the following figures the aggregated results for the sets B, C, I80 and I640 test graphs were presented for the BCSTP heuristic for each one of the three cases considered in the Step 1 (construction of the graph G_2): i) load cost (BCSTP-LC); ii) hop count (BCSTP-HC); iii) a linear combination (BCSTP-LComb) of the two metrics. There are also results for $k = 2$ (2-10) and $k = 3$ (3-10) shortest paths between each pair of nodes of G_2 graph. For comparison purposes the results obtained by Kou et al. algorithm using the load cost metric (in figures 1 and 2) or the hop count (in figure 3) were included. The results obtained by the heuristic without parallel arcs in G_2, designated as BCSTP-xx (1-1) in the figures, were also considered.

In figure 1 it is presented, for each heuristic, the percentage of number of times for which the optimal solution was obtained for each set of networks. As it can be seen in the figure, the optimum was obtained by BCSTP-LC (2-10) in 55% of the cases, for B networks. However, for I80 and I640 networks a very small number of optimal solutions was obtained by BCSTP-LC (x-x) because these networks are very difficult Steiner problems. Note that in most of the cases there were no optimal values obtained with BCSTP-HC (x-x), using the hop count in Step 1, because these were the heuristic variants which lead to the best hop count solutions. The linear combination of the metrics, BCSTP-LComb (x-x), gives in general intermediate values between BCSTP-LC (x-x) and BCSTP-HC (x-x), as expected. Using 2 or 3 shortest paths in Step 1, BCSTP-LC (2-10) and BCSTP-LC (3-10), can

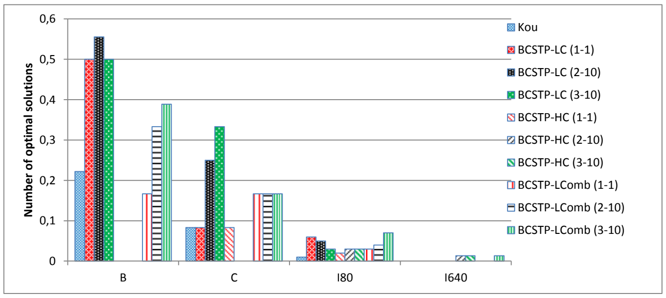

Fig. 1. Percentage of optimal solutions.

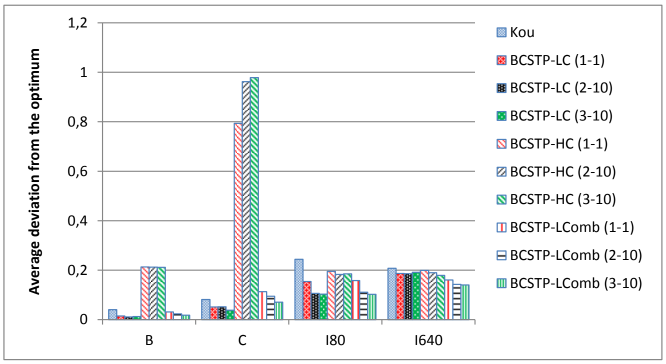

Fig. 2. Average relative deviation from the optimum (sets of networks B and C).

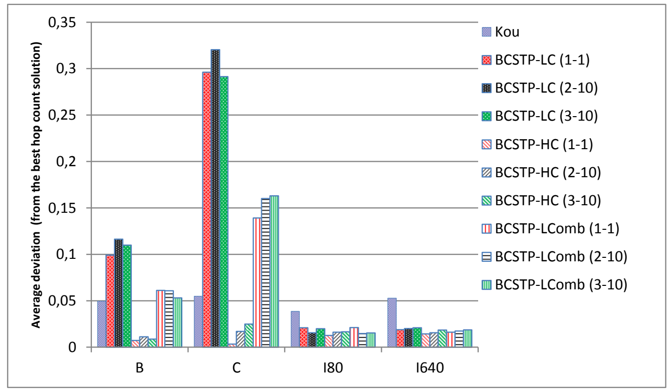

Fig. 3. Average relative deviation from the minimum obtained hop count.

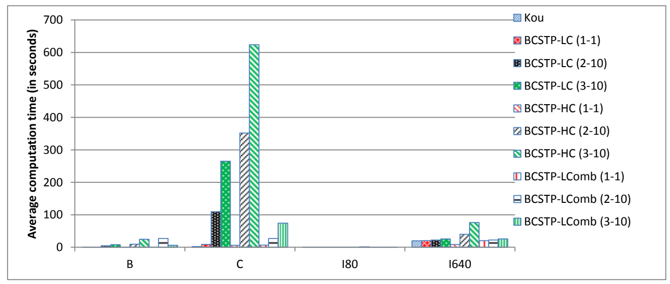

Fig. 4. Average computation time in ms (sets of networks B and I80).

give good results but the better results depend on the network test set. Another aspect is that Kou and BCSTP-LC (1-1) give very similar results for C networks but for B and I80 networks BCSTP-LC (1-1) give better results than Kou's heuristic which shows the effectiveness of the second metric proposed for Step 2 of the heuristic and the bi-criteria approach. The use of parallel arcs leads in general to better results than the ones obtained by BCSTP-LC (1-1) as can be seen in all the figures. The use of more than 3 parallel arcs in G_2 was also tested and led to better results in some networks but the improvement was not significant specially if we take into account the increase in the computational effort.

In figure 2 the average relative deviations from the optimal value, obtained by all the heuristics, are presented. The results are similar to the previous ones. Note that the worst values for all networks are obtained by BCSTP-HC (x-x) (variants that lead in average to the best hop count solutions, as can be seen in figure 3), followed by the Kou's heuristic. It can also be concluded that when we are looking for Steiner trees with low load cost, this metric should also be used in Step 1 because this leads to the best values, as expected.

In figure 3 the average relative deviations from the minimum hop count obtained by all the heuristics, are presented. Note that in this case the optimal value is not known. As already mentioned the better results were obtained by BCSTP-HC (x-x). Finally it is important to note that each BCSTP heuristic gives a (small) set of solutions but only the best one (for each case) was presented in the previous analysis.

In figure 4 the average computation time is presented for all the implemented heuristics. The main result is that in I networks the computation time is very small because there are very few paths with the same cost between the terminal nodes which leads to fewer potential solutions in Step 2 than in the cases of networks C. The other important aspect to be mentioned is that the computation time depends also on the number of terminal nodes and also depends on the number of parallel arcs in G_2 graph. It is important to note that the C networks with more than 83 nodes were not considered for this experimental study. Also, in the case of I640, the instances with more than 50 terminal nodes were not considered because the number of nodes involved in most of practical mp2mp connections should be no more than 50 terminal nodes.

V. Conclusion

In this paper a bi-criteria Steiner tree problem is formulated in order to find 'good' compromise mp2mp virtual connections in transport networks in terms of load cost and hop count. The metrics used in this formulation reflect the main aspects that each transport network service provider should take into account for an efficient bandwidth management. A heuristic to solve this NP-hard problem was presented which can give solutions that dominate the solutions obtained by Kou's heuristic in terms of load cost and hop count. The complexity of this heuristic depends mainly on the number of terminal nodes and not on the network size (which only matters for the MPS k–shortest path algorithm), which makes it suitable for network management interactive use. A more realistic experimental study is under evaluation by considering real network data.

References

[1] N. W. Group, B. Niven-Jenkins, D. Brungard, M. Betts, N. Sprecher, and S. Ueno, "RFC 5654 requirements of an mpls transport profile," September 2009.

[2] M. E. Forum, "MEF 10.1 – Ethernet Services Attributes Phase 2," November 2006.

[3] J. Clímaco and J. Craveirinha, *Multiple Criteria Decision Analysis: State of the Art Surveys*, ser. International Series in Operations Research & Management Science. New York: Springer Science, 2005, vol. 78, ch. Multicriteria Analysis in Telecommunication Planning and Design - problems and issues, pp. 899–951.

[4] 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*, D. O. Y. Shi and A. Stam, Eds. IOS Press, 2007, vol. edited in honor of Milan Zeleny, ch. 1, pp. 17–46.

[5] L. Martins, J. Craveirinha, J. Clímaco, and J. Lopes, "Network performance improvement through evaluation of bi-criteria routing methods in transport networks," *INESC – Coimbra working paper*, 2011.

[6] T. Koch, A. Martin, and S. Voß, "SteinLib: An updated library on steiner tree problems in graphs," Konrad-Zuse-Zentrum für Informationstechnik Berlin, http://elib.zib.de/steinlib, Tech. Rep. ZIB-Report 00-37, 2000.

[7] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows - Theory, Algorithms and Applications*. Prentice-Hall, Inc., 1993.

[8] P. Winter, "Steiner problem in networks: A survey," *Networks*, vol. 17, no. 2, pp. 129–167, 1987.

[9] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *Acta Informatica*, vol. 15, no. 2, pp. 141–145, June 1981.

[10] K. Mehlhorn, "A faster approximation algorithm for the steiner problem in graphs," *Information Processing Letters*, vol. 27, no. 3, pp. 125 – 128, 1988.

[11] A. Zelikovsky, "An 11/6-approximation algorithm for the network steiner problem," *Algorithmica*, vol. 9, pp. 463–470., 1993.

[12] I. R. P. Rabanal and F. Rubio, "Applying river formation dynamics to the steiner tree problem," in *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*, 2010, pp. 704 –711.

[13] M. Gendreau, J. F. Larochelle, and B. Sansó, "A tabu search heuristic for the steiner tree problem," *Networks*, vol. 34, no. 2, pp. 162–172, 1999.

[14] S. Ahn and D. H. C. Du, "A multicast tree algorithm considering maximum delay bound for real-time applications," in *Proceedings of the 21st Annual IEEE Conference on Local Computer Networks*, ser. LCN '96. Washington, DC, USA: IEEE Computer Society, 1996, pp. 172–181.

[15] J. Crichigno and B. Baran, "Multiobjective multicast routing algorithm for traffic engineering," in *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*, oct. 2004, pp. 301–306.

[16] S. Ruzika and H. W. Hamacher, "A survey on multiple objective minimum spanning tree problems," in *Algorithmics of Large and Complex Networks*, ser. Lecture Notes in Computer Science, J. Lerner, D. Wagner, and K. A. Zweig, Eds., vol. 5515. Springer, 2009, pp. 104–116.

[17] H. Hamacher and G. Ruhe, "On spanning tree problems with multiple objectives," *Annals of Operations Research*, vol. 52, pp. 209–230, 1994.

[18] S. Steiner and T. Radzik, "Computing all efficient solutions of the biobjective minimum spanning tree problem," *Comput. Oper. Res.*, vol. 35, no. 1, pp. 198–211, January 2008.

[19] N. Katoh, T. Ibaraki, and H. Mine, "An algorithm for finding k minimum spanning trees," *SIAM J. Comput.*, vol. 10, no. 2, pp. 247–255, 1981.

[20] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, "Bicriteria network design problems," *Journal of Algorithms*, vol. 28, no. 1, 1998.

[21] M. Bueno and G. Oliveira, "Analyzing the effects of neighborhood crossover in multiobjective multicast flow routing problem," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, 2010, pp. 4354 –4361.

[22] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 756 – 767, May 2002.

[23] E. Q. V. Martins, M. M. B. Pascoal, and J. L. E. Santos, "Deviation algorithms for ranking shortest paths," *International Journal of Foundations of Computer Science*, vol. 10, pp. 247–263, 1999.

[24] J. M. Coutinho-Rodrigues, J. N. Clímaco, and J. R. Current, "An interactive bi-objective shortest path approach: searching for unsupported nondominated solutions," *Computers & Operations Research*, no. 26, pp. 789–798, 1999.