

A Bi-Criteria Approach for Steiner's Tree Problems in Communication Networks

Lúcia Martins

Department of Electrical and Computer Engineering
Faculty of Sciences and Technology, University of Coimbra
Institute for Systems and Computers Engineering at Coimbra
Rua Antero de Quental, 199 3000-033 Coimbra Portugal
Email: lucia@deec.uc.pt

Nuno Gomes Ferreira

Institute for Systems and Computers Engineering at Coimbra
Rua Antero de Quental, 199 3000-033 Coimbra Portugal
Email: nfgferreira@hotmail.co.uk

Abstract—In this paper, an improved version of a previously proposed heuristic that finds ‘good’ compromise solutions for a bi-criteria Steiner trees problem is presented. This bi-criteria formulation of the Steiner’s tree problem is well suited for application in telecommunication networks whenever it is important to find the minimum amount of resources to connect a given subset of network nodes. In fact there are some (additive) metrics that may not lead to a tree with the minimum number of Steiner nodes when used in the single criterion Steiner’s tree problem. In this case it can be advantageous to consider also the minimisation of the hop count as a second criteria in the problem formulation. The performance of the new heuristic is evaluated and compared with the previous version by recurring to reference networks from a library of Steiner’s tree problems.

I. INTRODUCTION

In telecommunication networks there are optimization problems in which it is necessary to find the minimum amount of resources to connect a given subset of network nodes. The shortest network of ducts for the optical fibre cables to connect a given set of network users is one of such problems, which can be formulated as the Steiner’s tree problem in the Euclidean plane. To solve this problem it is usually necessary to introduce additional points (Steiner points) in order that the obtained network has the shortest length possible. The Steiner problem in the Euclidean plane is a well known NP-hard problem[1].

Multicast services can give rise to other kinds of network optimization problems similar to the previous one because usually it is necessary to find the minimum network, according to a given metric, to connect the subset of network nodes that must be reached by the service. This type of network optimization problem can be solved by Dijkstra’s algorithm[2] if the shortest paths tree is chosen to connect the nodes involved. However, if the minimum tree is required then the problem that must be solved is the Steiner’s tree problem in graphs which is a different version of the previously mentioned Steiner’s tree problem. In this problem the network can be represented as an undirected graph $G(N, A)$, where N is a set of $1, 2, \dots, |N|$ vertices (or nodes), and A is the set of arcs

(i, j) (or links) connecting the vertex i to the vertex j , with a cost function $C : A \rightarrow \mathbb{R}$ that assigns a cost $C_{i,j} \in \mathbb{R}$ to each arc $(i, j) \in A$. Given a subset $S \subset N$ of nodes (terminal nodes), the problem of finding the shortest tree T_S spanning all nodes in S and possibly some optional nodes in $N \setminus S$ (Steiner nodes) is known as the Steiner’s tree problem in graphs which is a NP-complete problem [1].

The Steiner’s tree problem (STP) in graphs has several other applications in telecommunication networks and depending on the metric that is associated with the cost function C , could be advantageous to formulate some of them as bi-criteria Steiner’s tree problems (BCSTP). If the cost $C_{i,j}$ of each arc (or link) (i, j) does not lead to a tree with a minimum number of Steiner nodes a second metric can be considered in order to get a minimum amount of network resources involved in the tree. This is the case when $C_{i,j}$ represents the degree of bandwidth occupation in each link as proposed, for instance, in [3]. The (single criteria) STP solution is a tree T_S which has the minimum cost C_{T_S} :

$$C_{T_S} = \sum_{i,j:(i,j) \in T_S} C_{i,j} \quad (1)$$

In this case the existence in the network of other trees with a lower number of (Steiner) nodes and with a higher value of C_{T_S} is possible. Concerning network management it could be more convenient to choose one of these trees instead of the minimum Steiner tree because the difference in cost is not big enough to justify choosing the tree with more nodes. In this case a bi-criteria formulation of the optimization problem makes it possible to obtain a set of non-dominated solutions which are potentially the best compromise solutions with respect to the two objective functions: the minimization of the original cost and the minimization of the tree hop count. A non-dominated solution (or Pareto optimal solution) is a feasible solution such that there is no other feasible solution that can improve the value of one objective function without worsening the value of at least one of the other objective functions. Note that the complexity of this problem is very high and there is no assurance that the set (or the complete set) of non-dominated solutions will be found in all cases.

In a previous work [4] a bi-criteria heuristic to find

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

multipoint-to-multipoint virtual connections in transport networks was proposed. This heuristic can give solutions that dominate the solutions obtained by the Kou et al. heuristic [5] in terms of the two metrics involved. Moreover, this heuristic can also find trees with a higher cost and a lower hop count which can be important in some transport networks management scenarios.

In this paper, an improved version of the bi-criteria heuristic, that can obtain ‘good’ compromise Steiner’s trees for the previous mentioned telecommunication networks optimization problems, is presented. A comparative performance evaluation of these heuristics using some reference networks [6], is discussed.

This paper is organized as follows. In section II an overview of the previous work is presented. In section III the improved heuristic for a bi-criteria Steiner’s tree problem is formulated and in section IV the performance evaluation of the improved heuristic is described. Finally, the conclusions are outlined in the final section.

II. PREVIOUS WORK

The Steiner’s tree problem in graphs is a combinatorial problem for which several heuristics and meta-heuristics such as in [5], [7], [8], [9], [10] have been proposed.

The particular case where all network nodes are terminal nodes is known as the minimum spanning tree (MST) problem which can be solved by polynomial-time algorithms such as Kruskal, Prim or Sollin’s algorithms [11].

In the bi-criteria formulation of the MST problem there are two costs $(C_{i,j}^1, C_{i,j}^2)$ associated with each 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 $C_T = (C_T^1, C_T^2)$ is represented as a vector-valued function, such that

$$C_T^l = \sum_{i,j:(i,j) \in T} C_{i,j}^l, \quad l = 1, 2 \quad (2)$$

The set of efficient trees $\{T\} \subset \mathcal{T}$ is such that, for every $T' \in \mathcal{T}$, $C_T^l \leq C_{T'}^l$, with $l = 1, 2$, $T \neq T'$ and $C_T^l < C_{T'}^l$ for, at least, one value of l . The number of efficient spanning trees in a graph is exponentially large in the number of vertices with a maximum number of $|N|^{|N|-2}$ efficient trees therefore the bi-criteria MST problem is intractable and NP-hard [12]. Nevertheless, the number of supported efficient spanning trees, that correspond to breakpoints of the non-dominated frontier, is polynomially bounded by $|A|^2$ for the bi-criteria case and can be efficiently computed using the weighted sum method proposed in [13]. Supported efficient spanning trees are the solutions of the multiple criteria MST problem that can be obtained through the minimization of any strictly positive weighted-sum of the objective functions. The remaining efficient trees are called unsupported [12]. If the entire set of efficient solutions of the bi-criteria MST is to be computed as in [14], then the non supported solutions may be obtained, for instance, by recurring to a k -minimum spanning

tree algorithm such as the one in [15]. Note that each MST of \mathcal{T} always has $N - 1$ arcs. For an overview of multiple objective MST problems see [12].

Concerning the bi-criteria Steiner’s tree problem there are a few heuristics and meta-heuristic proposals such as [16], [17], [18] but as the complexity of the problem is very high there is no single methodological approach that must be followed. In [19] a multicast adaptive multiple constraints routing algorithm is presented that guarantees QoS to the multicast members in an efficient manner but without leading always to a multicast tree.

A bi-criteria STP was formulated in a previous paper [4] with two additive metrics and so the tree cost is given by the cost vector $C_{T_S} = (C_{T_S}^1, C_{T_S}^2)$, such that:

$$C_{T_S}^l = \sum_{i,j:(i,j) \in T_S} C_{i,j}^l, \quad l = 1, 2 \quad (3)$$

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. For this reason we can use the hop count as a second metric in this problem whenever the first metric does not conduct by itself to a tree with a minimum number of arcs.

A. A Bi-Criteria Steiner Trees Heuristic

The previous proposed heuristic for the bi-criteria STP is based on the Kou et al. [5] heuristic for the single criterion problem. This procedure can be described through 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 $C_{i,j}$ (the cost of arc (i, j) in graph G_1) this is given by $C_{i,j} = \sum_{(k,l) \in r_{i,j}} C_{k,l}$;
- Step 2: Find the MST T_1 of G_1 , for instance with Kruskal’s or Prim’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 a set of non-dominated solutions for the bi-criteria Steiner trees problem was based on the previous heuristic 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 [13], considering the hop count as the second function to minimise. A cost pair $(C_{i,j}^1, C_{i,j}^2)$ for every arc in G_1 was then taken into account, with $C_{i,j}^2$ obtained by computing the hop count of each path $r_{i,j}$ in G . Note that Hamacher et al. algorithm does not obtain the entire set of efficient trees for the bi-criteria

MST problem and because of that it has only polynomial-time complexity $\mathcal{O}(|A_1|^2(|A_1| + |S| \log |S|))$ [13], [12].

Step 5 was also adapted to the bi-criteria case because all non-dominated solutions in G must be memorized and the dominated ones must be discarded. Note that 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 is not completely explored.

The results obtained with this strategy were very encouraging because with this heuristic it is possible to find solutions in some networks which dominate the solution obtained by the original Kou heuristic, using only the first metric. This means that with this new approach in some cases it is possible to find a solution that has lower values for C_{TS}^1 and C_{TS}^2 than the solution obtained by Kou et al. heuristic [5] because the minimum first cost solution in G_1 may not be the minimum first cost in G (see steps 4 and 5).

The heuristic proposed in [4] for the bi-criteria STP has the following improvements as related to the basic previous procedure:

- The use of parallel arcs in G_1 obtained through the k -shortest paths between each pair of nodes calculated by the MPS algorithm [20]. G_1 is then replaced by $G_2(S, A_2)$ with $|A_2| = k|S|(|S| - 1)/2$ where k represents the k -shortest paths between each pair of nodes;
- 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, in general, it includes arcs in G_2 which have in common a greater number of arcs in G hence promoting the consideration of lower cost Steiner trees also with a lower hop count.

In order to find as many non-dominated solutions as possible, three cases were considered for the computation of the G_2 graph: i) the use of cost $C_{i,j}^1$ in MPS algorithm on graph G ; ii) the use of cost $C_{i,j}^2$ in MPS algorithm in graph G ; iii) the use of a linear combination of $C_{i,j}^1$ and $C_{i,j}^2$ in MPS algorithm in graph G . Note that in this last case a bi-criteria approach for the shortest path problem [21], [22] is considered.

The new second metric used in Hamacher et al. algorithm is explained next. 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 the k paths r_{od} considered 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} . Therefore, the new second cost of each arc in G_2 , $C_{(o,d)^{r_{od}}}^2$ is given by:

$$C_{(o,d)^{r_{od}}}^2 = \frac{\sum_{(i,j) \in r_{od}} \sum_{R_{st} \subset R \setminus R_{od}} (N_{ij}^{st} / |R_{st}|)}{\text{hop count}(r_{od})} \quad (4)$$

In Step 5 the original costs, C_{TS}^1 and C_{TS}^2 (hop count), are used in order to evaluate all the obtained solutions and discard

all the dominated ones.

One parameter that had to be tuned first was the number of parallel arcs that must be considered in the heuristic. At first approximation k should be a small number such as 2 or 3 because the optimal solution for the (single criterion) STP is usually composed of paths of low cost order in the cases where the shortest path for some node pairs is not the best option. In order to consider all alternative paths with the same cost that might exist between each pair of terminal nodes 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 parallel arcs considered, which includes the alternative solutions that may exist for each path order number. This last parameter was set to 10.

In the results analysis presented in section IV the performance of this previous heuristic is compared with the new version which is presented in the next section.

III. NEW BI-CRITERIA STEINER TREE HEURISTIC

The new BCSTP heuristic is mainly based on a different way of computing the G_2 graph. In spite of the parallel arcs still being used for the construction of the graph, the following improvements can be seen:

- There are always only two parallel arcs between each pair of terminal nodes. In the previous heuristic there could be a variable number of parallel arcs between each pair of terminal nodes;
- There is a set of G_2 graphs instead of a single one, each one being obtained recurring to a special Steiner node $x \in X$, where X is an ordered set of Steiner nodes;
- For each G_2 graph the first path between each pair of terminal nodes is always the shortest path in G but the second path is obtained using a node $x \in X$ for all the second paths in the graph as follows:
 - The tree of the shortest paths between the node x and each terminal node is obtained using Dijkstra algorithm;
 - The second path between each pair of nodes $o-d$ in G is computed as the union of the paths between o and x and between x and d . Note that the union of such paths might have cycles.

Next the new version of the BCSTP heuristic is formalised.

A. BCSTP Heuristic

- 1) For each $x \in N \setminus S$ the tree of the shortest paths between node x and each terminal node is computed in G using Dijkstra algorithm and its cost $C_{T_x}^1$ (or $C_{T_x}^2$, depending on the metric used in the computation of G_2) is calculated. After ordering the Steiner nodes by the costs $C_{T_x}^1$ (or $C_{T_x}^2$), X is obtained by selecting the first K nodes;
- 2) For each $x \in X$:
 - a) Construct the complete undirected graph $G_2(S, A_2)$ with 2 parallel arcs between each pair of terminal nodes as previously described;

- b) Compute the second cost of each arc in G_2 as follows. Let N_{ij} be the number of times that the arc $(i, j) \in r_{od}$ appears in the set R . The cost of each arc in G_2 is given by:

$$C_{(o,d)r_{od}}^2 = - \frac{\sum_{(i,j) \in r_{od}} \sum_{R \setminus R_{od}} N_{ij}}{\text{hop count}(r_{od})} \quad (5)$$

- c) Find the set of efficient supported MST, $\{T_2\}$, of G_2 with the Hamacher et al. algorithm;
d) 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 ;
e) Find the set $\{T_{S'}\}$ of new MST by removing the cycles in each $G_{S'}$ in an adequate way as explained next;
f) 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.
- 3) Remove all dominated solutions from the set of all stored $\{T_S\}$.

The elimination of the cycles in this heuristic is also guided by a bi-criteria decision in order to obtain the maximum possible number of ‘non-dominated’ Steiner trees in G . In order to do that all possibilities were tested which means that when there is more than one cycle involved, every combination of arcs that can be removed in all cycles is tested.

As in the case of the previous version of the heuristic, in order to find more non-dominated solutions two cases were considered for the computation of the G_2 graph using Dijkstra algorithm on graph G : the use of cost $C_{i,j}^1$ and the use of cost $C_{i,j}^2$. The use of a linear combination of $C_{i,j}^1$ and $C_{i,j}^2$, as it gives intermediate results, was not considered for comparisons in the results analysis.

It is important to note that the set X obtained at the beginning of the heuristic makes it possible to test a small number of Steiner nodes instead of all of them, leading to similar results for almost all the networks tested with a much lower computation time, as is shown in the next section.

Concerning the complexity of this heuristic as compared with the previous one the main aspect to be considered is the complexity of the G_2 graph computation and the number of times that this graph must be computed.

The computation of the G_2 graph for the first heuristic is done by recurring to a k-shortest path algorithm (MPS [20]) but it could be done with Yen’s algorithm [23] which is the k-shortest path algorithm with the lowest worst–case complexity $\mathcal{O}(|N|^3)$. However, in [24] experimental results show that in practical situations this algorithm is more efficient than Yen’s, in terms of CPU time and RAM space. Therefore, considering the Yen’s algorithm the worst–case complexity associated with the computation of the G_2 complete graph with at most 10 parallel arcs between each node pair is $\mathcal{O}(|S|^2|N|^3)$. The computation of the G_2 graph for the second heuristic is done using only Dijkstra’s algorithm [2] which has a worst–case complexity of $\mathcal{O}(|A| + |N| \log |N|)$. For the construction of

the first G_2 graph, Dijkstra’s algorithm is called S times. For the computation of the G_2 (complete) graph with only one arc between each pair of nodes (corresponding to the shortest paths in G) Dijkstra’s algorithm needs to be called $S - 1$ times. It is necessary to call the algorithm once more to obtain the rooted tree from the Steiner node considered for all the terminal nodes S . The overall complexity for the calling of Dijkstra’s algorithm is $\mathcal{O}(|S||A| + |S||N| \log |N|)$. Next the second path between each pair of terminal nodes is simply obtained by the union of the shortest path between each terminal node and the Steiner node. This operation has a worst–case complexity of $\mathcal{O}(|S|^2|N|)$. For the second heuristic the G_2 graph needs to be computed $|N| - |S|$ times in the worst case where all the Steiner nodes are used in the heuristic but only the second paths in each G_2 graph needs to be computed because the first paths were the same for all of these graphs and because of that Dijkstra’s algorithm is called $|S| - 1 + (|N| - |S|) = |N| - 1$ times in the worst case for this heuristic.

The complexity associated with the computation of the cost $C_{(o,d)r_{od}}^2$ in both heuristics is $\mathcal{O}(|A||S|^2|N|)$. In the second heuristic these costs must be computed $|N| - |S|$ times in the worst case.

Also associated with the complexity of the second heuristic is the Hamacher et al. algorithm which is computed only once in the first heuristic and $|N| - |S|$ times in the second one.

However, as is explained in the results analysis, the $N - S$ term was replaced in the second heuristic by a fixed number of Steiner nodes (50) for all networks which eliminates this factor from the complexity analysis and leads to the lowest overall complexity for the second heuristic (if the elimination of cycles procedure considered for this heuristic is not considered). The elimination of cycles has a complexity of $\mathcal{O}\left(\left(\frac{|A|}{c}\right)^c\right)$, where c is the number of cycles. However, the elimination of cycles is rarely required and has little impact on the quality of the extremes solutions for each metric, being mainly important for the set of the non-dominated solutions obtained.

IV. RESULTS ANALYSIS

The proposed BCSTP heuristic was evaluated by using benchmark graphs from the the Steinlib Testdata Library [6].

In the following figures the aggregated results for all the networks of the sets B, C, I80 and I640 test graphs were presented for the two versions of BCSTP heuristic. In order to simplify the analysis in the case of the first version only two cases were considered: i) the case where graph G_2 is obtained with only the shortest path between each pair of nodes (designated as *BCSTP 1-1* in the figures); ii) the case where graph G_2 is obtained with the two shortest paths between each pair of nodes within a maximum of 10 paths (designated as *BCSTP 2-10* in the figures). For comparison purposes the results obtained by Kou et al. algorithm were also included.

In figure 1 for each heuristic the percentage of number of times for which the optimal solution was obtained for each set of networks is presented. As can be seen in the figure,

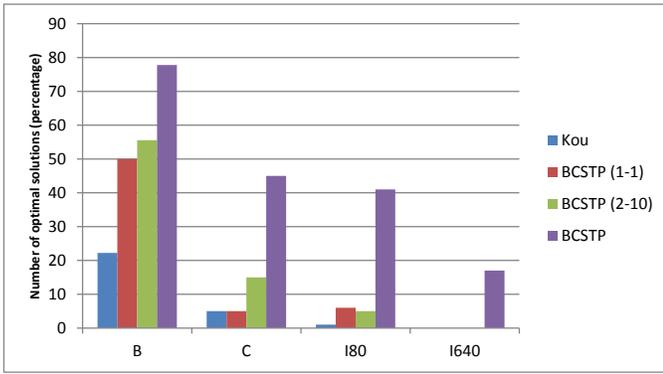


Fig. 1. Percentage of optimal solutions (G_2 obtained with the first metric).

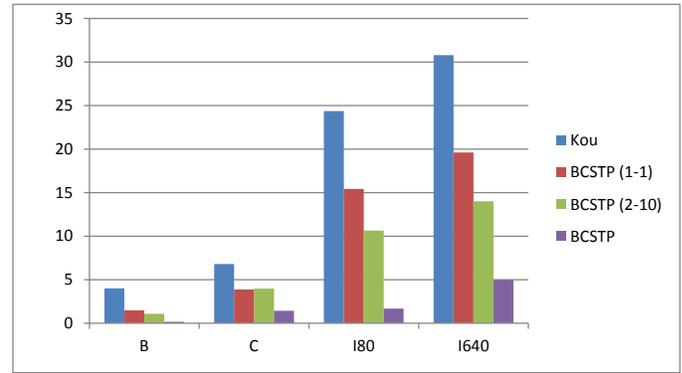


Fig. 2. Average relative deviation from the optimum (G_2 obtained with the first metric).

the optimum value was obtained with BCSTP (1-1) and with BCSTP (2-10) in 50% and 55% of the cases for B networks respectively. With the new version of the heuristic the optimum value was obtained in 78% of the cases. For C networks the results obtained with BCSTP (1-1) and with Kou's heuristic were similar and, as expected, were worse than those obtained with BCSTP (2-10). Note that there are many paths with the same cost between each pair of nodes in these networks and BCSTP (2-10) heuristic has the advantage of considering 10 parallel arcs for each pair of nodes while BCSTP (1-1) only considers one arc. For I80 and I640 networks a very small number of optimal solutions was obtained by BCSTP (1-1) and BCSTP (2-10), which shows that these networks are very difficult Steiner problems. However, the new heuristic can obtain 41% of optimum values in I80 networks and 17% in I640 networks which is a very good improvement.

In figures 2 and 3 the average and the maximum relative deviations from the optimum value obtained by each heuristic are presented. In these results the advantage of considering parallel arcs in G_2 graph is clearer than in figure 1 because the results obtained by BCSTP (2-10) are in general better than the ones obtained by BCSTP (1-1). Note that the best solutions for the first cost obtained with the new heuristic only differ from the optimum value by about 13% in the worst case of the I640 networks while BCSTP (2-10) differs by almost 35%.

It is important to note that the G_2 graph used in the BCSTP (1-1) heuristic is the same graph that is used by Kou et al. heuristic but the results obtained by BCSTP (1-1) are in general better. This means that this bi-criteria approach can be a good strategy in some (single criterion) Steiner's tree problems mainly if it also makes it possible to obtain better trees in terms of the hop count metric.

In figures 4 and 5 the average and the maximum relative deviations from the minimal hop count value obtained by all the heuristics are presented. Note that the optimum hop count value is not known. In this case, as we are looking for the best trees in terms of hop count, Kou algorithm was adapted to use this metric in Step 1 and in the MST computation. Moreover, in both BCSTP heuristics G_2 was computed with the hop count metric. As can be seen in the figures the BCSTP heuristic gives the best results for all the networks. The differences between

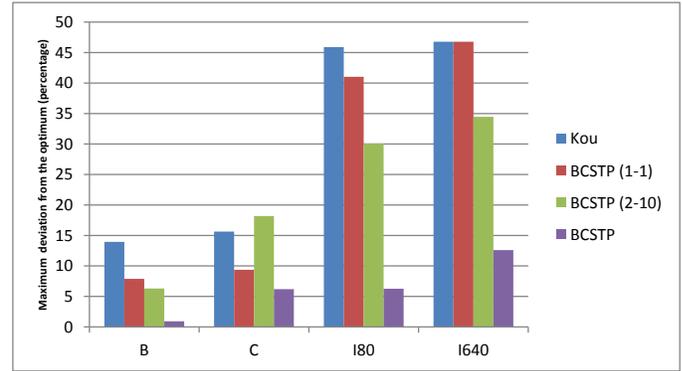
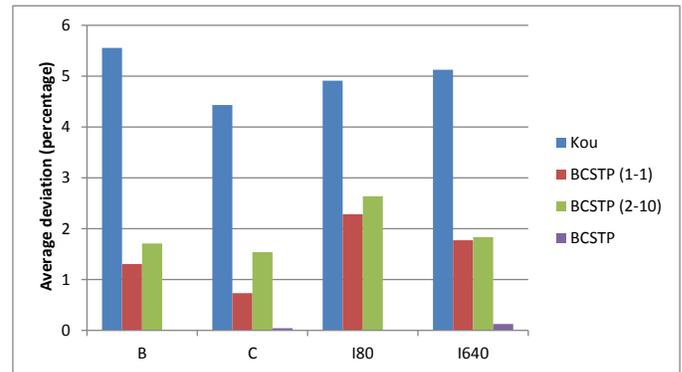


Fig. 3. Maximum relative deviation from the optimum (G_2 obtained with the first metric).

BCSTP (1-1) and BCSTP (2-10) are very small (less than 1% in figure 4) and the only case where BCSTP (2-10) gives better results than BCSTP (1-1) is for the maximum relative deviation in I640 networks (see figure 5), which are hardest Steiner problems.

Another aspect that was important to consider in the new BCSTP heuristic was the number of Steiner nodes (K) that must be included in the set X . Initially all the Steiner nodes were put into X in order to evaluate the performance of the



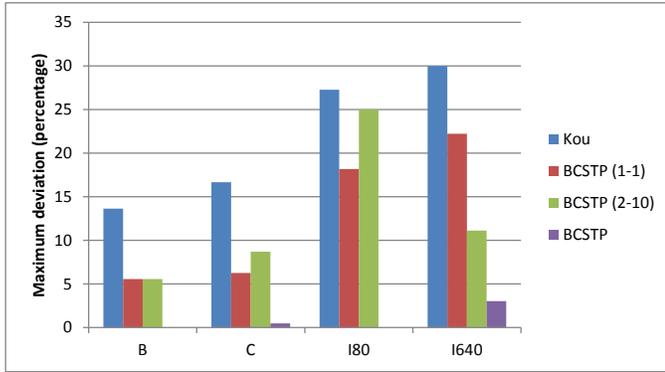


Fig. 5. Maximum relative deviation from the minimum obtained hop count (G_2 obtained with the hop count).

heuristic. These were the best results that could be obtained with this heuristic. As in this case the computation time was high, an evaluation of each node was done through the cost of the rooted tree C_{T_x} , as previously mentioned, in order to discard some of them from X . In figures 6 and 7 different numbers of Steiner nodes were considered in the set X (10, 20, 50, 100, 150, 200 and 250), presented in the figures as a percentage of the total Steiner nodes for each set of networks. The percentage of solutions which were equal to the best solutions obtained with the complete set of Steiner nodes was registered for each number of Steiner nodes in X . In figure 6 the results for the best first cost solution are presented (obtained with G_2 computed with the first metric) while the best results for the hop count are presented in figure 7 (obtained with G_2 computed with the hop count). The main conclusion that can be drawn from these results is that the solutions are consistently better as the number of Steiner nodes increases in X in both figures. This means that the ordering of these nodes increases the efficiency of the algorithm as it allows the restriction of the number of Steiner nodes involved.

Note that all the results previously presented for the BCSTP algorithm (in figures 1, 2, 3, 4 and 5) were obtained with 50 Steiner nodes in X which represents approximately 89% of the Steiner nodes for B networks, 13% for C networks, 75% for I80 networks and 9% of I640 networks. In the case of B networks all the best solutions were already obtained with 44% of the Steiner nodes for the first metric (see figure 6) and for the hop count (see figure 7). In the case of the first metric presented in figure 6 only 65% of best solutions were obtained for C networks with 50 Steiner nodes (13% of all nodes). However the obtained solutions for the remaining 35% (of C networks) have a maximum relative deviation from the respective best solution of 1.5%. This cannot be seen in figure 6 but is reflected in the previous results presented in figure 2 and 3. Moreover in the case of I640 networks 91% of the best solutions were already obtained with 50 Steiner nodes and the remaining solutions have a maximum relative deviation from the best solution of only 0.83%. Similar values were found for I80 networks and also in the case of the hop count presented in figure 7. In this case and for I640 networks 77% of the best

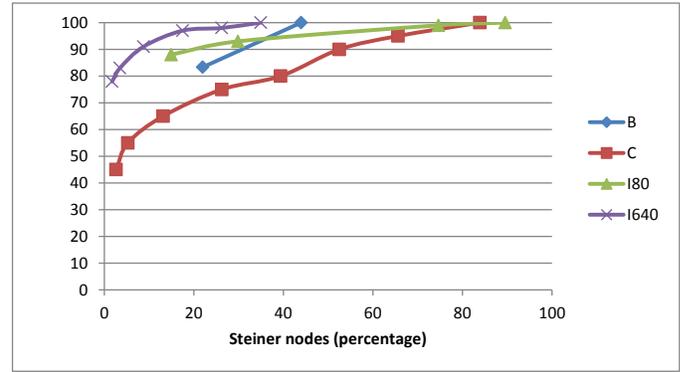


Fig. 6. Evolution of the percentage of best solutions found X (G_2 obtained with the first metric).

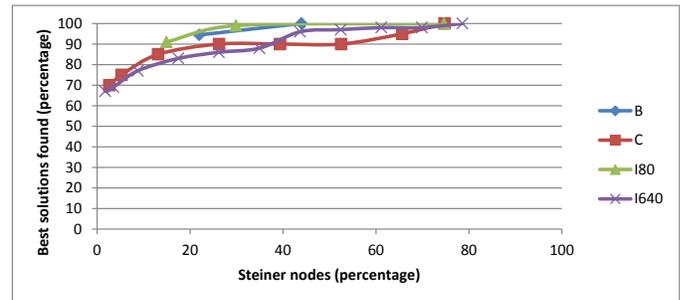


Fig. 7. Evolution of the percentage of best solutions found (G_2 obtained with the hop count).

solutions were already obtained with 50 Steiner nodes and the remaining 23% of the solutions have a maximum relative deviation from the best solution of 4.3%.

The results presented so far are the best results for each metric that can be obtained by each algorithm with G_2 graph computed accordingly with that metric. However in general a bi-criteria algorithm can give a set of non-dominated solutions instead of only one. In figures 8 and 9 the average number of 'non-dominated' solutions obtained by each algorithm for each set of networks are presented. The results for BCSTP algorithm using all Steiner nodes in X are also presented. As can be seen, the number of non-dominated solutions is very small in spite of the hundreds of trees that are computed by Hamacher et al. algorithm in some G_2 graphs. Most of these trees are discarded because they lead to dominated solutions. The solutions that are obtained in the end are the ones that dominate all the other solutions with regards to the first metric and to the hop count in graph G . The maximum number of non-dominated solutions (almost 10, in average) is obtained for I640 networks with G_2 computed with the first metric. As can be also seen with all the Steiner nodes in the BCSTP algorithm the number of non-dominated solutions does not change significantly when compared with the solutions obtained with 50 Steiner nodes.

The sets of the 'non-dominated' solutions obtained by each algorithm can be seen in more detail through the analysis of the results obtained for I80–123 network. In figures 10, 11,

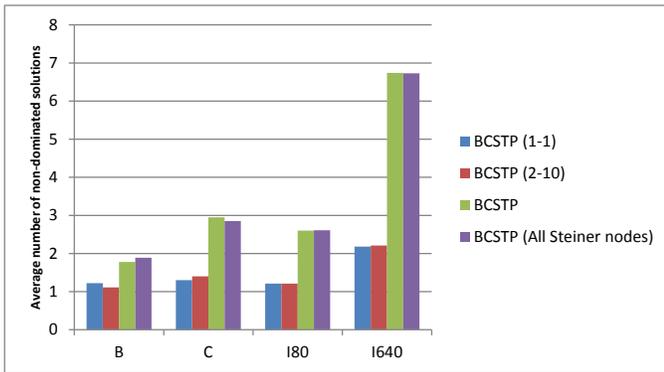
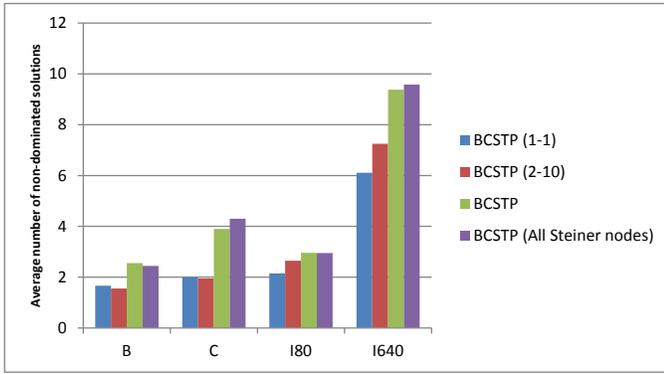


Fig. 9. Average number of non-dominated solutions (G_2 obtained with the hop count).

12 and 13 all the trees obtained are represented. In table I all the sets of non-dominated solutions obtained by each heuristic are summarised.

In table II the upper bounded values for the minimum, the average and the maximum CPU time obtained with a Intel Quad Core processor (2.33 GHz) are presented for each set of networks for the heuristics BCSTP and BCSTP (2-10). The algorithms were developed in Java because they will be integrated into a web-based telecommunications network management system. Note that for BCSTP heuristic the computation time was obtained with 50 Steiner nodes in X . An important conclusion is that the improvement of the heuristic was achieved without worsening the computational time. Note that the CPU time obtained for each heuristic was not optimized and some results were obtained in graphical

	G_2 -first metric	G_2 -hop count
BCSTP (2-10)	(1857,8) (2061,7)	(2100,7)
BCSTP	(1569,8) (2061,7)	(1569,8) (2100,7)
BCSTP (all Steiner nodes)	(1569,8) (2061,7)	(1569,8) (2100,7)

TABLE I

NON-DOMINATED SOLUTIONS OBTAINED FOR I80-123 NETWORK.

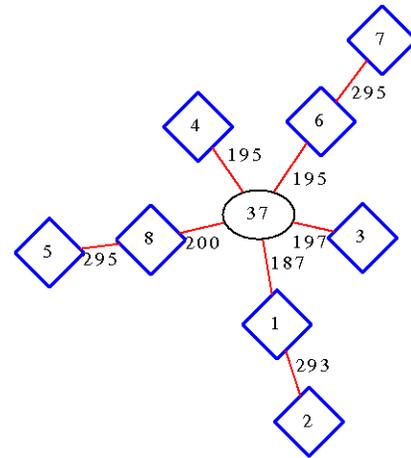


Fig. 10. Network I123 – BCSTP (2-10) minimum first metric solution (1857, 8).

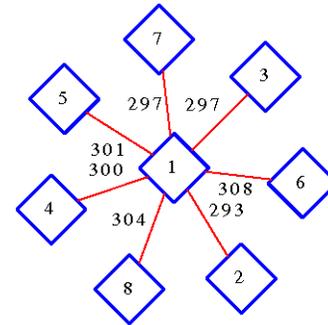


Fig. 11. Network I123 – BCSTP (2-10) minimum hop count solution (2100, 7).

mode.

V. CONCLUSION

In this paper a bi-criteria Steiner's trees problem is formulated that makes it possible to obtain 'good' compromise solutions in terms of the two metrics involved. The first metric is an additive metric for which the optimal tree, of the single criterion Steiner's tree problem, may be a tree with a number

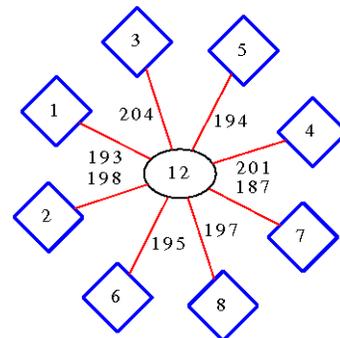


Fig. 12. Network I133 – Steinlib optimal solution and BCSTP minimum first metric solution (1569, 8).

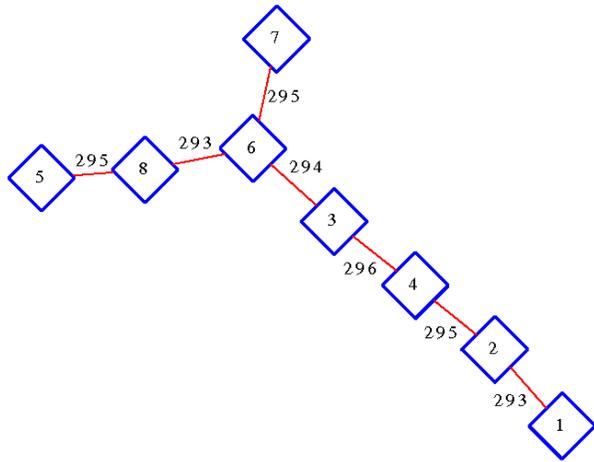


Fig. 13. Network I123 – BCSTP minimum hop count solution (2061, 7).

	Min.	Average	Max.
	BCSTP		
B	0.2	5	25
C	0.5	1874	32865
I80	0.1	0.5	3
I640	0.8	258	2295
	BCSTP (2-10)		
B	0.1	8	65
C	0.1	14260	99490
I80	0.01	0.5	8
I640	0.1	2258	22584

TABLE II
COMPUTATION TIME (IN SECONDS).

of Steiner nodes greater than the minimum possible. The second metric that can be added in this case is the hop count. This bi-criteria formulation of the Steiner's tree problem is well suited for application in telecommunication networks whenever it is important to find the minimum amount of resources to connect a given subset of network nodes. This is a very complex problem for which a heuristic resolution is proposed. The performance of this heuristic, which is an improved version of a previously proposed heuristic, was evaluated by recurring to reference networks from a library of Steiner's tree problems. The results show that this heuristic can find the optimal solution for the single criterion STP in some very complex Steiner's tree problems. Moreover, it can also find other solutions with a higher cost but a lower hop count that can be more advantageous than the optimal single criterion solution in some practical communication network conditions.

ACKNOWLEDGMENT

The authors would like to thank Professor Teresa Gomes for her implementation of MPS algorithm and also for fruitful discussions that allowed the improvement of the paper.

REFERENCES

- [1] M. R. Garey and D. S. Johnson, *Computers and Intractability – A Guide to the Theory of NP-Completeness*, V. Klee, Ed. Bell Telephone Laboratories, Incorporated, 1979.
- [2] E. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [3] 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.
- [4] L. Martins, N. Ferreira, and J. Craveirinha, "A bi-criteria algorithm for multipoint-to-multipoint virtual connections in transport networks," in *CNSM 2011, 7th International Conference on Network and Service Management*, Accepted to be published as short paper 2011.
- [5] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *Acta Informatica*, vol. 15, no. 2, pp. 141–145, June 1981.
- [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] K. Mehlhorn, "A faster approximation algorithm for the steiner problem in graphs," *Information Processing Letters*, vol. 27, no. 3, pp. 125 – 128, 1988.
- [8] A. Zelikovskiy, "An 11/6-approximation algorithm for the network steiner problem," *Algorithmica*, vol. 9, pp. 463–470., 1993.
- [9] 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.
- [10] H. Esbensen, "Computing near-optimal solutions to the steiner problem in a graph using a genetic algorithm," *Networks*, vol. 26, pp. 173–185, 1995.
- [11] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows - Theory, Algorithms and Applications*. Prentice-Hall, Inc., 1993.
- [12] 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.
- [13] H. Hamacher and G. Ruhe, "On spanning tree problems with multiple objectives," *Annals of Operations Research*, vol. 52, pp. 209–230, 1994.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] F. Kuipers and P. V. Mieghem, "MAMCRA: a constrained-based multicast routing algorithm," *Computer Communications*, vol. 25, pp. 801–810, 2002.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, July 1971.
- [24] E. Q. Martins, M. Pascoal, and J. Santos, "An algorithm for ranking loopless paths," CISUC, <http://www.mat.uc.pt/marta/Publicacoes/mps2.pdf>, Tech. Rep. 99/007, 1999.