Acta Math. Univ. Comenianae
Vol. LX, 1(1991), pp. 105–121

THREE NEW HEURISTICS FOR THE STEINER PROBLEM IN GRAPHS

M. DIANÉ AND J. PLESNÍK

Abstract. Three practically successful heuristics are presented and analysed. They include the known spanning tree heuristic (STH). One of the heuristics chooses Steiner vertices by STH and the other two heuristics according to their sum of all distances to the special vertices. The theoretical worst-case performance ratio remains the same as for STH.

Given a connected graph $G = (V, E)$ (undirected, without loops and multiple edges) with positive edge costs (called also lengths) and a set $Z \subset V$ of special (distinguished) vertices, the Steiner problem on graphs (networks) asks for a minimum cost tree within G that spans all members of Z. If $|Z| = 2$ we have the shortest path problem and if $Z = V$ we get the minimum spanning tree problem, which are well known problems solvable in polynomial time. The same is true for any fixed cardinality $p := |Z|$. However, in general, the Steiner problem is NP-hard. Nevertheless, a tree that is not more than $2 - 2/p$ times as expensive as an optimal tree can be computed in polynomial time. On the other hand no polynomial time approximation algorithm is known to have worst-case performance that is bounded by $2 - \varepsilon$ times the cost of an optimal tree, for $\varepsilon > 0$. The Steiner problem has an extensive literature and numerous applications, such as the design of integrated circuits and telephone networks. For good surveys on the Steiner problem see Hwang and Richards [**3**] and Winter [**13**].

Many exact and approximation methods have been developed for this NP-hard problem. There are also several graph polynomial time heuristics for the Steiner problem [**13,3**] and it is the purpose of this paper to present three new such heuristics which practically compare favorably to several known ones (including the spanning tree heuristic [**1,6,9**], the path heuristic [**12**] and the average distance heuristic [**10,11**]) and have the same theoretical worst-case performance. First, in Section 1 we give a heuristic based on the spanning tree heuristic. In Section 2 we present a heuristic which chooses vertices for a tree according to their sum of all distances to the special vertices. Since a vertex with the minimum sum is often called a median vertex, our heuristic is said to be median. The third heuristic deletes vertices with largest sum of distances and is called an antimedian heuristic. It is presented in Section 3.

Received February 6, 1990; revised November 7, 1990.
1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 68E10.
Key words and phrases. Steiner tree, approximation algorithm, worst-case analysis.

We now introduce some notation and terminology. Put $n := |V|$, $m := |E|$ and $p := |Z|$. A Z-vertex is a vertex of Z. The cost of an edge $e = ij$ is denoted by $c(e)$ or c_{ij}. The cost of a subgraph is the sum of all its edge costs. The distance of two vertices u and v is denoted by $d(u, v)$ where the lengths of edges are equal to their costs. (For standard graph algorithms see e.g. Lawler [**7**] or Even [**2**].) Given a subset $B \subset V$, by $K(B)$ we denote the complete graph on B where the cost of an edge ij is equal to the distance $d(i, j)$ measured in G. Further let $T(B)$ denote any minimum cost spanning tree of $K(B)$.

Given a Steiner tree T, i.e. a tree spanning Z, we can sometimes prune it and thus obtain a better Steiner tree. By pruning T we mean deleting all leaves of T (i.e. vertices of degree one) which are not Z-vertices (one at a time). In the sequel we use the following known spanning tree heuristic (STH) for the Steiner problem [**6**]: Find a minimum cost spanning tree T of $K(Z)$, then replace each its edge by a corresponding shortest path in G, then in the resulting graph G' find a minimum cost spanning tree T' and finally prune T'. The resulting tree is the output.

For each our heuristic H we shall prove:

Theorem A. *For any instance of the Steiner problem we have*

$$c(T_H) \leq (2 - 2/p)c^*,$$

where T_H is a tree produced by H and c^* is the cost of an optimal solution. Moreover, for any real $\delta > 0$ there is an instance such that

$$c(T_H) > (2 - \delta)c^*.$$

In all our heuristics we have included STH, hence $c(T_H) \leq c(T(Z))$. Thus the first inequality of Theorem A is implied by a result on STH [**1,6**] and the crucial problem is to find a bad example for proving the latter inequality.

1. Multiple spanning tree heuristic (MSTH).

Recall that any nonspecial vertex of a Steiner tree is called a Steiner vertex. It is well known (see e.g. [**4**]) that the number of Steiner vertices in a Steiner minimum tree does not exceed $p - 2$ whenever G is a complete graph whose edge costs form a metric (p is the number of special vertices).

In the following heuristic, we select at most $p - 2$ candidates for the Steiner vertices. More precisely, as there are at most $n - p$ nonspecial vertices, we select $q := \min\{p - 2, n - p\}$ candidates. In detail, this works as follows.

Step 1: Find a minimum spanning tree $T(Z)$.

Step 2: For every $v \in V - Z$ construct a minimum spanning tree $T(Z \cup \{v\})$. Select $q := \min\{p-2, n-p\}$ cheapest trees and set them in order according to the costs. Thus under a proper labelling we have: $c(T(Z \cup \{v_1\})) \leq c(T(Z \cup \{v_2\})) \leq \cdots \leq c(T(Z \cup \{v_q\}))$.

Step 3: Construct minimum spanning trees $T(Z \cup \{v_1, v_2\}), \ldots, T(Z \cup \{v_1, v_2, \ldots, v_q\})$. Then, among the trees

$T(Z), T(Z \cup \{v_1\}), T(Z \cup \{v_1, v_2\}), \ldots, T(Z \cup \{v_1, v_2, \ldots, v_q\})$, choose one of the smallest cost and denote it by $\hat{T}$.

Step 4: Construct a subgraph H of G by replacing each edge in $\hat{T}$ by the corresponding shortest path in G (ties are broken arbitrarily).

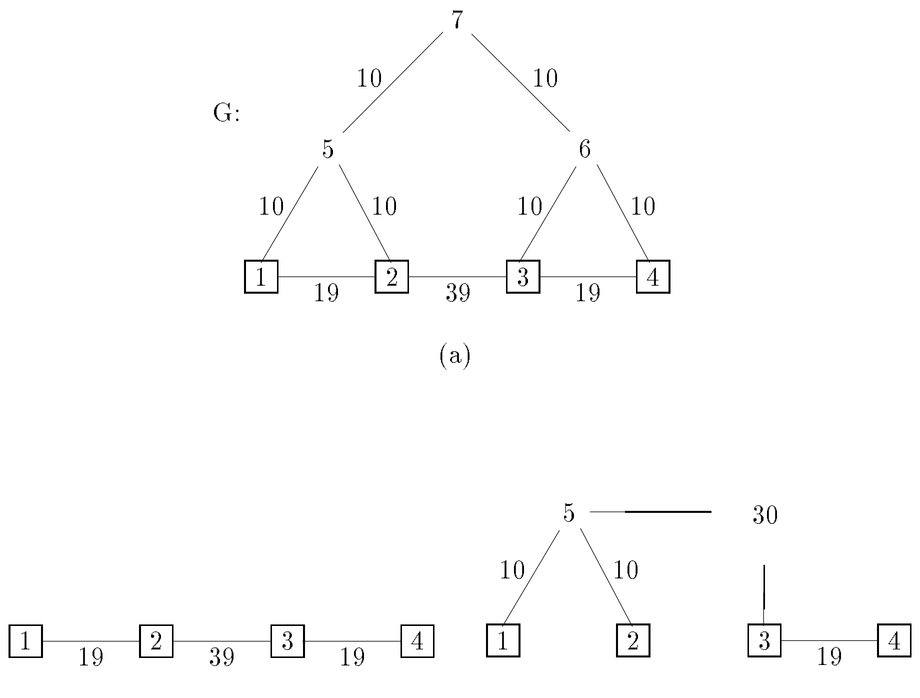
Step 5: Determine a minimum spanning tree T' of H.

Step 6: Prune T' by deleting all Steiner vertices of degree one (one at a time). The resulting tree, denoted by T_{MSTH}, is the solution. STOP.

The algorithm is explained by the following example.

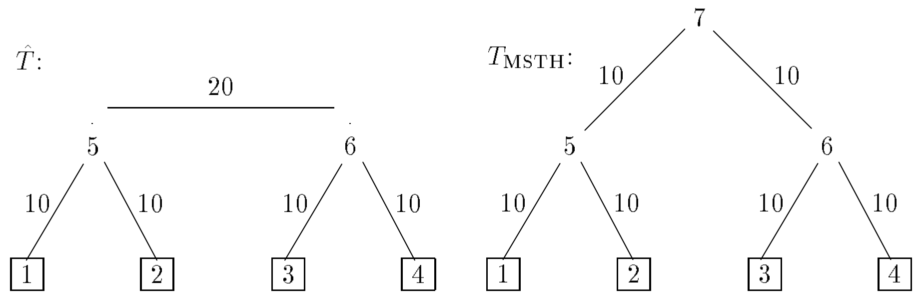
Example 1. Let G be the graph in Fig. 1(a) with edge costs as labelled, and $Z = \{1, 2, 3, 4\}$ (Z-vertices are depicted by squares).

Figure 1



(a)

(b) (c)



(d) (e)

Fig. 1(b) shows $T(Z)$ obtained in Step 1. Its cost $c(T(Z)) = 77$. In accordance with Step 2 we obtain 3 trees with costs: $c(T(Z \cup \{5\})) = c(T(Z \cup \{6\})) = 69$ and $c(T(Z \cup \{7\})) = 78$. Since $q = \min\{p-2, n-p\} = 2$, we shall consider only the first two of them. In Step 3 we deal with $T(Z)$ (see Fig. 1(b)), $T(Z \cup \{5\})$ (see Fig. 1(c)) and $T(Z \cup \{5, 6\})$ (see Fig. 1(d)). Thus $\hat{T} = T(Z \cup \{5, 6\})$, of cost 60. Further we see that $H = T' = T_{\mathrm{MSTH}}$ (cf. Steps 4 to 6) which is the tree from Fig. 1(e). Even in this case T_{MSTH} is an optimal solution.

To estimate the complexity of our heuristic we may suppose that at first $K(V)$, or equivalently, the distance matrix is computed. This can be done in $O(n^3)$ time. Using Prim's algorithm of complexity $O(n^2)$ we can find the required spanning trees in Step 1, 2, 3 and 5 in total time $[1 + (n-p) + (q-1) + 1]\, O(n^2)$, i.e. in $O(n^3)$ time. Since all other operations can be done in a smaller amount of time, the total time complexity of MSTH is $O(n^3)$.

Note, however, that the spanning tree heuristic of Kou, Markowski and Berman [**6**] can be implemented to run in $O(m + n \log n)$ time (see Kou and Makki [**5**] and Melhorn [**8**]). This means that our heuristic MSTH can be implemennted to run in $O(mn + n^2 \log n)$ time.

We have seen (Example 1) that MSTH can give an optimal solution. Now we are going to show that there are examples of the Steiner problem where MSTH gives a weak approximation. More precisely, we show that $c(T_{\mathrm{MSTH}})/c^*$ can tend to 2.

Let us consider the construction illustrated in Fig. 2, where "the upper part" of G is a binary tree of depth $k = 2t + 1$ with $t \geq 1$. Each of its $2^{k+1} - 2$ edges is of cost 1. Its leaves form the set Z with $p = |Z| = 2^k$.

In accordance with Fig. 2, the root of the binary tree forms a level set W^0, its two sons form a level set W^1, etc. Finally $W^k = Z$. Further with each vertex $v \in W^{t+1}$ the subtree lying under it is associated and called an envelope. Each envelope is a binary tree of depth t and contains 2^t special vertices. In each envelope we choose one Z-vertex (in Fig. 2 the leftmost vertex in an envelope) and

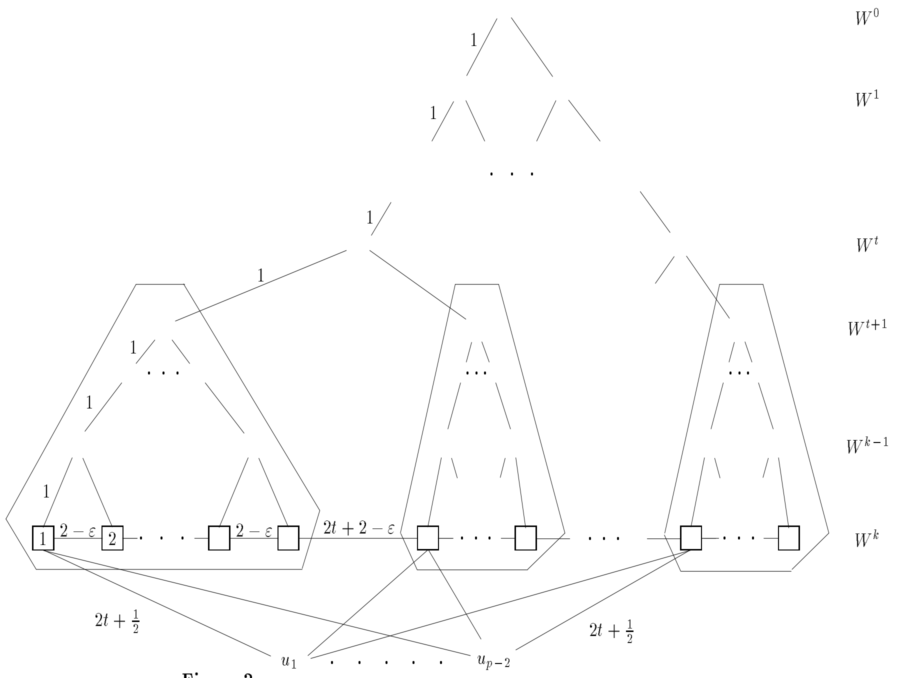
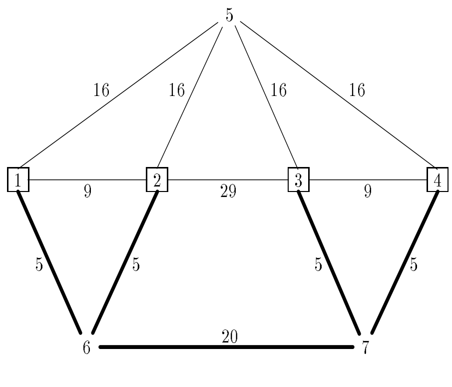

W^0
W^1
W^t
W^{t+1}
W^{k-1}
W^k
$2-\varepsilon$
$2t+2-\varepsilon$
$2t+\frac{1}{2}$
u_1
u_{p-2}

Figure 2.


5
16
16
16
16
9
29
9
5
5
5
5
20
6
7

joint it to vertices $u_1, \dots, u_{p-2}$ by edges of cost $2t + 1/2$ each. This is "the lower part" of G. Finally we add $2^k - 1$ edges forming a path P on set Z and joining the first Z-vertex and the last Z-vertex. The cost of such an edge $(i, i+1)$ is equal to the distance between i and $i+1$ in the binary tree minus ε, where $\varepsilon > 0$ is sufficiently small. If a vertex v belongs to W^{k-j} we say that v is of altitude j.

Computing tree $T(Z)$ by Kruskal's algorithm we include 2^{k-i} edges of cost $2i - \varepsilon$ for each i with $1 \le i \le k-1$ and one edge of cost $2k-1$. Note that edge $(p/2, p/2+1)$ of cost $2k - \varepsilon$ is not used in $T(Z)$ because there is an edge of cost $2k-1$. Thus

$$c(T(Z)) = \sum_{i=1}^{k-1} (2i - \varepsilon) 2^{k-i} + 2k - 1. \tag{1}$$

Now we consider trees $T(Z \cup \{v\})$ where $v \in W^{k-j}$, $1 \le j \le k$. Instead of calculating their costs we are going to give the cost differences against the path P. Clearly,

$$c(P) = c(T(Z)) + 1 - \varepsilon. \tag{2}$$

Let $S(i,j)$ denote the number of edges of cost $2i - \varepsilon$ under a vertex of altitude j in the binary tree. Let $v \in W^{k-j}$ $(1 \le j \le k)$. Then instead of an edge with cost $2i - \varepsilon$ taken for P we use in $T(Z \cup \{v\})$ an edge of $K(V)$ with cost:

a) j if $2i - \varepsilon > j$. There are $S(i,j) = 2^{j-i}$ such edges.
b) $2r + j$ if $2i - \varepsilon > 2r + j$. There are $\lceil S(i, r+j)/2 \rceil = 2^{r+j-i-1}$ such edges for every r with $1 \le r \le k - j$.

Thus (by a)) the edges under v which are used in P but not in $T(Z \cup \{v\})$ have total cost

$$c_1(j) :\, = \sum_{\substack{i=1 \\ 2i-\varepsilon > j}}^{j} (2i - \varepsilon) 2^{j-i} = \sum_{i=\lfloor j/2 \rfloor + 1}^{j} (2i - \varepsilon) 2^{j-i} \tag{3}$$

Instead of these edges we have in $T(Z \cup \{v\})$ other edges of total cost

$$c_2(j) :\, = j + \sum_{\substack{i=1 \\ 2i-\varepsilon > j}}^{j} j 2^{j-i} = j \left(1 + \sum_{i=\lfloor j/2 \rfloor + 1}^{j} 2^{j-i} \right) \tag{4}$$

Using b) we see that those edges which are not under v and belong to P but do not belong to $T(Z \cup \{v\})$ have total cost

$$
\begin{aligned}
c_1'(j)\colon &= \sum_{r=1}^{k-j} \left[2(r+j) - \varepsilon + \sum_{\substack{i=1 \\ 2i-\varepsilon > 2r+j}}^{r+j-1} (2i-\varepsilon) 2^{r+j-i-1} \right] \\
&= 2 \sum_{r=1}^{k-j} \left[r + j + \sum_{i=r+\lfloor j/2 \rfloor+1}^{r+j-1} i 2^{r+j-i-1} \right] \\
&\quad - \varepsilon \left[\sum_{r=1}^{k-j} \left(1 + \sum_{i=r+\lfloor j/2 \rfloor+1}^{r+j-1} 2^{r+j-i-1} \right) \right].
\end{aligned} \tag{5}
$$

Instead of these edges $T(Z \cup \{v\})$ uses other edges of total cost

$$
\begin{aligned}
c'_2(j)\colon &= \sum_{r=1}^{k-j} (2r+j) \left[1 + \sum_{\substack{i=1 \\ 2i-\varepsilon > 2r+j}}^{r+j-1} 2^{r+j-i-1} \right] \\
&= \sum_{r=1}^{k-j} (2r+j) \left[1 + \sum_{i=r+\lfloor j/2 \rfloor+1}^{r+j-1} 2^{r+j-i-1} \right].
\end{aligned} \tag{6}
$$

Since

$$
c(P) - c(T(Z \cup \{v\})) = c_1(j) - c_2(j) + c_1'(j) - c_2'(j) \tag{7}
$$

estimating the right hand side from above we obtain a lower bound on $c(T(Z \cup \{v\}))$. Before calculating the bound, we handle the trees $T(Z \cup \{u_j\})$, $j = 1, 2, \ldots, p-2$. One can easily verify that every such a tree contains 2^{t-i} edges of cost $2i - \varepsilon$ each, $i = 1, 2, \ldots, t$ in every envelope. There are 2^{t+1} envelopes and one (the leftmost) vertex in each enevelope is joined to vertex u_j by an edge of cost $2t + 1/2$. Hence

$$
\begin{aligned}
c(T(Z \cup \{u_j\})) &= \left[\sum_{i=1}^{t} (2i-\varepsilon) 2^{t-i} + 2t + \frac{1}{2} \right] 2^{t+1} \\
&= \left[4 \sum_{i=1}^{t} i 2^{t-i} + 4t + 1 \right] 2^t - \varepsilon \sum_{i=1}^{t} 2^{2t+1-i}.
\end{aligned} \tag{8}
$$

The following lemma is well known in difference calculus and the reader may prove it by the induction.

Lemma 1. *For any integers a, b with $0 \le a \le b$ we have*

$$
\sum_{i=a}^{b} 2^i = 2^{b+1} - 2^a, \quad \sum_{i=a}^{b} 2^{-i} = \frac{1}{2^{a-1}} - \frac{1}{2^b},
$$

$$
\sum_{i=a}^{b} i 2^i = (b-1) 2^{b+1} - (a-2) 2^a,
$$

and

$$\sum_{i=a}^{b} i2^{-i} = \frac{a+1}{2^{a-1}} - \frac{b+2}{2^b}.$$

Lemma 2. *For every j, $1 \le j \le k$, and very small $\varepsilon > 0$ we have*

$$(c_1(j) - c_2(j)) + (c_1'(j) - c_2'(j)) < 7 \cdot 2^t - 4t - 7 - \varepsilon(2^{t+1} - 2).$$

Proof. Using Lemma 1 in (3) to (6) our task reduces to prove that

$$(2\lfloor j/2 \rfloor - j + 4)(2 + k - j)2^{j - \lfloor j/2 \rfloor - 1} - 2(k+2) - \varepsilon\left[2^{j-\lfloor j/2 \rfloor - 1}(2 + k - j) - 1\right] < 7 \cdot 2^t - 4t - 7 - \varepsilon\left[2^{t+1} - 2\right],$$

where $k = 2t + 1$.

It is sufficient to prove that

$$(2\lfloor j/2 \rfloor - j + 4)(2 + k - j)2^{j - \lfloor j/2 \rfloor - 1} - 2(k+2) < 7 \cdot 2^t - 4t - 7$$

which is equivalent to

$$(2\lfloor j/2 \rfloor - j + 4)(2 + k - j)2^{j - \lfloor j/2 \rfloor - 1} < 7 \cdot 2^t - 1 \tag{9}$$

To prove (9) we distinguish two cases.
Case 1: j is even. Then $\lfloor j/2 \rfloor = j/2$ and (9) reduces to

$$2(2t + 3 - j)2^{j/2} < 7 \cdot 2^t - 1. \tag{10}$$

Introducing the function

$$f(j) \colon = (2t - j + 3)2^{j/2}$$

we see that its derivative

$$\begin{aligned} f'(j) &= -2^{j/2} + (2t - j + 3)2^{j/2}\log_e 2 \\ &= \left[(2t - j + 3)\log_e 2 - 1\right]2^{j/2} > 0 \end{aligned}$$

whenever $1 \le j \le 2t$. Hence $f(j)$ is an increasing function and it suffices to verify (10) for $j = 2t$, which is easy.
Case 2: j is odd. Then $\lfloor j/2 \rfloor = (j-1)/2$ and (9) reduces to

$$3(2t + 3 - j)2^{(j-1)/2} < 7 \cdot 2^t - 1. \tag{11}$$

Since f and thus also the left hand side in (11) is increasing whenever $1 \le j \le 2t$, it suffices to verify (11) for $j = 2t - 1$ and $j = 2t + 1$, which is immediate. $\square$

Lemma 3. *For any j with $1 \le j \le k$ and any vertex $v \in W^{k-j}$ we have*

$$c(T(Z \cup \{v\})) > c(T(Z)) - 7 \cdot 2^t + 4t + 8 - \varepsilon(2^{t+1} - 1).$$

Proof. By (2), (7) and Lemma 1. □

Lemma 4. *For any j with $1 \le j \le p-2$ we have*

$$c(T(Z \cup \{u_j\})) = 8 \cdot 2^{2t} - 7 \cdot 2^t - \varepsilon(2^{2t+1} - 2^{t+1}).$$

Proof. Immediately by (8) and Lemma 1. □

Lemma 5.

$$c(T(Z)) = 8 \cdot 2^{2t} - 4t - 7 - \varepsilon(2^{2t+1} - 2).$$

Proof. By (1) and Lemma 1. □

Lemma 6. *For any vertex $v \in V - Z - \{u_1, \ldots, u_{p-2}\}$ and u_j with $1 \le j \le p-2$ we have*

$$c(T(Z \cup \{u_j\})) < c(T(Z))$$

and

$$c(T(Z \cup \{u_j\})) < c(T(Z \cup \{v\})).$$

Proof. The first inequality is implied by Lemmas 4 and 5. The second follows from Lemmas 3 and 4. □

Now we see that in Step 3 of MSTH we choose $\hat{T}$ as a cheapest tree among trees $T(Z), T(Z \cup \{u_1\}), T(Z \cup \{u_1, u_2\}), \ldots, T(Z \cup \{u_1, u_2, \ldots, u_{p-2}\})$. It is a matter of routine to verify that $\hat{T} = T(Z \cup \{u_1\})$ and that $T_{\mathrm{MSTH}} = \hat{T}$. Hence we have

Theorem 1. *For every real $\varepsilon > 0$ and integer $t \ge 1$ there exists an instance of the Steiner problem such that an optimal solution T^* is of cost*

$$c(T^*) \le 2^{2t+2} - 2$$

whereas heuristic MSTH produces a tree T_{MSTH} of cost

$$c(T_{\mathrm{MSTH}}) = 2^{2t+3} - 7 \cdot 2^t - \varepsilon \left[2^{2t+1} - 2^{t+1}\right].$$

As a consequence we see that Theorem A holds for MSTH.

2. The median heuristic (MH).

One can observe that a minimum cost tree spanning Z often involves a vertex with minimum sum of distances from the vertices of Z, i.e., a median vertex. The following heuristic finds a median vertex among nonspecial vertices only, whereas a search among special vertices is substituted by STH [**6**]. This ensures a good

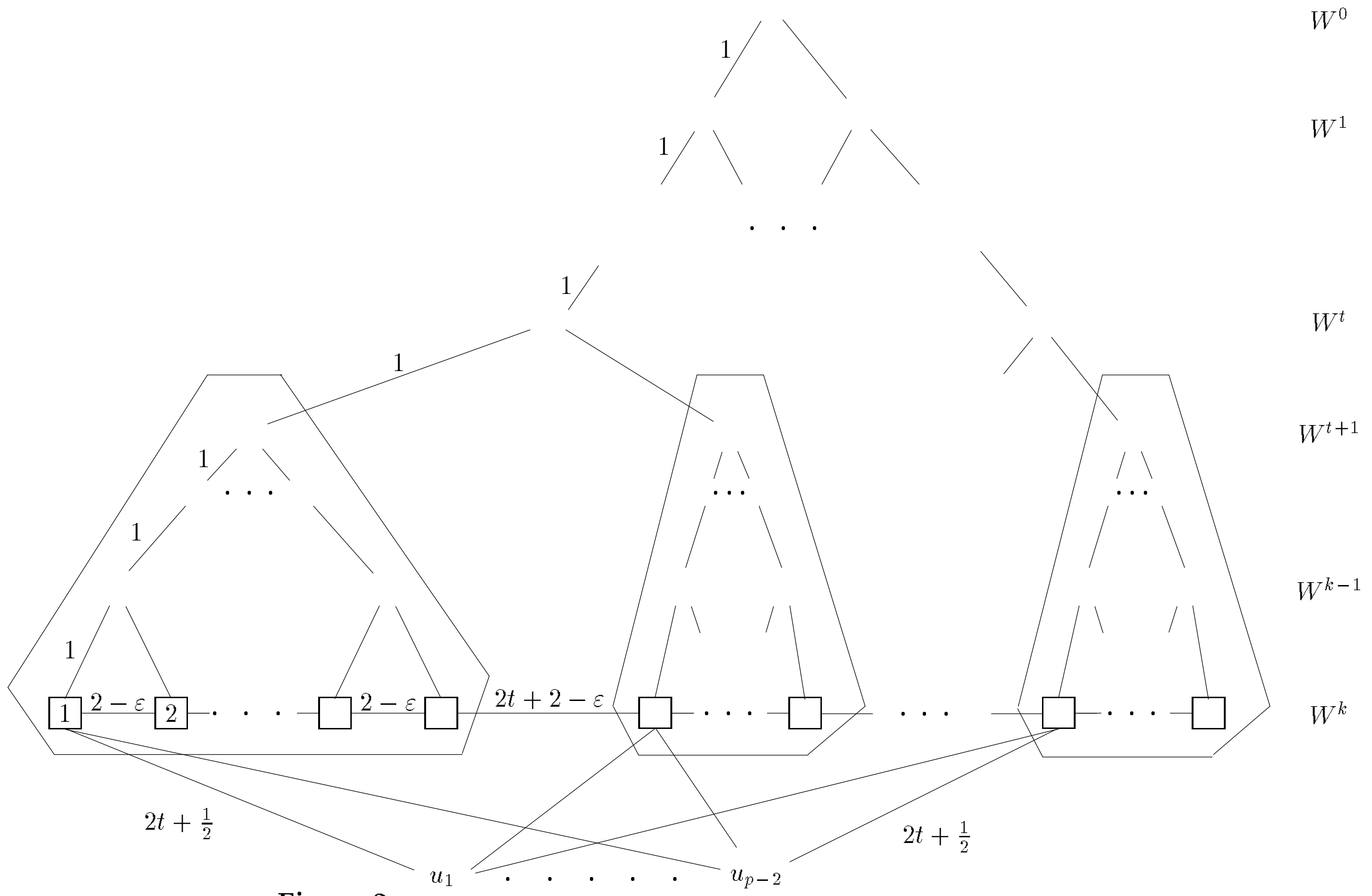


Figure 2.

practical and theoretical performance and consumes only small amount of time. In detail, the heuristic works as follows.

Step 1: For each vertex $u \in V - Z$ calculate

$$\mu(u) := \sum_{v \in Z} d(u, v) \tag{12}$$

(by Dijkstra's algorithm) and then find a vertex $u^0 \in V - Z$ which minimizes $\mu(u)$. Let T^0 be a tree consisting of shortest $u^0 - v$ paths, $v \in Z$ (provided by Dijkstra's algorithm).

Step 2: Determine a minimum cost spanning tree in the induced subgraph $G(V(T^0))$ and prune it. Denote the resulting tree by T^1.

Step 3: Determine a tree T^2 spanning Z in G by the spanning tree heuristic.

Step 4: The cheapest of trees T^1 and T^2 is the solution T_{MH}. STOP.

To illustrate MH we Consider the following instance of the Steiner problem.

Example 2. Let G be the graph in Fig 3 with the edge costs as shown and with 4 special vertices depicted as squares.

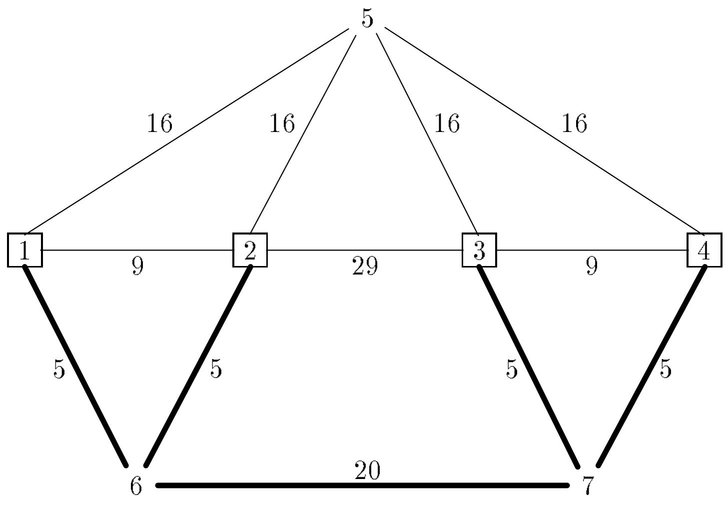


Figure 3.

One sees that $\mu(5) = 64$ and $\mu(6) = \mu(7) = 60$. Thus we can take $u^0 = 6$ and we find T^1 in $G(\{1, 2, 3, 4, 6, 7\})$. Clearly, T^1 is the tree depicted by heavy lines in Fig. 3 and $c(T^1) = 40$. Step 3 results in tree T^2 of cost $9 + 29 + 9 = 47 > 40$. Thus $T_{\mathrm{MH}} = T^1$ which is also an optimal solution.

Let us estimate the time complexity of MH. Step 1 can be done in $O((n-p)n^2)$ time ($n-p$ applications of Dijkstra's algorithm). Using Prim's agorithm to obtain T^1 in Step 2 we see that Step 2 requires no more than $O(n^2)$ time. The original implementation of Step 3 ran in $O(pn^2)$ time [**6**], but there are also ones of $O(m+n\log n)$ time [**5,8**]. Thus the overall worst-case time complexity of MH is $O(n^3)$.

Now we are going to present a bad example for MH. Let us consider a binary tree B of depth $k=2t+1$ (see "the upper part" of Fig. 4), where each edge is of cost 1. Let the levels $W^i \quad (i=0,1,\ldots,k)$ denote the sets of vertices at distance i from the root. The tree has 2^k leaves and we define them to be the special vertices. Further there are also edges forming a path P on vertices W^k. Each edge $(j,j+1)$ of P has cost equal to the distance between j and $j+1$ in the binary tree minus ε, where $\varepsilon>0$ is sufficiently small. Finally our graph G has a vertex u^* which is joined to every special vertex by an edge of cost $k-\varepsilon$. We say that the vertices v of the binary subtree of B with root u lie under u $(v \neq u)$.

In accordance with Step 1 of MH we calculate $\mu(u)$ for every $u \in V-Z$. First, let $u^j \in W^{k-j}$ and Z_j denote the set of special vertices which lie under u^j, $1 \leq j \leq k$.

$$\begin{aligned}\mu(u_j) &= \sum_{v\in Z_j} d(u^j,v) + \sum_{v\in Z-Z_j} d(u^j,v)\\ &= j2^j + \sum_{r=1}^{k-j}(2r+j)2^{r+j-1}.\end{aligned}$$

Using Lemma 1 we get

$$\mu(u^j) = (2k-j-2)2^k + 2^{j+1}, \qquad 1 \leq j \leq k \tag{13}$$

Evidently, we can write

$$\mu(u^*) = (k-\varepsilon)2^k. \tag{14}$$

Now we assert that

Lemma 7. *For every j with $o \leq j \leq k$ we have*

$$\mu(u^j) > \mu(u^*).$$

Proof. Introducing the function

$$f(j) := \mu(u^j) - \mu(u^*)$$

(see (13) and (14)) we have to prove that

$$f(j) > 0 \qquad 1 \leq j \leq k. \tag{15}$$

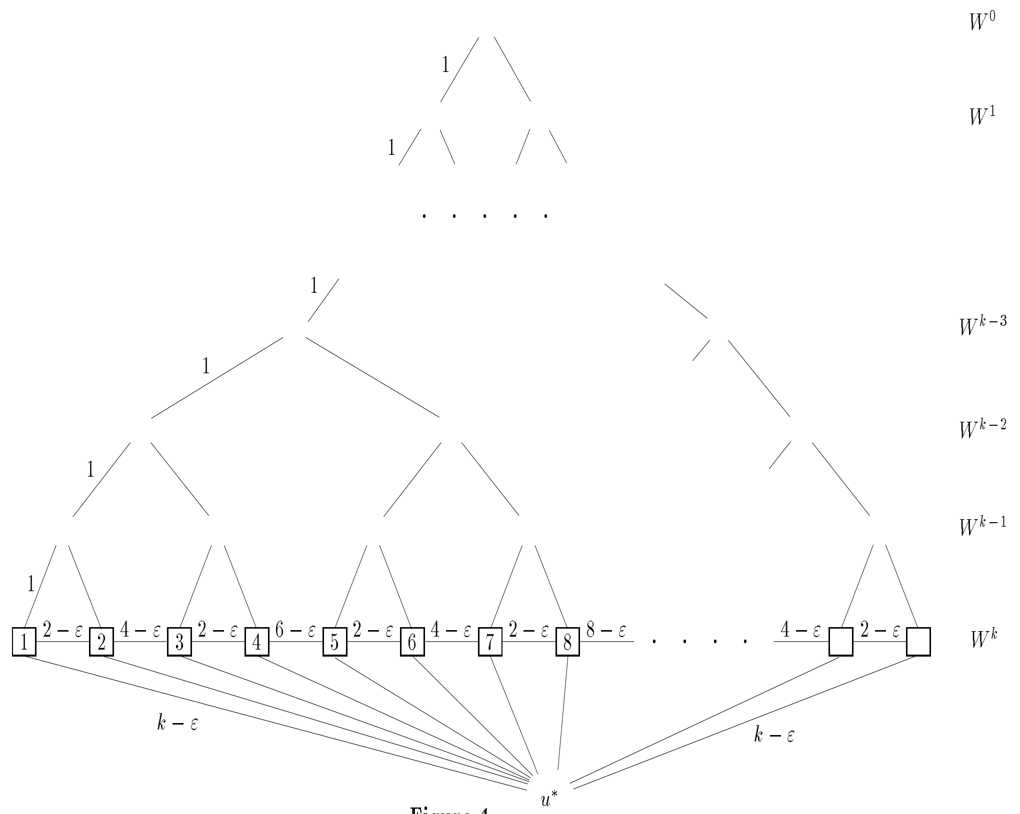


Figure 4.

Since $f(k) > 0$ it remains to prove (15) when $1 \le j \le k-1$. As $0 < \log_e 2 < 1$, derivative

$$f'(j) = -2^k + 2^{j+1}\log_e 2 \le 2^k(-1+\log_e 2) < 0, \qquad 1 \le j \le k.$$

Thus it suffices to verify (15) when $j = k-1$, which is immediate. □

Hence in Step 1 of MH we choose $u^0 = u^*$. One can easily see that $V(T^0) = Z \cup \{u^*\}$ and that T^1 is a tree consisting of some edges iu^* with $i \in Z$ and those edges of path P whose cost is less than $k-\varepsilon$. More precisely, T^1 contains 2^{k-i} edges of cost $2i-\varepsilon$ each, $i = 1, 2, \ldots, t$ and $2^k - \sum_{i=1}^{t} 2^{k-i}$ edges of cost $k-\varepsilon$ each. Thus

$$\begin{aligned} c(T^2) &= \sum_{i=1}^{t}(2i-\varepsilon)2^{k-i} + (k-\varepsilon)\left[2^k - \sum_{i=1}^{t} 2^{k-i}\right] \\ &= 2^{2t+3} - 3\cdot 2^{t+1} - \varepsilon\cdot 2^{2t+1}, \end{aligned} \tag{16}$$

where Lemma 1 was used (recall that $k = 2t+1$).

Further one sees that in accordance with Step 3 we get T^2 which differs from path P only in one edge. Namely, instead of (the most expensive) edge, say v_1v_2, with cost $2k-\varepsilon$ lying in P, the tree T^2 contains two edges v_1u^* and u^*v_2 of total cost $2k-2\varepsilon$. Since

$$\begin{aligned} c(P) &= \sum_{i=1}^{k}(2i-\varepsilon)2^{k-i} = 2^{k+2} - 2(k+2) - 2^k\varepsilon + \varepsilon, \\ c(T^4) &= 2^{k+2} - 2(k+2) - 2^k\varepsilon. \end{aligned} \tag{17}$$

Comparing (16) and (17) we get $T_{\mathrm{MH}} = T^1$.

To complete these considerations let us note that the binary tree B is an optimal solution. However, we shall not prove this assertion, noting only that the cost of an optimal tree T does not exceed the cost of B which is $2^{2t+2}-2$. Hence we have

Theorem 2. *For every integer $t \ge 1$ and sufficiently small real number $\varepsilon > 0$ there exists an instance of the Steiner problem in graphs such that a minimum cost Steiner tree T^* fulfils*

$$c(T^*) \le 2^{2t+2} - 2$$

whereas heuristic MH produces a Steiner tree T_{MH} with cost

$$c(T_{\mathrm{MH}}) = 2^{2t+3} - 3\cdot 2^{t+1} - \varepsilon\cdot 2^{2t+1}.$$

Consequently, Theorem A holds for heuristic MH.

We note that the Step 3 in MH is important for ensuring the first inequality of Theorem A and therefore cannot be deleted. This can be seen with the aid of Fig. 5 where we have a graph G consisting of p special vertices (squares) lying on a path P and p other paths of length k joining root u_0 with the special vertices.

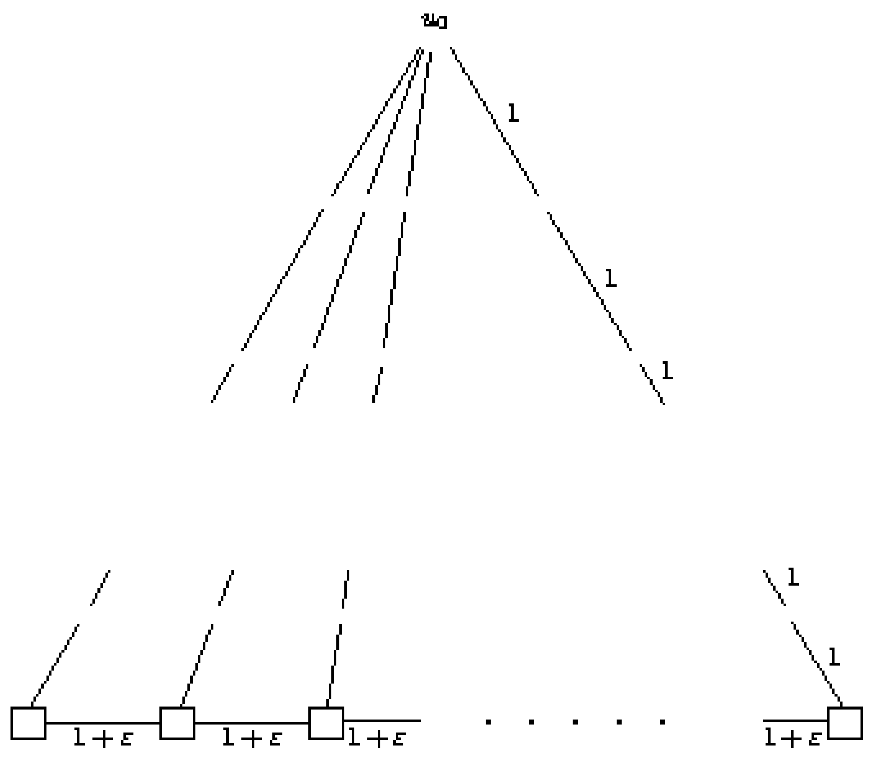


Figure 5

Here $\varepsilon > 0$ is sufficiently small. One can verify that $\mu(u_0)$ is minimal when p is sufficiently large and thus T^1 is a tree of cost pk, whereas the minimum cost of a Steiner tree is equal to $(p-1)(1+\varepsilon)$. Consequently, $c(T_{\mathrm{MH}})/c^*$ can tend to k, which can be arbitrarily large.

3. Antimedian heuristic (AMH).

A vertex $u \in V - Z$ with the maximum sum $\mu(u)$ of distances to all vertices of Z is called an antimedian vertex. The basic idea of AMH is that we successively delete vertices with large $\mu(u)$ without violating the connectedness of the remaining graph. Then the spanning tree heuristic (STH) is applied and the result is compared with that of STH on Z only. We assume that $Z \neq V$.

Our heuristic AMH is simply the following:

Step 1: Put $G' \colon = G$ and $i \colon = 1$. For each vertex $u \in V - Z$ calculate

$$\mu(u) \colon = \sum_{v \in Z} d(u, v)$$

(by Dijkstra's algorithm) and denote the vertices $u \in V - Z$ such that

$$\mu(u_1) \geq \mu(u_2) \geq \cdots \geq \mu(u_q)$$

where $q\colon = |V - Z|$.

Step 2: If Z is connected in $G' - u_i$, then put $G'\colon = G' - u_i$. If $i < q$, then put $i\colon = i + 1$ and go to Step 2.

Step 3: Determine a minimum cost spanning tree in G' and prune it. Denote the resulting tree by T'.

Step 4: Determine a tree T'' spanning Z in G by the spanning tree heuristic.

Step 5: The cheapest of trees T' and T'' is the solution T_{AMH}. STOP.

The time complexity of Step 1 is $O((n-p)n^2)$. The connectedness can be tested in $O(m)$ time and thus Step 2 is of complexity $O((n-p)m)$. Step 3 requires $O(n^2)$ time and Step 4 $O(pn^2)$ [**6**] (or even only $O(m + n \log n)$ time [**5**,**8**]). Thus the overall time complexity of heuristic AMH is $O(n^3)$.

To illustrate AMH let us consider the following example.

Example 3. Let G be the graph in Fig. 6 with edge costs as shown and the set of special vertices $Z = \{1, 2, 3, 4\}$ (in Fig. 6 depicted as squares).

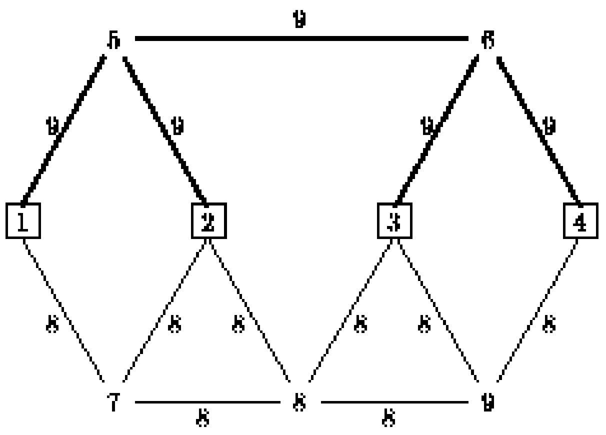


Figure 6.

We see that $\mu(7) = \mu(9) = 56$, $\mu(5) = \mu(6) = 54$ and $\mu(8) = 48$. Thus we successively delete vertices 7 and 9. Then neither 5 nor 6 can be deleted (connectedness!) and thus 8 is deleted. The result G' is even a tree T' (depicted by heavy lines in Fig. 6). Step 4 provides a tree T'' of cost 48. Since $c(T') = 45$, we have $T_{\mathrm{AMH}} = T'$, which is even an optimal solution.

Now we present a bad example for AMH. Our graph G is a complete graph on $p + 1$ vertices as shown in Fig. 7.

Any edge joining two special vertices (squares) is of cost $2-\varepsilon$ and each edge incident with vertex u is of cost 1. Clearly, AMH produces a tree of cost $(p-1)(2-\varepsilon)$

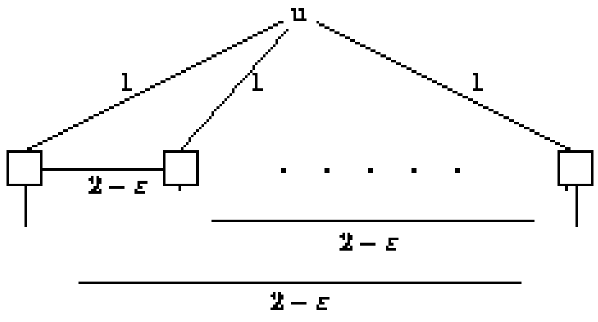


Figure 7.

but the cost of the optimal tree is p. Consequently, Theorem A holds for heuristic AMH.

Note that AMH without Step 4 does not imply the first inequality of Theorem A as the instance from Fig. 8 shows.

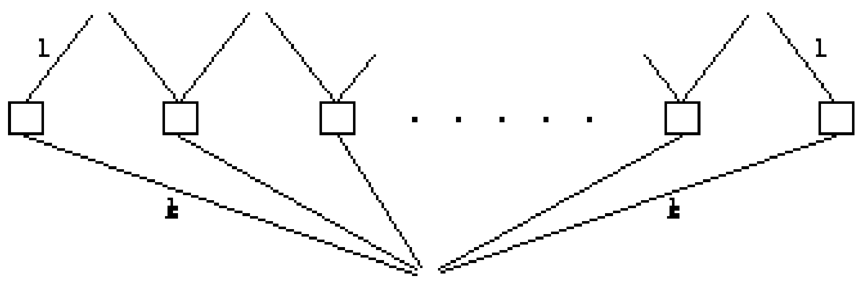


Figure 8.

In "the upper part" each edge is of cost 1 and in "the lower part" each edge is of cost k. It is a matter of routine to verify that in Step 2 of AMH all the "upper vertices" are deleted whenever k is fixed and p is sufficiently large and thus AMH wihout Step 4 provides a tree of cost pk. However, the cost of the optimal Steiner tree is equal to $2(p-1)$. Thus $c(T_{\mathrm{AMH}})/c^*$ can tend to $k/2$.

References

1. El-Arbi C., *Une heuristique pour le problème de l'arbre de Steiner. R.A.I.O.*, Operations Research **12** (1978), 207–212.
2. Even S., *Graph algorithms*, Computer Science Press, Potomac, 1979.
3. Hwang F. K. and Richards D. S., *Steiner tree problems*, preprint.

4. Jarník V. and Kössler M., *O minimálních grafech obsahujících n daných bodů*, Časopis Pěst. Mat. Fyz. **63** (1934), 223–235.
5. Kou L. T. and Makki T., *An even faster approximation algorithm for the Steiner tree problem in graphs*, Congr. Numer. **59** (1987), 147–154.
6. Kou L., Markowsky G. and Berman L., *A fast algorithm for Steiner trees*, Acta Informatica **15** (1981), 141–145.
7. Lawler E. L., *Combinatorial optimization: networks and matroids*, (Holt, Rinehart and Winston), New York, 1976.
8. Mehlhorn K., *A faster approximation algorithm for the Steiner problem in graphs*, Inf. Process. Lett. **27** (1988), 125–128.
9. Plesník J., *A bound for the Steiner tree problem in graphs*, Math. Slovaca **31** (1981), 155–163.
10. Rayward-Smith V. J., *The computation of nearly minimal Steiner trees in graphs*, Int. J. Math. Educ. Sci. Technol. **14** (1983), 15–23.
11. Rayward-Smith V. J. and Clare A., *On finding Steiner vertices*, Networks **16** (1986), 283–294.
12. Takahashi H. and Matsuyama A., *An approximate solution for the Steiner problem in graphs*, Math. Japonica **24** (1980), 573–577.
13. Winter P., *Steiner problem in networks: a survey*, Networks **17** (1987), 129–167.

M. Diané, 1097 Conakry, Guinea

J. Plesník, Katedra numerických a optimalizačných metód MFF Univerzity Komenského, Mlynská dolina, 842 15 Bratislava, Czechoslovakia