Acta Math. Univ. Comenianae
Vol. LX, 2(1991), pp. 269–284

WORST–CASE RELATIVE PERFORMANCES OF HEURISTICS FOR THE STEINER PROBLEM IN GRAPHS

J. PLESNÍK

Abstract. The Steiner problem asks for a minimum cost tree spanning a given subset of vertices in a graph (network) with positive edge costs. First we modify the Rayward-Smith heuristic and prove that this does not change its worst-case performance, but the number of iterations is often reduced. Then 9 heuristics are theoretically analysed as to their worst-case relative performances.

1. Introduction

In the Steiner problem in graphs (networks) we are given a graph (undirected, without loops and multiple edges) $G = (V, E)$, a positive-valued cost (length) function $c\colon E \to \mathbb{R}^+$, and $Z \subseteq V$. We are asked to find a minimum cost tree $T \subseteq G$ spanning Z, where the cost of T, $c(T)$, is the sum of its edge costs. Denote $n := |V|$, $m := |E|$ and $p := |Z|$.

At the present time there are more than 100 papers related to this Steiner problem. Most of them are surveyed by Winter in the excellent paper [**19**]. For a further work see the very recent survey by Hwang and Richards [**7**] which gives a vast bibliography. The Steiner problem is NP-hard even in some special cases. Moreover, in these surveys no polynomial time approximation algorithm A is given with the worst-case error ratio $c(T_A)/c_*$ that is bounded by $2-\varepsilon$, for $\varepsilon > 0$. (T_A is a Steiner tree produced by A and $c_* := c(T_*)$ is the cost of a minimum cost Steiner tree T_* for the same instance.) Note that recently Zelikovsky [**21**] announced an 11/6-approximation algorithm. However, we do not study his heuristic. Also very recent "combined" (2-approximation) heuristics from [**5, 14, 20**] remain undiscussed here.

There are several experimental studies [**12, 14, 16, 20**] comparing various heuristics. One of the best heuristics is that developed by Rayward-Smith [**11, 12**]. In Section 3 we present and analyse a modification which usually takes a less number of iterations than the original Rayward-Smith heuristic does. Note that before actually solving the Steiner problem in practice, some tests may reduce the

---

Received May 13, 1991; revised September 2, 1991.
1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 68E10.
Key words and phrases. Graph, network, Steiner tree, approximation algorithm, heuristic, worst case.

problem size (by eliminating vertices or edges from the graph) [**6**]. Nevertheless we consider heuristics in their pure form.

The core of our contribution is Section 4 which gives a comparative theoretical analysis of 9 heuristics. This extends and strengthens results of Widmayer [**18**] and Plesník [**10**].

Some results of the present paper were reported at the Fourth Czechoslovak Symposium on Combinatorics held in Prachatice, June 10–16, 1990.

2. A Survey of Heuristics

Here we shall consider 8 known heuristics. Some of them are published in well available journals and therefore no description is given. A further heuristic (ADHF) will be added in Section 3. Given a heuristic H, its worst-case time complexity is denoted by τ_H and its worst-case performance ratio by ρ_H.

STH (the minimum spanning tree heuristic): This heuristic is usually attributed to Kou, Markowsky and Berman [**8**], but it was developed by Choukhmane [**4**] and then several times rediscovered (up to slight differences) (see [**7, 19**]). The version we consider here can be found in [**8, 10, 19**]. $\tau_{\mathrm{STH}} = O(pn^2)$ (for faster versions see references in [**7**]) and $\rho_{\mathrm{STH}} = 2 - 2/p$.

PH (the minimum path heuristic): It was developed by Takahashi and Matsuyama [**13**]. $\tau_{\mathrm{PH}} = O(pn^2)$ and $\rho_{\mathrm{PH}} = 2 - 2/p$. For a description of PH see e.g. [**10, 13, 19**].

CH (the contraction heuristic): It was developed be Plesník [**9**]. Here we consider this heuristic in the form specified in [**10, 19**]. $\tau_{\mathrm{CH}} = O(n^3)$ and $\rho_{\mathrm{CH}} = 2 - 2/p$ [**10, 19**]. Even a bit better (but more complicated) bound than $2 - 2/p$ was derived in [**10**].

CHR (the revised contraction heuristic): It was suggested and analysed by Plesník [**10**]. The values of parameters τ and ρ are the same as for CH.

ADH (the minimum average distance heuristic): This heuristic was suggested by Rayward-Smith [**11, 12**]. It can be found also in [**10, 19**], but for our aims in Section 3 we give its full description here.

Step 1: Begin with the collection F of single vertex trees consisting of the p Z-vertices. For the sake of simplicity F is called a forest.

Step 2: For every vertex $v \in V$ relabel the trees in the current forest $F = \{T_1, \dots, T_k\}$ such that they are in nondecreasing order of their distance from v (i.e. $d(v, T_1) \leq d(v, T_2) \leq \cdots \leq d(v, T_k)$) and for each r, $2 \leq r \leq k$, compute mean distance

$$\mu(v, r) := \frac{\sum_{j=1}^{r} d(v, T_j)}{r - 1}$$

Define $f(v) := \min\{\mu(v, r) \,|\, 2 \leq r \leq k\}$ and choose $\bar{v}$ minimizing $f(v)$.

Step 3: Join the corresponding trees T_1 and T_2 nearest to $\bar{v}$ by a shortest walk through $\bar{v}$ forming a new tree T'. Put $F := (F - \{T_1, T_2\}) \cup \{T'\}$. If F contains at least two trees go to Step 2, else the single tree in F is the solution T_{ADH}. STOP. $\tau_{\text{ADH}} = O(n^3)$. ρ_{ADH} is bounded from above by $2 - 2/p$ and can tend to 2, as shown by Waxman and Imase [**17**].

We fully describe also the following three heuristics because they are not very known and the original sources are not easily accessible.

2-TH (the minimum 2-tuple heuristic of Wang [**15**]): It is similar to PH which is Prim based, but 2-TH is Kruskal based (cf. [**3**]).

Step 1: Begin with the collection F of single vertex trees consisting of the p Z-vertices.

Step 2: If $|F| = 1$, then the single tree in F is the solution $T_{\text{2-TH}}$, STOP. Else find two trees T_1, $T_2 \in F$ with minimal distance $d(T_1, T_2)$ in G and join them by a shortest path (between T_1 and T_2) forming a new tree T'. Put $F := (F - \{T_1, T_2\}) \cup \{T'\}$ and go to Step 2.

Wang [**15**] described an implementation of this heuristic with $\tau_{\text{2-TH}} = O(pn^2)$. The analysis by Widmayer [**18**] shows that $\rho_{\text{2-TH}} = 2 - 2/p$. In Section 4 we shall see that $\rho_{\text{2-TH}}$ can tend to 2.

3-TH (the minimum 3-tuple heuristic of Chen [**2**]): It is a natural analogue of 2-TH and runs as follows.

Step 1: Begin with the collection F of single vertex trees consisting of the p Z-vertices.

Step 2: If $|F| = 1$, then the single tree in F is the solution $T_{\text{3-TH}}$, STOP. If $|F| = 2$, then join the two trees of F by a shortest path forming a single tree, which is the solution $T_{\text{3-TH}}$, STOP.
Else find 3 trees T_1, T_2 and T_3 in F such that a minimum Steiner tree T' connecting them in G has least cost. Put $F := (F - \{T_1, T_2, T_3\}) \cup \{T'\}$ and go to Step 2.

According to [**2, 18**] 3-TH can be implemented in such a way that $\tau_{\text{3-TH}} = O(mp^2 \log n)$. Widmayer [**18**] analysed this heuristic and proved that $\rho_{\text{3-TH}} \leq 2 - 2/p$. In Section 4 we shall show that $\rho_{\text{3-TH}}$ can tend to 2.

P3-TH (the minimum path and minimum 3-tuple heuristic of Chen [**2**]): This heuristic combines ideas from PH and 3-TH. Assume $p \geq 3$.

Step 1: Find 3 Z-vertices such that a minimum Steiner tree T for them has least cost.

Step 2: If T includes all the Z-vertices, then $T_{\text{P3-TH}} := T$ is the solution, STOP. Else change T as follows.

2.1: Find a Z-vertex v outside T which is nearest to a vertex u of T.

2.2: For each edge e of T incident with u do the following.

2.2.1: Remove e from T and prune the obtained two subtrees of T (i.e. successively delete all vertices of degree 1 not belonging to Z), yielding trees T_1 and T_2.

2.2.2: Find a minimum Steiner tree T' connecting T_1, T_2 and v in G.

2.3: Let T be a T' with least cost among those obtained in Step 2.2. Go to Step 2.

An implementation [**2, 18**] of this heuristic provides $\tau_{\text{P3-TH}} = O(mnp \log n)$. The analysis done by Widmayer [**18**] shows that $\rho_{\text{P3-TH}} \leq 2 - 2/p$ and in Section 4 we shall see that $\rho_{\text{P3-TH}}$ can be arbitrarily close to 2.

3. A Modification of ADH.

Winter [**19,** p. 148] suggested to investigate versions of ADH with Step 3 permitting to connect several (and not only two) trees of F. Here we present such a version of ADH and give its worst-case performance analysis.

ADHF (the minimum average distance heuristic with full connection): This heuristic differs from ADH only in Step 3 which is replaced by

Step 3': Choose $\bar{r}$, $2 \leq \bar{r} \leq k$, minimizing $\mu(\bar{v}, r)$. Join (successively) each tree T_j of the corresponding $\bar{r}$ trees nearest to $\bar{v}$ by a shortest path P_j to $\bar{v}$ forming a new tree T'. Put $F := (F - \{T_1, \ldots, T_{\bar{r}}\}) \cup \{T'\}$. If F contains at least two trees go to Step 2. Else the single tree in F is the solution T_{ADHT}. STOP.

One can easily verify that the complexity of ADHF is $O(n^3)$. Notice that Step $3'$ is less "precautious" than Step 3. However, our few computational results indicate that this does not affect the quality of the solutions, but ADHF requires often less iterations than ADH does.

Note that ADHF is not new in full because Bern and Plassmann [**1**] already considered it and proved that it is a 4/3-approximation algorithm for Steiner problem on complete graphs with edge lengths 1 and 2. But Bern and Plassmann thought that they had considered ADH. In Section 4 we shall show that in general ADH and ADHF do not necessarily give solutions of the same cost.

The following result shows that ADHF is similar to several other heuristics with respect to its worst-case performance [**7, 18**].

Theorem 1. *For any instance of the Steiner problem we have*

$$c(T_{\mathrm{ADHF}}) \leq (2 - 2/p)c_*.$$

Moreover, for any $\varepsilon > 0$ *there is an instance of the Steiner problem such that*

$$c(T_{\mathrm{ADHF}}) > (2 - \varepsilon)c_*.$$

Proof. Since the "bad" example given by Waxman and Imase [**17**] for ADH works also for ADHF, it remains to prove the first inequality. The proof is much easier than that for ADH [**17**]. We show that each iteration of ADHF can be associated and compared with a few iterations of a minimum spanning tree algorithm processing an auxiliary complete graph. This will yield the required inequality.

Let us consider the i-th iteration of ADHF and the corresponding forest $F^{(i)}$. Thus we have chosen a vertex $\bar{v}^{(i)}$ and determined a number $\bar{r}^{(i)}$. The corresponding mean distance $\mu(\bar{v}^{(i)}, \bar{r}^{(i)})$ is denoted by $\mu^{(i)}$. Let $P_j^{(i)}$ denote a shortest $\bar{v}^{(i)} - T_j^{(i)}$ path, hence $P_j^{(i)}$ has cost $c(P_j^{(i)}) = d(\bar{v}^{(i)}, T_j^{(i)})$. Define $\alpha^{(i)}$ to be the minimum distance between two Z-vertices from distinct trees of $F^{(i)}$. In the i-th iteration trees $T_1^{(i)}, \ldots, T_{\bar{r}^{(i)}}^{(i)}$ are connected into one new tree. Hence $\alpha^{(i+1)}$ of the $(i+1)$-st iteration fulfils

$$\alpha^{(i)} \leq \alpha^{(i+1)} \tag{1}$$

By the rules of ADHF we have

$$\mu^{(i)} \leq \alpha^{(i)} \tag{2}$$

Given an instance of the Steiner problem processed by ADHF, construct the complete graph $K(Z)$ on Z and define its edge cost function c' successively for every iteration i of ADHF as follows. In the i-th iteration of ADHF put $c'(uw) := \alpha^{(i)}$ whenever u and w belong to distinct trees of $\{T_1^{(i)}, \ldots, T_{\bar{r}^{(i)}}^{(i)}\} \subseteq F^{(i)}$. Now apply to $K(Z)$ with c' the Kruskal minimum spanning tree algorithm (MSTA) (see e.g. [**3**]).

Each iteration i of ADHF connecting trees $T_1^{(i)}, \ldots, T_{\bar{r}^{(i)}}^{(i)}$ can be associated with $\bar{r}^{(i)} - 1$ iterations of MSTA as follows. For each $j = 1, \ldots, \bar{r}^{(i)}$ choose (arbitrarily) a Z-vertex u_j in $T_j^{(i)}$. Then add $\bar{r}^{(i)} - 1$ edges, one per iteration of MSTA, to form a spanning tree on vertices $u_1, \ldots, u_{\bar{r}^{(i)}}$. Each of these $\bar{r}^{(i)} - 1$ edges has its

c'-cost equal to $\alpha^{(i)}$. Since (1) holds, this can be done without violating the rules of MSTA. Denote by T_{MSTA} the output of MSTA for $K(Z)$ with c'. We are going to show that $c(T_{\mathrm{ADHF}}) \leq c'(T_{\mathrm{MSTA}})$.

In the i-th iteration of ADHF we add paths $P_1^{(i)}, \ldots, P_{\bar{r}^{(i)}}^{(i)}$, which are not necessarily edge disjoint. Thus the c-cost contribution does not exceed

$$\sum_{j=1}^{\bar{r}^{(i)}} c(P_j^{(i)}) = \sum_{j=1}^{\bar{r}^{(i)}} d(\bar{v}(i), T_j^{(i)}) = (\bar{r}^{(i)} - 1)\mu^{(i)}$$

On the other hand, the corresponding $\bar{r}^{(i)} - 1$ iterations of MSTA give c'-cost contribution equal to $(\bar{r}^{(i)} - 1)\alpha^{(i)} \geq (\bar{r}^{(i)} - 1)\mu^{(i)}$ (by (2)). Hence $c(T_{\mathrm{ADHF}}) \leq c'(T_{\mathrm{MSTA}})$.

Now consider the cost function c'' for $K(Z)$ with $c''(uw) = d_G(u, w)$ (the distance in G) for any two Z-vertices u and w. It is well known (see e.g. [**4, 8**]) that the cost of a minimum c''-cost spanning tree of $K(Z)$ does not exceed $(2 - 2/p)c_*$. Since $c'(uw) \leq d_G(u, w)$, $c'(T_{\mathrm{MSTA}}) \leq (2-2/p)c_*$ too and the proof is completed.□

4. Relative Performances of Heuristics

In the literature on can find examples demonstrating that a heuristic provides a better solution than another heuristic does. Several such examples are hidden in computational experiments. Explicitly presented examples are known only for some pairs of heuristics. The first systematic study if this question is due to Widmayer [**18**] who showed that for any pair of distinct heuristics (H_1, H_2) from set {STH, PH, 2-TH, 3-TH, P3-TH} there is an instance of the Steiner problem for which H_1 yields a better solution than H_2 does. His cost ratios $c(T_{H_2})/c(T_{H_1})$ of the corresponding solutions are close to 1 (5/4 or so). Nevertheless, they exceed 1 for any choice of solutions. Therefore Widmayer claimed that any two of the above five heuristics are strongly incomparable. Being unaware of the work of Widmayer, we studied [**10**] the heuristics from set {STH, PH, CH, CHR, ADH} and except for four pairs gave ratios arbitrarily close to 2. Here we strengthen and extend the above results. The following nine heuristics are considered: STH, PH, CH, CHR, ADH, ADHF, 2-TH, 3-TH and P3-TH. We say that a heuristic H_1 wins over a heuristic H_2 with ratio $r > 1$ if there is an instance of the Steiner problem such that for any outputs their corresponding costs fulfill

$$c(T_{H_2})/c(T_{H_1}) \geq r$$

Theorem 2. *For any small $\varepsilon > 0$ and any non-empty (H_1, H_2) entry of Table* 1 *H_1 wins over H_2 with ratio $2 - \varepsilon$.*

Proof. In [**10**] we proved the cases if $H_1, H_2 \in$ {STH, PH, CH, CHR, ADH}. The examples given in [**10**] can also be used for other pairs as indicated in Table 1.

E.g. (ADHF, CH) entry is $e1$. This means that in Example 1 of [**10**] $c(T_{\text{CH}})$ is equal to nearly two times $c(T_{\text{ADHF}})$. Really, for any solutions T_{CH} and T_{ADHF} we have $c(T_{\text{CH}}) = (p-1)(2-\delta)$ and $c(T_{\text{ADHF}}) = p$. Thus $c(T_{\text{CH}})/c(T_{\text{ADHF}})$ tends to 2 whenever δ tends to zero and p goes to infinity. All such verifications are easy and therefore are left to the reader. Thus it remains to deal with entries equal to Ej, which is the reference to Example j in the sequel.

Table 1. Row heuristics win over column heuristics. Symbols Ej and ej mean Example j of this paper and paper [**10**], respectively.

	STH	PH	CH	CHR	ADH	ADHF	2-TH	3-TH	P3-TH
STH		e5			e4	e4	e4	E6	E6
PH	e3		e3	e3	e4	e4	e4	E6	E6
CH		e5			e4	e4	e4	E6	E6
CHR	e2	e5	e2		e4	e4	e2	E6	E6
ADH	e1	e1	e1	e1		E2	e1	E5	E5
ADHF	e1	e1	e1	e1	E1		e1	E5	E5
2-TH	E3	E3	E3	E3	E4	E4		E3	E6
3-TH	e1	e1	e1	e1	E4	E4	e1		E7
P3-TH	E3	E3	E3	E3	E4	E4	e1	E3	

Example 1. Consider the graph G^k defined in Fig. 1, where $k \geq 2$. The "beak" part is realized as shown, the Z-vertices are denoted by black circles and the costs of edges are marked ($\delta > 0$ is sufficiently small, say, $\delta < 1/10$). One can easily verify that ADHF yields for G^k a unique solution T^k_{ADHF} consisting of "the upper part" of G^k (see Fig. 2 if $k = 2$) and $c(T^k_{\text{ADHF}}) = (k+6)2^{k+1} - 15 \cdot 2^k \cdot \delta$. On the other hand ADH yields a unique solution T^k_{ADH} formed by "the lower part" of G^k (see Fig. 3 if $k = 2$) and $c(T^k_{\text{ADH}}) = (2k+5)2^{k+1} - (33 \cdot 2^k - 17)\delta$. Thus $c(T^k_{\text{ADH}})/c(T^k_{\text{ADHF}})$ tends to 2 whenever k tends to infinity.

Example 2. To prove that ADH can win over ADHF we use similar graphs as in Example 1. Let $\tilde{G}^k$ be the graph obtained from G^k by overturning all "the beaks" as can be seen in Fig. 4 where $\tilde{G}^2$ is depicted. It is left to the reader to verify that ADHF produces a unique solution $\tilde{T}^k_{\text{ADHF}}$ consisting of "the lower part" of $\tilde{G}^k$ while the unique solution $\tilde{T}^k_{\text{ADH}}$ produced by ADH consists of "the upper part" of $\tilde{G}^k$. As $c(\tilde{T}^k_{\text{ADHF}}) = (2k+5)2^{k+1} - (32 \cdot 2^k - 17)\delta$ and $c(\tilde{T}^k_{\text{ADH}}) = (k+6)2^{k+1} - 16 \cdot 2^k \cdot \delta$, the proof follows.

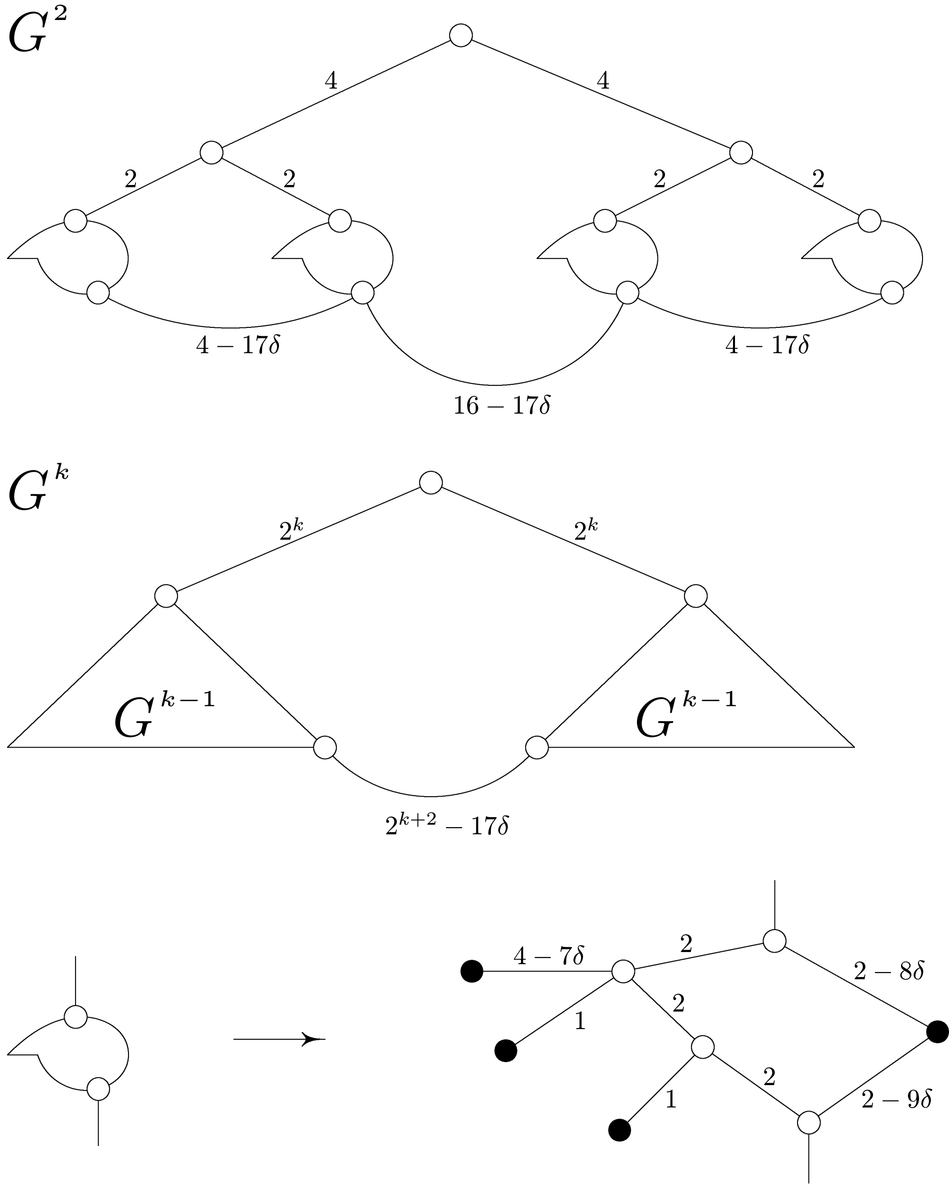
G^2
4
4
2
2
2
2
$4-17\delta$
$4-17\delta$
$16-17\delta$
G^k
2^k
2^k
G^{k-1}
G^{k-1}
$2^{k+2}-17\delta$
$4-7\delta$
2
$2-8\delta$
1
2
1
2
$2-9\delta$

Figure 1.

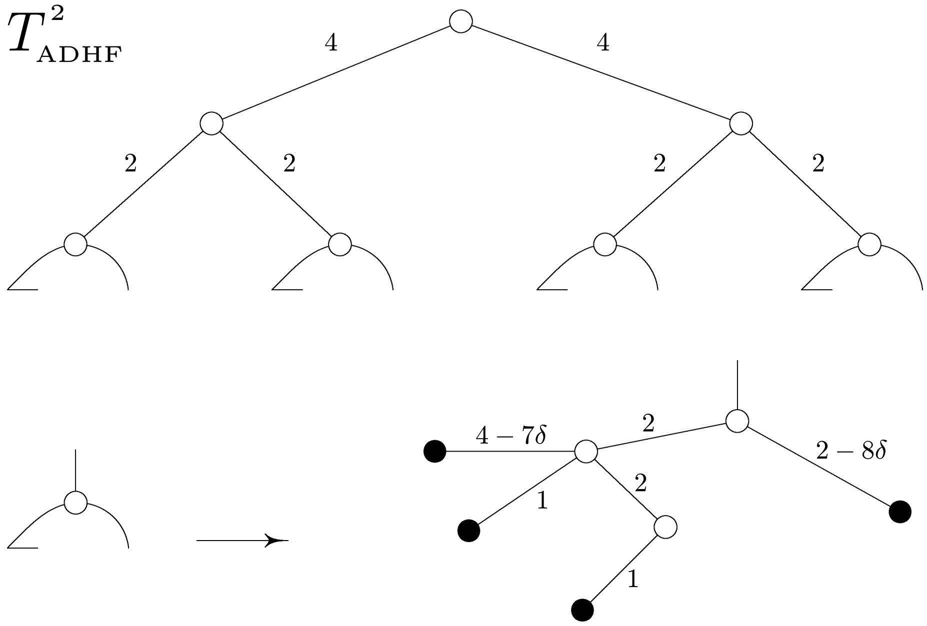

Figure 2.

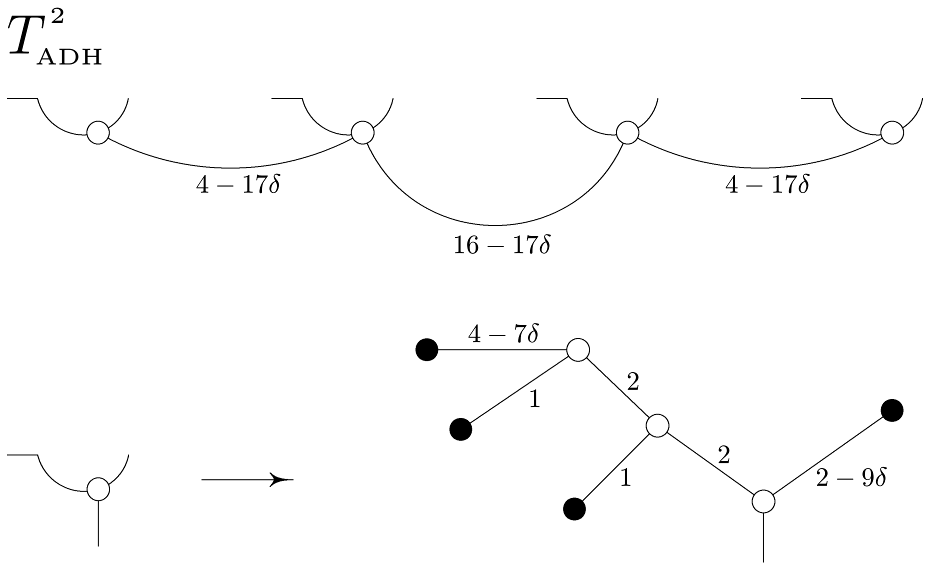

Figure 3.

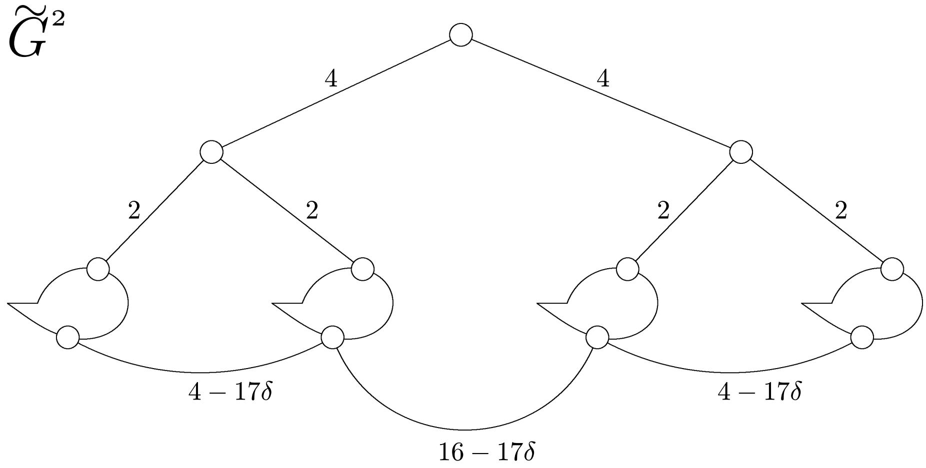

Figure 4.

Example 3. Consider the graph of Fig. 5. In general, the top vertex is assumed to be of degree $k \to \infty$ and $\delta \to 0^+$. One can easily verify that the following heuristic solutions are uniquely determined and that

$$c(T_{\mathrm{STH}}) = c(T_{\mathrm{PH}}) = c(T_{\mathrm{CH}}) = c(T_{\mathrm{CHR}}) = c(T_{\text{3-TH}}) = (k-1)(2+\delta) + 4k\delta,$$
$$c(T_{\text{2-TH}}) = c(T_{\text{P3-TH}}) = k + 4k\delta.$$

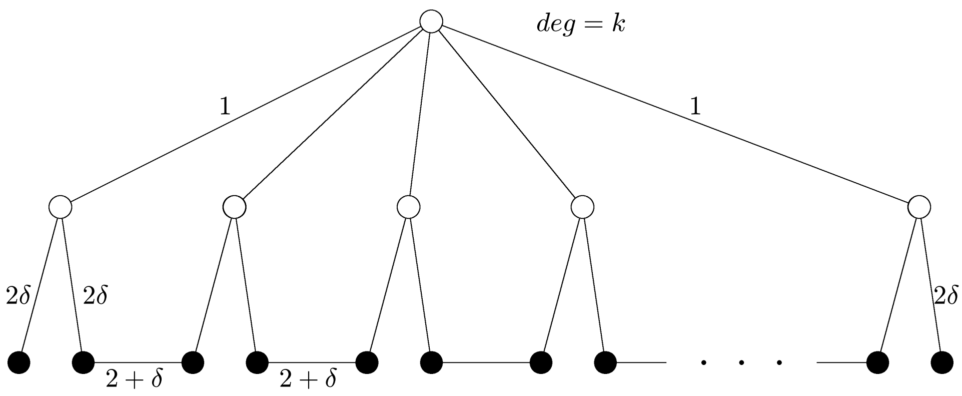

Figure 5.

Example 4. Consider a binary tree of depth k with an appendage as shown in Fig. 6 for $k = 4$. Depending on the level, an edge has cost $1, 1, 2, 4, \ldots,$ or 2^{k-2}. In the appendage the "vertical" edges have cost $3/2 - \delta$ each, the shortest horizontal edges have cost $8 - \delta$ each, the second shortest edges have $16 - \delta$, … , and the longest edge has cost $2^k - \delta$. Again, $k \to \infty$ and $\delta \to 0^+$. One can verify that $T_{2\text{-TH}}$ is the (upper) binary tree with

$$c(T_{2\text{-TH}}) = (k+1)2^{k-1}.$$

Further, $T_{3\text{-TH}} = T_{\text{P3-TH}}$ differs from $T_{2\text{-TH}}$ only in three edges and

$$c(T_{3\text{-TH}}) = c(T_{\text{P3-TH}}) = (k+1)2^{k-1} + 3/2 - 3\delta.$$

On the other hand, ADH and ADHF yield the lower tree equal to the appendage. Thus

$$c(T_{\text{ADH}}) = c(T_{\text{ADHF}}) = (2k-1)2^{k-1} - (2^k + 2^{k-2} - 1)\delta.$$

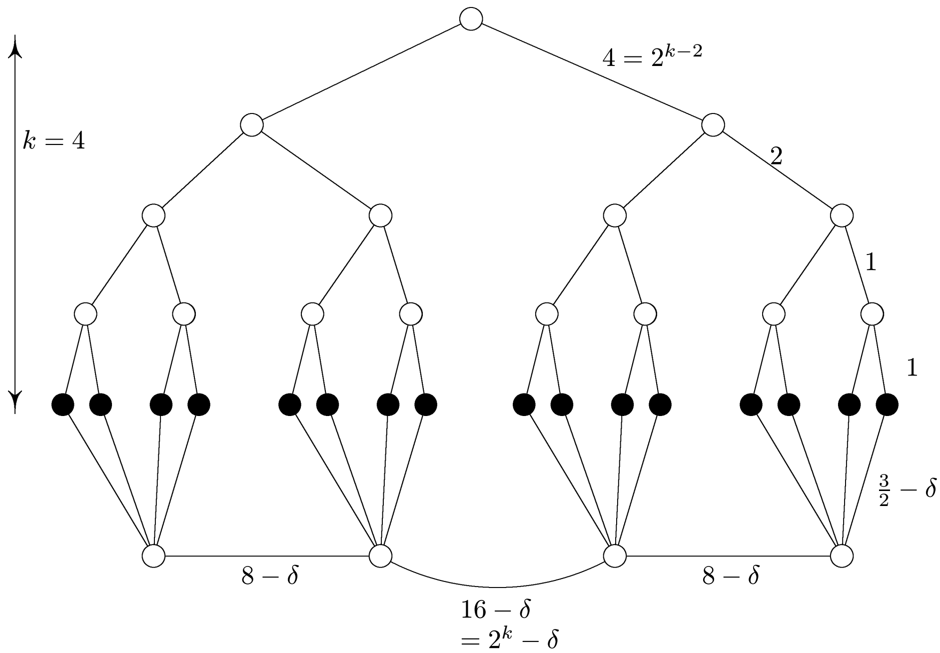

Figure 6.

Example 5. Consider the graph of Fig. 7, where the top vertex is of degree $k \to \infty$ and $\delta \to 0^+$. It can be seen that the tree $T_{\text{ADH}} = T_{\text{ADHF}}$ contains the top vertex and

$$c(T_{\text{ADH}}) = c(T_{\text{ADHF}}) = k + 5k\delta.$$

On the other hand, the tree $T_{3\text{-TH}} = T_{\text{P3-TH}}$ does not contain the top vertex and

$$c(T_{3\text{-TH}}) = c(T_{\text{P3-TH}}) = (2k-1) + (4k+1)\delta.$$

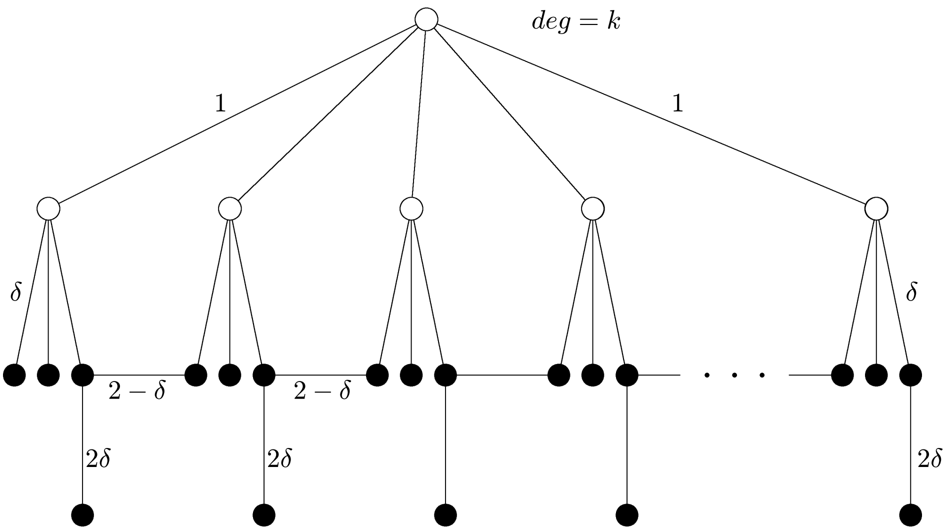

Figure 7.

Example 6. Now consider the graph of Fig. 8 which consists of a binary tree of depth k without one leaf together with an appendage of degree three vertices. Here $k \to \infty$ and $\delta < \frac{1}{2^k}$. One can verify that STH, PH, CH, CHR, and 2-TH yield "the upper tree". On the other hand "the lower tree" (the appendage) is the solution yielded by 3-TH and P3-TH (the leftmost three Z-vertices are connected first). Thus

$$c(T_{\mathrm{STH}}) = c(T_{\mathrm{PH}}) = c(T_{\mathrm{CH}}) = c(T_{\mathrm{CHR}}) = c(T_{\text{2-TH}}) = (2k+1)2^{k-1} \\ - 2 + (2^{k-1} - 1)\delta + 2^{k-2}(2^{k-1}+1)\delta^2$$

and

$$c(T_{\text{3-TH}}) = c(T_{\text{P3-TH}}) = (4k-4)2^{k-1} - \delta^2.$$

Example 7. Finally consider the graph consisting of a ternary tree of depth k and some "lower" edges as shown in Fig. 9 for $k = 3$. The edges of each level have the same costs as indicated. Again $k \to \infty$ and $\delta \to 0^+$. One can verify that 3-TH yields the upper ternary tree as a unique solution. The output of P3-TH is the tree consisting of the "lower" edges and the edges of cost 1. Thus

$$c(T_{\text{3-TH}}) = k \cdot 3^k$$

and

$$c(T_{\text{P3-TH}}) = (2k-2)3^k + 3 - (3^{k-1} - 1)\delta.$$

Having covered all nonempty entries of Table 1, the proof is complete. □

Remark 1. In Table 1 every entry is covered by at most one example, but the reader has certainly observed that some entries can be supplied by several our examples (e.g. (2-TH, 3-TH) entry is E3 and also E6).

Remark 2. As noted in Section 2, Widmayer [**18**] proved that $\rho_{\text{2-TH}}$, $\rho_{\text{3-TH}}$, and $\rho_{\text{P3-TH}}$ are bounded above by $2 - 2/p$. Our examples show that these parameters can tend to 2 (see e.g. E5 and e1).

Remark 3. One sees that 2-TH and 3-TH can be continued to receive a minimum q-tuple heuristic q-TH with a fixed $q \geq 4$. Evidently, q-TH is a polynomial time heuristic. Unfortunately $\rho_{q\text{-TH}}$ can also tend to 2 as one can verify on a graph similar to that in Example 3 (Fig. 5). (In the upper tree, at each vertex with 2 edges of cost 2δ consider $q-1$ such edges.)

At the end we note that the four empty entries in Table 1 are open problems. Even no wins with ratios $1 + \varepsilon$ ($\varepsilon > 0$) are known.

References

1. Bern M. and Plassmann P., *The Steiner problem with edge lengths 1 and 2*, Inform. Process. Lett. **32** (1989), 171–176.

2. Chen N. P., *New algorithms for Steiner tree on graphs*, Proceedings of the IEEE International Symposium on Circuits and Systems, 1983, pp. 1217–1219.

3. Cheriton D. and Tarjan R. E., *Finding minimum spanning trees*, SIAM J. Computing **5** (1976), 724–742.

4. Choukhmane E.-A., *Une heuristique pour le probleme d l'arbre de Steiner*, RAIRO Operations Res. **12** (1978), 207–212.

5. Diané M. and Plesník J., *Three new heuristics for the Steiner problem in graphs*, Acta Math. Univ. Comenianae **60** (1991), 105–121.

6. Duin C. W. and Volgenant A., *Reduction tests for the Steiner problem in graphs*, Networks **19** (1989), 549–567.

7. Hwang F. K. and Richards Dana S., *Steiner tree problems*, preprint.

8. Kou L., Markowsky G. and Berman L., *A fast algorithm for Steiner trees*, Acta Informatica **15** (1981), 141–145.

9. Plesník J., *A bound for the Steiner tree problem in graphs*, Math. Slovaca **31** (1981), 155–163.

10. ______, *On heuristics for the Steiner problem in graphs* (to appear).

11. Rayward-Smith V. J., *The computation of nearly minimal Steiner trees in graphs*, Int. J. Math. Educ. Sci. Technol. **14** (1983), 15–23.

12. Rayward-Smith V. J. and Clare A., *On finding Steiner vertices*, Networks **16** (1986), 283–294.

13. Takahashi H. and Matsuyama A., *An approximate solution for the Steiner problem in graphs*, Math. Japonica **24** (1980), 573–577.

14. Voss S., *Steiner's problem in graphs: heuristic methods*, Discrete Appl. Math. (to appear).

15. Wang S. M., *A multiple source algorithm for suboptimum Steiner trees in graphs*, Proc. International Workshop on Graphtheoretic Concepts in Computer Science (H. Noltemeier, ed.), Trauner, Würzburg, 1985, pp. 387–396.

16. Waxman B. M., *Routing of multipoint connections*, IEEE J. Select. Areas Comm. **6** (1988), 1617–1622.

17. Waxman B. M. and Imase M., *Worst-case performance of Rayward-Smith's Steiner tree heuristic*, Inform. Process. Lett. **29** (1988), 283–287.

18. Widmayer P., *Fast approximation algorithms for Steiner's problem in graphs* (Habilitation thesis, 1986), Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Universität Karlsruhe, 1987 (Bericht 185).

19. Winter P., *Steiner problem in networks: a survey*, Networks **17** (1987), 129–167.

20. Winter P. and Smith MacGregor J., *Path-distance heuristics for the Steiner problem in undirected networks*, Algorithmica (to appear).

21. Zelikovsky A. Z., *An 11/6-approximation algorithm for the Steiner problem on networks*, In: Proc. 4th Czechoslovak Symp. on Combinatorics (to appear).

J. Plesník, Department of Numerical and Optimization Methods, Faculty of Mathematics and Physics, Comenius University, 842 15 Bratislava, Czechoslovakia