Algorithmica (1993) 9: 463–470

Algorithmica

An 11/6-Approximation Algorithm for the Network Steiner Problem

A. Z. Zelikovsky[1]

Abstract. An instance of the Network Steiner Problem consists of an undirected graph with edge lengths and a subset of vertices; the goal is to find a minimum cost Steiner tree of the given subset (i.e., minimum cost subset of edges which spans it). An 11/6-approximation algorithm for this problem is given. The approximate Steiner tree can be computed in the time $O(|V|\,|E| + |S|^4)$, where V is the vertex set, E is the edge set of the graph, and S is the given subset of vertices.

Key Words. Steiner tree, Approximation algorithm.

1. Introduction. Let $G = (V, E, d)$ be a graph with nonnegative edge lengths $d: E \to R^+$ and let S be a set of distinguished vertices. A *Steiner tree* is a tree which spans all members of S. The *Network Steiner Problem* asks for a minimum cost Steiner tree $T_{\min}$, where the cost of a set of edges is the sum of lengths of its elements. Since this problem is NP-complete (see [1]) and has important applications, faster approximation algorithms are valuable.

We use the following notations. The *metric closure* of G is a complete graph $\bar{G} = (V, \bar{E}, \bar{d})$ which has edge lengths equal to the shortest path distances in G. G_M is the subgraph of G induced by a vertex subset $M \subseteq V$. In particular, $\bar{G}_S$ is the subgraph of $\bar{G}$ induced by S. Let us denote by $d(A)$ the cost of A and by $mst(G)$ the cost of a minimum spanning tree (MST) of G. The famous *MST algorithm* approximates $T_{\min}$ with an MST of $\bar{G}_S$ with edges replaced by the corresponding shortest paths in G. Kou *et al.* [3] showed that

$$mst(\bar{G}_S)/d(T_{\min}) \le 2. \tag{1.1}$$

They proved that 2 is the supremum for the left ratio in (1.1). There are some other approximation algorithms for the Network Steiner Problem (see the survey [5]), but none is proven to have a better approximation ratio $d(T_{\text{approximate}})/d(T_{\min})$.

Mehlhorn [4] and Kou [2] proposed a faster implementation of the MST algorithm which computes the approximate tree in time $O(|V| \log|V| + |E|)$.

In this paper we present an 11/6-approximation algorithm with the implementation time $O(|V|\,|E| + |S|^4)$. This algorithm improves the MST algorithm by taking into consideration some additional vertices outside S.

[1] Institute of Mathematics, Grosul 5, Kishinev 277028, Moldova.

Received October 29, 1990; revised May 15, 1991. Communicated by Christos H. Papadimitriou.

2. The Approximation Algorithm. Some preliminary definitions: Given a metricly closed graph $G = (V, E, d)$, we may *contract* an edge e, i.e., reduce its length to 0. We use $G[e] = (V, E, d')$ to denote the resulting graph. For any triple $z = \{u, v, w\}$ the graph $G[z]$ equals $G[(u, v)][(v, w)]$, i.e., it results from two contractions.

Our algorithm goes as follows

ALGORITHM 2.1.

```
(1) $F \leftarrow \bar{G}_S$; $W \leftarrow \varnothing$; $Triples \leftarrow \{z \subset S: \#z = 3\}$
(2) For every $z \in Triples$ do
        find $v$ which minimizes $\sum_{s \in z} d(v, s)$
        $v(z) \leftarrow v$
        $d(z) \leftarrow \sum_{s \in z} d(v(z), s)$
(3) repeat forever
      (a) find $z \in Triples$ which maximizes $win = mst(F) - mst(F[z]) - d(z)$
      (b) if $win \leq 0$ then exit repeat
      (c) $F \leftarrow F[z]$; insert $(W, v(z))$
(4) Find a Steiner tree for $S \cup W$ in graph $G$ using the MST algorithm.
```

In a few words, the algorithm presented (see step (3)) finds, as long as possible, the best reduction of the cost of MST of F, which initially coincides with $\bar{G}_S$, by adding to an MST of F three G-edges with a common end and removing the longest edges from each resulting cycle. After the triple contraction we get the next F.

The main result of this paper is the following:

THEOREM 2.2. $mst(\bar{G}_{S \cup W})/d(T_{\min}) \leq 11/6$.

In the next section we investigate the quality of the greedy approximation of the best possible reduction of MST cost by triples. We show that the approximation ratio equals 2. Section 4 is devoted to the approximation of the minimum Steiner tree by the tree which can be obtained by the best possible reduction of MST cost by triples. We show that in this case the approximation ratio is no more than 5/3. These two estimates imply the statement of Theorem 2.2. In the last section a faster implementation (in time $O(|V||E| + |S|^4)$) of Algorithm 2.1 is given.

3. The Win of a Greedy Sequence of Triples. Let $F = \langle V, E, d_F \rangle$ be a graph and let d be a cost function defined on the set of triples of vertices. For a set Z of triples, $d(Z)$ is the sum of costs of its elements.

For a set A consisting of edges and triples we define $F[A]$ recursively: $F[\varnothing] = F$ and $F[A \cup e] = F[A][e]$. (For brevity, we denote a singleton $\{x\}$ as x.) For a set Z of triples, we define $win_F(Z) = mst(F) - mst(F[Z]) - d(Z)$.

Let $z_1, \ldots, z_m$ be a sequence of triples. We say that this sequence is *greedy in F* if it satisfies the following conditions:

if $win_F(z) \leq 0$ for every triple z, then $m = 0$,
otherwise, $win(z_1) \geq win(z)$ for every triple z;
the sequence $z_2, \ldots, z_m$ is greedy in $F[z_1]$.

In this section we prove the following:

THEOREM 3.1. *If H is the set of elements of a greedy sequence of triples, then, for every set of triples Z,*

$$2win_F(H) \geq win_F(Z). \tag{3.1}$$

PROOF. For an edge e of F define $save_F(e) = mst(F) - mst(F[e])$.

LEMMA 3.2. *Let $F_{\leq x}$ be a graph with the same vertices and edges as F, except that the edges of length larger than x are removed. Then $save_F(e)$ is the minimal value x such that both ends of e are in the same component of $F_{\leq x}$.*

PROOF. Let x be defined as above. Consider an MST of $F[e]$ which contains e, say T. $T - e$ has two connected components, each containing one of the ends of e. By definition, there exists a path in F which connects ends of e together, such that all its edges have length at most x. One edge on this path, say a, must cross from one component to the other, thus $T \cup a - e$ is a spanning tree of F with cost at most $mst(F[e]) + x$.

On the other hand, consider T', an MST of F, and the unique path in T' which connects the ends of e. By definition, one edge on this path must have cost at least x. Removing this edge from T' and inserting the contracted edge e results in a spanning tree of $F[e]$ with cost at most $mst(F) - x$. □

LEMMA 3.3. *For every set of edges A and any edge e, we have*

$$save_{F[A]}(b) \leq save_F(b).$$

PROOF. The proof follows directly from the above characterization, because the set of edges $F[A]_{\leq x}$ contains the set of edges $F_{\leq x}$. □

LEMMA 3.4. *Let Z be a set of triples. Then, for every edge e, either $win_{F[e]}(Z) = win_F(Z)$ or there exists $z \in Z$ such that*

$$win_{F[e]}(Z - z) \geq win_{F[z]}(Z - z). \tag{3.2}$$

PROOF. Let T be an MST of F. The unique path P in T which connects the ends of e contains an edge a such that $d_F(a) = save_F(e)$. Let U and W be the two connected components of $T - a$. Consider now T', an MST of $F[Z]$. On the unique path P' in T' connecting the ends of e at least one edge, say b, connects U and W. The choice of b assures $save_F(b) = save_F(b)$.

Assume, first, that $d_{F[Z]}(b) = d_F(b)$. Then $save_{F[Z]}(e) \geq d_{F[Z]}(b) \geq save_F(e)$, consequently $save_{F[Z]}(e) = save_F(e) = c$. As a result,

$$\begin{aligned} win_{F[e]}(Z) &= mst(F[e]) - mst(F[e \cup Z]) - d(Z) \\ &= [mst(F) - c] - [mst(F[Z]) - c] - d(Z) \\ &= win_F(Z). \end{aligned}$$

Now we may assume that $d_{F[Z]}(b) < d_F(b)$. In this case $b \subset z$ for some triple $z \in Z$. We will show that such a z satisfies (3.2).

Because $save_F(b) = save_F(e)$, we know that $mst(F[e]) = mst(F[b])$. This allows us to rewrite the left-hand side of (3.2) as follows:

$$\begin{aligned} win_{F[e]}(Z - z) &= mst(F[e]) - mst(F[e \cup Z - z]) - d(Z - z) \\ &= mst(F[b]) - mst(F[Z - z][e]) - d(Z - z). \end{aligned}$$

Without loss of generality we may assume that $z = b \cup c$, where c is an edge which belongs to $T' - P'$. Thus removing b and c from T' creates three components, and two of them can be connected either by b or by e. This shows that $save_{F[Z-z]}(b) = save_{F[Z-z]}(e)$. We use this identity to rewrite the right-hand side of (3.2) as follows:

$$\begin{aligned} win_{F[z]}(Z - z) &= mst(F[z]) - mst(F[Z]) - d(Z - z) \\ &= mst(F) - save_F(b) - save_{F[b]}(c) - mst(F[Z]) - d(Z - z) \\ &= mst(F[b]) - save_{F[b]}(c) - mst(F[Z - z][b][c]) - d(Z - z) \\ &= mst(F[b]) - save_{F[b]}(c) - mst(F[Z - z][b]) \\ &\quad + save_{F[Z-z][b]}(c) - d(Z - z) \\ &= mst(F[b]) - save_{F[b]}(c) - mst(F[Z - z][e]) \\ &\quad + save_{F[Z-z][b]}(c) - d(Z - z). \end{aligned}$$

Thus it sufficies to prove that $save_{F[b]}(c) \geq save_{F[Z-z][b]}(c)$, which follows immediately from Lemma 3.2. $\square$

Now we can prove Theorem 3.1 by induction on $\#H$. If $H = \varnothing$, then $win_F(Z) \leq 0$. This follows from the fact that in this case, for every triple z, $win_F(z) \leq 0$ and $win_F(Z - z) = win_F(Z) - win_F(z)$.

In the inductive step, let h be the first element of a greedy sequence. For some two edges a and b we have $h = a \cup b$. By Lemma 3.4, there exists a subset Y of Z with at most two elements, such that

$$\begin{aligned} win_{F[h]}(Z) &= win_{F[a][b]}(Z) \geq win_{F[Y]}(Z - Y) \\ &= win_F(Z) - win_F(Y) \geq win_F(Z) - 2\, win_F(h). \end{aligned}$$

The last inequality follows trivially if Y is an empty or singleton set. If $Y = \{y, z\}$, then, by Lemma 3.3,

$$\begin{aligned} win_F(Y) &= win_F(y) + win_F(z) \\ &= win_F(y) + save_{F[y]}(z) - d(z) \leq win_F(y) + save_F(z) - d(z) \\ &= win_F(y) + win_F(z) \leq 2\, win_F(h). \end{aligned}$$

Note that $H - h$ is the set of elements of a greedy sequence in $F[h]$. By inductive hypothesis and inequality $win_{F[h]}(Z) \geq win_F(Z) - 2\,win_F(h)$, we may conclude that

$$2\,win_F(H) = 2\,win_{F[h]}(H - h) + 2\,win_F(h) \geq win_{F[h]}(Z) + 2\,win_F(h) \geq win_F(Z). \quad \square$$

4. The Set of Triples Induced by a Minimum Steiner Tree

LEMMA 4.1. *There is a set of triples Z such that*

$$3(mst(\bar{G}_S) - win(Z)) \leq 5d(T_{\min}). \tag{4.1}$$

PROOF. Given an instance of the Steiner tree problem and the respective minimum Steiner tree, we transform this instance by replicating certain vertices, so that the distance between copies of the same vertex are zero. If a group replicates a distinguished vertex, some of the copies must be distinguished. This way we can assure that a minimum Steiner tree $T_{\min}$ has the form of the complete binary tree, with the set of leaves S coinciding with the set of distinguished vertices.

Let r be the depth of $T_{\min}$ and let $B = \{0, 1\}$. We label vertices of $T_{\min}$ with words from $B^* = \{w \in B^* \colon |w| \leq r\}$: λ is the root, while w has children $w0$, $w1$. We use L to denote the set of labels of inner vertices of $T_{\min}$.

We recursively define ρ_w to be 0 if w is a leaf, and $d(w, w0) + \rho_{w0}$ otherwise. We relabel vertices in such way that, for every w, we have $\rho_w \leq d(w, w1) + \rho_{w1}$. This assures that ρ_w is the length of the shortest tree path from w to a leaf in its subtree, and $s_w = w0^{r-|w|}$ is such a leaf.

For $w \in L$ we define edge $e_w = \{s_{w0}, s_{w1}\}$ and we use d_w to denote $d(e_w)$. Note that $T_S = \{e_w \colon w \in L\}$ is a spanning tree of S. We will show that we can sufficiently "improve" the tree T_S using triples of the form

$$z_{wb} = e_{wb} \cup e_w = \{s_{wb0}, s_{wb1}, s_{w\bar{b}}\}.$$

Triple z_{wb} has a minimum Steiner tree which has cost $d_{wb} + d_w - save_{wb}$. Figure 1 shows that this cost can be estimated as follows:

$$d_{wb} + d_w - save_{wb} \leq h_{wb} + h_w + \rho_{wb0} + \rho_{wb} + \rho_{w\bar{b}}, \tag{wb}$$

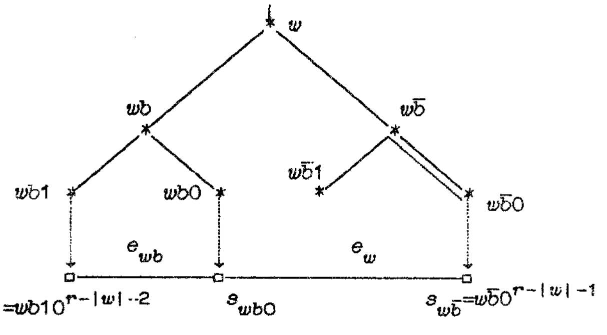

Fig. 1. The estimated cost of the minimum Steiner tree of triple z_{wb}.

where h_w is a shorthand for $d(w, w0) + d(w, w1)$. By adding inequalities (wb) for all $wb \in L - \lambda$ we obtain

$$(4.2)\qquad 3 \sum_{w \in L} d_w - d_\lambda - \sum_{w \in L - \lambda} save_w \leq 3 \sum_{w \in L} h_w - h_\lambda + 2 \sum_{w \in L} \rho_w - 2\rho_\lambda.$$

Observe that

$$(4.3)\qquad \sum_{w \in L} h_w = d(T_{\min}).$$

We already know

$$(4.4)\qquad \sum_{w \in L} d_w = d(T_S).$$

Moreover, we can prove by induction that $\sum_{w \in L \cap uB^*} \rho_w \leq \sum_{w \in L \cap uB^*} h_w - \pi_u$, where π_u is the length of the tree path from u to $u1^{r-|u|}$ which is the longest path from u to a leaf in its subtree. Therefore

$$(4.5)\qquad \sum_{w \in L} \rho_w \leq d(T_{\min}) - \pi_\lambda$$

and

$$(4.6)\qquad d_\lambda \leq 2\pi_\lambda.$$

Inequalities (4.2)–(4.6) yield

$$(4.7)\qquad 3d(T_S) - \sum_{w \in L - \lambda} save_w \leq 5d(T_{\min}).$$

Given wb, we can replace, in T_S, edges e_{wb} and e_w by a minimum Steiner tree of z_{wb}, this subtracts $save_{wb}$ from the tree cost. If we perform such replacement for every $wb \in A$, we subtracts $\sum_{w \in A} save_w$ from the tree cost, provided that no such replacements affect the same edge. In this case we say that set A is legal.

Consider the following graph: vertices form set $L - \lambda$, edges are of the form $\{w, w0\}$, $\{w, w1\}$, and $\{w0, w1\}$. It is easy to see that A is legal if and only if A is an independent set in this graph. It is also easy to see that this graph is 3-colorable, thus $L - \lambda$ can be partitioned into three legal sets, say A_1, A_2, and A_3. Let $Z_i = \{z_w : w \in A_i\}$. From the above remarks we can conclude that

$$(4.8)\qquad \sum_{i=1}^{3} (mst(\bar{G}_S) - win(Z_i)) \leq 3d(T_S) - \sum_{w \in L - \lambda} save_w.$$

Inequalities (4.7) and (4.8) imply that one of the sets Z_1, Z_2, Z_3 satisfies the claim of the lemma. $\square$

Proof of Theorem 2.2. To prove this theorem it is enough to write the sequence of the following inequalities for a greedy sequence of triples H:

$$\begin{aligned} 6\, mst(\bar{G}_{S\cup W}) &\le 6(mst(\bar{G}_S) - w(H)) = 3\ mst(\bar{G}_S) + 3[mst(\bar{G}_S) - 2w(H)] \\ &\overset{(3.1)}{\le} 3\ mst(\bar{G}_S) + 3[mst(\bar{G}_S) - w(Z)] \overset{(4.1)}{\le} 3\ mst(\bar{G}_S) + 5d(T_{\min}) \\ &\overset{(1.1)}{\le} 6d(T_{\min}) + 5d(T_{\min}) = 11d(T_{\min}). \end{aligned}$$

□

5. The Faster Approximation Algorithm. The *Voronoi region* $V(s) \subseteq V$ of a vertex $s \in S$ is the set of vertices which are nearer to s than to any other vertex of S. If the distances between a vertex and several vertices of S are the same, then we assign this vertex to exactly one Voronoi region. The vertex set of G is partitioned by Voronoi regions into disjoint subsets $V = \bigcup\{V(s) \mid s \in S\}$.

In our algorithm each triple z is considered together with its *center* $v(z)$. We say that a pair of the form $(z, v(z))$ is a *star* and elements of z are *ends* of this star. Consequently, we view a greedy sequence of triples as a greedy sequence of stars.

Lemma 5.1. *There is a greedy sequence of stars in which every center belongs to the Voronoi region of one of its star ends.*

Proof. Let $F = \bar{G}_S$. Assume that we have already greedily selected stars forming set H, that (z, v) maximizes $win_{F[H]}(z)$, and that $win_{F[H]}(z) > 0$. Consider an MST of $F[H \cup z]$, say T. Because the win of z is positive, T contains two edges which span z. After we remove these edges from T, the set S splits into three connected components, where each contains a different element of z. Assume that $v \in V(s)$. Let w be the element of z which is in the same component as s and let $z' = (z - w) \cup s$. It is easy to see that $mst(F[H \cup z']) = mst(F[H \cup z])$. Moreover, $d(z', v) = d(z, v) - d_{\bar{G}}(v, w) + d_{\bar{G}}(v, s)$ which cannot be larger than $d(z, v)$. Hence (z', v) is also a valid greedy choice, while $v \in V(s)$ and $s \in z$. □

The following procedure finds the save matrix of a Steiner tree T in time $O(|S|^2)$.

```
findsave($T$)
  if $T \neq e$ then
        $e \leftarrow$ an edge of $T$ with maximum length; $x \leftarrow d(e)$
        $T_1, T_2 \leftarrow$ the connected components of $T - e$
        for each vertex $v_1$ of $T_1$ and each vertex $v_2$ of $T_2$ do
        $save(v_1, v_2) \leftarrow x$
        findsave($T_1$); findsave($T_2$)
```

Algorithm 5.2.

(1) $F \leftarrow \bar{G}_S$: $W \leftarrow \varnothing$; $Triples \leftarrow \{z \subset S\colon \#z = 3\}$
(2) For every $s \in S$ find Voronoi region $V(s)$

```
(3) For every $z \in Triples$ do
        find $v \in \bigcup_{s \in z} V(s)$ which minimizes $\sum_{s \in z} d(v, s)$
        $v(z) \leftarrow v$
        $d(z) \leftarrow \sum_{s \in z} d(v(z), s)$
(4) repeat forever
    (a) $T \leftarrow$ an MST of $F$; findsave($T$)
    (b) find $z \in Triples$ which maximizes

            $win = \max_{e \subset z} save(e) + \min_{e \subset z} save(e) - d(z)$

    (c) if $win \leq 0$ then exit repeat
    (d) $F \leftarrow F[z]$; insert($W$, $v(z)$)
(5) Find a Steiner tree for $S \cup W$ in graph $G$ using the MST algorithm.
```

By Lemma 5.1, Algorithms 5.2 and 2.1 give the same trees.

THEOREM 5.3. *The implementation time of Algorithm* 5.3 *is* $O(|V|\,|E| + |S|^4)$.

PROOF. The following implementation time bounds for the steps of Algorithm 5.2 are already known or obvious:

(1) $O(|V|\,|E| + |S|^3)$;
(2) $O(|E|)$ (see [4]);
(3) $O(|V|\,|S|^2)$;
    (a) $O(|S|^2)$;
    (b) $O(|S|^3)$ and, therefore,
(4) $O(|S|^4)$;
(5) $O(|V| \log|V| + |E|)$. □

Acknowledgments. I am deeply indebted to the referee whose suggestions considerably improved the final version of this paper. I would also like to thank Professors P. S. Soltan, V. D. Chepoi, and F. F. Dragan for helpful discussions of this work.

References

[1] R. M. Karp, Reducibility among combinatorial problems, in: R. E. Miller and J. W. Tatcher, eds., *Complexity of Computer Computation*, Plenum, New York, 1972, pp. 85–103.

[2] L. Kou, A faster approximation algorithm for the Steiner problem in graphs, *Acta Inform.*, **27** (1990), 369–380.

[3] L. Kou, G. Markowsky, and L. Berman, A fast algorithm for Steiner trees, *Acta Inform.*, **15** (1981), 141–145.

[4] K. Mehlhorn, A faster approximation algorithm for the Steiner problem in graphs, *Inform. Process. Lett.*, **27** (1988), 125–128.

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