Report Number: WUCS-88-13

1988-01-01

Worst Case Performance of Rayward-Smith's Steiner Tree Heuristic

Authors: Bernard M. Waxman and Makoto Imase

In this paper, we prove that the worst case performance of the Steiner tree approximation algorithm by Rayward-Smith is within two times optimal and that two is the best bound in the sense that there are instances for which RS will do worse than any value less than two.

WORST CASE PERFORMANCE OF RAYWARD-SMITH'S STEINER TREE HEURISTIC

Bernard M. Waxman and Makoto Imase

WUCS-88-13

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

Abstract

In this paper, we prove that the worst case performance of the Steiner tree approximation algorithm by Rayward-Smith is within two times optimal and that two is the best bound in the sense that there are instances for which RS will do worse than any value less than two.

This work supported by Bell Communications Research, Italtel SIT, and the National Science Foundation (DCI-8600947).

Makoto Imase is with NTT Software Laboratories. The work described here was performed while on leave at Washington University.

WORST CASE PERFORMANCE OF RAYWARD-SMITH'S STEINER TREE HEURISTIC

Bernard M. Waxman and Makoto Imase

1. Introduction

We prove that the worst case performance of the approximation algorithm for the Steiner tree problem in graphs due to Rayward-Smith (RS) [4] is within two times optimal and that its performance can be as bad as $2-\epsilon$ for any $\epsilon > 0$. In the Steiner problem we are given an undirected graph $G(V,E)$, a cost function $C : E \rightarrow \Re^+$, and $D \subseteq V$. (Throughout this paper we assume that G is connected.) We are asked to find a minimum cost spanning tree for the set D, where cost of a tree is defined in the obvious way.

The Steiner tree problem is NP-complete [3] and no polynomial time approximation algorithm is known [8] to have worst case performance that is bounded by $2-\epsilon$ times the cost of the minimum Steiner tree, for $\epsilon > 0$. RS is of particular interest in light of experimental studies [5,6] and analysis of probabilistic performance [7] in which RS compares favorably to several well known algorithms.

We give a brief description of RS referring the reader to [4,5] for a more detailed description. Construct a collection of single node trees $\mathcal{T}$ consisting of the nodes in D. Then repeat the following step for $1 \leq i \leq |D|-1$ until there is only one tree.

Let v be a vertex with smallest $f(v)$ where

$$f(v) \quad = \quad \min_{S \subseteq \mathcal{T},\, |S|>1} \left\{ \frac{1}{|S|-1} \sum_{T \in S} \operatorname{dist}(v,T) \right\}.$$

Let T_1 and T_2 be two closest trees to v.
Join T_1 and T_2 by a shortest path through v.

Informally, f is the average cost of making r joins to $r+1$ trees through a node v, where $r+1 = |S|$.

We make use of a second algorithm, which we refer to as the *minimum spanning tree* heuristic (MST). MST begins by constructing a complete graph $G[D]$ on D, where the distances between nodes in $G[D]$ correspond to the distances in G. MST constructs a minimum spanning tree for $G[D]$ using one of the standard algorithms. Finally MST translates each edge of this tree into a path in G to produce a solution. MST will produce a spanning tree for $G[D]$ which has cost within $2(|D|-1)/|D|$ times the cost of a *minimum Steiner tree*. For more details on MST and a proof of this bound see [1,2].

2. An Upper Bound on Worst Case Performance

We define a collection of algorithms $\{RS_k |\ 0 \leq k \leq |D|-1]\}$ such that RS_0 is equivalent to MST and $RS_{|D|-1}$ is equivalent to RS. In the following specification of RS_k let $V(p)$ be the set of nodes in graph p.

Create a collection of single node trees $\mathcal{T}$ consisting of the nodes in D.
do $|\mathcal{T}| > |D| - k \rightarrow$
Choose $v \in V$ such that $f(v)$ is minimum where

$$f(v) \quad = \quad \min_{S \subseteq \mathcal{T},\, |S|>1} \Big\{ \frac{1}{|S|-1} \sum_{T \in S} \mathrm{dist}(v, T) \Big\}.$$

Join two trees T_1 and T_2 closest to v by a shortest path through v.
od
$D_k := \bigcup_{T \in \mathcal{T}} V(T)$
do $|\mathcal{T}| > 1 \rightarrow$
Choose two trees T_1 and T_2 in $\mathcal{T}$ such that $g(T_1, T_2)$ is minimum where

$$g(T_1, T_2) \quad = \quad \min_{u \in V(T_1) \cap D_k,\, v \in V(T_2) \cap D_k} \mathrm{dist}(u, v)\ .$$

Join the trees T_1, T_2 by a shortest path with its endpoints in D_k.
od

Note that the function g in the second **do** statement is equivalent to f for $|S| = 2$ if the endpoints are restricted to nodes in D_k. We also note that RS_k does not specify what action to take in case more than one choice is possible at some point. For simplicity we initially assume that all choices are deterministic.

For an instance (G, C, D) of the Steiner problem let $MST(G, C, D)$ equal the cost of a minimum spanning tree produced by MST on $G[D]$, $RS(G, C, D)$ the cost of a solution produced by RS,and let

$$RS_k(G, C, D) \;=\; \sum_{i=1}^{|D|-1} \mathrm{cost}(p_{k,i})$$

where $\{p_{k,i} | 1 \le i \le b-1\}$ are the paths selected by RS_k. Clearly $RS_k(G, C, D) \ge RS(G, C, D)$ for $k = b-1$. We will prove that $RS(G, C, D) \le MST(G, C, D)$ by proving that $RS_k(G, C, D) \le MST(G, C, D)$.

Lemma 2.1 *For an instance (G, C, D) of the Steiner problem let $\{p_{k,i} | \, 1 \le i \le b-1\}$ be the sequence of paths generated by RS_k. Then $\mathrm{cost}(p_{k,j}) \le \mathrm{cost}(p_{k,j+1})$ for $k < j < b-1$.*

Proof: After the selection of path $p_{k,k}$ all paths selected by RS_k have their endpoints in D_k. Therefore, all possible path choices for the remaining steps are known with RS_k selecting paths of least cost.
□

Lemma 2.2 *Let $G(V, E)$ be a graph with subgraphs H and H' consisting of γ and γ' components, respectively. Let $P = \{p_1, p_2, \ldots p_{\gamma-1}\}$ be a collection of paths such that $H \cup (\bigcup_{p \in P} p)$ is a connected graph and each path in P has its endpoints and only its endpoints in H. If H is a subgraph of H' then there exists some $P' \subseteq P$ with $|P'| = \gamma' - 1$ such that each $p \in P'$ has its endpoints in two distinct components of H'.*

Proof: We prove this lemma by giving an algorithm for constructing a set P'. Let $C = \{c_1, c_2, \ldots c_{\lambda'}\}$ be the components of H', let $S = \{c_1\}$, and initialize $P' = \emptyset$.

do $S \neq C \rightarrow$
  select a path $p \in P$ such that p has one endpoint in some $c \in S$
    and the other the other endpoint in some $c' \in C - S$
  $S := S \cup \{c'\}$
  $P' := P' \cup \{p\}$
od

This algorithm will terminate after selecting $\gamma' - 1$ distinct paths from P since $H \cup (\bigcup_{p \in P} p)$ is connected.
□

Theorem 2.1 *For all instances (G, C, D) of the the Steiner tree problem in graphs*

$$RS(G, C, D) \;\le\; MST(G, C, D)\,.$$

Thus, $RS(G, C, D)/OPT(G, C, D) \le 2(|D| - 1)/|D|$, where OPT(G,C,D) is the cost of a minimum Steiner tree.

Proof: Let $b = |D|$, $\{p_{j,i} \mid 1 \leq i \leq b-1\}$ be the set of paths selected by RS_j in sequence, and the phrase *step i* indicate the loop in which RS_j selects the i^{th} path $p_{j,i}$. Since RS_0 is identical to MST and $RS(G,C,D) \leq RS_{b-1}(G,C,D)$ it is sufficient to show that

$$(\forall k,\ 0 < k \leq b-1)\ (RS_k(G,C,D) \leq RS_{k-1}(G,C,D))\,. \tag{1}$$

For $0 < k \leq b-1$ the first $k-1$ paths selected by RS_{k-1} and RS_k are identical. Therefore,

$$\sum_{i=1}^{k-1} \text{cost}(p_{k,i}) \;=\; \sum_{i=1}^{k-1} \text{cost}(p_{k-1,i})\,. \tag{2}$$

Let x be the node selected by RS_k in step k with $r_k = |S|-1$. From step k through step $m_k = k+r_k-1$, RS_k will select paths that have a total cost no more than $C_k = r_k f(x)$. By the definition of RS_k and lemma 2.1 we know that $f(x) \leq \text{cost}(p_{k-1,i})$, $k \leq i \leq b-1$. Thus

$$C_k \;\leq\; \sum_{i=k}^{m_k} \text{cost}(p_{k-1,i})\,. \tag{3}$$

Let $P = \{p_{k-1,k}, p_{k-1,k+1}, \ldots p_{k-1,b-1}\}$ and let H_i be the intermediate solution generated by RS_k at the completion of step i, i.e. the graph consisting of all trees in $\mathcal{T}$ at this point. Clearly, $H_{k-1} \cup P$ is a connected graph, each path in P has its endpoints and only its endpoints in H_{k-1}, and H_{k-1} is a subgraph of H_i for $m_k < i \leq b-1$. Therefore, we can apply lemma 2.2 to show the existence of a set $P' \subset P$, $|P'| = b-1-i$ where the endpoints of each path in P' are in two distinct components of H_i. Then $\text{cost}(p_{k,i}) \leq \min_{p \in P'} \text{cost}(p)$ since the paths in P' are possible choices for RS_k in step i. Applying lemma 2.1 we have

$$\Big(\forall i,\ m_k < i \leq b-1\Big)\Big(\text{cost}(p_{k,i}) \;\leq\; \text{cost}(p_{k-1,i})\Big)\,. \tag{4}$$

Finally combining (2), (3), and (4) we have $RS_k(G,C,D) \leq RS_{k-1}(G,C,D)$.
□

In those cases where RS_k has more than one choice at some point we make an arbitrary selection among the available possibilities. We modify our definition of $RS_k(G,C,D)$ so that

$$RS_k(G,C,D) \;=\; \text{lub}\left\{\sum_{i=1}^{|D|-1} \text{cost}(p_{k,i})\right\}$$

over all possible path sets generated by RS_k. As long as RS_k and RS_{k-1} make the same choices through step $k-1$ the proof of theorem 2.1 still holds with complete generality.

3. A Worst Case Example

Given any $\epsilon > 0$ we show that there exists an instance (G,C,D) (see figure 1) of the Steiner problem such that $RS(G,C,D)/OPT(G,C,D) \;>\; 2-\epsilon$. G consists of a

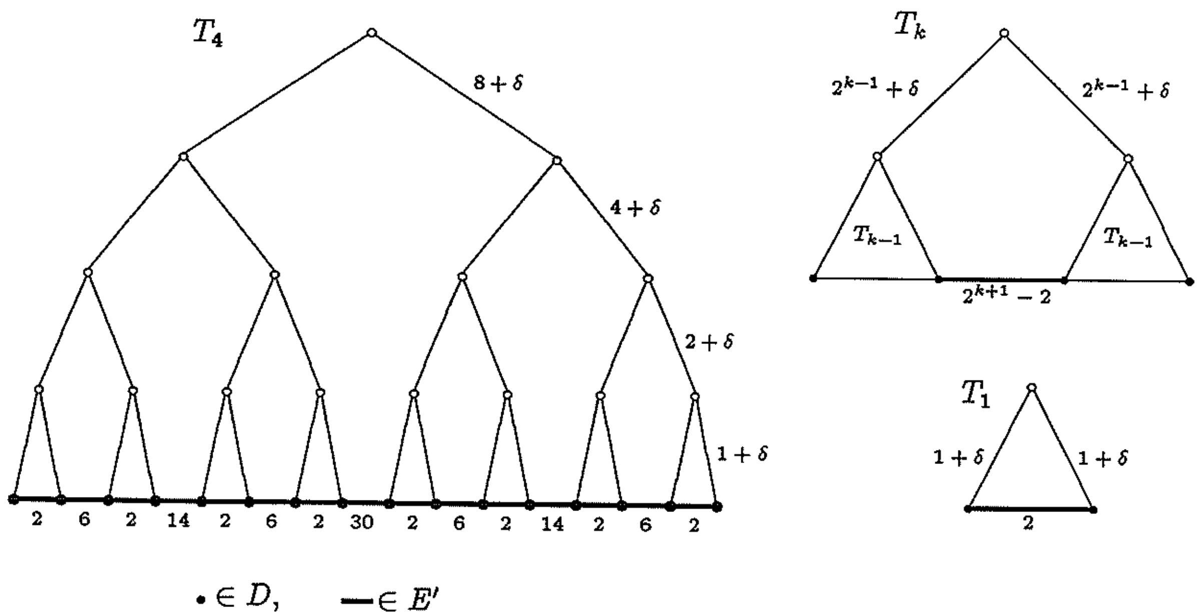

Figure 1: A worst case example for *RS*

spanning tree T plus a set of $2^k - 1$, $k \in Z^+$ additional edges $E' = \{e_1, e_2, \ldots e_{2^k-1}\}$. To construct G we begin with the root of T at level $l = k$. Join a left and a right subtree to the root each by an edge of cost $2^{l-1} + \delta$, $\delta > 0$. Repeat this step recursively for each subtree at level $l := l - 1$ until reaching the subtrees at level 0. The level 0 subtrees are just the leaves of T. Label the leaf nodes with the numbers 1 through 2^k from left to right. For each pair of nodes i and $i+1$, $1 \le i < 2^k$ add an edge e_i. For each edge e_i with $i \equiv 2^{j-1} \pmod{2^j}$, where $0 < j \le k$, set the edge cost to $2^{j+1} - 2$. Since $i \equiv 2^{j-1} \pmod{2^j}$ implies that $i \equiv 0 \pmod{2^{j'}}$ for any $j' < j$, this mapping is well defined.

If we let D be the set of leaf nodes of T, then the solution generated by RS will be the tree $H(D, E')$. We can easily show, for a fixed k, that

$$\begin{aligned} RS(G,C,D) &= \sum_{j=1}^{k} 2^{k-j}(2^{j+1} - 2) \\ &= 2\sum_{j=1}^{k}(2^k - 2^{j-1}) \\ &= 2(k2^k - 2^k + 1)\,. \end{aligned}$$

The *minimum Steiner tree* will just be the tree T. Therefore,

$$\begin{aligned} OPT(G,C,D) &= \sum_{j=1}^{k} 2^j(2^{k-j}+\delta) \\ &= k2^k + (2^{k+1}-1)\delta\,. \end{aligned}$$

We then have that

$$\begin{aligned} \frac{RS(G,C,D)}{OPT(G,C,D)} &= \frac{2}{1+(2^{k+1}-1)\delta/k2^k} - \frac{2^{k+1}-2}{k2^k+(2^{k+1}-1)\delta} \\ &> \frac{2}{1+2\delta/k} - \frac{2}{k}\,. \end{aligned}$$

Thus, given any $\epsilon > 0$ there exists $k \in Z^+$ such that $RS(G,C,D)/OPT(G,C,D) > 2-\epsilon$ for any fixed $\delta > 0$.

References

[1] K. Bharath-Kumar and J.M. Jaffe. "Routing to multiple destinations in computer networks." *IEEE Transactions on Communications*, COM-31(3):343–51, March 1983.

[2] L. Kou, G. Markowsky, and L. Berman. "A fast algorithm for Steiner trees." *Acta Informatica*, 15:141–5, 1981.

[3] R.E. Miller and J.W. Thatcher. *Complexity of Computer Computations*, chapter Reducibility among combinatorial problems - by R.M. Karp, pages 85–103. Plenum Press, 1972.

[4] V.J. Rayward-Smith. "The computation of nearly minimal Steiner trees in graphs." *International Journal of Math. Ed. Sci. Tech.*, 14(1):15–23, 1983.

[5] V.J. Rayward-Smith and A. Clare. "On finding Steiner vertices." *Networks*, 16:283–94, 1986.

[6] B.M. Waxman. "Thesis proposal: routing of multipoint connections." Washington University, Department of Computer Science, St. Louis, MO, WUCS-87-10, 1987.

[7] B.M. Waxman. "Probable performance of Steiner tree algorithms." Washington University, Department of Computer Science, St. Louis, MO, WUCS-88-4, 1988.

[8] P. Winter. "Steiner problem in networks: a survey." *Networks*, 17:129–67, 1988.