アルゴリズム 26－1
（1992. 3. 23）

辺重みが制限されているグラフのスタイナー木の近似

ハルダスソン、マグナス
東京工業大学工学部情報工学科

上野 修一　中尾 洋　梶谷 洋司
東京工業大学工学部電気・電子工学科

あらまし

グラフのスタイナー木問題に対する平均距離法と呼ばれる近似解法の近似比を解析する．完全グラフの辺の重みが2値 $\{\alpha, \beta\}$ であるとき，及び区間 $[\alpha, \beta]$ にあるときには，近似比のほぼ最適な上界と下界を証明する．この近似比は，他の解析されている近似アルゴリズムの近似比と比較して約 $1/e$ であることを示す．

Approximating Steiner trees in graphs with restricted weights

Magnús M. Halldórsson
Department of Computer Science
Tokyo Institute of Technology

Shuichi Ueno　Hiroshi Nakao　Yoji Kajitani
Department of Electrical and Electronic Engineering
Tokyo Institute of Technology

Abstract

We analyze the approximation ratio of the *average distance heuristic* for the Steiner tree problem on graphs, and prove nearly tight bounds for the cases of complete graphs with binary weights $\{\alpha, \beta\}$, or weights in the interval $[\alpha, \beta]$, where $\alpha \leq 2 \cdot \beta$. The improvement over other analyzed algorithms is a factor of about e.

1 Introduction

Given a graph with real-valued edge weights, and a subset of the vertices distinguished as terminals, the *Steiner tree problem* involves finding a tree of minimum weight that spans all terminal vertices. It has attracted a great deal of attention in recent decades, partly due to its natural application to minimizing the lengths of communication paths, for example in VLSI layout and telephone switching networks.

In this paper we consider a restriction of the problem when the network is a complete graph and the ratio between the smallest and the largest edge weight is small. We distinguish between two cases: *binary* weights, when the graph contains only two weights α or β, and the more general *interval* weights, when the edge weights fall in the interval $[\alpha, \beta]$. For both types, we restrict our attention in this paper to those cases where $\alpha \leq 2 \cdot \beta$.

The algorithm that we shall consider for approximately solving the Steiner problem is known as the Average Distance Heuristic (ADH), and was introduced by Rayward-Smith in [5]. Only later was it shown by Waxman and Imase [7] that the *performance ratio* of the algorithm, or the worst-case ratio between the length of the solution it generates to the length of the optimal Steiner tree, is asymptotically two. Empirical and average case results [6, 8] also indicate excellent performance in practice. Our line of work was started by Bern and Plassman [2] who considered the performance of ADH on the complete graph with binary weights 1 and 2, and proved a ratio of 4/3. They also showed that the problem was MAX SNP-hard, and thus unlikely to be approximable in polynomial time within arbitrary constant.

We first show that the Steiner problem is NP-complete on binary weighted graphs, for any $\alpha \neq \beta$. We then give several bounds on the performance of ADH on graphs with binary and interval weights. In both cases the precise bounds are shown to be of the form $1 + \frac{1}{ek} + \mathcal{O}(\frac{1}{k^2})$, where e is the basis of the natural logarithm, and the constants behind the lower order term are small. This improves on the $1 + \frac{1}{k}$ performance of other analyzed methods.

The paper is organized as follows. The Average Distance Heuristic is described in next section and basic definitions given. Section 3 contains the NP-completeness proof, and sections 4 and 5 contain the analysis of the performance of the heuristic in the binary and interval case, respectively. Section 6 describes evaluation of the formulas obtained in sections 4 and 5, closing off with a discussion of related problems and methods in the final section.

2 Definitions

We are given a weighted, undirected graph $G = (V, E)$, and a set S of *terminal* vertices. The elements of $V - S$ will be referred to as *optional* vertices. In the remainder of the paper, E will be assumed to be the complete graph $V \times V$. Let n denote $|S|$.

The objective of the Steiner tree problem is to connect the terminals using minimum total sum of edge weights. The ADH attacks the problem by repeatedly shrinking a terminal subset to a single terminal, choosing each subset greedily so as to minimize the average cost of connections spent on each reduced terminal. Let $AD(v, X)$ denote the average *distance* of vertex v to a set of terminals X, namely

$$AD(v, X) = \frac{\sum_{s \in X} d(v, s)}{|X| - 1}$$

where $d(v, s)$ denotes the weight of the edge (v, s). The algorithm chooses the pair (v, X) that minimizes $AD(v, X)$. The vertex v can either be a terminal in X, or an optional vertex. In the former case, we may assume without loss of generality that two vertices are reduced at a time. In the latter case, these vertices form a *star* centered at v.

Let α (β) denote the minimum (maximum) edge weight in the graph, respectively, and let k be a real value such that $\beta/\alpha = 1 + 1/k$. Let K denote $\lfloor k \rfloor$. Let p denote the number of optional nodes in an optimum Steiner tree.

Let $HEU(G)$ denote the size of the solution found by the algorithm on G (assuming worst-case tie-breaking), and $OPT(G)$ denote the size of the optimal Steiner tree for G. Let $r_k(G)$ denote the ratio of $HEU(G)$ to $OPT(G)$, and r_k denote the minimum such ratio over all graphs under consideration. We are interested only in the asymptotic ratios, as the size of the input grows, although the differences are minor. Hence, we ignore all terms that do not grow linearly with the size of the input.

Finally, let H_k denote the k-th harmonic number, $\sum_{i=1}^{k} 1/i$.

3 NP-completeness

Theorem 1 *The Steiner Tree problem on a complete graph with binary values is NP-complete for any choice of values $\alpha \neq \beta$.*

Proof. The problem is clearly in $\mathcal{NP}$. For $\beta \geq 2\alpha$, hardness has already been established, for instance in the MAX SNP-hardness proof of [2]. We shall prove it here for the remaining values of β and α. Let κ be the least integer such that $\beta/\alpha > 1 + 1/\kappa$. The proof is by a reduction from "Exact Cover by κ-Sets" [3, p.221].

We are given an input (C, X) to Exact Cover, consisting of a basis set X, and collection C of subsets of X of size κ each, for which the objective is to decide if there exists a subcollection C' of C of mutually disjoint sets whose union is the basis set X. From this we construct a network $G = (V, E, S)$ as follows.

The graph contains a terminal vertex for each element of the basis set, and an optional vertex for each set in C, with the vertices labelled accordingly. The edges with weight alpha are those between two optional vertices, as well as those between an optional vertex t and terminal vertex o such that the label of o contains the label of t. More formally,

$$wt(u,v) = \begin{cases} \alpha & \text{if } v \text{ and } u \in O \text{ OR} \\ & \text{if } u \in O,\ v \in T, \text{ and } l(v) \in l(u) \\ \beta & \text{otherwise} \end{cases}$$

We now claim that the weight of the optimal Steiner tree of the graph and the question whether the set system has an exact cover are have a strong relationship, namely that

$$OPT(G) = |T| \cdot \alpha + (q-1) \cdot \alpha = \alpha((\kappa+1) \cdot q - 1)$$
$$\text{iff} \quad (C, X) \text{ has an exact cover}$$

Assume first that (C, X) has a cover C'. Then if we consider the restriction of the graph to the alpha edges between the terminal set along with those optional vertices associated with C', this graph is connected and thus has a Steiner tree of size $(|T| + |C'| - 1) \cdot \alpha$. Moreover, if the cover is exact, the size of C' is only $|T|/\kappa = q$, proving the if part.

On the other hand, consider a Steiner tree with r beta edges. At least $n - r$ terminals must be connected via at least $(n-r)/\kappa$ optional vertices for a cost of $[(\kappa+1)q - 1]\alpha - (\kappa+1)/\kappa r\alpha + r\beta$. This is at most $[(\kappa+1)q-1]\alpha$ only if $r = 0$. Moreover, if n/κ optional vertices sufficed to cover the n terminals, then they form an exact cover by κ sets of the set system. ∎

4 Binary weights

When dealing with binary weights, there is a natural way of mapping the weighted problem on a complete graph to incomplete, unweighted graphs. The edges of this graph are the lightweight α edges only. We focus our attention on this graph G'. One useful observation is that no two terminals will be adjacent in a worst-case instance.

We can rephrase some of the notation in terms of this graph. A t-star is an optional vertex in G' adjacent to at least t terminal nodes. Also, p, defined as the number of optional nodes in an optimum Steiner tree, is equivalent to the size of the minimum dominating set of the unweighted graph.

In what follows, we start by generalizing both the upper and lower bound of [2] to arbitrary k. These bounds have been included here primarily for the basis they provide for further intuition, as well as for their simplicity. We then improve both bounds to functions that converge as k grows.

A lower bound

For each positive integer t, we construct the following graph named *t-rake*, and denoted R_t: A sequence of optional vertices are linked in a path, with t terminals hanging off each optional vertex as a leaf node. A 3-rake is shown in figure 1.

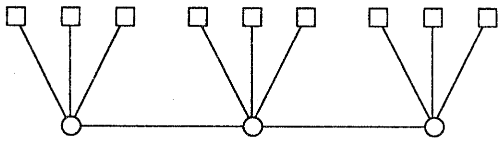

Figure 1: A short 3-rake

Theorem 2 *For k integral, $r_k \geq 1 + \dfrac{1}{k(k+2)}$*

Proof. For a t-rake, it is easy to verify that

$$HEU(R_t) = \beta(n-1)$$

when $t \leq \lfloor k \rfloor + 1$, and that

$$OPT(R_t) = \alpha \cdot \frac{t+1}{t} \cdot n$$

when $t \geq \lfloor k \rfloor + 1$. Now if we consider the $(k+1)$-rake,

$$\frac{HEU(R_{k+1})}{OPT(R_{k+1})} = \frac{\beta n}{\alpha n(k+2)/(k+1)} = \frac{k+1}{k}\frac{k+1}{k+2} = 1 + \frac{1}{k(k+2)}$$

∎

An upper bound

The following observations were made in [2] in a somewhat lesser generality.

Observation 1 *1. If G contains a $t{+}1$-star, then the algorithm makes progress towards $\frac{t+1}{t}$ ratio. More precisely, the $\frac{t+1}{t}$ ratio will be attained if ADH attains that ratio on the smaller graph equivalent to G with the $t{+}1$-star reduced to a single terminal.*

2. $OPT(G) \geq \alpha(n+p-1)$, where p is the number of optional vertices in the optimal solution.

3. If G contains no $t+1$-star, then $p \geq \lceil n/t \rceil$.

4. If the minimum tree contains q t-stars, then at least $2q$ nodes will be covered in t-reductions by the heuristic.

A simple application of observation 1 part 1-3 yields the following bound.

Lemma 1 $r_k \leq 1 + \frac{1}{\lceil 2k \rceil}$

Proof. Assume G contains no $t+1$-star. Then $OPT(G) \geq \alpha(n+p-1) \geq \alpha(n+n/t-1)$, while $HEU(G) \leq \beta(n-1)$. Hence for $t = \lceil 2k \rceil$, $r_k(G) \leq \frac{1+1/k}{1+1/\lceil 2k \rceil} = 1 + \frac{\lceil 2k \rceil - k}{k(\lceil 2k \rceil + 1)} \leq 1 + \frac{1}{\lceil 2k \rceil + 1}$.

On the other hand, if G does contain a $\lceil 2k \rceil + 1$-star, then we can make progress towards a $1 + \frac{1}{\lceil 2k \rceil}$ ratio, by observation 1.1. ∎

By counting just how many $t+1$-stars the graph contains, and showing that the adversary maximizes the ratio when that number is zero, we can strike a better balance between the two options of the previous proof.

Assume the graph contains no $t+1$-stars. Let q denote the number of (disjoint) t-stars in an optimal solution. Then $n \leq qt + (p-q)(t-1) = p(t-1) + q$, and thus $p \leq \frac{n-q}{t-1}$.

From the above, and observation 1.2,

$$OPT(G) \geq \alpha(n+p) \geq \alpha(n+\frac{n-q}{t-1}) = \alpha\frac{t}{t-1}(n-\frac{q}{t})$$

Let m be the number of t-reductions or larger, and let s be the number of nodes reduced in them. Then $m \leq s/t$. Moreover, since at least 2 nodes from a given star must be reduced in order to decrease the star, we have that $s \geq 2q$ (see observation 1.4).

$$\begin{aligned} HEU(G) &\leq \beta(n+m-s) + \alpha s \\ &= \alpha\frac{k+1}{k}(n+m-s(1-\frac{k}{k+1}) \\ &\leq \alpha\frac{k+1}{k}(n+\frac{s}{t}-\frac{s}{k+1}) \\ &\leq \alpha\frac{k+1}{k}(n-2q(\frac{1}{k+1}-\frac{1}{t})) \end{aligned}$$

We have that the ratio between the two is at most $\frac{k+1}{k} \cdot \frac{t-1}{t}$, as long as $2q(\frac{1}{k+1} - \frac{1}{t}) \geq \frac{q}{t}$, which is satisfied when $t \geq \frac{3}{2}(k+1)$.

If we now set $t = \lceil 2k \rceil + 1$, the ratio is bounded by $\frac{k+1}{k}\frac{\lceil 2k \rceil}{\lceil 2k \rceil + 1} = 1 + \frac{\lceil 2k \rceil - k}{k(\lceil 2k \rceil + 1)} \leq 1 + \frac{1}{2k+1}$. Finally, recall that if there is a $(t+1)$-star, the same bound holds (obs. 1.2). Thus we obtain the following theorem.

Theorem 3 $r_k \leq 1 + \frac{1}{\lceil 2k \rceil + 1}$

An improved lower bound

Construct a graph $T_{k,R}$, for an integer R, as an adjustment of the $\lfloor k \rfloor + 1$-rake in the following way. Let each center node (an internal node in the "spine" of the rake) cover R terminals. Let the extra $R - (\lfloor k \rfloor + 1)$ terminals be connected to some additional optional vertices as follows: A specific terminal node X is adjacent to every new optional vertex. Ensure that in all cases, the next star will have a) no more than one terminal from any center node (except possibly X), b) the node X, and c) exactly as many terminals as the currently largest star has.

This is constructible and has the property that the sizes of the reductions will slowly decrease from R down to $\lfloor k \rfloor + 2$. In particular, there will be about p/t reductions of size $t{+}1$, except for size R for which there will be $2p/(R-1)$.

The upper part of figure 2 shows the graph $T_{1,3}$ as an example. The additional optional vertices are labeled with the round in which they will be reduced (according to the reduction order we have chosen). The lower half is a snapshot after all the star reductions have taken place. The heuristic cost of each star in this particular example, for $k = 1$, is $4/3 + 4/3 + 2 = 14/3$, versus the optimal cost of 4, making this a poor $7/6$ lower bound. In fact, this type of construction can not attain the optimal bound for $k = 1$, but as we can see, is quite strong in the more general case.

$$\begin{aligned} & HEU(T_{k,R}) \\ &= \alpha[\frac{p \cdot R}{R-1} + \frac{p \cdot R}{R-1} + \frac{p(R-1)}{R-2} + \cdots \\ &\quad + \frac{p \cdot (\lfloor k \rfloor + 2)}{\lfloor k \rfloor + 1}] + \beta \cdot \lfloor k \rfloor \cdot p \end{aligned}$$

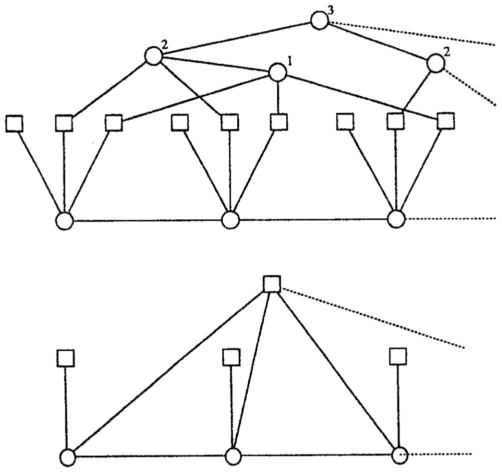

Figure 2: Lower bound example

$$
\begin{aligned}
&= \alpha p[(R - \lfloor k \rfloor) + \frac{1}{R-1} + \frac{1}{R-1} + \frac{1}{R-2} \\
&\quad + \cdots + \frac{1}{\lfloor k \rfloor + 1} + (\lfloor k \rfloor + \lfloor k \rfloor / k)] \\
&= \alpha p[(R+1) + \frac{1}{R(R-1)} + \mathcal{H}_R - \mathcal{H}_{\lfloor k \rfloor} \\
&\quad + (\lfloor k \rfloor / k - 1)]
\end{aligned}
$$

(We have ignored a single α cost in the initial reduction, and several ceilings here and there, which will account for no more than R extra α costs. Both of these are negligible for large n.)

However, $OPT(T_{k,R}) = \alpha p(R+1)$, hence

$$
r(T_{k,R}) \geq 1 + \frac{(\mathcal{H}_R - \mathcal{H}_{\lfloor k \rfloor}) + \frac{1}{R(R-1)} + \frac{\lfloor k \rfloor}{k} - 1}{R+1} \qquad (1)
$$

An improved upper bound

If p is the size of the minimum dominating set, at most $(k+1)p$ nodes will remain for beta-reductions, and the rest must be reduced by star-reductions. The size of a star reduction is the size of the largest star available, or at least $R = \lceil n/p \rceil$. Since each t-reduction decreases the count of terminals by $t-1$, in order to decrease $\lceil n/p \rceil$ by one, the p terminals must be reduced in at most $p \cdot t/(t-1)$ reductions.

$$
\begin{aligned}
&HEU(G) \\
&\leq \alpha[\frac{pR}{R-1} + \frac{p(R-1)}{R-2} + \cdots + \frac{p(\lfloor k \rfloor + 2)}{\lfloor k \rfloor + 1}] \\
&\quad + \beta p(\lfloor k \rfloor + 1) \\
&= \alpha p[(R - (\lfloor k \rfloor + 1)) + \frac{1}{R-1} + \frac{1}{R-2} + \cdots \\
&\quad + \frac{1}{k+1}] + \alpha(1 + 1/k)(\lfloor k \rfloor + 1)p \\
&= \alpha p[(R+1) + (\mathcal{H}_{R-1} - \mathcal{H}_{\lfloor k \rfloor}) + \frac{\lfloor k \rfloor + 1 - k}{k}]
\end{aligned}
$$

And, $OPT(G)$ is the same as before. Hence,

$$
r_k \leq 1 + \max_{R \in \mathrm{N}} \frac{(\mathcal{H}_{R-1} - \mathcal{H}_{\lfloor k \rfloor}) + \frac{\lfloor k \rfloor + 1 - k}{k}}{R+1}) \qquad (2)
$$

5 Interval weights

We now turn our attention to graphs for which only the ratio between the largest and the smallest weight $\frac{\beta}{\alpha} = 1 + \frac{1}{k}$ is given. We generalize the notion of a *t-star* to a set of terminals of an average distance at most $\frac{t}{t-1}$ from an optional vertex.

We obtain an exact, albeit non-trivial, bound for the approximation ratio for these graphs. To distinguish it from the ratio for binary weighted graphs, we refer to the performance ratio as r'_k.

Theorem 4 $r'_k = 1 + \max_{\epsilon} \max_{x} G_k(\epsilon, x)$
where

$$
G_k(x, \epsilon) = \frac{(1+\epsilon)(\mathcal{H}_{x-1} - \mathcal{H}_{\lfloor (1+\epsilon)k \rfloor}) + \frac{\lfloor (1+\epsilon)k \rfloor - (1+\epsilon)k + 1}{k}}{x + 1 + \epsilon}
$$

Let us further define $g_k(\epsilon) = \max_x F_k(x)$.

The lower bound

We construct a graph $Z_{k,R,\epsilon}$, as a long R-rake with slightly modified weights on edges between vertices in the same star. If each star consists of terminals $t_1, \ldots, t_R$ and an internal vertex v, and we let k' denote $\lfloor (1+\epsilon)k \rfloor$, the weights of the edges are given by:

$$
\begin{aligned}
d(v, t_i) &= \begin{cases} (1 + \epsilon/k')\alpha & i = 1, 2, \ldots k' \\ \alpha & i = k' + 1, \ldots, R \end{cases} \\
d(t_i, t_j) &= \begin{cases} \alpha \frac{i+1+\epsilon}{i} & j = i+1, i = k' \ldots R \\ \beta & \text{otherwise} \end{cases}
\end{aligned}
$$

Figure 3 gives an example of a modification of a 3-rake with no ϵ weight, which, in fact, yields a lower bound of 1.375 for the case $k = 1$, i.e. weights in the range $[1, 2]$.

The construction ensures that the terminals will be reduced first, in inverse order of their introduction. The weight of an added edge (t_i, t_{i-1}) will be

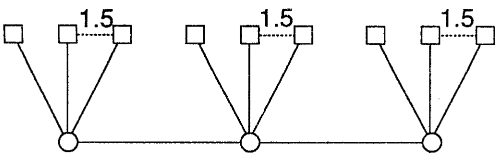

Figure 3: The graph $Z_{1,3,0}$.

equal to the average distance at the optional vertex at the time when the edge is reduced.

Let w denote the largest value of i for which $\frac{i+\epsilon}{i-1} \geq \beta/\alpha$, or $\frac{1+\epsilon}{i-1} \geq \frac{1}{k}$, or $w = 1 + \lfloor (1+\epsilon)k \rfloor$.

If we now focus only on the cost of each star, we have that

$$\begin{aligned} & HEU(Z_{k,R,\epsilon}) \\ = & \sum_{i=w+1}^{R} d(t_i, t_{i-1}) + \beta w \\ = & \alpha(\sum_{i=w+1}^{R} \frac{i+\epsilon}{i-1} + (1+\frac{1}{k})w) \\ = & \alpha(R + (1+\epsilon)\sum_{i=w}^{R-1} \frac{1}{i} + \frac{w}{k}) \\ = & \alpha(R + (1+\epsilon)(H_{R-1} - H_{w-1}) + \frac{w}{k}) \end{aligned}$$

On the other hand, $OPT(Z_{k,R,\epsilon}) = \alpha(R+1+\epsilon)$. Hence,

$$r(Z_{k,R,\epsilon}) = 1 + \frac{(1+\epsilon)(H_{R-1} - H_{\lfloor (1+\epsilon)k \rfloor}) + \frac{\lfloor (1+\epsilon)k \rfloor + 1 - (1+\epsilon)k}{k}}{R+1+\epsilon}$$

Thus, $r'_k \geq 1 + \max_R r(Z_{k,R,\epsilon}) = 1 + g_k(\epsilon)$, for any $\epsilon \geq 0$.

The upper bound

To observe that the above bound is tight, we first make the crucial observation that on some worst-case instance the heuristic will reduce only pair of terminals. The idea is that if some star has a low average cost, we can pass it on to the edges between those terminals without affecting the heuristic or optimal costs adversely.

Lemma 2 *For any instance G, there is an instance G' with identical optimal and heuristic values to G, for which the heuristic reduces only pairs of terminals.*

Proof. Let X be a star consisting of a set of terminals $T_1, \ldots, T_l$ and an internal node v, and assume it is of a minimum average weight. Now set the weight of the edges connecting the terminals to $AD(v, X)$. That value can be no less than the original edge weight, hence the heuristic cost is not affected and the optimal cost not increased. These edges will now be 2-sets of minimum weight, hence the order of reduction remains the same. Apply this argument recursively to obtain the instance G'. ∎

This implies that we can assume that the stars of the optimal Steiner tree are disjoint, and thus consider each separately. Note that it is important here to allow for continuous weights – the binary case is actually more complicated for this reason.

From now focus on a given star, which we assume has R terminals, sum C of edge weights to the internal node, additional weight $\epsilon\alpha$, and average distance a_i (at the optional vertex) before the i-th terminal (of this star) has been reduced (to another terminal in this star). Denote the cost of reduction i by c_i.

Some straightforward relationships are $C = \alpha(R+\epsilon)$, and $c_i \leq a_i$. The important thing is that in a given reduction, the sum of weights to the internal node must decrease by at least α, while the number of terminals decreases by at most one. Hence, the average cost of the i-th reduction is

$$a_i \leq \frac{C - \alpha(i-1)}{R-i}$$

which simplifies to $\alpha(1 + \frac{1+\epsilon}{R-i})$.

Assume w beta reductions are performed. Now everything falls in place.

$$\begin{aligned} & HEU(G) \\ \leq & \sum_{i=0}^{R-w-1} c_i + \beta w \\ \leq & \alpha[(T-w) + \sum_{i=0}^{R-w-1} \frac{1+\epsilon}{R-1-i} + \frac{k+1}{k} w] \\ = & \alpha[R + (1+\epsilon)(\mathcal{H}_{R-1} - \mathcal{H}_{w-1}) + \frac{w}{k}] \end{aligned}$$

And, $OPT(G) = C + \alpha = \alpha(R + \epsilon + 1)$.

We know that w is the largest integer for which the average distance of the remaining w terminals exceeds β. That is, $\frac{\alpha(w+\epsilon)}{w-1} = 1 + \frac{1+\epsilon}{w-1} \geq 1 + \frac{1}{k}$. Hence, $w \leq (1+\epsilon)k + 1$, and since w is the largest integral value satisfying that bound, we have that

$$w = \lfloor (1+\epsilon)k \rfloor + 1.$$

Thus, $r'_k \leq 1 + \max_\epsilon \max_R G_k(R, \epsilon)$, completing the proof of theorem 1.

6 Evaluation

Asymptotic evaluation

Theorem 5 *The performance ratio of the average distance heuristic on complete graphs with either binary weights* $\{\alpha, \alpha(1+1/k)\}$, *or interval weights* $[\alpha, \alpha(1+1/k)]$, *is* $1 + 1/(ek) + \mathcal{O}(1/k^2)$.

Let us first consider the case of interval weights. A simple approximation of the harmonic number $\mathcal{H}_n$ is by $\ln n + \gamma + \mathcal{O}(1/n)$, where γ is a constant. This gives us

$$G_k(x,\epsilon) = \frac{(1+\epsilon)(\ln(x-1) - \ln(1+\epsilon)k + \mathcal{O}(\frac{1}{k}))}{x+1+\epsilon}$$

The additive terms can be conveniently hidden in the lower order term, and the term involving ϵ eventually factored out.

$$\begin{aligned}
&\max_x g_k(\epsilon)\\
&= \max_{x,\epsilon} \frac{(1+\epsilon)\ln\frac{x}{(1+\epsilon)k}}{x+1+\epsilon} + \mathcal{O}(\tfrac{1}{k^2})\\
&= \max_{\epsilon, y = xk(1+\epsilon)} \frac{(1+\epsilon)\ln y}{y(1+\epsilon)k + (1+\epsilon)} + \mathcal{O}(\tfrac{1}{k^2})\\
&= \max_y \frac{\ln y}{yk} + \mathcal{O}(\tfrac{1}{k^2})\\
&= \frac{1}{ek} + \mathcal{O}(\tfrac{1}{k^2})
\end{aligned}$$

Hence, $r'_k = 1 + \frac{1}{ek} + \mathcal{O}(\frac{1}{k^2})$.

In the case of binary weights, the lower bound in equation (1) differs from the upper bound (2) by at most $\frac{1}{\bar{k}+1}[\frac{1}{k} - \frac{1}{\bar{k}} - \frac{1}{\bar{k}(\bar{k}-1)}] = \mathcal{O}(\frac{1}{k^2})$. Since the upper bound equals $1 + g_k(0)$, the equality also holds for the binary case, by the above derivation.

Empirical observations

The function $G(x,\epsilon)$ we have obtained is not a simple one, and, in particular, it depends on the maximization of two parameters, x and ϵ. The following results have been observed experimentally. Let f_k denote $g_k(0) = \max_x \frac{\mathcal{H}_{x-1} - \mathcal{H}_{\lfloor k \rfloor} + \frac{\lfloor k \rfloor + 1 - k}{k}}{x+1}$, and f'_k denote $g_k(\frac{\lceil k \rceil}{k} - 1)$.

We find that for any given ϵ, $g_k(\epsilon)$ is monotone decreasing in the interval $[\lceil k \rceil / k - 1, \infty)$. In fact, the maxima of $g_k(\epsilon)$ occurs at one of two specific values of ϵ.

Claim 1 $g_k(\epsilon) = \max(f_k, f'_k)$

Thus when k is an integer, g_k assumes a maximum when $\epsilon = 0$. The actual winner of the two depends subtly on the size of the fractional part of k, with the exact tradeoff being a slowly decreasing function approaching $\frac{e-1}{e}$.

Claim 2 *1.* $f_k > f'_k$ *when* $k - \lfloor k \rfloor \le \frac{e-1}{e} \approx 0.62$.

2. $f_k < f'_k$ *when* $k - \lfloor k \rfloor \ge 2/3 = 0.6\bar{6}$.

3. The difference between f'_k *and* f_k *amounts to less than* 0.5% *of the relative value of* r'_k, *and for* $k \ge 52$, *it is less than* 0.001%.

Note that f_k is exactly the upper bound we obtained in the binary case. Even for that special case of g_k, we have been unable to obtain a closed form expression. By experimentation, we find that f_k is maximized when $x = \mathsf{round}(e(k - .5))$. Note f_k has only a single maxima for x in $[k, \infty)$, and is therefore easily computable.

Current bounds for specific cases

Table 1 lists the current best bounds for some specific values of k, along with the relative improvement over minimum spanning tree based methods.

Binary weights

k	Upper bnd	Lower bnd	$\frac{MST-1}{ADH_I-1}$
1	$1.\bar{3}$	$1.\bar{3}$	3
2	$1.18\bar{3}$	1.15(*)	$3.\bar{3}$
3	1.121786	1.100952	3.302
4	1.0913029	1.0790349	3.163
10	1.0366375	1.0344664	2.901
100	1.003677	1.0036539	2.737

Interval weights

k	r'_k	MST	$\frac{MST-1}{ADH-1}$
1	1.375	2.0	$2.\bar{6}$
2	1.183333	1.5	2.72727
3	1.121786	$1.\bar{3}$	2.73705
4	1.0913029	1.25	2.73814
10	1.0366375	1.1	2.72944
100	1.003677	1.01	2.71966

Table 1: Some performance ratio bounds.

Note that the lower bound for $k = 2$ in the binary case was obtained from a construction slightly different from those of section 4.

7 Discussion

Comparison with other heuristics

The binary weighted network corresponding to the graph consisting of a single, huge star shows that the performance of the Minimum Spanning Tree heuristic is β/α, for $\beta \leq 2\alpha$. It is well-known that this ratio never exceeds 2, and until recently, that was the best result known. Most other methods with a comparable performance ratio have been found to simulate the MST construction either directly or indirectly.

A recent breakthrough by Zelikovsky [9] improved this ratio to 11/6. His method finds optimal Steiner trees of all 3-element subsets of S, greedily adding them the solution. This was further generalized by Berman and Ramaiyer [1] to t-element sets of terminals. They obtained a ratio of 16/9 for $t = 4$, and improvements with every increase in t. Nevertheless, the limiting ratio is still above 5/3, and the time complexity grows at least as fast as n^t.

It turns out that these advanced techniques yield little improvement over MST on the restriction of the Steiner problem considered in this paper. For instance, Zelikovsky's method performs no better than MST-based methods on graphs with $\beta \leq 4/3 \cdot \alpha$. More generally, we state the following observation.

Observation 2 *If a heuristic considers sets of terminals of size at most k, then on binary weighted graphs with $\beta/\alpha \leq 1 + 1/k$, its performance ratio is at least β/α.*

Thus ADH yields a "relative" factor of e improvement over the competition on these problems[1].

Extensions

We have also considered the case of binary weights $\{\alpha, d\alpha\}$, when the ratio $d = \beta/\alpha$ is greater than 2 [4]. For the case $d = 2.5$, we have a tight bound of 1.5, and when $d = 3$, the performance ratio lies between $30/19 \approx 1.57$ and $5/3 = 1.\overline{6}$. In general, we have a lower bound of $2 - 4/(3 \log d + 5)$ when d is a power of 2. It is an interesting question if the convergence to the asymptotic bound of 2 is only logarithmic in the weight ratio.

While we have been able to generalize the technique of the proof of theorem 3 to obtain reasonable upper bounds for d up to 3, going beyond that will require different techniques as we must take into account weights of pairs of edges.

[1] Partly because of comparisons like these, the performance ratio measure is often defined as one less than our definition.

References

[1] P. Berman and V. Ramaiyer. Improved approximations for the Steiner tree problem. In *Proc. of the Third ACM-SIAM Symp. on Discrete Algorithms*, pages 325–334, 1992.

[2] M. Bern and P. Plassman. The Steiner problem with edge lengths 1 and 2. *Inform. Process. Lett.*, 32:171–176, Sept. 1989.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractibility: A Guide to the Theory of NP-completeness.* Freeman, 1979.

[4] M. M. Halldorsson, S. Ueno, N. Hara, and Y. Kajitani. Further bounds on the average distance heuristic in restricted weight graphs. In preparation.

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

[6] B. M. Waxman. Probable performance of Steiner tree algorithms. Technical Report WUCS-88-4, Dept. Computer Science, Washington University, St. Louis, 1988.

[7] B. M. Waxman and M. Imase. Worst-case performance of Rayward-Smith's Steiner tree heuristic. *Inform. Process. Lett.*, 29:283–287, Dec. 1988.

[8] P. Winter and J. M. Smith. Path-distance heuristics for the Steiner problem in undirected networks. *Algorithmica*, 7:309–327, 1992.

[9] A. Z. Zelikovsky. An 11/6 approximation algorithm for the Steiner problem on networks. to appear in Information and Computation.