Approximating Steiner trees in graphs with restricted weights

Magnús M. Halldórsson* Shuichi Ueno† Hiroshi Nakao† Yoji Kajitani†

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 $\{1, d\}$, or weights in the interval $[1, d]$, where $d \leq 2$. The improvement over other analyzed algorithms is a factor of about $e \approx 2.718$.

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 1 and d, and the more general *interval* weights, when the edge weights fall in the interval $[1, d]$. For both types, we restrict our attention in this paper to those cases where $d \leq 2$. These restricted problems are likely to occur in the construction of communication networks when the cost per link is largely independent of its length. The binary weights may then correspond to whether link contains some existing partial installation.

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 [6]. It was shown by Waxman and Imase [8] 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 [7, 9] also indicate excellent performance in practice. Our line of work was started by Bern and Plassman [3] who considered the performance of ADH on complete graphs with binary weights 1 and 2, and proved a ratio of 4/3. They also showed that no polynomial time approximation scheme existed for this problem; namely, they showed the existence of a constant $c > 1$ such that it is $\mathcal{NP}$-hard to approximate the problem within a ratio of c.

We describe the Steiner tree problem and the Average Distance Heuristic in the next section along with attendant notation. We then proceed in Section 3 to derive a sequence of bounds on the performance of ADH on graphs with binary weights, and do the same for interval weights in Section 4. In both cases the precise bounds are shown to be of the form $1 + \frac{1}{ek} + O(\frac{1}{k^2})$, where k is such that $d = 1 + \frac{1}{k}$, and e is the basis of the natural logarithm. The constants behind the lower order

*Corresponding author. Science Institute, University of Iceland, IS-107 Reykjavik, Iceland. mmh@rhi.hi.is

†Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Meguro-ku, 152 Tokyo, Japan.

term are small. This improves on the $1 + \frac{1}{k}$ performance of other analyzed methods. Comparison and evaluation of the function describing the performance ratio is given in Section 5. Finally, we complement the analysis of ADH by showing, in Section 6, the binary weighted Steiner tree problem to be $\mathcal{NP}$-hard to approximate within a factor of c, for some fixed constant $c > 1$. The paper closes off with a discussion of related problems and methods.

2 The Steiner tree problem and the Average Distance Heuristic

The *Steiner tree problem on graphs* is defined as follows.

> **Given:** A graph $G = (V, S, E, d)$, where V is a set of vertices, containing a subset $S \subseteq V$ of *terminals*, $E = V \times V$ is the set of edges, and d is a weight function $d : E \mapsto \mathbb{R}$.
> **Find:** A set of edges $T \subset E$ that connects together the elements of S (and possibly some of $V - S$) into a tree, such that the cost of T, $\sum_{e \in T} d(e)$, is minimized.

The elements of $V - S$ will be referred to as *optional* vertices.

Average Distance Heuristic The objective of the Steiner tree problem is to connect the terminals using minimum total sum of edge weights. Like many Steiner tree heuristics, the ADH attacks the problem by performing a sequence of *reductions*, each contributing a small set of edges to the resulting tree and slightly reducing the size of the instance. In each step, it chooses a subset X of terminals and a vertex v, adds the edges from v to the elements of X to the solution, and merges v with the vertices of X to a single new terminal. As with most of the studied heuristics, ADH selects the set to be reduced according to a greedy rule. The characterizing feature of ADH is that it considers all sets of terminals as candidates, independent of their cardinality, whereas other heuristics [9, 10, 2] consider only constant-sized sets of terminals.

The algorithm chooses a set of terminals X and a vertex v that minimizes the quantity

$$AvgDist(v, X) \doteq \frac{\sum_{x \in X} d(v, x)}{|X| - 1}$$

referred to as the *average distance* of v to X. The reduction replaces the terminals in X by a single terminal, thus reducing the number of terminals by $|X| - 1$ at the cost of adding the edges from v to X, or $\sum_{x \in X} d(v, x)$. $AvgDist(v, X)$ thus represents the average cost spent per terminal eliminated. The algorithm is given in Figure 1. For details of implementation and time complexity, refer to [6, 9].

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, the edges added to the tree induce a *star* centered at v.

Notation Let n denote $|S|$. Let $d(u, v)$ denote $d(e)$ where $e = (u, v)$. We assume that the weights in the input have been scaled so that the minimum weight is 1. Let d denote the maximum weight, k denote the real value such that $d = 1 + 1/k$, and $K = \lfloor k \rfloor$. Let p denote the number of optional nodes in an optimum Steiner tree. Let $\mathcal{H}_z$ denote the z-th harmonic number, $\sum_{i=1}^{z} 1/i$.

Let HEU(G) denote the cost of the solution found by the algorithm (ADH) on instance G, maximized over all possible tie-breaks. OPT(G) denotes the cost of an 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.

```
ADH(G)
  T ← ∅
  while |S| > 1 do
    Choose X ⊂ S and v ⊂ V that minimizes AvgDist(v, X)
    S ← (S − X) ∪ {v}
    V ← (V − X) ∪ {v}
    for each w ∈ V − X
      d(v, w) ← min_{x∈X∪{v}} d(x, w)
    T ← T ∪ {(x, v) : x ∈ X}
  od
  output T
```

Figure 1: The Average Distance Heuristic

3 Binary weights

When considering binary weights, we change our perspective of weighted graphs to that of related unweighted, not-complete graphs. For an instance $G = (V, S, E, d)$, the related unweighted graph $G' = (V, S, E')$ is defined on the same vertex set such that

$$d(u, v) = 1 \iff (u, v) \in E', \qquad d(u, v) = d \iff (u, v) \notin E'.$$

We shall only speak of these unweighted graphs for the remainder of this section.

We say that the graph *contains a t-star*, if it contains an optional vertex adjacent to t or more terminals. One useful observation is that no terminal vertices are adjacent in a worst-case instance. The heuristic therefore performs only *star-reductions* – represented by a particular optional node and all adjacent terminals – and *heavy reductions* that merges two terminals of distance d together. Once it starts performing heavy reductions, we may assume that it does so throughout.

In what follows, we start by generalizing both the upper and lower bound of [3] 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.

3.1 Simple bounds

For each positive integer t, we consider the following graph which we name *t-rake*: $R_t = R_{t,p}$: A sequence of p optional vertices are linked in a path, with t terminals hanging off each optional vertex as leaf nodes. A 3-rake is shown in Figure 2.

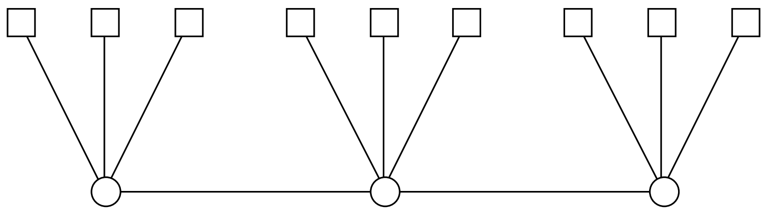

Figure 2: A short 3-rake

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

Proof. In a t-rake, the *average distance* of any node to any subset of terminals is at least $(t+1)/t$. Thus, when $t \leq K+1 = \lfloor k \rfloor + 1$, the heuristic may simply reduce the terminals in heavy reductions, for a cost of

$$\mathrm{HEU}(R_t) = d(n-1) = \frac{k+1}{k}(n-1). \tag{1}$$

On the other hand, when $t \geq K+1$,

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

In particular, in the case of $k+1$-rake with k integral,

$$\frac{\mathrm{HEU}(R_{k+1})}{\mathrm{OPT}(R_{k+1})} = \frac{dn}{n(k+2)/(k+1)} = \frac{k+1}{k}\frac{k+1}{k+2} = 1 + \frac{1}{k(k+2)}.$$

∎

We generalize some observations of [3].

Observation 1

1. *If G contains a $(t+1)$-star, then the algorithm makes progress towards $\frac{t+1}{t}$ ratio. More precisely, it will add a set of edges to the solution of cost c, obtaining a new graph H with the property that $HEU(G) = HEU(H) + c$ and $OPT(G) \geq OPT(H) + c\frac{t+1}{t}$.*
2. *$OPT(G) \geq n + p - 1$, 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 disjoint t-stars, then at least $2q$ nodes will be covered in t-reductions by the heuristic.*

A simple application of Observation 1 parts 1-3 yields the following upper bound.

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

Proof. Let $t = \lceil 2k \rceil$. The proof is by induction on the size of G. The statement holds trivially for the single-node graph.

Assume G contains no $(t+1)$-star. Then $\mathrm{OPT}(G) \geq n + p - 1 \geq n + n/t - 1$, while $\mathrm{HEU}(G) \leq d(n-1)$. Hence,

$$r_k(G) \leq \frac{1 + 1/k}{1 + 1/\lceil 2k \rceil} = 1 + \frac{\lceil 2k \rceil - k}{k(\lceil 2k \rceil + 1)} \leq 1 + \frac{1}{\lceil 2k \rceil + 1}.$$

On the other hand, if G does contain a $(t+1)$-star, by Obs. 1.1, ADH performs a reduction of cost c and obtains a new graph H with the property that $\mathrm{HEU}(G) = \mathrm{HEU}(H) + c$ and $\mathrm{OPT}(G) \geq \mathrm{OPT}(H) + c\frac{t+1}{t}$. By the induction hypothesis, $\mathrm{HEU}(H) \leq (1 + \frac{1}{t})\mathrm{OPT}(H)$, so

$$\mathrm{HEU}(G) = (1 + \frac{1}{t})(\mathrm{OPT}(H) + c\frac{t+1}{t}) \geq (1 + \frac{1}{t})\mathrm{OPT}(G).$$

∎

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.

Theorem 2 $r_k \le 1 + \dfrac{1}{\lceil 2k \rceil + 1}$

Proof. Let $t = \lceil 2k \rceil + 1$. If there is a $(t+1)$-star, then by Obs. 1.1) we make progress towards a $1 + 1/t$ ratio. Thus, assume that the graph contains no $(t+1)$-stars.

Let q denote the number of (disjoint) t-stars in an optimal solution. Then $n \le qt + (p-q)(t-1) = p(t-1) + q$, and thus $p \ge \frac{n-q}{t-1}$.

From the above, and Observation 1.2,

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

Let m be the number of reductions with t or more terminals, and let s be the number of terminals participating in them. Then $m \le s/t$. Moreover, since at least 2 nodes from a given star must be reduced in order to decrease the star, we have that $s \ge 2q$ (see Observation 1.4).

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

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

Using that $t = \lceil 2k \rceil + 1$, the ratio is bounded by $\frac{k+1}{k}\frac{\lceil 2k \rceil}{\lceil 2k \rceil + 1} = 1 + \frac{\lceil 2k \rceil - k}{k(\lceil 2k \rceil + 1)} \le 1 + \frac{1}{2k+1}$. ∎

3.2 Asymptotically tight bounds

We now derive a new lower and an upper bound on the performance ratio of ADH, which are asymptotically tight as k grows.

Theorem 3

$$r_k \ge 1 + \max_{R \in \mathbb{N}} \frac{(\mathcal{H}_R - \mathcal{H}_{\lfloor k \rfloor}) + \frac{1}{R(R-1)} + \frac{K}{k} - 1}{R+1} \tag{2}$$

Proof. Let R be an integer and let p be a multiple of $(R-1)!/K!$ We construct a family of graph $\{G_{R,k,p}\}$ parameterized by R, representing the sizes of stars in the instance, p, representing the number of optional nodes in the optimal solution, and k, the weight factor. The graphs are a modification of the R-star, arranged so that the algorithm will reduce the degree of all optional vertices by one at a time, until it reaches $K+1$, at which point heavy-reductions take over. In the proper context, we shall simply refer to an instance of the family as G.

Define

$$f(i,z) = \begin{cases} \lceil i/(z-1) \rceil, & \text{when } z = R \\ \lceil i/z \rceil & \text{otherwise.} \end{cases}$$

The sets of terminals and optional nodes of G are given by

$$\begin{aligned}
S(G) &= \{T_{i,j} : i = 1, \ldots, p, j = 1, \ldots, R\} \cup \{\omega\} \\
V(G) - S(G) &= \{s_i : i = 1, \ldots, p\} \cup \{x_y^z : z = K+1, \ldots R, y = 1, \ldots, f(p,z)\}.
\end{aligned}$$

The edges of G are given by

$$\begin{aligned} E(G) &= \{(s_i, T_{i,j}), (x^z_{f(i,z)}, T_{i,z}) : i = 1, \ldots, p, j = 1, \ldots, R, z = K+1, \ldots, R\} \\ &\quad \cup \{(s_i, s_{i+1}) : i = 1, \ldots, p-1\} \\ &\quad \cup \{(x^z_y, \omega) : z = K+1, \ldots, R, y = 1, \ldots, f(p, z)\}. \end{aligned}$$

Observe that each x^z_y vertex is adjacent to $j+1$ terminals (including ω) when $j < R$, but to j terminals when $j = R$.

The key to analyzing the worst-case cost of the Steiner tree computed by the algorithm is to fix a particular sequence of reductions.

Claim 1 *One possible sequence of reductions that the algorithm may perform is:*

$$x^R_1, x^R_2, \ldots, x^R_{p/(R-1)}, x^{R-1}_1, x^{R-1}_2, \ldots, x^{R-1}_{p/(R-1)}, \ldots, x^{K+1}_1, x^{K+1}_2, \ldots, x^{K+1}_{p/(K+1)}$$

followed by heavy reductions.

Initially, all nodes are adjacent to at most R terminals. This includes the x^R_y nodes. Observe that since the x-nodes are not adjacent to each other or to any other optional node, no sequence of x-node reductions can increase the number of terminals adjacent to any given node. Thus, we may assume that the x^R_y nodes, $y = 1, 2, \ldots$ are all reduced in the first round. Call the resulting network $H_{K,p,R-1}$.

After this round, all $T_{i,R}$ nodes, $i = 1, \ldots, p$, have been merged into the single terminal ω. The optional nodes s_i, $i = 1, \ldots, p$ are still adjacent to R terminals, and so are the x^{R-1}_y optional nodes, $y = 1, \ldots, p/(R-1)$. Thus, we may assume that the x^{R-1}_y nodes are reduced in sequence. Now, observe that this results in the network $H_{K,p,R-2}$. Since the above argument holds independent of R, for integer R, $R \geq K+1$, we have that, by induction, the reduction sequence continues as claimed: the x^{R-2}_y nodes, x^{R-3}_y nodes, down to the x^{K+1}_y nodes.

The network $H_{K,p,K}$ consists of the K-rake along with an additional terminal ω adjacent to all the optional nodes s_i, $i = 1, \ldots, p$.. Each optional node is then adjacent to at most $K+1$ terminals. Then, we may assume that the remainder is reduced by heavy reductions, for a cost of $dn = dpK$ (see (1)).

If we sum up the costs of the reductions in each round, we obtain

$$\mathrm{HEU}(G) \geq R\frac{p}{R-1} + R\frac{p}{R-1} + (R-1)\frac{p}{R-2} + \cdots + (K+1)\frac{p}{K} + \mathrm{HEU}(H_{K,p,K}).$$

Simplify the sum using the harmonic function $\mathcal{H}_t$,

$$\begin{aligned} \mathrm{HEU}(G_{k,p,R}) &\geq p[(R-K) + \frac{1}{R-1} + \frac{1}{R-1} + \frac{1}{R-2} + \cdots + \frac{1}{K+1}] + \frac{k+1}{k}pK \\ &= p[R + \frac{1}{R(R-1)} + \mathcal{H}_R - \mathcal{H}_K + K/k]. \end{aligned} \tag{3}$$

The optimal solution reduces the s_i vertices, $i = 1, \ldots, p$, for a cost of

$$\mathrm{OPT}(G) = (R+1)p - 1. \tag{4}$$

The ratio of (3) to (4), maximized over all values of R yields (2) and the theorem. ∎

Following the pattern in the above argument, we are led to a similar argument for the upper bound on the performance ratio.

Theorem 4

$$r_k \leq 1 + \max_{R \in \mathbb{N}} \frac{(\mathcal{H}_{R-1} - \mathcal{H}_K) + \frac{K+1-k}{k}}{R+1} \tag{5}$$

Proof. If p is the size of the minimum dominating set, at most $(k+1)p$ nodes will remain for heavy-edge 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.

$$\begin{aligned}
\mathrm{HEU}(G) &\leq \frac{pR}{R-1} + \frac{p(R-1)}{R-2} + \cdots + \frac{p(K+2)}{K+1} + dp(K+1) \\
&= p[(R-(K+1)) + \frac{1}{R-1} + \frac{1}{R-2} + \cdots + \frac{1}{k+1}] + (1+1/k)(K+1)p \\
&= p[(R+1) + (\mathcal{H}_{R-1} - \mathcal{H}_K) + \frac{K+1-k}{k}]
\end{aligned}$$

From Observation 1 (part 2 and 3), $\mathrm{OPT}(G) \geq n(1+1/R) \approx p(R+1)$. The performance ratio of the algorithm is at most the ratio between these two values, maximized over the possible values of R. ∎

Combining Theorems 3 and 4, we obtain a characterization of the performance ratio that is tight asymptotic with k.

Corollary 1 $r_k = 1 + \frac{1}{ek} + O(\frac{1}{k^2})$

Proof. Observe that the difference between the bounds of the two ratios is $\frac{1}{k} - \frac{1}{R} - \frac{1}{R(R-1)} = O(\frac{1}{k^2})$. Recall the approximation of $\mathcal{H}_t$ as $\ln t + \gamma + O(1/t)$, where γ is a constant. Denote $x = (R+1)/k$. Then, we have that

$$\begin{aligned}
r_k &= 1 + \max_x \frac{\mathcal{H}_{xk} - \mathcal{H}_k + O(1/k)}{xk} \\
&= 1 + \max_x \frac{\ln xk - \ln k + O(1/k)}{xk} \\
&= 1 + \frac{1}{k}(\max_x \frac{\ln x}{x}) + O(\frac{1}{k^2}) \\
&= 1 + \frac{1}{ek} + O(\frac{1}{k^2})
\end{aligned}$$

∎

4 Interval weights

We now turn our attention to graphs for which the only restriction is on the ratio between the largest and the smallest edge weight. 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 5 $r'_k = 1 + \max_{x,\epsilon \geq 0} G_k(x,\epsilon)$
where

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

We generalize the notion of a *t-star* to a set of terminals of an average distance at most $t/(t-1)$ from an optional vertex.

We spend the remainder of this section proving Theorem 5, delaying further evaluation of the approximation to the next section.

The lower bound For an integer R and a real value $\epsilon \geq 0$, we construct a graph $Z_{k,R,\epsilon}$. The graph is 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, the weights of the edges are given by:

$$d(v, t_i) = \begin{cases} 1 + \epsilon/\lfloor(1+\epsilon)k\rfloor & i = 1, \ldots, \lfloor(1+\epsilon)k\rfloor \\ 1 & i = \lfloor(1+\epsilon)k\rfloor + 1, \ldots, R \end{cases}$$

$$d(t_i, t_j) = \begin{cases} \frac{i+1+\epsilon}{i} & j = i+1, i = \lfloor(1+\epsilon)k\rfloor, \ldots, R \\ d & \text{otherwise} \end{cases}$$

Figure 3 gives an example of a modification of a 3-rake without ϵ weights, which, in fact, yields a lower bound of 1.375 for the case $k = 1$, i.e. weights in the interval $[1, 2]$.

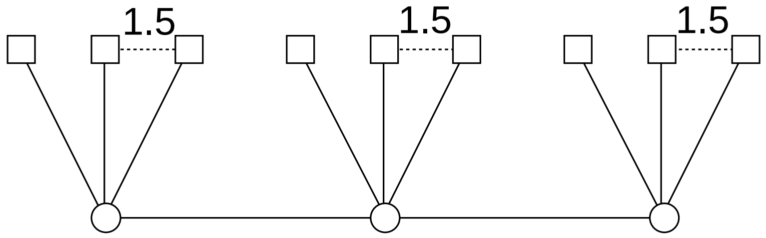

Figure 3: The graph $Z_{1,3,0}$.

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 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 $\frac{i+\epsilon}{i-1} \geq d$, 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

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

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

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

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 to our problem, there is another instance with identical optimal and heuristic values, for which the heuristic reduces only pairs of terminals.*

Proof. Take a star of minimum average weight, and reset the weight of the edges between the terminals to the average distance of the star (see definitions). That value can be no less than the original edge weight, hence the heuristic cost is not affected and the optimal cost not increased. Each of these edges will now be terminal pairs of minimum weight, hence the order of reduction remains the same. Apply this argument recursively to obtain the claimed instance. ∎

Lemma 3 *For any instance to our problem, there is another instance with identical optimal and heuristic values, in which edges of cost less than d that are incident on optional vertices induce a forest with terminals as leaves.*

Proof. Apply the transformation of the previous lemma, to ensure that the heuristic only reduces pairs of terminals. Now set the weight of all edges neither in the optimal nor heuristic solution as d. This affects neither the optimal nor the heuristic solutions. The only remaining edges incident on optional vertices of cost less than d are those from the optimal solution, thus necessarily forming a forest (possibly a tree). ∎

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 W of weights of edges from terminals to the internal node, additional weight ϵ, and average distance a_i (at the optional vertex) before the i-th terminal (of this star) is reduced (to another terminal in this star). Denote the cost of the i-th reduction by c_i.

Some straightforward relationships are $W = R + \epsilon$, and $c_i \leq a_i$. The crucial observation is that in a given reduction, the sum of weights to the internal node must decrease by at least 1, while the number of terminals decreases by at most one. Hence, the average cost of the i-th reduction is

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

which simplifies to $a_i \leq 1 + \frac{1+\epsilon}{R-i}$.

Let w be the number of heavy reductions are performed. Now everything falls in place.

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

And, $\mathrm{OPT}(G) = W + 1 = R + \epsilon + 1$.

We know that w is the largest integer for which the average distance of the remaining w terminals exceeds d. That is, $\frac{w+\epsilon}{w-1} = 1 + \frac{1+\epsilon}{w-1} \geq 1 + \frac{1}{k}$. 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 5.

5 Evaluation

Asymptotic evaluation

Theorem 6 *The performance ratio of ADH complete graphs with weights in the interval* $[1, 1+1/k]$ *is* $1 + 1/(ek) + O(1/k^2)$*, for any* $k \geq 1$*.*

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

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

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

$$\begin{aligned}
\max_{x,\epsilon} G_k(x,\epsilon) &= \max_{x,\epsilon} \frac{(1+\epsilon)\ln\frac{x}{(1+\epsilon)k}}{x+1+\epsilon} + O(\frac{1}{k^2}) \\
&= \max_{\epsilon, y = x/[k(1+\epsilon)]} \frac{(1+\epsilon)\ln y}{y(1+\epsilon)k + (1+\epsilon)} + O(\frac{1}{k^2}) \\
&= \max_{y} \frac{\ln y}{yk} + O(\frac{1}{k^2}) \\
&= \frac{1}{ek} + O(\frac{1}{k^2})
\end{aligned}$$

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

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. Define $g_k(\epsilon) = \max_x G_k(x, \epsilon)$.

We found that g_k is monotone decreasing for ϵ 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 2 $\max_\epsilon g_k(\epsilon) = \max(g_k(0), g_k(\frac{\lceil k \rceil}{k} - 1))$

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 3

1. $g_k(0) > g_k(\frac{\lceil k \rceil}{k} - 1)$ *when* $k - \lfloor k \rfloor \leq \frac{e-1}{e} \approx 0.62$.
2. $g_k(0) < g_k(\frac{\lceil k \rceil}{k} - 1)$ *when* $k - \lfloor k \rfloor \geq 2/3 = 0.6\overline{6}$.

3. The difference between $g_k(\frac{\lceil k \rceil}{k} - 1)$ *and* $g_k(0)$ *amounts to less than* 0.5% *of the relative value of* r'_k*, and for* $k \geq 52$*, it is less than* 0.001%.

Note that $g_k(0)$ 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 $g_k(0)$ is maximized when $x = \mathsf{round}(e(k - .5))$. Note $g_k(0)$ 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				Interval weights	
k	Upper bnd	Lower bnd	$\frac{MST-1}{ADH_{lb}-1}$	MST	r'_k	$\frac{MST-1}{ADH-1}$
1	$1.\overline{3}$	$1.\overline{3}$	3	2.0	1.375	$2.\overline{6}$
2	$1.18\overline{3}$	1.15(*)	$3.\overline{3}$	1.5	1.183333	2.72727
3	1.121786	1.100952	3.302	$1.\overline{3}$	1.121786	2.73705
4	1.0913029	1.0790349	3.163	1.25	1.0913029	2.73814
10	1.0366375	1.0344664	2.901	1.1	1.0366375	2.72944
100	1.003677	1.0036539	2.737	1.01	1.003677	2.71966

Table 1: Some bounds on the performance ratio.

We should note that the construction that yielded the lower bound for $k = 2$ in the binary case was obtained from a construction not given in this paper.

6 Hardness

Theorem 7 *There exists a constant* $c > 1$*, such that the Steiner Tree problem on a complete graph with distinct binary weights* $\mathcal{NP}$*-hard to approximate within a factor of* c.

Proof. Assume without loss of generality that the edge weights are either 1 or d. For $d \geq 2$, hardness has already been established by the hardness proof of [3] for the case $d = 2$. We shall prove it here for the remaining values of d. Let κ be the least integer such that $d > 1 + 1/(\kappa/2)$.

The proof is by a reduction from *κ-Set Packing.* Given an set system $(Y, \mathcal{C})$, consisting of a basis set Y, and collection $\mathcal{C}$ of subsets of Y of size κ each, the problem asks if there exists a subcollection $\mathcal{C}'$ of $\mathcal{C}$ of mutually disjoint sets whose union is the basis set Y. From an instance $(Y, \mathcal{C})$ to κ-Set Packing 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 $\mathcal{C}$, with the vertices labelled accordingly. Edges between optional vertices are given a unit weight, while those between terminal vertices are assigned weight d. For edges between an optional vertex v and a terminal vertex u, the weight assigned will be 1 if the label of u is contained in the label of v, and d otherwise. In other words, unit weight is assigned if the basis element (in Y) associated with u is a member of the subset (in $\mathcal{C}$) associated with v.

The weight of the optimal Steiner tree of the graph is strongly related to the question whether the set system has an exact cover, or a perfect set packing. Namely, it is easy to verify that

$$\mathrm{OPT}(G) = n + n/\kappa - 1 \text{ iff } (\mathcal{C}, Y) \text{ has a packing of } n/\kappa \text{ sets} \tag{6}$$

In this case, the *average cost* per terminal is $1 + 1/\kappa$.

On the other hand, suppose the set system does not contain a packing of more than $\frac{n}{\kappa}(1-\delta)$ sets. Then, any Steiner tree has at most that many internal nodes that cover κ terminals. The remaining δn nodes must be covered by internal nodes covering $\kappa - 1$ or fewer nodes, or by edges of weight d. In either case, the average cost for each of the δn nodes is at least $1 + 1/(\kappa - 1)$, and the total cost of the tree is at least

$$(1-\delta)n(1+\frac{1}{\kappa}) + \delta n(1+\frac{1}{(\kappa-1)\kappa}) - 1 = n[1+\frac{1}{\kappa}+\frac{\delta}{\kappa(\kappa-1)}] - 1$$

It is known that there is a fixed $\delta > 0$, such that it is $\mathcal{NP}$-hard to decide whether a k-Set Packing instance has an exact cover, or if it contains no packing of $(1-\delta)n/\kappa$ sets [5]. It follows for each binary weighted Steiner problem, it is $\mathcal{NP}$-hard to decide whether there is a Steiner tree of cost $n(1+1/\kappa)-1$, or if every Steiner tree has cost at least $n(1+1/\kappa+\delta/\kappa(\kappa-1))-1$. The ratio between the two values is

$$\frac{1+1/\kappa+\delta/\kappa(\kappa-1)}{1+1/\kappa} = 1 + \frac{\delta}{(\kappa-1)(\kappa+1)}.$$

Thus, for every fixed d, there exists a constant $\delta' > 0$, such that it is hard to approximate the binary weighted Steiner tree problem within a factor of $1 + \delta'$. ∎

In particular, this shows that our problem is hard to approximate within a factor or $1 + \Omega(k^{-2})$. It is known that, for some $\epsilon > 0$, k^ϵ approximation of k-Set Packing is hard [1], which implies that our Steiner problem is hard within $1 + \Omega(k^{-2+\epsilon})$. The hardness of the general Set Packing problem [4] indicate that k-Set-Packing is even hard to approximate within $k^{1-\epsilon}$, for *any* $\epsilon > 0$. That would would imply a $1 + \Omega(k^{-1-\epsilon})$ hardness for our problem, suggesting that our bounds are close to the best possible.

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 d, for $d \leq 2$. 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 breakthrough by Zelikovsky [10] 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 [2] 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 $d \leq 4/3$. 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 $d \leq 1 + 1/k$, its performance ratio is at least d.*

Thus ADH yields a "relative" factor of e improvement over known algorithms whose polynomiality does not depend on k [1].

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

Extensions We have also considered the case of binary weights when the ratio d is greater than 2. 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 2 to obtain reasonable upper bounds for d up to 3, going beyond that will require different techniques involving weights of pairs of edges.

8 Acknowledgments

We thank Chris Long for his generous efforts to evaluate the approximation function.

References

[1] N. Alon, U. Feige, A. Wigderson, and D. Zuckerman. Derandomized graph products. *Computational Complexity*, 5(1):60–75, 1995.

[2] P. Berman and V. Ramaiyer. Improved approximations for the Steiner tree problem. *Journal of Algorithms*, 17(3):381–408, Nov. 1994.

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

[4] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proc. 37th Ann. IEEE Symp. on Found. of Comp. Sci.*, pages 627–636, Oct. 1996.

[5] E. Petrank. The hardness of approximation: Gap location. *Comput. Complexity*, 4:133–157, 1994.

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

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

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

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

[10] A. Zelikovsky. An 11/6-approximation algorithm for the network Steiner problem. *Algorithmica*, 9:463–470, 1993.