A Factor 3/2 Approximation for Generalized Steiner Tree Problem with Distances One and Two

Piotr Berman* Marek Karpinski† Alex Zelikovsky‡

Abstract

We design a 3/2 approximation algorithm for the Generalized Steiner Tree problem (GST) in metrics with distances 1 and 2. This is the first polynomial time approximation algorithm for a wide class of non-geometric metric GST instances with approximation factor below 2.

1 Introduction

We design a 3/2 approximation algorithm for constructing generalized Steiner trees (Steiner Forests) for metrics with distances 1 and 2. With the exception of geometric metrics [5], there were no wide classes of instances known with approximation ratios better than 2. This was in contrast to similar problems like Traveling Salesman and Steiner Tree Problems [2], [3].

2 Definitions and Notation

A metric with distances 1 and 2 can be represented as a graph with edges being pairs of distance 1 and non-edges being pairs of distance 2. We will use GST[1,2] to denote the Generalized Steiner Tree Problem restricted to such metrics.

*Department of Computer Science & Engineering, Pennsylvania State University, University Park, PA 16802. Research partially done while visiting Department of Computer Science, University of Bonn and supported by DFG grant Bo 56/174-1. Email: berman@cse.psu.edu

†Department of Computer Science and the Hausdorff Center for Mathematics, University of Bonn, 53117 Bonn. Supported in part by DFG grants, Procope grant 31022, and the Hausdorff Center grant EXC59-1. Email: marek@cs.uni-bonn.de

‡Department of Computer Science, Georgia State University, Atlanta, GA 30303. Email: alexz@cs.gsu.edu

The problem instance of GST(1,2) will be a graph $G = (V, E)$ that defines a metric in this way, and a collection $\mathcal{R}$ of subsets of V called *required sets*. We say that $\cup_{R\in\mathcal{R}} R$ is the set of *terminals*. In a *proper* instance, the required sets do not overlap and have more than one element. It is obvious that for every family of requirements $\mathcal{R}$, there exists a unique family $prop(\mathcal{R})$ that is equivalent and proper.

A valid solution is a set of unordered node pairs F such that each R_i is contained in a connected component of (V, F). The objective is to minimize $|F \cap E| + 2|F - E|$.

We will use in the sequel some notation and terminology introduced in [3].

A basic building block of our solutions is an s-star consisting of a non-terminal c, called the center, s terminals $t_1, \dots, t_s$ and edges $(c, t_1), \dots, (c, t_s)$. In [3] we used also a more general version of a building block, an (r, s)-comet consisting of a non-terminal center c, non-terminal fork nodes $f_1, \dots, f_s$ plus $r + 2s$ terminals, the center is connected to r terminals and all the fork nodes, while each fork node is connected to two terminals of its own.

If $s < 3$ we say that the star is *degenerate*, and *proper*, otherwise.

In the analysis of our algorithm, we will view its selections as transformations of an input instance, so after each phase we have a partial solution and a residual instance. We formalize these notions as follows.

A partition Π of V induces a graph $(\Pi, E(\Pi))$ where $(A, B) \in E(\Pi)$ if $(u, v) \in E$ for some $u \in A, v \in B$). We say that (u, v) is a representative of (A, B).

Similarly, Π induces required sets. Let $R_\Pi = \{A \in \Pi : A \cap R \neq \emptyset\}$, then $\mathcal{R}_\Pi = prop(\{R_\Pi : R \in \mathcal{R}\})$.

In our algorithms, we augment initially empty solution F. Edge set F defines partition $\Pi(F)$ into connected components of (V, F). In a step, we identify a connected set A in the induced graph $(\Pi(F), E(\Pi(F)))$ and we augment F with representatives of edges that form a spanning tree of A. We will call it "collapsing A", because A will become a single node of $(\Pi(F), E(\Pi(F)))$.

Thus, if we select some "building block" C, F is going to be augmented by the representatives of the edges in C, and this changes the "residual" graph in which we make our next selection. For that reason we will use terms "select" and "collapse" as synonyms,

3 Analyzing Greedy Heuristics

We introduce a new way of analyzing greedy heuristics for our problem, and in this section we illustrate it on the example of the Rayward-Smith heuristic [6] for STP[1,2]. This heuristic has approximation ratio of exactly 4/3, as demonstrated by Bern and Plassman [4]. However, the new analysis method is tighter (see Theorem 1) and characterizes the effect of more general classes of greedy choices, as we will show in the next section. We have reformulated the Rayward-Smith heuristic as follows.

While there is more than one terminal
perform the first possible operation from the following list:

1. **Preprocessing:** Collapse an edge between terminals.
2. **Collapsing of stars:** Collapse an s-star S with maximum s.
3. **Finishing:** Connect two terminals with a non-edge.

If we can perform a step of Preprocessing, the approximation ratio can only improve since such the collapsed edge can be forced into the optimal solution. Thus it suffices to analyze the case when no two terminals share a cost-1 connection.

Let T^* be an optimal Steiner tree and let $T = T^* \cap E$ be its *Steiner skeleton* consisting of its edges (cost-1 connections),

Let T_{RS} be the Steiner tree given by Rayward-Smith heuristic. We are going to prove the following

Theorem 1 $cost(T_{RS}) < cost(T^*) + \frac{1}{3}cost(T)$.

In the analysis of the Collapsing of stars and Finishing, we update the following three values after each iteration:

CA = the total cost F, the set of edges collapsed so far, initially, $CA = 0$;

CR = the cost of the *reference solution* T_{ref} derived from the optimum solution T^*; T_{ref} is a solution of the residual problem in (V_F, E_F);

P = the sum of potentials distributed among objects, which will be defined later.

The sum $CA + CR + P$ will be the *promised cost*, *PromCost*.

We will define the potential satisfying the following conditions:

(a) initially, $P < cost(T)/3 \leq cost(T^*)/3$;

(b) after each star collapse, *PromCost* will be unchanged or decreased.

(c) at termination, $CR = 0$ (T_{ref} will be empty) and $P = 0$.

These properties clearly imply our claim, as the initial *PromCost* would satisfy the statement of the theorem, *PromCost* cannot increase and at the termination we return a solution with that cost.

Initially, $T_{ref} = T = T^* \cap E$. In the analysis, we also use the *skeleton* of T_{ref}, $T_{ref}^{sk} = T_{ref} \cap E$, the set of 1-cost connections of T_{ref}. The potential is given to the following objects:

- edges of T_{ref}^{sk};
- *C-comps* which are connected components of T_{ref}^{sk};
- *S-comps* which are Steiner full components of T_{ref}^{sk}.

The total potential of edges, C-comps and S-comps is denoted PE, PC and PS respectively. **At all times**, the potential of each edge $e \in T^{sk}_{ref}$ is $p(e) = \frac{1}{3}$.

Initially, the potential of each C-comp and S-comp is zero.

A Steiner tree is called *bridgeless* if no two Steiner points are adjacent and each Steiner point has degree at least 3.

Lemma 1 *Without increasing PromCost, we can transform the optimum solution T^* into a bridgeless reference solution T_{ref}, while the new potential p satisfies*

(i) each C-comp C has $p(C) \geq -\frac{2}{3}$ and if C has fewer than 3 edges, $p(C) = 0$;

(ii) each S-comp S has $p(S) = 0$;

Proof. Because CA and PS remain zero, to see that *PromCost* does not increase it suffices that each transformation of T_{ref} and p satisfies $\Delta CR + \Delta PE + \Delta PC \leq 0$. The bridgeless Steiner tree is obtained using the following two types of steps.

Path step. Suppose that T^{sk}_{ref} contains a Steiner point v of degree 2. We remove two edges incident to v from T^{sk}_{ref} adding a non-edge (cost-2 connection) to T_{ref}. The potential for the both resulting C-comps is set to 0. One can see that $\Delta CR = 0$, $\Delta PE = -\frac{2}{3}$ (two edges removed) and $\Delta PC \leq \frac{2}{3}$ ($\Delta PC = \frac{2}{3}$ if the component C that was split had $p(C) = -\frac{2}{3}$.

If the removal of edges in a Path step creates Steiner points of degree 1, we remove them; this can only decrease *PromCost*.

Bridge Step. Suppose that we cannot perform a Path step and $e \in T^{sk}_{ref}$ is a *bridge*, i.e., an edge $e = (u, v)$ between Steiner points. We remove this edge from T_{ref} (replacing with a non-edge between terminals); this splits a C-comp C into C_0 and C_1. Each new C-comp has at lest two edges since u and v originally have degree at least 3. We set $p(C_0) = p(C)$ and $p(C_1) = -\frac{2}{3}$. Thus $\Delta CR = 1$ (the cost is increased by 1), $\Delta PE = -\frac{1}{3}$ and $\Delta PC = -\frac{2}{3}$ (one more C-comp with potential $-\frac{2}{3}$.

Note that if we create a C-comp with two edges, we can apply a Bridge Step; this is because we assume than there are no edges between terminals. ❒

From now on our reference Steiner tree T_{ref} is assumed to be bridgeless.
Now we will prove

Lemma 2 *After collapsing an s-star S, $s > 3$, conditions (i)-(ii) of Lemma 1 are satisfied and PromCost does not increase.*

Proof. Suppose that the terminals of S be in a C-comps. To break cycles created in T_{ref} when we collapse S, we replace $s - 1$ connections, of which $a - 1$ are cost-2 connections between different C-comps and $s - a$ edges within C-comps.

If this is the entire modification, $\Delta CA = s$, $\Delta CR = -s-a+2$, $\Delta PE = -\frac{1}{3}(s-a)$ (for edges removed from T^{sk}_{ref}) while $\Delta PC \leq \frac{2}{3}(a-1)$ (for removing potential of $a-1$ C-comps, each $-\frac{2}{3}$ or 0) hence

$$\Delta PromCost = s - s - a + 2 - \tfrac{1}{3}(s-a) + \tfrac{2}{3}(a-1) = \tfrac{1}{3}(4-s) \leq 0.$$

However, the new C-comp that we create can be trivial; in this case we need to increase the estimate of ΔPC by $\frac{2}{3}$. If that C-comp had but one edge left, this edge would be removed from T_{ref} and T^{sk}_{ref}, which decreases the estimate of ΔCR by 1 and ΔCE by $\frac{1}{3}$. If that C-comp had two edges left, we would remove them from T^{sk}_{ref} using a Path step, this does not change CR but decreases CE by $\frac{2}{3}$. Therefore our estimate of *PromCost* does not increase. ❐

Once we collapsed s-stars for $s > 3$ we redistribute potential between C-comps and S-comps by increasing potential of each nontrivial C-comp by $\frac{1}{6}$ bringing it to $-\frac{1}{2}$ and decreasing potential of one of its S-comps to $-\frac{1}{6}$. This will replace conditions (i)-(ii) with

(i') each C-comp C has $p(C) \geq -\frac{1}{2}$ and each trivial C-comp (with at most one edge) has $p(C) = 0$;

(ii') each S-comp S has $p(S) \geq -\frac{1}{6}$;

Lemma 3 *After collapsing a 3-star, conditions (i')-(ii') are satisfied and PromCost does not increase.*

Proof. Suppose that the terminals of the selected star S belong to 3 different C-comps. Then we replace two cost-2 connections from T_{ref} with 3 collapsed edges, while we decrease the number of C-comps by 2, thus

$$\Delta PromCost = \Delta CA + \Delta CR + \Delta PC \leq 3 - 4 + 2\tfrac{1}{2} = 0.$$

Suppose that the terminals of S belong to 2 different C-comps. $\Delta CR = 3$ because we remove one cost-2 connection from T_{ref} and one edge from an S-comp. This S-comp becomes a 2-star, hence we remove it from T using a Path Step, so together we remove 3 edges from T^{sk}_{ref} and $\Delta PE = 1$.

One S-comp disappears, so $\Delta PS = -\frac{1}{6}$. Because we collapse two C-comps into one, $\Delta PC = -\frac{1}{2}$. Consequently,

$$\Delta PromCost = 3 - 3 - 3\tfrac{1}{3} + \tfrac{1}{2} + \tfrac{1}{6} < 0.$$

If the terminals of the selected star belong to a single C-comp and we remove 2 edges from a single S-comp, we also remove the third edge of this S-comp and $\Delta CR = -3$, while $\Delta PE = -1$, $\Delta PS = \frac{1}{6}$, and if its C-comp degenerates to a single node, we have $\Delta PC = \frac{1}{2}$ (otherwise, zero). This yields the same change in *PromCost* as the previous case.

Finally, if the terminals of the selected star belong to a single C-comp and we remove 2 edges from two S-comps, we have $\Delta CA + \Delta CR = 1$. Because we apply Path Steps to those two S-comps, $\Delta PE = -2$. while $\Delta PS = \frac{1}{3}$ and $\Delta PC \leq \frac{1}{2}$. Thus $\Delta PromCost$ is at most $-\frac{1}{6}$. ❐

To complete the proof of Theorem 1 it suffices to see that when no more star collapsing is possible, T_{ref} consists of cost 2-connections, $T^{sk}_{ref} = \varnothing$ and thus the remaining potential is zero. Each finishing step increases CA by 2 and decreases CR by 2, with no changes in *PromCost*. When we terminate, we have a solution with cost $CA = PromCost$.

4 3/2 Approximation for GST with Distances 1 and 2

In the heuristic for STP[1,2]we could start with **Preprocessing** in which we collapsed every edge (cost-1 connection) between terminals, arguing that such an edge can be forced as a component of an optimum solution T^*. In GST[1,2]this is no longer valid, because this could be an edge between different connected components of T^*. Indeed, we need to increase our potential and thus *PromCost* to create a "budget" for this class of wrong selections: connecting sets that should not be connected.

Instead, we can start with the following preprocessing that is safe in the context of GST[1,2]:

> **G-Preprocessing:** While there exists an edge or an s-star (with $s \geq 3$) contained in one of the required sets R_i, collapse it.

We can also normalize the optimum solution T^* to assure these two properties: Steiner nodes have degree at least 3 and cost-2 connections connect only pairs of terminals from the same required set R_i. Steiner nodes of degree 1 can be obviously removed, Steiner nodes of degree 2 and cost-2 connections can be removed, and reconnection, if needed, can be achieved by connecting terminals that have to be connected.

Because the terminals of the edge (s-star) selected by G-Preprocessing is surely contained in a single connected component of the optimum forest T^*, while we increase CA by 1 (s), we decrease the cost of the T^* of the residual problem by 1 ($s-1$), thus preserving the approximation ratio of 1/1 ($s/(s-1)$).

Thus we can proceed with the assumption that no steps of G-Preprocessing can be performed. After the preprocessing, we can perform normalization of the "reference tree" T_{ref} that we initialize with T^*. Because T_{ref} has multiple connected component and it may also contain edges between terminals, we introduce two new notions:

- *F-comps* which are connected components of the forest T_{ref};
- *T-comps* which are connected components of the subgraph of T^{sk}_{ref} that is induced by the terminals.

We also introduce the second component of the potential, p_g, such that the sum of all p(object) and p_g(object) does not exceed $\frac{1}{2}cost(T^*)$. We will use p_g to cover the cost of connections made between different F-comps, the class of errors that are specific to the generalized problem.

We can give $p_g(e) = \frac{1}{6}$ for every edge of $T^* \cap E$, for edges inside a T-comp we can increase it to $p_g(e) = \frac{1}{2}$. For each non-edge e' in T^* we can give $p_g(e') = 1$. Moreover, to each initial C-comp C we can give $p_g(C) = -\frac{2}{3}$. Let $p_g(F)$ be the sum of p_g potentials of objects contained in an F-comp F.

We can define $PromCost' = PromCost + PF$ where PF is the sum of all $p_g(F)$'s. Our goal is to build a solution by collapsing selected connections without increasing

PromCost'. When we make a connection within an F-comp, we do not increase *PromCost* and PF does not increase either.

When we make a selection that connects two F-comps, say F_1, F_2 into $F = F_1 \cup F_2$, we can cover the cost of that connection using $p_g(F_1)$, and F can use $p_g(F_2)$ for a future connection with another F-comp. Because we will not connect distinct F-comps with non-edges, such a connection costs at most $\frac{3}{2}$ (this is the cost of connections made by a 3-stars, larger stars and edges make connections with a smaller cost). This is safe if $p_g(F_i) \geq \frac{3}{2}$.

Lemma 4 *If a required set of terminals R_i has more than 2 nodes, it is contained in an initial F-comp F such that $p_g(F) \geq \frac{3}{2}$.*

Proof. Tree F may contain three kinds of connections: e is a T-connection if it is an edge between terminals, and $p_g(e) = \frac{1}{2}$, a 2-connection if it is a non-edge, and $p_g(e) = 1$ and a C-connection, any other edge, and $p_g(e) = \frac{1}{6}$.

If F contains a C-connection, it contains a Steiner node, and thus at least 3 C-connections and a C-comp; those objects alone give p_g of $3\frac{1}{6} + \frac{2}{3} = \frac{3}{2} - \frac{1}{6}$. If there are no other connections in F, it is a 3-star, but in this case all terminals of that star are in R_i and we would collapse it in G-Preprocessing. And any other connection would increase $p_g(F)$ to at least $\frac{3}{2}$.

Now we assume that F does not contain C-connections. If it contains a 2-connection, it must contain another connection as well, and the least possible $p_g(F)$ is $1 + \frac{1}{2}$ if this other connection is a T-connection. In the remaining case, F has some a terminals, $a - 1$ T-connections and $p_g(F) = (a-1)\frac{1}{2}$. Again, if $a > 3$ then $p_g(F)$ is sufficiently high and if $a = 3$ then only R_i are terminals of F and the T-connections would be collapsed in G-preprocessing. ❒

We can also observe that

Lemma 5 *If a required set of terminals R_i has 2 nodes, it is contained in an initial F-comp F such that $p_g(F) \geq 1$.*

For this reason, it always safe to collapse edges between terminals. However, the status of the resulting merged sets of terminals requires some reasoning. Let us say that a set of terminals F is *safe* if it has $p_g(F) \geq \frac{3}{2}$. When we merge two sets of terminals, F_1 and F_2 using a connection with cost c, the union $F = F_1 \cup F_2$ will get $p_g(F) = p_g(F_1) + p_g(F_2) - c$. If $c = 1$, then union is safe as long as at least one of F_1, F_2 is safe, but not otherwise. However, suppose that after a union creating a larger unsafe set F an edge (of the residual graph) is contained in F. Then the balance of F is more favorable, by 1, then our pessimistic reasoning that deemed F unsafe, and this suffices to tag is safe.

Thus we can perform a bit bolder preprocessing if we keep track which resulting requirement sets are safe, and which are not.

GE-Preprocessing (G-preprocessing, extended version): Tag each required set of terminals R_i as safe if $|R_i| > 2$ and unsafe otherwise. While you can, do the following: collapse s-star contained in a required set and collapse any edge between two terminals. In the latter case, if these two terminals were in two different required sets and thus the collapsing replaces them with their union, tag the union safe if at least one of the merged requirement was safe. Moreover, if the collapsed edge is contained in some requirement set, tag that set safe.

We are now left with the problem: what to do with the unsafe sets that remain after the GE-Preprocessing. To address this problem, we need a stronger version of Lemma 1.

Lemma 6 *Without increasing PromCost we can transform the optimum solution T^* into a reference solution that satisfies the conditions of Lemma 1 and in which each T-comp has a cost-1 connection to at most one Steiner point.*

Proof. The reasoning is the same as in Lemma 1, except that we need to perform Bridge Step in the situation when we have a T-comp connected to more then one Steiner node. If we cannot remove such a connection as a Path step, we break a C-comp into two, so each of the resulting parts has at least two edges adjacent to Steiner nodes (the sufficient premise for reasoning of the Bridge Step). ❒

Now we can justify

Annihilation of unsafe sets: After GE-Preprocessing, break each unsafe set of requirements into original requirements, connect them individually with cost-2 connections and remove from further consideration.

Lemma 7 **Annihilation of unsafe sets** *does not increase PromCost′.*

Proof. Consider an unsafe requirement R'. It is created from a union of some p pair requirements $R_1, \ldots, R_p$ (each with two terminals). Because R' remains unsafe, GE-preprocessing did not collapsed exactly $p-1$ 1-cost connections inside R', so it consists of $p+1$ connected components (T-comps). We increase CA by $p+1$ by replacing these $p-1$ connections with p non-edges.

A pair R_i that is connected separately in T_{ref} contributes 3 to *PromCost′* and after annihilation uses only the correct cost, 2, so this case has a surplus. One can see that the most tight case is when in T_{ref} every T-comp of R' is connected by an edge to some Steiner node (a connection to a terminal would be performed already). Thus we remove those connections and decrease CR by $p+1$, remove the connections made by GE-Preprocessing and decrease CA by $p-1$ and reconnect with p non-edges; this does not change $CA + CR$, while the sum of potentials can only decrease. ❒

We conclude that after the Annihilation of unsafe sets we can proceed with the heuristic described in the previous section without increasing $PromCost'$, and $PromCost'$ is initialized as not larger than $\frac{3}{2}cost(T^*)$.

We construct now our approximation algorithm to consist of GE-Preprocessing followed by Annihilation of unsafe sets and followed by Rayward-Smith heuristics.

With the above we have the following main result.

Theorem 2 *There exists a polynomial time 3/2-approximation algorithm for the* GST[1,2].

References

[1] A. Agrawal, P.N. Klein, and R. Ravi, *When trees collide: An approximation algorithm for the generalized Steiner tree problem on networks*, (preliminary version appeared in Proc. 23nd STOC, (1991), 134-144), the journal version appeared in SIAM J. Comput. 24 (1995), 440-456.

[2] P. Berman and M. Karpinski, *8/7-Approximation Algorithm for (1,2)-TSP*, Proc. 17th ACM-SIAM SODA (2006), 641-648.

[3] P. Berman, M. Karpinski, and A. Zelikovsky, *1.25 Approximation Algorithm for the Steiner Tree Problem with Distances One and Two*, CoRR abs/0810.1851 (2008).

[4] M. Bern and P. Plassmann, *The Steiner problem with edge lengths 1 and 2*, Information Processing letters **32**:171-176, 1989.

[5] C. Borradaile, P. Klein and C. Mathieu, *A Polynomial Time Approximation Scheme for Euclidean Steiner Forest*, Proc. 49th IEEE FOCS (2008).

[6] V.J. Rayward-Smith and A.R. Clare, *The computation of nearly minimal Steiner trees in graphs*, Internat. J. Math. Educ. Sci. Tech. **14**:15-23, 1983.

[7] G. Robins, A. Zelikovsky, *Tighter Bounds for Graph Steiner Tree Approximation*, SIAM Journal on Discrete Mathematics, **19**(1):122-134 (2005). (Preliminary version appeared in Proc. SODA (2000), 770-779).