On the terminal Steiner tree problem

Guohui Lin a,1, Guoliang Xue b,*,2

a Department of Computing Science, University of Alberta, Edmonton, AB, T6G 2E8, Canada
b Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287-5406, USA

Received 29 October 2001; received in revised form 28 December 2001
Communicated by F.Y.L. Chin

Abstract

We investigate a practical variant of the well-known graph Steiner tree problem. In this variant, every target vertex is required to be a leaf vertex in the solution Steiner tree. We present hardness results for this variant as well as a polynomial time approximation algorithm with performance ratio $\rho + 2$, where $\rho$ is the best-known approximation ratio for the graph Steiner tree problem.

Keywords: Approximation algorithms; Steiner minimum tree; Terminal Steiner tree

1. Introduction

In the well-known graph Steiner tree problem (STP), we are given a positively edge-weighted graph and a subset of target vertices in the graph. The goal is to find a minimum weight subgraph interconnecting all target vertices, where the weight of the subgraph is the sum of the weights of the edges therein. From the nature of the problem, it is assumed that the given graph is connected. The minimization objective implies that the interconnection subgraph is a tree. To distinguish from the minimum spanning tree (MST) problem, the tree computed for STP may contain some non-target vertices for the sake of reducing weight. These non-target vertices are called Steiner vertices and the tree achieving the minimum weight is called a Steiner minimum tree (SMT) for the target vertices.

The STP, as well as the MST problem, has applications in many practical problems such as VLSI global and local routing, telecommunications, and transportation. It has received a lot of attention in the last several decades and some of the important results can be found in [9,3,2,6,12,1,5,4,8,11], which include hardness results, (in)approximabilities, and quite a few approximation algorithms (see also [7] for a survey) together with their performance analyses.

In some applications, the target vertices are required to be leaves in the Steiner tree. For example, in VLSI global routing, the target vertices correspond to pins and gates that are not allowed to overlap with the Steiner vertices introduced to reduce the total weight. In telecommunications, the message senders and re-
receivers are not allowed to act as transmitters which correspond to the Steiner vertices. Therefore, in the solution tree, it is desirable to have all target vertices as leaf vertices. We study the following terminal Steiner tree problem or TES, for short, which is a graph TSP problem with an additional requirement that every target vertex must be a leaf vertex in the output interconnection tree.

For feasibility, we assume that there exists an optimal solution tree in which every target vertex appears as a leaf vertex. Therefore, in the given graph every target vertex should be adjacent to some non-target vertex. Furthermore, we may compute, for each pair of vertices $u$ and $v$, a shortest $u\rightarrow v$ path (with weight denoted by $w(u, v)$) that does not contain any target vertex as an internal vertex. We then add the edge $(u, v)$ (or substitute if the edge is already in) with weight $w(u, v)$ into the graph. This operation will not change the corresponding Steiner tree problem. As a result, we may assume without loss of generality that the given graph is complete and the edge weights satisfy the triangle inequality, that is, for every three vertices $u, v, x$: $w(u, v) \leq w(u, x) + w(x, v)$. Throughout the paper, an approximation algorithm means a polynomial time approximation algorithm. We use $\rho$ to denote the best-known approximation ratio for the TSP.

Let $N(V, w, T)$ be a positively edge-weighted complete graph with vertex set $V$, edge weighting function $w$ satisfying the triangle inequality, and target set $T \subset V$. The terminal Steiner tree problem (TES) asks for a minimum weight subgraph interconnecting all target vertices such that every target vertex appears as a leaf vertex in the subgraph.

In Section 2, we will show that the TES problem is NP-hard and MAX SNP-hard (APX-hard) [10]. In Section 3, we will present a polynomial time approximation algorithm for TES with a approximation ratio $\rho + 2$, where $\rho$ is the best-known approximation ratio for STP. We conclude this paper in Section 4.

2. Hardness

In this section, we will show that the TES problem is MAX SNP-hard. Because the techniques used here are standard reductions, we will present a high level description only. The proof is done via an $L$-reduction from STP on graphs whose edge weights are either 1 or 2. A graph whose edge weights are either 1 or 2 is called a 1, 2-edge-weighted graph.

**Lemma 2.1** [3]. STP restricted to the 1, 2-edge-weighted graphs is MAX SNP-hard.

Given a 1, 2-edge-weighted complete graph $G$ with vertex set $U$ and target set $S \subset U$, for each vertex $a \in S$ create a new vertex $t(a)$ and a new edge $(a, t(a))$ where $t(a)$ is taken as a target vertex in the TES instance $I$. The target set in $I$ is $T = \{ t(a), a \in S \}$ and the vertex set is $V = U \cup T$. Note that $a$ is no longer a target in instance $I$. The weight of edge $(u, v)$ for $u, v \in U$ is not changed. The weight of edge $(a, t(a))$ is set at $\varepsilon$, which is a sufficiently small positive number less than $1/|S|$. Completing the above graph gives the instance $I$ of TES, denoted by $N(V, w, T)$.

Based on the above polynomial time transformation, it is easy to see that whenever there is a Steiner tree $S$ interconnecting vertices in $S$ in graph $G$, we may trivially construct a Steiner tree $T$ interconnecting target vertices in $T$ in graph $N$ by adding edges in the form $(a, t(a))$ for all $a \in S$ into $S$. One can verify that $w(T) = c(S) + |S|\varepsilon < c(S) + 1$. On the other hand, given any feasible Steiner tree $T_1$ interconnecting target vertices in $T$ in graph $N$ (every target appears as a leaf in $T_1$), we can get another feasible Steiner tree $T$, with weight no more than $w(T_1)$, interconnecting targets in $T$ in which every target vertex appears as a leaf in $T$. Therefore we have the following hardness result.

**Theorem 2.2.** The TES problem is MAX SNP-hard.

3. Approximations

In this section, we will present two provably good polynomial time approximation algorithms for TES. The first approximation algorithm is easy to understand, with an approximation ratio of $1 + 2\rho$. The sec-
ond approximation algorithm is more involved, with an improved approximation ratio of $2 + \rho$.

Our first approximation algorithm consists of three phases: shrink, frame, and assembly. Let $N(V, w, T)$ be a given instance of the TESST problem. In the shrink phase, we search for each target $t \in T$ a Steiner vertex $s(t) \in V - T$ such that $w(t, s(t))$ is the minimum over all choices of Steiner vertices (ties are broken arbitrarily). In the frame phase, we restrict our attention to computing an approximate Steiner tree in the induced subgraph $N[V - T]$ where the set of “target” vertices is $\{s(t), t \in T\}$. We can employ the best known approximation algorithm for STP and let $T_0$ denote the output Steiner tree. In the assembly phase, we attach $|T|$ edges in the form $(t, s(t))$ for all targets $t \in T$ to $T_0$ to form a genuine tree $T$ interconnecting all the target vertices in $T$. The high-level description of the algorithm is depicted in Fig. 1.

**Lemma 3.1.** Suppose there is an approximation algorithm for STP with performance ratio $\rho$, then there is a corresponding approximation algorithm for TESST with performance ratio $1 + 2\rho$.

**Proof.** Let $T^*$ denote an optimal interconnection tree for target vertices in $T$ such that every target in $T$ is a leaf in $T^*$. Let $s^*(t)$ denote the Steiner vertex in $T^*$ to which target $t$ is adjacent. By definitions,

$$w(s^*(t), t) \geq w(s(t), t).$$

(3.1)

Therefore,

$$w(T^*) \geq \sum_{t \in T} w(s^*(t), t) \geq \sum_{t \in T} w(s(t), t).$$

(3.2)

From $T^*$, if we delete edges of form $(s^*(t), t)$ and add in edges $(s^*(t), s(t))$ correspondingly, for all $t \in T$, then we will get a Steiner tree interconnecting vertices in $\{s(t), t \in T\}$ in the induced subgraph $N[V - T]$. Denote this Steiner tree using $T'$. It follows from triangle inequality $w(s^*(t), s(t)) \leq w(s^*(t), t) + w(t, s(t))$ that

$$w(T') \leq w(T^*) + \sum_{t \in T} w(s(t), t).$$

(3.3)

Let $T^*$ denote a minimum weight Steiner tree interconnecting vertices in $\{s(t), t \in T\}$ in the induced subgraph $N[V - T]$ and recall that $T_0$ denotes the Steiner tree that we compute to interconnect vertices in $\{s(t), t \in T\}$ in the induced subgraph $N[V - T]$. Therefore, $w(T_0) \leq \rho w(T^*) \leq \rho w(T')$,

where $\rho$ is the performance ratio of the (best-known) approximation algorithm for STP that we use to compute tree $T_0$ (the best value of $\rho$ aware to us is about 1.550 [11]). Note that $T$ denotes the tree obtained by attaching all edges of form $(s(t), t)$ onto tree $T_0$. It follows that

$$w(T) = w(T_0) + \sum_{t \in T} w(s(t), t)$$

$$\leq \rho w(T') + \sum_{t \in T} w(s(t), t)$$

$$\leq \rho w(T^*) + (1 + \rho) \sum_{t \in T} w(s(t), t)$$

$$\leq (1 + 2\rho) w(T^*).$$

This proves the lemma. □

The number of operations required by the shrink phase is bounded by the number of edges in the graph and the number of operations required by the assembly phase is bounded by $O(|T|)$. Therefore the
**Algorithm II**

| Shrink-and-partition phase:
| --- |
| for each target \( t \in T \), find vertex \( s(t) \);
| compute a minimum weight perfect matching \( M \) for \( T \) in \( N[T] \);
| partition \( T \) into \( T_1 \) and \( T_2 \) according to matching \( M \);

| Frame phase:
| --- |
| compute a Steiner tree \( T_0 \) interconnecting \( \{s(t_1), t_1 \in T_1\} \) in the induced subgraph \( N[V - T] \);

| Assembly phase:
| --- |
| add edges of form \( (t_1, s(t_1)) \), for all \( t_1 \in T_1 \), to \( T_0 \);
| add edges of form \( (t_2, s(t_1)) \), for all \( t_2 \in T_2 \) such that \( (t_1, t_2) \in M \), to \( T_0 \);

**Output** the resultant tree \( T \).

Fig. 2. A high-level description of the second approximation.

The time complexity of Algorithm I is bounded by the time complexity of the algorithm used for computing a \( \rho \)-approximation to the STP in the frame phase, which varies depending on the approximation algorithm being used.

In the following, we will present an improved approximation algorithm with a performance ratio of \( 2 + \rho \). Unless specified otherwise, the notations used in the following are those used in our first approximation algorithm. This improved approximation again has three phases: shrink-and-partition, frame, and assembly.

In the shrink-and-partition phase, we first determine the Steiner vertex \( s(t) \) for every target \( t \) in \( T \) and then partition the target set \( T \) into two subsets \( T_1 \) and \( T_2 = T \setminus T_1 \), to be specified. When \(|T|\) is even, we can compute a minimum weight perfect matching \( M \) of \( T \) in the induced subgraph \( N[T] \) (recall that \( N \) is complete). Two target vertices connected by an edge in \( M \) form a pair. We assume that for every edge \( (t_1, t_2) \in M \), \( w(s(t_1), t_1) \leq w(s(t_2), t_2) \); and further let \( T_1 = \{t_1, (t_1, t_2) \in M \} \). When \(|T|\) is odd, we arbitrarily pick a target \( t \in T \), after performing the partition \( T \setminus \{t\} = T_1 + T_2 \) as in the above, we add \( t \) into \( T_1 \).

In the frame phase, we compute a \( \rho \)-approximation \( T_0 \) to the Steiner minimum tree interconnecting \( \{s(t_1) \mid t_1 \in T_1\} \) in the induced subgraph \( N[V - T] \). This differs from the frame phase in the first algorithm because we are now interconnecting only half of the “targets” \((|T_1| = |T|/2)\). In the assembly phase, we add edges in the form \((t_1, s(t_1))\) to \( T_0 \) for every \( t_1 \in T_1 \). We also add edges in the form \((t_2, s(t_1))\) to \( T_0 \) if \((t_1, t_2) \in M \). This turns \( T_0 \) into a feasible Steiner tree \( T \), which is our approximate solution. The high-level description of the algorithm is depicted in Fig. 2. In the following we will analyze the worst-case performance of the algorithm.

Recall that \( T^* \) denotes an optimal interconnection tree for target set \( T \) such that every target in \( T \) is a leaf. By duplicating edges in tree \( T^* \) and constructing an Euler traversal of the duplicate and original edges, we can further shortcut a Hamiltonian cycle for the target vertices (excluding all Steiner vertices). This indicates that

\[
w(M) \leq w(T^*).
\]

A little different from the first approximation algorithm, this time let \( T' \) denote the Steiner tree obtained from tree \( T^* \) by deleting edges of form \((s^*(t_1), t)\) for all \( t \in T \) and adding in edges \((s^*(t_1), s(t_1))\) for all \( t_1 \in T_1 \). Let \( T'^\rho \) denote a minimum weight Steiner tree interconnecting vertices in \( \{s(t_1), t_1 \in T_1\} \) in the induced subgraph \( N[V - T] \) and recall that \( T_0 \) denotes a \( \rho \)-approximation to the Steiner minimum tree interconnecting vertices in \( \{s(t_1), t_1 \in T_1\} \) in the induced subgraph \( N[V - T] \). Then,

\[
w(T_0) \leq \rho w(T'^\rho) \leq \rho w(T')
\]

\[
\leq \rho \left( w(T^*) + \sum_{(t_1, t_2) \in M} (w(s(t_1), t_1) - w(s^*(t_2), t_2)) \right)
\]

\[
\leq \rho w(T^*),
\]

since \( w(s(t_1), t_1) \leq w(s(t_2), t_2) \leq w(s^*(t_2), t_2) \).

After we get tree \( T_0 \), we attach both edges \((s(t_1), t_1)\) and \((s(t_1), t_2)\) to Steiner vertex \( s(t_1) \), for every target
pair $t_1$ and $t_2$ (we attach edge $(s(t_1), t_1)$ to Steiner vertex $s(t_1)$ if $t_1$ is not paired up). The achieved tree $T$ is a feasible Steiner tree since every target appears as a leaf. To estimate the weight of $T$, we need the inequalities

$$ w(s(t_1), t_2) \leq w(s(t_1), t_1) + w(t_1, t_2) $$

$$ \leq w(s(t_2), t_2) + w(t_1, t_2). $$

We have

$$ w(T) = w(T_0) + \sum_{(t_1, t_2) \in M} \left( w(s(t_1), t_1) + w(s(t_1), t_2) \right) $$

$$ \leq w(T_0) + \sum_{(t_1, t_2) \in M} \left( w(s(t_1), t_1) + w(s(t_2), t_2) + w(t_1, t_2) \right) $$

$$ = w(T_0) + \sum_{t \in T} w(s(t), t) + w(M) $$

$$ \leq (\rho + 2)w(T^*), $$

according to Eqs. (3.2) and (3.4). Therefore we have proved the following theorem.

**Theorem 3.2.** Suppose there is an approximation algorithm for STP with performance ratio $\rho$, then there is a corresponding approximation algorithm for TeST with performance ratio $\rho + 2$.

Note that $2 + \rho < 1 + 2\rho$ where $\rho$ is the best-known approximation ratio (close to 1.550) for STP [11]. Therefore the approximation ratio of ALGORITHM II is better than that of ALGORITHM I. Like ALGORITHM I, the worst-case running time of ALGORITHM II is also bounded by the time complexity of the $\rho$-approximation algorithm for STP that is being used in the frame phase of our algorithm.

### 4. Conclusions

In this paper, we have studied TeST, the terminal Steiner tree problem, which is a variant of the well-known graph Steiner tree problem. In TeST, every target vertex has to be a leaf in the solution tree. We proved that the TeST problem is NP-hard and MAX SNP-hard. We then presented a polynomial time approximation algorithm with performance ratio $\rho + 2$, where $\rho$ is the best-known approximation ratio for STP. It would be interesting and challenging to design better approximation algorithms.

### Acknowledgements

We thank the handling editor and two anonymous referees for their helpful comments and suggestions on an earlier version of this paper.

### References