An Improved Approximation Algorithm for the Terminal Steiner Tree Problem

Yen Hung Chen

Department of Computer Science, Taipei Municipal University of Education,
Taipei 10042, Taiwan, R.O.C.
yhchen@tmue.edu.tw
http://tmue.edu.tw/~yhchen/

Abstract. Given a complete graph $G = (V, E)$ with a length function on edges and a subset R of V, the terminal Steiner tree is defined to be a Steiner tree in G with all the vertices of R as its leaves. Then the terminal Steiner tree problem is to find a terminal Steiner tree in G with minimum length. In this paper, we present an approximation algorithm with performance ratio $2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}$ for the terminal Steiner tree problem, where ρ is the best-known performance ratio for the Steiner tree problem with any $\alpha \geq 2$. When we let $\alpha = 3.87 \approx 4$, this result improves the previous performance ratio of 2.515 to 2.458.

Keywords: Approximation algorithms, NP-complete, Steiner tree, terminal Steiner tree problem, multicast routing, evolutionary tree reconstruction in biology, telecommunications.

1 Introduction

Given an arbitrary graph $G = (V, E)$, a subset $R \subseteq V$ of vertices, and a length (or weight) function $\ell : E \rightarrow R^+$ on the edges, a *Steiner tree* is an acyclic subgraph of G that spans all vertices in R. The given vertices R are usually referred to as *terminals* and other vertices $V \setminus R$ as *Steiner* (or *optional*) vertices. The length of a Steiner tree is defined to be the sum of the lengths of all its edges. The *Steiner tree problem* (STP for short) is concerned with the determination of a Steiner tree with minimum length in G [6, 8, 16]. This problem has been shown to be NP-complete [11] and MAX SNP-hard [2]. So, many approximation algorithms with constant ratios have been proposed [1, 3, 13, 18, 23–28] instead of exact algorithms. It has been shown that STP has many important applications in VLSI design, network communication, computational biology and so on [4, 6, 8, 9, 12, 16, 17, 19].

Motivated by the reconstruction of evolutionary tree in biology, Lu, Tang and Lee studied a variant of the Steiner tree problem, called as the *full Steiner tree problem* [21]. Independently, motivated by VLSI global routing and telecommunications, Lin and Xue defined the *terminal Steiner tree problem* (TSTP for short), which is equal to the full Steiner tree problem [20]. A Steiner tree is a terminal Steiner tree if all terminals are the leaves of the Steiner tree [3, 16, 21].

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 141–151, 2011.
© Springer-Verlag Berlin Heidelberg 2011

The TSTP is concerned with the determination of a terminal Steiner tree for R in G with minimum length. The problem is shown to be NP-complete and MAX SNP-hard [20], even when the lengths of edges are restricted to be either 1 or 2 [21]. However, Lu, Tang and Lee [21] gave a $\frac{8}{5}$-approximation algorithm for the TSTP with the restriction that the lengths of edges are either 1 or 2, and Lin and Xue [20] presented a $(\rho + 2)$-approximation algorithm for the TSTP if the length function is *metric* (i.e., the lengths of edges satisfy the triangle inequality), where ρ is the best-known performance ratio for the STP whose performance ratio is $1 + \frac{\ln 3}{2} \approx 1.55$ [24]. Then Chen, Lu and Tang [5], Fuchs [10], Drake and Hougardy [7], designed 2ρ-approximation algorithms for the TSTP if the length function is *metric*, independently. Martineza, Pinab, Soares [22] proposed an approximation algorithm to improve the performance ratio to $2\rho - (\frac{\rho}{3\rho-2})$ which is the best-known performance ratio. For other related results, Chen, Lu and Tang [5] also gave an $O(|E| \log |E|)$ time algorithm to optimally solve the *bottleneck terminal Steiner tree problem*. Drake and Hougardy [7] proved that TSTP does not exist an approximation algorithm with the performance ratio better than $(1-o(1))ln\ n$ unless $NP = DTIME(|V|^{O(loglog|V|)})$ when the length function is not *metric*. In this paper, we present an approximation scheme with performance ratio of $2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}$ for the TSTP, where $\alpha \geq 2$. This algorithm is more general than Martineza, Pinab, Soares' algorithm [22]. When we let $\alpha = 2$, this algorithm achieves a performance ratio of $2\rho - (\frac{\rho}{3\rho-2})$. When we let $\alpha = 4$ (respectively, $\alpha = 3$), this algorithm has a performance ratio of $2\rho - (\frac{6\rho}{10\rho-1})$ (respectively, $2\rho - (\frac{3\rho}{6\rho-2})$), which improves the previous result $2\rho - (\frac{\rho}{3\rho-2})$ of [22].

The rest of this paper is organized as follows. In Section 2, we describe a $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation algorithm for the TSTP. We make a conclusion in Section 3.

2 A $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-Approximation Algorithm for the TSTP

TSTP (Terminal Steiner Tree Problem)
Instance: A complete graph $G = (V, E)$ with $\ell : E \to R^+$, and a proper subset $R \subset V$ of terminals, where the length function ℓ is metric.
Question: Find a terminal Steiner tree for R in G with minimum length.

In this section, we will present a $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation algorithm to solve the above TSTP, whose length function is metric, in polynomial time. By definition, any terminal Steiner tree T for R in $G = (V, E)$ contains no edge in $E_R = \{(u, v)|u, v \in R, u \neq v\}$. Hence, throughout the rest of this paper, we assume that G contains no edge in E_R (i.e., $E \cap E_R = \phi$). We use $L(H)$ to denote the length of any subgraph H of G (i.e., $L(H)$ equals to the sum of the lengths of all the edges of H). Let T_{OPT} be the optimal terminal Steiner tree for R in G. For convenience, we let $N_G(r)$ be the set of the neighbors of $r \in R$ in a graph G (i.e., $N_G(r) = \{v|(r,v) \in E\}$ and its members are all Steiner vertices) and $\hat{n}_r$ be the nearest neighbor of r in G (i.e., $\ell(r, \hat{n}_r) = \min\{\ell(r,v)|v \in N_G(r)\}$). We also use $E_{\hat{N}}$ to denote the edge set that contains edges $(r, \hat{n}_r)$ for all $r \in R$. Let $\mathcal{A}_{STP}$ be the best-known approximation algorithm for the STP, whose performance ratio $\rho = 1 + \frac{\ln 3}{2} \approx 1.55$ [24]. For any real number $\alpha \geq 2$, our approximation algorithm first constructs two terminal Steiner trees T_{APX1} and T_{APX2}. Then output the minimum length between T_{APX1} and T_{APX2}. It will show that if $L(E_{\hat{N}}) > (\frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}) * L(T_{OPT})$, we have

$$L(T_{APX1}) \leq (2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})L(T_{OPT}).$$

Otherwise,

$$L(T_{APX2}) \leq (2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})L(T_{OPT}).$$

Hence, we construct the terminal Steiner tree T_{APX1} that is a $2\rho - \frac{L(E_{\hat{N}})}{L(T_{OPT})}$-approximation solution of the TSTP in subsection 2.1. Then subsection 2.2 presents a $(\rho + (\frac{\alpha^2\rho+\alpha\rho-4\alpha+2}{\alpha^2-\alpha}) * \frac{L(E_{\hat{N}})}{L(T_{OPT})})$-approximation algorithm that outputs another terminal Steiner tree T_{APX2} for the TSTP. Finally, we show that the minimum length terminal Steiner tree between T_{APX1} and T_{APX2} is a $(2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation solution in subsection 2.3.

2.1 The Performance Ratio of $(2\rho - \frac{L(E_{\hat{N}})}{L(T_{OPT})})$ for the TSTP

In this section, we show a $(2\rho - \frac{L(E_{\hat{N}})}{L(T_{OPT})})$-approximation algorithm that is the same as in the previous result [5, 7, 10, 22]. First, we use algorithm $\mathcal{A}_{STP}$ to G and construct a Steiner tree $S = (V_S, E_S)$ for R in G. Note that if all vertices of R are leaves in S, then S is also a terminal Steiner tree of G. If not, we apply Algorithm 1 to transform it into a terminal Steiner tree. By definition, $N_S(r)$ is the set of the neighbors of $r \in R$ in S (i.e., $N_S(r) = \{v|(r,v) \in E_S\}$). Let ${n'}_r$ be the nearest neighbor of r in S (i.e., $\ell(r, n'_r) = \min\{\ell(r,v)|v \in N_S(r)\}$). We let $star(r)$ be the subtree of S induced by $\{(r,v)|v \in N_S(r)\}$. Fig. 1 shows the definitions of $star(r)$, n'_r and $N_S(r)$. Dashed edges are the edges in E not in E_S.

Algorithm 1. Method of transforming S into a terminal Steiner tree

For each $r \in R$ with $|N_S(r)| \geq 2$ in S **do**

1. Delete all the edges in $star(r) \setminus \{(r, n'_r)\}$ from S.
2. Let $G[N_S(r)]$ be the subgraph of G induced by $N_S(r)$. Then construct a minimum spanning tree $MST(N_S(r))$ of $G[N_S(r)]$, and add all the edges of $MST(N_S(r))$ into S.

end for

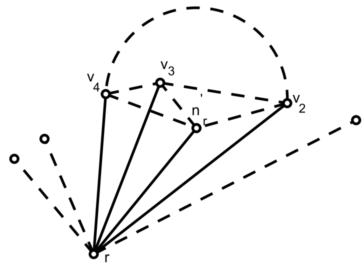

Fig. 1. The definitions of $star(r)$, n'_r and $N_S(r)$. $N_S(r) = \{n'_r, v_2, v_3, v_4\}$ and $star(r)$ is represented by solid edges and $r \cup N_S(r)$.

After running Algorithm 1, S becomes a terminal Steiner tree. By the previous result [5], the time-complexity of Algorithm 1 is $O(|V|^3)$.

Now, for clarification, we construct the terminal Steiner trees T_{APX1} for the TSTP as follows.

Algorithm $APX1$
Input: A complete graph $G = (V, E)$ with $\ell : E \rightarrow R^+$ and a set $R \subset V$ of terminals, where we assume that G contains no edge in E_R and the length function is metric.
Output: A terminal Steiner tree T_{APX1} for R in G.
1. /* Find a Steiner tree S in G */
Construct a Steiner tree S in G by using the currently best-known approximation algorithm $\mathcal{A}_{STP}$ for the STP.
2. /* Transform S into a terminal Steiner tree T_{APX1} */
If S is a terminal Steiner tree **then**
Let the Steiner tree S be T_{APX1}.
else
Transform S into a terminal Steiner tree by using Algorithm 1 and let T_{APX1} be such a terminal Steiner tree.

Theorem 1. *Algorithm APX1 is a* $(2\rho - \frac{L(E_{\hat{N}})}{L(T_{OPT})})$*-approximation algorithm for the TSTP.*

Proof. Note that the time-complexity of Algorithm APX1 is dominated by the cost of the step 1 for running the currently best-known approximation algorithm for the STP [24]. Let S_{OPT} be the optimal Steiner tree for R in G. Note that we use the currently best-known approximation algorithm $\mathcal{A}_{STP}$ for the STP to find a Steiner tree S for R in G. Hence, we have $L(S) \leq \rho * L(S_{OPT})$, where ρ is the performance ratio of $\mathcal{A}_{STP}$. Since T_{OPT} is also a Steiner tree for R in G, we have $L(S_{OPT}) \leq L(T_{OPT})$ and hence $L(S) \leq \rho * L(T_{OPT})$. Let R_1 be the set of all leaf terminals in S and R_2 is the set of all non-leaf terminals in S. Recall that in each iteration of Algorithm 1, we transform each terminal r in R_2 into a leaf by first deleting all the edges, except (r, n'_r), and then adding all the edges in $MST(N_S(r))$. Let k be $|N_S(r)|$. For all $r \in R_2$, let n''_r denote the second nearest neighbor of r in $N_S(r)$ and let $P = (v_1 \equiv n'_r, v_2, \ldots, v_k \equiv n''_r)$ be any arbitrary path visiting each vertex in $N_S(r)$ exactly once and both n'_r and n''_r are its end-vertices. By triangle inequality, we have the following inequalities.

$$\begin{aligned}\ell(v_1, v_2) &\le \ell(r, v_1) + \ell(r, v_2)\\ \ell(v_2, v_3) &\le \ell(r, v_2) + \ell(r, v_3)\\ &\vdots\\ \ell(v_{k-1}, v_k) &\le \ell(r, v_{k-1}) + \ell(r, v_k).\end{aligned}$$

By above inequalities, we have

$$\ell(v_1, v_2) + \ell(v_2, v_3) + \ldots + \ell(v_{k-1}, v_k) \le 2 * L(star(r)) - \ell(r, v_1) - \ell(r, v_k).$$

Consequently we have,

$$L(P) \le 2 * L(star(r)) - \ell(r, n'_r) - \ell(r, n''_r).$$

It is clear that $L(MST(N_S(r))) \le L(P)$ since $MST(N_S(r))$ is a minimum spanning tree of $G[N_S(r)]$. In other words, we have $L(MST(N_S(r))) \le 2 * L(star(r)) - \ell(r, n'_r) - \ell(r, n''_r)$. By construction of T_{APX1}, we have

$$\begin{aligned}L(T_{APX1}) &= L(S) + \sum_{r\in R_2} (L(MST(N(r))) - L(star(r)) + \ell(r, n'_r))\\ &\le L(S) + \sum_{r\in R_2} (L(star(r)) - \ell(r, n''_r)).\end{aligned}$$

Note that for any two terminals $r_i, r_j \in R$, $star(r_i)$ and $star(r_j)$ are edge-disjoint in S. Hence, we have $\sum_{r\in R_2} L(star(r)) \le L(S) - \sum_{r\in R_1} \ell(r, n'_r)$. As a result, we have

$$\begin{aligned}L(T_{APX1}) &\le 2 * L(S) - \sum_{r\in R_1} \ell(r, n'_r) - \sum_{r\in R_2} \ell(r, n''_r)\\ &\le 2 * L(S) - \sum_{r\in R} \ell(r, n'_r)\\ &\le 2\rho * L(T_{OPT}) - L(E_{\hat{N}}),\end{aligned}$$

and the result follows. □

2.2 The Performance Ratio of $(\rho + \frac{(\alpha^2\rho+\alpha\rho-4\alpha+2)}{(\alpha^2-\alpha)} * \frac{L(E_{\hat{N}})}{L(T_{OPT})})$ for the TSTP

In this section, we describe a $(\rho + \frac{(\alpha^2\rho+\alpha\rho-4\alpha+2)}{(\alpha^2-\alpha)} * \frac{L(E_{\hat{N}})}{L(T_{OPT})})$-approximation algorithm that is more general than the previous one [22]. To show the performance, we first construct a Steiner tree $S = (V_S, E_S)$ for R in G by using algorithm

$\mathcal{A}_{STP}$. Recall that if all vertices of R are leaves in S, then S is also a terminal Steiner tree of G. If not, we apply Algorithm 2 to transform it into a terminal Steiner tree. The definitions of $N_S(r)$ and $star(r)$ are also the same as in the previous section. We let $star(\hat{n}_r)$ be the subtree of G induced by $\{(\hat{n}_r, v)|v \in N_S(r)\}$. Note that $\hat{n}_r$ is the nearest neighbor of r in G (maybe not in $star(r)$). Fig. 2 shows the definition of $star(\hat{n}_r)$. Dashed edges and thick dashed edges are the edges in E not in E_S.

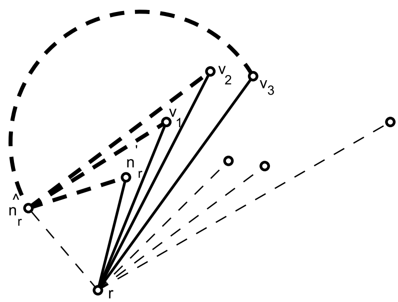

Fig. 2. The definition of $star(\hat{n}_r)$. $N_S(r) = \{n'_r, v_1, v_2, v_3\}$ and $star(\hat{n}_r)$ is represented by thick dashed edges and $\hat{n}_r \cup N_S(r)$.

Algorithm 2. Method of transforming S into a terminal Steiner tree
For each $r \in R$ with $|N_S(r)| \geq 2$ in S **do**
  1. Delete all the edges in $star(r)$ from S.
  2. add all the edges in $star(\hat{n}_r) \cup \{(r, \hat{n}_r)\}$ into S.
end for

Algorithm 2 is similar to the Algorithm 1, except adding all edges in $star(\hat{n}_r)$ instead of $MST(N_S(r))$ and $(r, \hat{n}_r)$ instead of (r, n'_r). Let $\tilde{S}$ be the Steiner tree after running Algorithm 2. Clearly, $\tilde{S}$ is a terminal Steiner tree. Since there are at most $|R|$ non-leaf terminals in S, there are at most $|R|$ iterations in Algorithm 2. For step 1, its total cost is $O(|E|)$ time since for any two non-leaf terminals r_i and r_j in S, we have $\{(r_i, v)|v \in N_S(r_i)\} \cap \{(r_j, v)|v \in N_S(r_j)\} = \phi$. In step 2, we need to find a $star(\hat{n}_r)$ in each iteration, which can be done in $O(|N_S(r)|)$ time. Hence, its total cost is $O(|V|^2)$ time. As a result, the time-complexity of Algorithm 2 is $O(|V||E| + |V|^2)$. Let $\alpha \geq 2$ be a real parameter. For each $r \in R$ and $v \in N_{\tilde{S}}(r)$, if $\ell(v, \hat{n}_r) \leq \ell(r, v) - \frac{\ell(\hat{n}_r, r)}{\alpha}$, we show that $\tilde{S}$ is a $(\rho + (1 - \frac{2}{\alpha})(\frac{L(E_{\hat{N}})}{L(T_{OPT})}))$-approximation solution of the TSTP by the next lemma.

Lemma 1. *For all $r \in R$ with $v \in N_S(r)$ and a real $\alpha \geq 2$, Algorithm 2 returns a terminal Steiner tree $\tilde{S}$ with $L(\tilde{S}) \leq L(S) + (1 - \frac{2}{\alpha}) * L(E_{\hat{N}})$ if $\ell(v, \hat{n}_r) \leq \ell(r, v) - \frac{\ell(\hat{n}_r, r)}{\alpha}$.*

Proof. Let R_1 be the set of all leaf terminals in S and R_2 is the set of all non-leaf terminals in S. For each $r \in R_2$, let k be $|N_S(r)|$. Let $(v_1, v_2, \ldots, v_k)$ be all vertices in $N_S(r)$. Since $\ell(v, \hat{n}_r) \leq \ell(r, v) - \frac{\ell(\hat{n}_r, r)}{\alpha}$, we have the following inequalities.

$$\ell(v_1, \hat{n}_r) \leq \ell(r, v_1) - \frac{\ell(\hat{n}_r, r)}{\alpha}$$
$$\ell(v_2, \hat{n}_r) \leq \ell(r, v_2) - \frac{\ell(\hat{n}_r, r)}{\alpha}$$
$$\vdots$$
$$\ell(v_k, \hat{n}_r) \leq \ell(r, v_k) - \frac{\ell(\hat{n}_r, r)}{\alpha}.$$

By above inequalities, we have $L(star(\hat{n}_r)) \leq L(star(r)) - \frac{k}{\alpha}\ell(\hat{n}_r, r)$ for $r \in R_2$. Recall that for any two terminals $r_i, r_j \in R$, $star(r_i)$ and $star(r_j)$ are edge-disjoint in S. By construction of $\tilde{S}$, we have

$$\begin{aligned} L(\tilde{S}) &= L(S) + \sum_{r \in R_2} (L(star(\hat{n}_r)) - L(star(r)) + \ell(\hat{n}_r, r)) \\ &\leq L(S) + \sum_{r \in R_2} \{(1 - \frac{k}{\alpha})\ell(\hat{n}_r, r)\} \\ &\leq L(S) + \sum_{r \in R_2} \{(1 - \frac{2}{\alpha})\ell(\hat{n}_r, r)\} \\ &\leq L(S) + (1 - \frac{2}{\alpha}) * L(E_{\hat{N}}). \end{aligned}$$ □

Since S is a ρ-approximation solution for the STP. By Lemma 1, $\tilde{S}$ is a $(\rho + (1 - \frac{2}{\alpha})(\frac{L(E_{\hat{N}})}{L(T_{OPT})}))$-approximation solution of the TSTP.

In the remaining paragraphs of this section, we construct a terminal Steiner tree T_{APX2} such that $L(T_{APX2}) \leq \rho * L(T_{OPT}) + \frac{(\alpha^2\rho + \alpha\rho - 4\alpha + 2)}{(\alpha^2 - \alpha)} * \frac{L(E_{\hat{N}})}{L(T_{OPT})}$. First, we modify the length function ℓ to a new length function $\tilde{\ell} : E \rightarrow R^+$ on the edges of G, such that each $r \in R$ and $v \in N_G(r)$, $\tilde{\ell}(v, \hat{n}_r) \leq \tilde{\ell}(r, v) - \frac{\tilde{\ell}(\hat{n}_r, r)}{\alpha}$. Then use Algorithm 2 to find a terminal Steiner tree $\tilde{S}$ that satisfies Lemma 1. Finally, we let $\tilde{S}$ be T_{APX2}. The new length function $\tilde{\ell}$ is defined by

$$\tilde{\ell}(u, v) = \begin{cases} \ell(u, v) + (\frac{1+\alpha}{\alpha - 1})\ell(u, \hat{n}_u) & \text{, if } u \in R \text{ and } v \in N_G(u) \\ \ell(u, v) & \text{, otherwise.} \end{cases} \quad (1)$$

For $r \in R$ and $v \in N_G(r)$, since $\ell(v, \hat{n}_r) \leq \ell(r, v) + \ell(\hat{n}_r, r)$ (i.e., metric), we have

$$\begin{aligned} \tilde{\ell}(v, \hat{n}_r) &= \ell(v, \hat{n}_r) \leq \ell(r, v) + \ell(\hat{n}_r, r) \\ &= \ell(r, v) + (\frac{1+\alpha}{\alpha - 1})\ell(\hat{n}_r, r) - \frac{\ell(\hat{n}_r, r) + (\frac{1+\alpha}{\alpha-1})\ell(\hat{n}_r, r)}{\alpha} \\ &= \tilde{\ell}(r, v) - \frac{\tilde{\ell}(\hat{n}_r, r)}{\alpha}. \end{aligned}$$

Now, for clarification, we construct the terminal Steiner trees T_{APX2} for the TSTP as follows.

Algorithm $APX2$
Input: A real $\alpha \geq 2$. A complete graph $G = (V, E)$ with $\ell : E \rightarrow R^+$ and a set $R \subset V$ of terminals, where we assume that G contains no edge in E_R and the length function is metric.
Output: A terminal Steiner tree T_{APX2} for R in G.
1. Use Eq. (1) to transform the length function ℓ to $\tilde{\ell}$.
2. /* Find a Steiner tree S in G with $\tilde{\ell}$ */
Use the currently best-known approximation algorithm $\mathcal{A}_{STP}$ for the STP to find a Steiner tree S in G with the length function $\tilde{\ell}$.
3. /* Transform S into a terminal Steiner tree T_{APX2} */
Use Algorithm 2 to transform S into a terminal Steiner tree $\tilde{S}$ and let $\tilde{S}$ be T_{APX2}.

Theorem 2. *Algorithm APX2 is a* $(\rho + \frac{(\alpha^2\rho+\alpha\rho-4\alpha+2)}{(\alpha^2-\alpha)} * \frac{L(E_{\hat{N}})}{L(T_{OPT})})$*-approximation algorithm for the TSTP.*

Proof. Note that the time-complexity of Algorithm APX2 is also dominated by the cost of the step 2 for running the currently best-known approximation algorithm for the STP [24]. Since we define a new length function $\tilde{\ell}$, let $\tilde{L}(H)$ and $\widetilde{T}_{OPT}$ denote the length of any subgraph H (i.e., $\tilde{L}(H)$ equals to the sum of the lengths of all the edges of H) and the optimal terminal Steiner tree for R in G with the length function $\tilde{\ell}$, respectively. It is clear that

$$\begin{aligned}\tilde{L}(S) &\leq \rho * \tilde{L}(\widetilde{T}_{OPT}) \leq \rho * \tilde{L}(T_{OPT}) \\ &\leq \rho * \{L(T_{OPT}) + \sum_{r \in R}(\frac{1+\alpha}{\alpha-1})\ell(\hat{n}_r, r)\} \\ &\leq \rho * \{L(T_{OPT}) + (\frac{1+\alpha}{\alpha-1})L(E_{\hat{N}})\}. \end{aligned} \quad (2)$$

By construction of T_{APX2}, we have

$$L(T_{APX2}) = L(\tilde{S}) \leq \tilde{L}(\tilde{S}) - \sum_{r \in R}(\frac{1+\alpha}{\alpha-1})\ell(\hat{n}_r, r) \leq \tilde{L}(\tilde{S}) - (\frac{1+\alpha}{\alpha-1})L(E_{\hat{N}}). \quad (3)$$

Since length function $\tilde{\ell}$ satisfies the property in Lemma 1, and hence we have

$$\tilde{L}(\tilde{S}) \leq \tilde{L}(S) + (1 - \frac{2}{\alpha})L(E_{\hat{N}}). \quad (4)$$

By Eqs. (2)–(4),

$$L(T_{APX2}) \leq \tilde{L}(\tilde{S}) - (\frac{1+\alpha}{\alpha-1})L(E_{\hat{N}})$$

$$\le \tilde{L}(S) + (1 - \frac{2}{\alpha})L(E_{\hat{N}}) - (\frac{1+\alpha}{\alpha - 1})L(E_{\hat{N}})$$
$$\le \rho * \{L(T_{OPT}) + (\frac{1+\alpha}{\alpha - 1})L(E_{\hat{N}})\} + \{\frac{\alpha - 2}{\alpha} - \frac{1+\alpha}{\alpha - 1}\}L(E_{\hat{N}})$$
$$\le \rho * L(T_{OPT}) + \frac{(\alpha^2\rho + \alpha\rho - 4\alpha + 2)}{(\alpha^2 - \alpha)}L(E_{\hat{N}}),$$

and the result follows. □

2.3 The Performance Ratio of $(2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$ for the TSTP

Finally, we present a $(2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation algorithm. First, we apply Algorithm APX1 and Algorithm APX2 to construct two terminal Steiner tree T_{APX1} and T_{APX2}, respectively. Then select a terminal Steiner tree of minimum length between T_{APX1} and T_{APX2}. For the completeness, we list the $(2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation algorithm as follows.

Algorithm APX
Input: A real $\alpha \ge 2$. A complete graph $G = (V, E)$ with $\ell : E \to R^+$ and a set $R \subset V$ of terminals, where we assume that G contains no edge in E_R and the length function is metric.
Output: A terminal Steiner tree T_{APX} for R in G.
1. Use Algorithm APX1 to find a terminal Steiner tree T_{APX1} that satisfies Theorem 1.
2. Use Algorithm APX2 to find a terminal Steiner tree T_{APX2} that satisfies Theorem 2.
3. Select a minimum length terminal Steiner tree T_{APX} between T_{APX1} and T_{APX2} (i.e., $L(T_{APX}) = \min\{L(T_{APX1}), L(T_{APX2})\}$).

Theorem 3. *Algorithm APX is a $(2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation algorithm to solve the TSTP, where ρ is the best-known performance ratio for the STP and $\alpha \ge 2$.*

Proof. Note that the time-complexity of Algorithm APX is also dominated by the cost of running the currently best-known approximation algorithm for the STP [24]. By Theorem 1 and Theorem 2, we have $L(T_{APX1}) \le 2\rho * L(T_{OPT}) - L(E_{\hat{N}})$ and $L(T_{APX2}) \le \rho * L(T_{OPT}) + \frac{(\alpha^2\rho + \alpha\rho - 4\alpha + 2)}{(\alpha^2 - \alpha)}L(E_{\hat{N}})$. Clearly, $L(T_{APX2})$ will increase when $L(E_{\hat{N}})$ increases. However, $L(T_{APX1})$ will decrease when $L(E_{\hat{N}})$ increases. Moreover, when $L(E_{\hat{N}}) = (\frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}) * L(T_{OPT})$,

$$\rho * L(T_{OPT}) + \frac{(\alpha^2\rho + \alpha\rho - 4\alpha + 2)}{(\alpha^2 - \alpha)}L(E_{\hat{N}}) = 2\rho * L(T_{OPT}) - L(E_{\hat{N}}).$$

Hence, when $L(E_{\hat{N}}) > (\frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}) * L(T_{OPT})$,

$$L(T_{APX1}) \leq (2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha + \alpha^2)(\rho - 1) + 2(\alpha - 1)^2})L(T_{OPT}).$$

Otherwise,

$$L(T_{APX2}) \leq (2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha + \alpha^2)(\rho - 1) + 2(\alpha - 1)^2})L(T_{OPT}).$$

However, Algorithm APX always outputs a minimum length terminal Steiner tree between T_{APX1} and T_{APX2} and hence the result follows. □

Now, let $\alpha = 3.87 \approx 4$ and $\rho \approx 1.55$, Algorithm APX achieves a performance ratio of 2.458 that improves the previous result 2.515 (i.e., let $\alpha = 2$). Note that if $\alpha \approx 3$, it achieves a performance ratio of 2.463.

3 Conclusion

In this paper, we presented an approximation algorithm with performance ratio $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$ for the TSTP under the metric space. An immediate direction for future research could involve finding a better approximation algorithm for the TSTP. Another direction for future research is whether we can apply our approximation algorithm to the partial terminal Steiner tree problem [14] (i.e, a more general terminal Steiner tree problem) or selected-internal Steiner tree problem [15] (i.e. a contrary problem of the partial terminal Steiner tree problem).

References

1. Berman, P., Ramaiyer, V.: Improved Approximations for the Steiner Tree Problem. Journal of Algorithms 17, 381–408 (1994)
2. Bern, M., Plassmann, P.: The Steiner Tree Problem with Edge Lengths 1 and 2. Information Processing Letters 32, 171–176 (1989)
3. Borchers, A., Du, D.Z.: The k-Steiner Ratio in Graphs. SIAM Journal on Computing 26, 857–869 (1997)
4. Caldwell, A., Kahng, A., Mantik, S., Markov, I., Zelikovsky, A.: On Wirelength Estimations for Row-Based Placement. In: Proceedings of the 1998 International Symposium on Physical Design (ISPD 1998), pp. 4–11. ACM, Monterey (1998)
5. Chen, Y.H., Lu, C.L., Tang, C.Y.: On the Full and Bottleneck Full Steiner Tree Problems. In: Warnow, T.J., Zhu, B. (eds.) COCOON 2003. LNCS, vol. 2697, pp. 122–129. Springer, Heidelberg (2003)
6. Cheng, X., Du, D.Z.: Steiner Tree in Industry. Kluwer Academic Publishers, Dordrecht (2001)
7. Drake, D.E., Hougardy, S.: On Approximation Algorithms for the Terminal Steiner Tree Problem. Information Processing Letters 89, 15–18 (2004)

8. Du, D.Z., Smith, J.M., Rubinstein, J.H.: Advances in Steiner Tree. Kluwer Academic Publishers, Dordrecht (2000)
9. Du, D.Z., Hu, X.: Steiner Tree Problems in Computer Communication Networks. World Scientific Publishing Company, Singapore (2008)
10. Fuchs, B.: A Note on the Terminal Steiner Tree Problem. Information Processing Letters 87, 219–220 (2003)
11. Garey, M.R., Graham, R.L., Johnson, D.S.: The Complexity of Computing Steiner Minimal Trees. SIAM Journal of Applied Mathematics 32, 835–859 (1997)
12. Graur, D., Li, W.H.: Fundamentals of Molecular Evolution, 2nd edn. Sinauer Publishers, Sunderland (2000)
13. Hougardy, S., Prommel, H.J.: A 1.598 Approximation Algorithm for the Steiner Problem in Graphs. In: Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1999), pp. 448–453. ACM/SIGACT-SIAM, Baltimore (1999)
14. Hsieh, S.Y., Gao, H.M.: On the Partial Terminal Steiner Tree Problem. The Journal of Supercomputing 41, 41–52 (2007)
15. Hsieh, S.Y., Yang, S.C.: Approximating the Selected-Internal Steiner Tree. Theoretical Computer Science 381, 288–291 (2007)
16. Hwang, F.K., Richards, D.S., Winter, P.: The Steiner Tree Problem. Annuals of Discrete Mathematics, vol. 53. North-Holland, Elsevier, Amsterdam (1992)
17. Kahng, A.B., Robins, G.: On Optimal Interconnections for VLSI. Kluwer Academic Publishers, Boston (1995)
18. Karpinski, M., Zelikovsky, A.: New Approximation Algorithms for the Steiner Tree Problems. Journal of Combinatorial Optimization 1, 47–65 (1997)
19. Kim, J., Warnow, T.: Tutorial on Phylogenetic Tree Estimation. Department of Ecology and Evolutionary Biology. Yale University, New Haven (1999) (manuscript)
20. Lin, G.H., Xue, G.L.: On the Terminal Steiner Tree Problem. Information Processing Letters 84, 103–107 (2002)
21. Lu, C.L., Tang, C.Y., Lee, R.C.T.: The Full Steiner Tree Problem. Theoretical Computer Science 306, 55–67 (2003)
22. Martineza, F.V., Pinab, J.C.D., Soares, J.: Algorithm for Terminal Steiner Trees. Theoretical Computer Science 389, 133–142 (2007)
23. Prommel, H.J., Steger, A.: A New Approximation Algorithm for the Steiner Tree Problem with Performance Ratio 5/3. Journal of Algorithms 36, 89–101 (2000)
24. Robins, G., Zelikovsky, A.: Improved Steiner Tree Approximation in Graphs. In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000), pp. 770–779. ACM/SIGACT-SIAM, San Francisco (2000)
25. Robins, G., Zelikovsky, A.: Tighter Bounds for Graph Steiner Tree Approximation. SIAM Journal on Discrete Mathematics 19, 122–134 (2005)
26. Zelikovsky, A.: An 11/6-Approximation Algorithm for the Network Steiner Problem. Algorithmica 9, 463–470 (1993)
27. Zelikovsky, A.: A Faster Approximation Algorithm for the Steiner Tree Problem in Graphs. Information Processing Letters 46, 79–83 (1993)
28. Zelikovsky, A.: Better Approximation Bounds for the Network and Euclidean Steiner Tree Problems. Technical report CS-96-06, University of Virginia (1996)