



The full Steiner tree problem[☆]

Chin Lung Lu^a, Chuan Yi Tang^{b,*}, Richard Chia-Tung Lee^c

^aDepartment of Biological Science and Technology, National Chiao Tung University, Hsinchu 300, Taiwan, ROC

^bDepartment of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, ROC

^cDepartment of Computer Science and Information Engineering, National Chi-Nan University, Puli, Nantou Hsien 545, Taiwan, ROC

Received 29 August 2002; received in revised form 11 February 2003; accepted 14 February 2003
Communicated by D.-Z. Du

Abstract

Motivated by the reconstruction of phylogenetic tree in biology, we study the full Steiner tree problem in this paper. Given a complete graph $G = (V, E)$ with a length function on E and a proper subset $R \subset V$, the problem is to find a full Steiner tree of minimum length in G , which is a kind of Steiner tree with all the vertices of R as its leaves. In this paper, we show that this problem is NP-complete and MAX SNP-hard, even when the lengths of the edges are restricted to either 1 or 2. For the instances with lengths either 1 or 2, we give a $\frac{8}{5}$ -approximation algorithm to find an approximate solution for the problem.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Full Steiner tree problem; Phylogenetic tree; Evolutionary tree; NP-complete; MAX SNP-hard; Approximation algorithm

1. Introduction

Given a graph $G = (V, E)$, a subset $R \subseteq V$ of vertices, and a length (or distance) function $d: E \rightarrow \mathbb{R}^+$ on the edges, a *Steiner tree* is a connected and acyclic subgraph of G which spans all vertices in R . The vertices in R are usually referred to as *terminals* and the vertices in $V \setminus R$ as *Steiner (or optional)* vertices. Note that a Steiner tree might

[☆] The preliminary version of this paper was presented at the Eighth Ann. Internat. Computing and Combinatorics Conf. (COCOON 2002). This research was partly supported by the National Science Council of the Republic of China under Grant NSC91-2213-E-009-147.

* Corresponding author.

E-mail addresses: cllu@mail.nctu.edu.tw (C.L. Lu), cytang@cs.nthu.edu.tw (C.Y. Tang), rctlee@ncnu.edu.tw (R.C.-T. Lee).

contain the Steiner vertices. The *length* of a Steiner tree is defined to be the sum of the lengths of all its edges. The so-called *Steiner tree problem* is to find a *Steiner minimum tree* (i.e., a Steiner tree of minimum length) in G . The Steiner tree problem has been extensively studied in the past years because it has many important applications in VLSI design, network routing, wireless communications, computational biology and so on [5–7,14]. This problem is well known to be NP-complete [15], even in the Euclidean metric [8] or rectilinear metric [9]. However, it has many approximation algorithms with constant performance ratios [11,14].

Motivated by the reconstruction of phylogenetic (or evolutionary) tree in biology, we study a variant of the Steiner tree problem, called the full Steiner tree problem (FSTP), in this paper. A Steiner tree is *full* if all terminals are the leaves of the tree [14]. The *FSTP* is to find a full Steiner tree with minimum length. If we restrict the lengths of edges to be either 1 or 2, then the problem is called the *(1,2)-full Steiner tree problem* (FSTP(1,2)). From the viewpoints of biologists, the terminals of a full Steiner tree \mathcal{T} can be regarded as the extant taxa (or species, morphological features, biomolecular sequences), the internal vertices of \mathcal{T} as the extinct ancestral taxa, and the length of each edge in \mathcal{T} as the evolutionary time along it. Then \mathcal{T} might correspond to an evolutionary tree of the extant species, which trends to minimize the tree length according to the principle of parsimony (i.e., nature always finds the paths that require a minimum evolution) [10]. Hence, the problem of reconstruction of such kind of phylogenetic tree can be considered as the FSTP, if the extant taxa and their possible ancestral taxa are given. We refer the readers to [12,16,19,20] for other models of evolutionary trees and time-complexities of their constructions.

In fact, the computation of the full Steiner tree also plays a very important role in the Steiner tree problem because any Steiner tree can be decomposed into many smaller components by splitting all the non-leaf terminals such that these components are full Steiner trees (hence they are called the *full components*) [11,14]. A Steiner tree is called *k-Steiner tree* if all of its full components contains at most k terminals. Obviously, every Steiner tree is a k -Steiner tree if k is the number of the terminals. Borchers and Du [4] showed that an optimum k -Steiner tree is a good approximation for a Steiner minimum tree if k is sufficiently large. Unfortunately, the problem of finding an optimum k -Steiner trees was shown to be NP-hard for $k \geq 4$ [11]. In this paper, we will show that the computation of an optimum full Steiner tree is not easier than that of an optimum k -Steiner tree.

To our knowledge, little work has been done on the FSTP. Hwang [13] gave a linear-time algorithm for constructing a relatively minimal full Steiner tree \mathcal{T} with respect to \mathcal{G} in the Euclidean metric, where \mathcal{G} is the given topology of \mathcal{T} . In this paper, we show that the FSTP is NP-complete and MAX SNP-hard, even when the lengths of edges are restricted to be either 1 or 2. However, we give a $\frac{8}{5}$ -approximation algorithm for the FSTP(1,2).

The rest of this paper is organized as follows. In Section 2, we give the formal definition of the problem and introduce some definitions used in this paper. Then we prove the NP-completeness and MAX SNP-hardness results of the problem in Section 3. In Section 4, we describe a $\frac{8}{5}$ -approximation algorithm for the FSTP(1,2). Finally, we give the concluding remarks in Section 5.

2. Preliminaries

To make sure that a full Steiner tree exists, we restrict the given graph $G = (V, E)$ to be complete and R to be a proper subset of V (i.e., $R \subset V$) in the FSTP.

FSTP (Full Steiner tree problem)

Instance: A complete graph $G = (V, E)$, a length function $d: E \rightarrow \mathbb{R}^+$ on the edges, a proper subset $R \subset V$, and a positive integer bound B .

Question: Is there a full Steiner tree \mathcal{T} in G such that the length of \mathcal{T} is less than or equal to B ?

The length function d is called a *metric* if it satisfies the following three conditions:

- (1) $d(x, y) \geq 0$ for any x, y in V , where equality holds if and only if $x = y$,
- (2) $d(x, y) = d(y, x)$ for any $x, y \in V$,
- (3) $d(x, y) \leq d(x, z) + d(z, y)$ for any x, y, z in V (triangle inequality).

If we restrict that all edge lengths are either 1 or 2 (i.e., $d: E \rightarrow \{1, 2\}$), then we call this restricted FSTP as FSTP(1,2), where such length function is a metric. For convenience, we use MIN-FSTP and MIN-FSTP(1,2) to be referred as the optimization problems of FSTP and FSTP(1,2), respectively. Their formal definitions are as follows.

MIN-FSTP (Minimum full Steiner tree problem): Given a complete graph $G = (V, E)$ with a length function $d: E \rightarrow \mathbb{R}^+$ on the edges and a proper subset $R \subset V$ of terminals, find a full Steiner tree of minimum length in G .

MIN-FSTP(1,2) (Minimum (1,2)-full Steiner tree problem): Given a complete graph $G = (V, E)$ with a length function $d: E \rightarrow \{1, 2\}$ on the edges and a proper subset $R \subset V$ of terminals, find a full Steiner tree of minimum length in G .

Given two optimization problems Π_1 and Π_2 , we say that Π_1 *L-reduces* to Π_2 if there are polynomial-time algorithms f and g and positive constants α and β such that for any instance I of Π_1 , the following conditions are satisfied:

- (1) Algorithm f produces an instance $f(I)$ of Π_2 such that $\text{OPT}(f(I)) \leq \alpha \text{OPT}(I)$, where $\text{OPT}(I)$ and $\text{OPT}(f(I))$ stand for the optimal solutions of I and $f(I)$, respectively.
- (2) Given any solution of $f(I)$ with cost c_2 , algorithm g produces a solution of I with cost c_1 in polynomial time such that $|c_1 - \text{OPT}(I)| \leq \beta |c_2 - \text{OPT}(f(I))|$.

A problem is said to be *MAX SNP-hard* if a MAX SNP-hard problem can be *L-reduced* to it. Arora et al. [1] showed that if any MAX SNP-hard problem has a PTAS (polynomial time approximation scheme), then $P = NP$, where a problem has a PTAS if for any fixed $\varepsilon > 0$, the problem can be approximated within a factor of $1 + \varepsilon$ in polynomial time [2]. In other words, it is very unlikely for a MAX SNP-hard problem to have a PTAS. On the other hand, if Π_1 *L-reduces* to Π_2 and Π_2 has a PTAS, then Π_1 has a PTAS [17].

3. Hardness results

In this section, we will only show that the optimization problem of FSTP(1,2), referred to as MIN-FSTP(1,2), is MAX SNP-hard by an *L-reduction* from the vertex cover- B problem (VC- B for short), which was shown to be MAX SNP-hard by

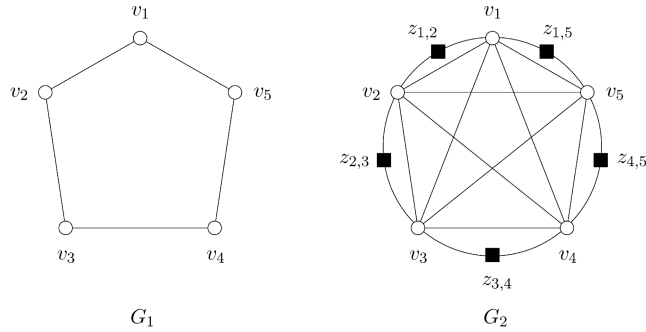


Fig. 1. An L -reduction of VC-B to MIN-FSTP(1,2), where only the edges of length 1 in G_2 are shown.

Papadimitriou and Yannakakis [17]. From the proof of this MAX SNP-hardness, it can be easy to see that the decision problem FSTP(1,2) is NP-hard.

VC-B (Vertex cover-B problem): Given a graph $G = (V, E)$ with degree bounded by a constant B , find a vertex cover of minimum cardinality in G .

Let $G_1 = (V_1, E_1)$ and B be an instance I_1 of VC-B with $V_1 = \{v_1, v_2, \dots, v_n\}$. (Without loss of generality, we assume that G_1 is connected and $n \geq 3$.) Then we transform I_1 into an instance I_2 of MIN-FSTP(1,2), say G_2 and R , as follows.

- A complete graph $G_2 = (V_2, E_2)$ with $V_2 = V_1 \cup \{z_{i,j} \mid (v_i, v_j) \in E_1\}$, and $R = V_2 \setminus V_1 = \{z_{i,j} \mid (v_i, v_j) \in E_1\}$.
- For each edge $e \in E_2$, $d(e) = \begin{cases} 1 & \text{if } e \in \mathcal{E}, \\ 2 & \text{otherwise,} \end{cases}$

where $\mathcal{E} = \{(v_i, v_j) \mid 1 \leq i < j \leq n\} \cup \{(v_i, z_{i,j}), (z_{i,j}, v_j) \mid (v_i, v_j) \in E_1\}$.

See Fig. 1 for an example of the reduction, where G_1 is a C_5 (i.e, a cycle of length 5).

Lemma 3.1. *Let \mathcal{T} be a solution of length c to MIN-FSTP(1,2) on the instance I_2 which is obtained from a reduction of an instance I_1 of VC-B. Then in polynomial time, we can find another solution \mathcal{T}' of length no more than c to MIN-FSTP(1,2) on instance I_2 such that \mathcal{T}' contains no edge of length 2.*

Proof. In the following, we only show how to replace an edge of length 2 from \mathcal{T} with some edges of length 1 in polynomial time without increasing the length of the resulting \mathcal{T} . Then by repeatedly applying this procedure to \mathcal{T} , we will finally obtain \mathcal{T}' in polynomial time. Let (x, y) be an edge of length 2 in \mathcal{T} . Since both x and y cannot belong to R or V_1 at the same time, one of them must be a terminal and the other must be a Steiner vertex according to the rules to construct G_2 . Without loss of generality, we assume that x is a terminal and y is a Steiner vertex. Since we assume that G_1 is connected and $n \geq 3$, x must be connected to some one terminal z with a path of two edges of length 1, say (x, v) and (v, z) . Let u be the Steiner vertex of \mathcal{T} which is adjacent to z . Then we consider the following two possibilities.

Case 1: $u = v$. Then we replace (x, y) with (x, u) of length 1.

Case 2: $u \neq v$. Then we replace (x, y) with (x, v) and (v, u) of length 1.

It is easy to see that the resulting \mathcal{T} is still a full Steiner tree of G_2 , which can be obtained in polynomial time, without increasing the length. \square

Theorem 3.1. *MIN-FSTP(1,2) is a MAX SNP-hard problem.*

Proof. Let f denote the polynomial-time algorithm (as described in the beginning of this section) to transform an instance I_1 of VC-B to the instance I_2 of MIN-FSTP(1,2) (i.e., $f(I_1) = I_2$). We design another polynomial-time algorithm g as follows. Given a full Steiner tree \mathcal{T} in G_2 of length c , we transform it into another full Steiner tree \mathcal{T}' using the method described in the proof of Lemma 3.1. Clearly, \mathcal{T}' contains no edge of length 2 and its length is no more than c , which implies that the number of vertices in \mathcal{T}' is less than or equal to $c + 1$ (since \mathcal{T}' is a tree). Then the collection of those internal vertices of \mathcal{T}' which are adjacent to the leaves of \mathcal{T}' corresponds to a vertex cover of G_1 whose size is less than or equal to $c - |E_1| + 1$. Next, we prove that algorithms f and g are an L -reduction from VC-B to MIN-FSTP(1,2) by showing the following two inequalities.

(1) $\text{OPT}(f(I_1)) \leq \alpha \text{OPT}(I_1)$, where $\alpha = 2B$. Note that $B\text{OPT}(I_1) \geq |E_1|$ since each vertex in G_1 covers at most B edges. Let u be a vertex in G_1 whose degree is B . Then we can build a star \mathcal{T} with u as its center and R as its leaves. Clearly, \mathcal{T} is a feasible solution of MIN-FSTP(1,2) on $f(I_1)$ whose length is $B + 2(|E_1| - B) = 2|E_1| - B$. Hence,

$$\text{OPT}(f(I_1)) \leq 2|E_1| - B \leq 2|E_1| = 2B \frac{|E_1|}{B} \leq 2B\text{OPT}(I_1).$$

(2) $|c_1 - \text{OPT}(I_1)| \leq \beta |c_2 - \text{OPT}(f(I_1))|$, where $\beta = 1$. Given a vertex cover \mathcal{C} in G_1 of size c , we can create a full Steiner tree \mathcal{T} in G_2 of length $c + |E_1| - 1$ in the following way. Connect each edge of E_1 (corresponding a terminal in G_2) to an arbitrary vertex in \mathcal{C} (corresponding a Steiner vertex in G_2) and connect all vertices of \mathcal{C} by $c - 1$ edges of length 1 in G_2 . Hence, $\text{OPT}(f(I_1)) \leq \text{OPT}(I_1) + |E_1| - 1$. Conversely, by algorithm g , a full Steiner tree \mathcal{T} of G_2 with length c_2 can be transformed into a vertex cover of G_1 of size c_1 less than or equal to $c_2 - |E_1| + 1$ (i.e., $c_1 \leq c_2 - |E_1| + 1$). Then

$$\begin{aligned} c_1 - \text{OPT}(I_1) &\leq (c_2 - |E_1| + 1) - \text{OPT}(I_1) \\ &= c_2 - (\text{OPT}(I_1) + |E_1| - 1) \\ &\leq c_2 - \text{OPT}(f(I_1)). \end{aligned}$$

Hence, $|c_1 - \text{OPT}(I_1)| \leq 1 \cdot |c_2 - \text{OPT}(f(I_1))|$. \square

Clearly, the proof of Theorem 3.1 can be applied to show that MIN-FSTP is still MAX SNP-hard, even though the length function is metric.

4. A $\frac{8}{5}$ -approximation algorithm for MIN-FSTP(1,2)

For MIN-FSTP(1,2), it is not hard to see that any star with an arbitrary Steiner vertex as its center and all terminals as its leaves is an approximate solution with performance ratio within 2 of the optimal one. In this section, we give a $\frac{8}{5}$ -approximation algorithm for MIN-FSTP(1,2) using the so-called average length (or distance) heuristics [3,18].

Let *Steiner star* be a star \mathcal{T} with a Steiner vertex as its $\text{center}(\mathcal{T})$ and the terminals as its $\text{leaves}(\mathcal{T})$, where $\text{center}(\mathcal{T})$ and $\text{leaves}(\mathcal{T})$ denote the center and the leaves of \mathcal{T} , respectively. For a Steiner star \mathcal{T} with $|\text{leaves}(\mathcal{T})| \geq 2$, we define its *average length* $f(\mathcal{T})$ as follows:

$$f(\mathcal{T}) = \frac{\sum_{v \in \text{leaves}(\mathcal{T})} d(\text{center}(\mathcal{T}), v)}{|\text{leaves}(\mathcal{T})| - 1}.$$

In fact, the above definition of average length is a kind of scoring function for Steiner stars, which will help us to distinguish the Steiner stars according to their average lengths such that we are able to design our approximation algorithm by a greedy method. For convenience, we use \mathcal{X}_k -star to denote a Steiner star \mathcal{T} with k leaves and $d(\text{center}(\mathcal{T}), v) = 1$ for each v of $\text{leaves}(\mathcal{T})$.

Lemma 4.1. *Let \mathcal{T} be a Steiner star with k terminals, where $k \geq 2$. If \mathcal{T} contains no leaf at distance 1 from $\text{center}(\mathcal{T})$, then $f(\mathcal{T}) = 2 + 2/(k - 1)$.*

Proof. By definition, $f(\mathcal{T}) = 2k/(k - 1) = 2 + 2/(k - 1)$. \square

Lemma 4.2. *Let \mathcal{T} be a Steiner star with k terminals, where $k \geq 2$. If \mathcal{T} contains only one leaf at distance 1 from $\text{center}(\mathcal{T})$, then $f(\mathcal{T}) = 2 + 1/(k - 1)$.*

Proof. By definition, $f(\mathcal{T}) = (1 + 2(k - 1))/(k - 1) = 2 + 1/(k - 1)$. \square

Lemma 4.3. *Let \mathcal{T} be a Steiner star with k terminals, where $k \geq 2$. If \mathcal{T} contains exactly two leaves at distance 1 from $\text{center}(\mathcal{T})$, then $f(\mathcal{T}) = 2$.*

Proof. By definition, $f(\mathcal{T}) = (2 + 2(k - 2))/(k - 1) = 2$. \square

Lemma 4.4. *Let \mathcal{T} be a \mathcal{X}_k -star with $k \geq 3$. Then $f(\mathcal{T}) = 1 + 1/(k - 1)$.*

Proof. By definition, $f(\mathcal{T}) = k/(k - 1) = 1 + 1/(k - 1)$. \square

Lemma 4.5. *Let \mathcal{T}_1 be an \mathcal{X}_k -star with $k \geq 3$ and let \mathcal{T}_2 be the Steiner star obtained from \mathcal{T}_1 by adding a new terminal z with $d(\text{center}(\mathcal{T}_1), z) = 2$. Then $f(\mathcal{T}_1) < f(\mathcal{T}_2)$.*

Proof. By definition, $f(\mathcal{T}_1) = k/(k - 1)$ and $f(\mathcal{T}_2) = (k + 2)/k$, and hence $f(\mathcal{T}_2) - f(\mathcal{T}_1) = (k - 2)/(k(k - 1)) > 0$. \square

Next, we describe our approximation algorithm for MIN-FSTP(1,2) in the following algorithm APX-FSTP(1,2).

APX-FSTP(1,2)
Input: A complete graph $G = (V, E)$ with $d: E \rightarrow \{1, 2\}$ and a set $R \subset V$ of terminals.
Output: A full Steiner tree \mathcal{T}_{APX} for R in G .
Step 1: Let \mathcal{E} be an empty set;
Step 2: /* Choose a Steiner star with the minimum average length */
 if there are two or more remaining Steiner vertices **then**
 Find a Steiner star \mathcal{T} with the minimum average length;
 if $f(\mathcal{T}) = 2$ **then** /* Transform \mathcal{T} into an \mathcal{X}_2 -star */
 Remove from \mathcal{T} those leaves at distance 2 from $\text{center}(\mathcal{T})$ if they exist;
 end if
 else
 Let \mathcal{T} be the Steiner star with the only Steiner vertex as its center and all remaining terminals as its leaves;
 end if
Step 3: /* Perform a reduction */
 Let $\mathcal{E} = \mathcal{E} \cup \{(\text{center}(\mathcal{T}), v) \mid v \in \text{leaves}(\mathcal{T})\}$;
 Replace the Steiner star \mathcal{T} by a single new terminal, say z ;
 Let $d(z, u) = d(\text{center}(\mathcal{T}), u)$ for each remaining vertex u ;
Step 4: **if** there is still more than one terminal **then**
 Go to Step 2;
 else
 Let \mathcal{T}_{APX} be the full Steiner tree induced by \mathcal{E} ;
 end if

According to Lemmas 4.1–4.5, our algorithm APX-FSTP(1,2) always selects an \mathcal{X}_k -star with maximum k , where $k \geq 3$, to do the reduction if it exists, since its average length must be minimum. If only \mathcal{X}_2 -stars are found in the (resulting) instance, then the average length of the minimum Steiner star must be 2 by Lemmas 4.1–4.3. In this case, the minimum Steiner star selected by APX-FSTP(1,2) might contain some leaves at distance 2 from the center. To avoid this situation, APX-FSTP(1,2) will transform it into an \mathcal{X}_2 -star without changing its average length by Lemma 4.3. If the (resulting) instance does not contain a \mathcal{X}_k -star with $k \geq 2$, then APX-FSTP(1,2) will perform only one reduction by Lemmas 4.1 and 4.2. As discussed above, we can find that APX-FSTP(1,2) is a greedy algorithm, which will always select an \mathcal{X}_k -star with maximum k , where $k \geq 2$, to do the reduction except the last one.

We analyze the time-complexity of APX-FSTP(1,2) as follows. Let n and m be the numbers of the terminals and the Steiner vertices in G , respectively (i.e., $n = |R|$ and $m = |V \setminus R|$). Clearly, the time complexity of APX-FSTP(1,2) is dominated by the cost of Step 2, which needs to find a Steiner star with minimum average length. It can

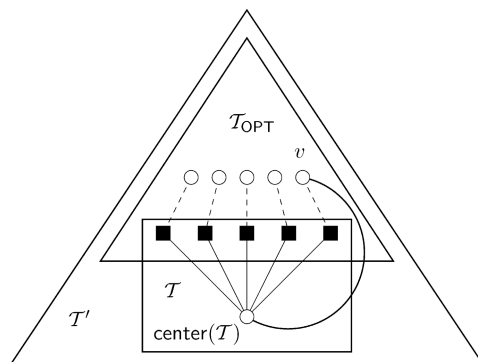


Fig. 2. Add edge $(\text{center}(\mathcal{T}), v)$ to build \mathcal{T}' after removing those dashed lines.

be implemented by first finding an optimal Steiner star with each Steiner vertex as the center and then selecting the best one among these optimal Steiner stars. For each Steiner vertex v , we can find an optimal Steiner star with v as its center in $\mathcal{O}(n')$ time, where n' denotes the number of the resulting terminals. The reason is that we just calculate the number of terminals at distance 1 from v and then we are able to know what its optimal Steiner star is by Lemmas 4.1–4.5. Suppose that there are m' Steiner vertices in each reduction. Then Step 2 can be done in $\mathcal{O}(n'm' + m')$ time. Since each reduction eliminates one Steiner vertex and at least one terminal, the number of the iterations is at most $\min\{n, m\}$ and hence the total time-complexity of APX-FSTP(1,2) is polynomial.

In the following, we analyze the performance ratio of our approximation algorithm APX-FSTP(1,2). Let the *performance ratio* of our approximation algorithm APX-FSTP(1,2) for instance I be $\text{ratio}(I) = \text{APX}(I)/\text{OPT}(I)$, where $\text{OPT}(I)$ denotes the length of an optimal full Steiner tree for I and $\text{APX}(I)$ denotes the length of \mathcal{T}_{APX} obtained by APX-FSTP(1,2). In the following, we assume that I is a worst-case instance among all instances. That is, $\text{ratio}(I') \leq \text{ratio}(I)$ for each $I' \neq I$.

Lemma 4.6. *If instance I contains an \mathcal{X}_k -star for $k \geq 6$, then $\text{ratio}(I) \leq \frac{3}{2}$.*

Proof. Let \mathcal{T} be an arbitrary \mathcal{X}_k -star in I whose k is maximum and let $E(\mathcal{T})$ be the set of its edges. Then by Lemmas 4.1–4.5, the first iteration of our algorithm APX-FSTP(1,2) will reduce \mathcal{T} since its average length $f(\mathcal{T})$ is minimum. Let I' be the resulting instance of APX-FSTP(1,2) after \mathcal{T} is reduced. Clearly, we have $\text{APX}(I') = \text{APX}(I) - k$.

Let \mathcal{T}_{OPT} be an optimal full Steiner tree of I , and let \mathcal{H} be the resulting graph obtained by adding the k edges of $E(\mathcal{T})$ to \mathcal{T}_{OPT} . Then by removing from \mathcal{H} some edges not in $E(\mathcal{T})$ and adding some one edge if possible, we can build a full Steiner tree \mathcal{T}' of I such that it contains all edges of $E(\mathcal{T})$. See Fig. 2 for an illustration to the worst case in which the k edges of $E(\mathcal{T})$ and the $\text{center}(\mathcal{T})$ are not in \mathcal{T}_{OPT} . Then we need to add edge $(\text{center}(\mathcal{T}), v)$ to build a full Steiner tree \mathcal{T}' , where v is a

Steiner vertex in \mathcal{T}_{OPT} . Clearly, the length of \mathcal{T}' is less than or equal to $\text{OPT}(I) + 2$ since $d(\text{center}(\mathcal{T}), v) \leq 2$. If we reduce \mathcal{T} in \mathcal{T}' , then we obtain a full Steiner tree \mathcal{T}'' of instance I' whose length is less than or equal to $\text{OPT}(I) - k + 2$. In other words, $\text{OPT}(I') \leq \text{OPT}(I) - k + 2$. Hence, $\text{ratio}(I') = \text{APX}(I') / \text{OPT}(I') \geq (\text{APX}(I) - k) / (\text{OPT}(I) - k + 2)$. Recall that $\text{ratio}(I') \leq \text{ratio}(I)$. Then we have

$$\begin{aligned} \frac{\text{APX}(I) - k}{\text{OPT}(I) - k + 2} &\leq \frac{\text{APX}(I)}{\text{OPT}(I)} \\ &\Leftrightarrow k \text{OPT}(I) \geq (k - 2) \text{APX}(I) \\ &\Leftrightarrow \frac{k}{k - 2} \geq \frac{\text{APX}(I)}{\text{OPT}(I)} = \text{ratio}(I). \end{aligned}$$

Hence, we have $\text{ratio}(I) \leq \frac{3}{2}$. \square

According to Lemma 4.6, if any instance I of MIN-FSTP(1,2) contains an \mathcal{X}_k -star with $k \geq 6$, then our algorithm APX-FSTP(1,2) always produces a full Steiner tree of length less than or equal to $\frac{3}{2} \text{OPT}(I)$. In the following, we assume that I contains no such an \mathcal{X}_k -star with $k \geq 6$ and we will then show that $\text{ratio}(I) \leq \frac{8}{5}$.

Given an instance I consisting of $G = (V, E)$ and $R \subset V$, we say that a vertex $v \in V$ 1-dominates (or dominates for simplicity) itself and all other vertices at distance 1 from v . For any $D \subseteq V$, we call it as a 1-dominating set (or dominating set) of R if every terminal in R is dominated by at least one vertex of D . A dominating set of R with minimum cardinality is called as a *minimum* dominating set of R .

Lemma 4.7. *Given an instance I of MIN-FSTP(1,2), let D be a minimum dominating set of R . Then $\text{OPT}(I) \geq n + |D| - 1$, where $n = |R|$.*

Proof. Let \mathcal{T}_{OPT} be an optimal full Steiner tree of I (i.e., $\text{OPT}(I) = |\mathcal{T}_{\text{OPT}}|$) and let $R' \subseteq R$ be the set of terminals that are dominated by the vertices of D' , where $D' \subseteq V \setminus R$ is the set of Steiner vertices in \mathcal{T}_{OPT} . See Fig. 3 for an illustration. Note that for those vertices in D' , \mathcal{T}_{OPT} needs to contain at least $|D'| - 1$ edges to connect them. Clearly, the length of \mathcal{T}_{OPT} is $|\mathcal{T}_{\text{OPT}}| \geq |R'| + (|D'| - 1) + 2|R \setminus R'| = |R| + |D'| + |R \setminus R'| - 1$. Since the union of D' and $R \setminus R'$ is a dominating set of R and they are disjoint, $|D| \leq |D' \cup (R \setminus R')| = |D'| + |R \setminus R'|$. In other words, we have $\text{OPT}(I) \geq |R| + |D| - 1 = n + |D| - 1$. \square

Let D be a minimum dominating set of R . Then we can partition R into many subsets in a way as follows. Assign each terminal z of R to a member of D which dominates it. If two or more vertices of D dominate z , then we arbitrarily assign z to one of them. Let $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_q$ be the partitions consisting of exactly 5 terminals. Clearly, we have $0 \leq q \leq \frac{n}{5}$.

Lemma 4.8. $\text{OPT}(I) \geq 5n/4 - q/4 - 1$.

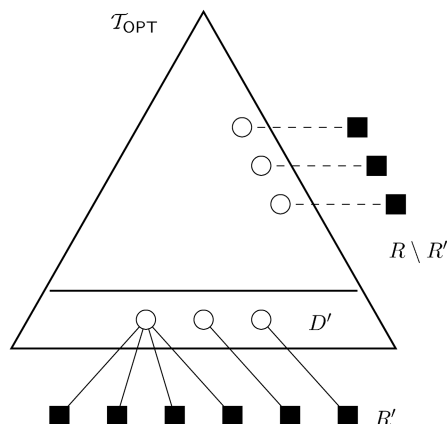


Fig. 3. The illustration of the proof of Lemma 4.7, where the solid lines are edges of length 1 and the dashed lines are edges of length 2.

Proof. According to the partition of R , we have $5q + 4(|D| - q) \geq n$, which means that $|D| \geq (n - q)/4$. Recall that $\text{OPT}(I) \geq n + |D| - 1$ by Lemma 4.7. Then,

$$\text{OPT}(I) \geq n + \frac{n - q}{4} - 1 = \frac{5n}{4} - \frac{q}{4} - 1.$$

Hence, we have $\text{OPT}(I) \geq 5n/4 - q/4 - 1$. \square

Lemma 4.9. *If instance I contains no \mathcal{X}_k -star with $k \geq 6$, then $\text{ratio}(I) \leq \frac{8}{5}$.*

Proof. Assume that APX-FSTP(1,2) totally reduces j \mathcal{X}_{k_i} -stars, where $1 \leq i \leq j$ and $k_i \geq 2$. Even though the instance I contains no \mathcal{X}_k -star with $k \geq 6$, the reduced \mathcal{X}_{k_i} -star may be an \mathcal{X}_6 -star for each $i \geq 2$. This is because that the new terminal z created by reducing $\mathcal{X}_{k_{i-1}}$ -star is at distance 1 from the center of an \mathcal{X}_5 -star in the instance I such that the reduced \mathcal{X}_{k_i} is an \mathcal{X}_6 -star formed by adding z into the \mathcal{X}_5 -star. However, it is impossible that \mathcal{X}_{k_i} -star is an \mathcal{X}_k -star with $k \geq 7$. The reason is that when an \mathcal{X}_6 -star is created by a reduction, it will be selected by our algorithm APX-FSTP(1,2) for the next reduction, which may create another \mathcal{X}_6 -star (i.e., there is at most one \mathcal{X}_6 -star in each resulting instance).

Note that \mathcal{X}_{k_i} is a subtree of the full Steiner tree \mathcal{T}_{APX} produced by APX-FSTP(1,2) and its length is k_i . Since the reduction of \mathcal{X}_{k_i} merges k_i old terminals into a new one, the number of the terminals is decreased by $k_i - 1$. After reducing \mathcal{X}_{k_j} , the number of the remaining terminals is $n - \sum_{i=1}^j (k_i - 1)$. To reduce these terminals, APX-FSTP(1,2) creates a Steiner star with length less than or equal to $2(n - \sum_{i=1}^j (k_i - 1))$. Hence, the total length of \mathcal{T}_{APX} is less than or equal to $(\sum_{i=1}^j k_i) + (2(n - \sum_{i=1}^j (k_i - 1))) = 2n - \sum_{i=1}^j (k_i - 2)$. In other words, we have $\text{APX}(I) \leq 2n - p$, where we let $p = \sum_{i=1}^j (k_i - 2)$.

Recall that we partition R into many disjoint subsets in which $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_q$, where $0 \leq q \leq \frac{n}{5}$, are the partitions with each consisting of exactly 5 terminals. In other words, there are at least q disjoint \mathcal{X}_5 -stars in I . Next, we claim that $p \geq 11q/10$. The best situation is that each partition \mathcal{C}_i , where $1 \leq i \leq q$, corresponds to an \mathcal{X}_{k_l} -star, where $1 \leq l \leq j$ and $k_l = 5$ or 6, which will be reduced by APX-FSTP(1,2). In this case, each such an \mathcal{X}_{k_l} -star contributes at least 3 to p and hence we have $p \geq 3q > 11q/10$. Otherwise, we consider the general case with the following five properties, where for simplicity of illustration, we assume that $q_2 \equiv 0 \pmod{2}$, $q_3 \equiv 0 \pmod{3}$, $q_4 \equiv 0 \pmod{4}$, $q_5 \equiv 0 \pmod{5}$ and $q_1 + q_2 + q_3 + q_4 + q_5 = q$.

- (1) There are q_1 partitions $\mathcal{C}_{i_1}, \mathcal{C}_{i_2}, \dots, \mathcal{C}_{i_{q_1}}$ in which each partition \mathcal{C}_{i_h} corresponds to an \mathcal{X}_{k_l} -star reduced by APX-FSTP(1,2), where $1 \leq h \leq q_1$.
- (2) There are q_2 partitions $\mathcal{C}_{i_{q_1+1}}, \mathcal{C}_{i_{q_1+2}}, \dots, \mathcal{C}_{i_{q_1+q_2}}$ in which every other two consecutive partitions $\mathcal{C}_{i_{q_1+h+1}}$ and $\mathcal{C}_{i_{q_1+h+2}}$ correspond to an \mathcal{X}_{k_l} -star reduced by APX-FSTP(1,2), where $0 \leq h \leq q_2 - 2$ and $h \equiv 0 \pmod{2}$.
- (3) There are q_3 partitions $\mathcal{C}_{i_{q_1+q_2+1}}, \mathcal{C}_{i_{q_1+q_2+2}}, \dots, \mathcal{C}_{i_{q_1+q_2+q_3}}$ in which every other three consecutive partitions $\mathcal{C}_{i_{q_1+q_2+h+1}}$, $\mathcal{C}_{i_{q_1+q_2+h+2}}$ and $\mathcal{C}_{i_{q_1+q_2+h+3}}$ correspond to an \mathcal{X}_{k_l} -star reduced by APX-FSTP(1,2), where $0 \leq h \leq q_3 - 3$ and $h \equiv 0 \pmod{3}$.
- (4) There are q_4 partitions $\mathcal{C}_{i_{q_1+q_2+q_3+1}}, \mathcal{C}_{i_{q_1+q_2+q_3+2}}, \dots, \mathcal{C}_{i_{q_1+q_2+q_3+q_4}}$ in which every other four consecutive partitions $\mathcal{C}_{i_{q_1+q_2+q_3+h+1}}, \mathcal{C}_{i_{q_1+q_2+q_3+h+2}}, \dots, \mathcal{C}_{i_{q_1+q_2+q_3+h+4}}$ correspond to an \mathcal{X}_{k_l} -star reduced by APX-FSTP(1,2), where $0 \leq h \leq q_4 - 4$ and $h \equiv 0 \pmod{4}$.
- (5) There are q_5 partitions $\mathcal{C}_{i_{q_1+q_2+q_3+q_4+1}}, \mathcal{C}_{i_{q_1+q_2+q_3+q_4+2}}, \dots, \mathcal{C}_{i_{q_1+q_2+q_3+q_4+q_5}}$ in which every other five consecutive partitions $\mathcal{C}_{i_{q_1+q_2+q_3+q_4+h+1}}, \mathcal{C}_{i_{q_1+q_2+q_3+q_4+h+2}}, \dots, \mathcal{C}_{i_{q_1+q_2+q_3+q_4+h+5}}$ correspond to an \mathcal{X}_{k_l} -star reduced by APX-FSTP(1,2), where $0 \leq h \leq q_5 - 5$ and $h \equiv 0 \pmod{5}$.

It is not hard to see that the reduction of \mathcal{X}_{k_l} -stars of property (1) (respectively, (2)–(5)) will contribute at least $3q_1$ (respectively, $3q_2/2$, $3q_3/3$, $3q_4/4$ and $3q_5/5$) to p and in the worst case, produce 0 (respectively, $q_2/2$, 0, $q_4/4$ and 0) \mathcal{X}_3 -star and produce 0 (respectively, 0, $2q_3/3$, $3q_4/4$ and $5q_5/5$) \mathcal{X}_4 -star in the remaining instance. In the worst case, the $(0 + q_2/2 + 0 + q_4/4 + 0 = (2q_2 + q_4)/4)$ produced \mathcal{X}_3 -stars and the $(0 + 0 + 2q_3/3 + 3q_4/4 + 5q_5/5 = (8q_3 + 9q_4 + 12q_5)/12)$ produced \mathcal{X}_4 -stars will further contribute $1((2q_2 + q_4)/4)/3$ and $2((8q_3 + 9q_4 + 12q_5)/12)/4$, respectively to p . Hence, we have

$$\begin{aligned}
 p &\geq 3q_1 + \frac{3q_2}{2} + \frac{3q_3}{3} + \frac{3q_4}{4} + \frac{3q_5}{5} + \frac{2q_2 + q_4}{12} + \frac{8q_3 + 9q_4 + 12q_5}{24} \\
 &= q_1 + \frac{480q_1 + 400q_2 + 320q_3 + 290q_4 + 264q_5}{240} \\
 &= q_1 + \frac{480(q_1 + q_2 + q_3 + q_4 + q_5) - (80q_2 + 160q_3 + 190q_4 + 216q_5)}{240} \\
 &\geq q_1 + \frac{264q}{240}
 \end{aligned}$$

$$\begin{aligned}
& (\text{since } q_1 + q_2 + q_3 + q_4 + q_5 = q \text{ and } 80q_2 + 160q_3 + 190q_4 + 216q_5 \\
& \leq 216q) \\
& \geq \frac{11q}{10}.
\end{aligned}$$

As discussed above, we have $p \geq 11q/10$. Recall that $\text{OPT}(I) \geq 5n/4 - q/4 - 1$ by Lemma 4.8. Then we have

$$\begin{aligned}
\text{ratio}(I) &= \frac{\text{APX}(I)}{\text{OPT}(I)} \leq \frac{2n - p}{5n/4 - q/4 - 1} = \frac{8n - 4p}{5n - q - 4} \leq \frac{8n - 4(11q/10)}{5n - q - 4} \\
&= \frac{40n - 22q}{25n - 5q - 20}.
\end{aligned}$$

It is easy to verify that $(40n - 22q)/(25n - 5q - 20) \leq \frac{8}{5}$ if $q \geq \frac{16}{7}$. Since $0 \leq q \leq n/5$, $\text{ratio}(I) \leq \frac{8}{5}$ for $n \geq \frac{80}{7}$. Note that for $n < \frac{80}{7}$, the optimal solution can be found by an exhaustive search in polynomial time. Hence, we have $\text{ratio}(I) \leq \frac{8}{5}$. \square

According to Lemmas 4.6 and 4.9, we have the following theorem immediately.

Theorem 4.1. *APX-FSTP(1,2) is a $\frac{8}{5}$ -approximation algorithm for MIN-FSTP(1,2).*

5. Conclusion

In this paper, we studied the algorithmic complexities of the problem of finding an optimal full Steiner tree for a given set of terminals and a complete graph with a length function on the edges. We showed that this problem is NP-complete and MAX SNP-hard, even when the lengths of the edges are restricted to either 1 or 2. If the lengths of edges are either 1 or 2, then we gave a $\frac{8}{5}$ -approximation algorithm for the problem. It would be interest to know if our $\frac{8}{5}$ -approximation algorithm for MIN-FSTP(1,2) can be further improved, and if there is an approximation algorithm within a constant ratio for MIN-FSTP (i.e., the FSTP without any constraint on the length function).

Acknowledgements

The authors would like to thank the anonymous referees for many constructive observations and suggestions for the presentation of this paper.

References

- [1] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, Proof verification and the hardness of approximation problems, *J. Assoc. Comput. Mach.* 45 (1998) 501–555.

- [2] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamelai, M. Protasi, *Complexity and Approximation—Combinatorial Optimization Problems and their Approximability Properties*, Springer, Berlin, 1999.
- [3] M. Bern, P. Plassmann, The Steiner problem with edge lengths 1 and 2, *Inform. Process. Lett.* 32 (1989) 171–176.
- [4] A. Borchers, D.Z. Du, The k -Steiner ratio in graphs, *SIAM J. Comput.* 26 (1997) 857–869.
- [5] X. Cheng, D.Z. Du, *Steiner Trees in Industry*, Kluwer Academic Publishers, Dordrecht, Netherlands, 2001.
- [6] D.Z. Du, J.M. Smith, J.H. Rubinstein, *Advances in Steiner Trees*, Kluwer Academic Publishers, Dordrecht, Netherlands, 2000.
- [7] L.R. Foulds, R.L. Graham, The Steiner problem in phylogeny is NP-complete, *Adv. Appl. Math.* 3 (1982) 43–49.
- [8] M.R. Garey, R.L. Graham, D.S. Johnson, The complexity of computing Steiner minimal trees, *SIAM J. Appl. Math.* 32 (1977) 835–859.
- [9] M.R. Garey, D.S. Johnson, The rectilinear Steiner problem is NP-complete, *SIAM J. Appl. Math.* 32 (1977) 826–834.
- [10] D. Graur, W.H. Li, *Fundamentals of Molecular Evolution*, 2nd Edition, Sinauer Publishers, Sunderland, MA, 2000.
- [11] C. Gröpl, S. Hougardy, T. Nierhoff, H.J. Prömel, Approximation algorithms for the Steiner tree problem in graphs, in: X. Cheng, D.Z. Du (Eds.), *Steiner Trees in Industry*, Kluwer Academic Publishers, Dordrecht, Netherlands, 2001, pp. 235–279.
- [12] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, Cambridge, 1997.
- [13] F.K. Hwang, A linear time algorithm for full Steiner trees, *Oper. Res. Lett.* 4 (1986) 235–237.
- [14] F.K. Hwang, D.S. Richards, P. Winter, The Steiner tree problem, in: *Annals of Discrete Mathematics*, Vol. 53, Elsevier, Amsterdam, 1992.
- [15] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), *Complexity of Computer Computations*, Plenum Press, New York, 1972, pp. 85–103.
- [16] J. Kim, T. Warnow, Tutorial on phylogenetic tree estimation, Manuscript, Department of Ecology and Evolutionary Biology, Yale University, 1999.
- [17] C. Papadimitriou, M. Yannakakis, Optimization, approximation and complexity classes, *J. Comput. System Sci.* 43 (1991) 425–440.
- [18] V. Rayward-Smith, The computation of nearly minimum Steiner trees in graphs, *Internat. J. Math. Ed. Sci. Tech.* 14 (1983) 15–23.
- [19] H.T. Wareham, On the computational complexity of inferring evolutionary trees, Tech. Report 93-01, Department of Computer Science, Memorial University of Newfoundland, 1993.
- [20] M.S. Waterman, *Introduction to Computational Biology: Maps, Sequences and Genomes*, Chapman & Hall, London, 1995.