Computational Geometry 48 (2015) 453–458

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

On full Steiner trees in unit disk graphs ☆

Ahmad Biniaz *, Anil Maheshwari, Michiel Smid

School of Computer Science, Carleton University, Ottawa, Canada

ARTICLE INFO

Article history:
Received 18 August 2014
Accepted 19 December 2014
Available online 9 February 2015

Keywords:
Approximation algorithms
Full Steiner tree
Unit disk graph

ABSTRACT

Given an edge-weighted graph $G = (V, E)$ and a subset R of V, a Steiner tree of G is a tree which spans all the vertices in R. A full Steiner tree is a Steiner tree which has all the vertices of R as its leaves. The full Steiner tree problem is to find a full Steiner tree of G with minimum weight. In this paper we consider the full Steiner tree problem when G is a unit disk graph. We present a 20-approximation algorithm for the full Steiner tree problem in G. As for λ-precise unit disk graphs we present a $(10 + \frac{1}{\lambda})$-approximation algorithm, where λ is the length of the shortest edge in G.

1. Introduction

Given a graph $G = (V, E)$ of n vertices with a weight function $w : E \to \mathbb{R}^+$ on edges and a subset R of V. The vertices in R are called the *terminals* and the vertices in $V \setminus R$ are called *Steiner vertices* (usually denoted by S, i.e., $S = V \setminus R$); see Fig. 1(a). A *Steiner tree* of G is a tree which contains all the vertices in R; see Fig. 1(b). The weight of a tree T is defined as the sum of the weights of all the edges in T; i.e., $w(T) = \sum_{e \in T} w(e)$. The *Steiner tree problem* is to find a Steiner tree T of G with minimum weight [13]. This problem is known to be MAX SNP-hard [1,2]. Robins and Zelikovsky [17] presented a 1.55-approximation algorithm for this problem. The approximation ratio was improved to $\ln(4) + \varepsilon < 1.39$ by Byrka et al. [3].

Motivated by the reconstruction of evolutionary trees in biology [15] and VLSI global routing and telecommunications [14], a *full Steiner tree* of G is defined as a Steiner tree which has all the vertices of R as its leaves; see Fig. 1(c). The *full Steiner tree problem* is to find a full Steiner tree T of G with minimum weight. This problem is also known as the *terminal Steiner tree problem*. In a full Steiner tree problem one may assume that G does not have any edge between the vertices of R.

A *metric graph* is defined as a complete graph, whose edge weights satisfy the *triangle inequality*, i.e., for any three vertices x, y, and z, $w(x, y) \le w(x, z) + w(y, z)$ [7,20]. Lin and Xue [14] showed that the full Steiner tree problem for metric graphs is NP-complete and MAX SNP-hard, even when the lengths of the edges are restricted to be either 1 or 2. Many approximation algorithms have been proposed for the full Steiner tree problem in a metric graph [4,5,7,8,11,14,16]. Lin and Xue [14] presented an approximation algorithm with performance ratio $2 + \rho$, where ρ is the approximation ratio for the Steiner tree problem. The currently best-known approximation ratio for the Steiner tree problem is 1.39 [3]. The approximation ratio was improved to 2ρ in [5,7,8], and further to $2\rho - (\frac{\rho}{3\rho-2})$ in [16], and $2\rho - \frac{(\rho\alpha^2-\rho\alpha)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}$ in [4] for any $\alpha \ge 2$. The straightforward 2ρ-approximation algorithms [5,7,8] start by computing a Steiner tree T of G which has

☆ Research supported by NSERC.

* Corresponding author.
E-mail address: ahmad.biniaz@gmail.com (A. Biniaz).

http://dx.doi.org/10.1016/j.comgeo.2015.02.004

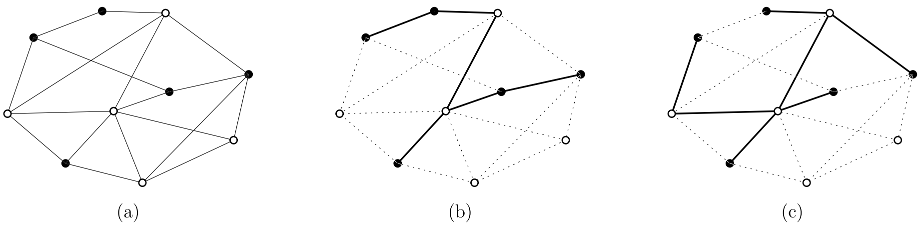

Fig. 1. (a) The input graph G; terminals are indicated by filled circles, non-terminals by non-filled circles. (b) A Steiner tree and (c) a full Steiner tree of G.

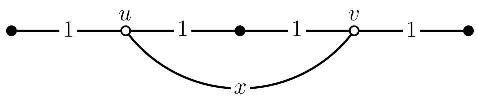

Fig. 2. A non-metric instance for the full Steiner tree problem.

no edge between any pair of terminals. Then, for each non-leaf terminal r in T, pick one of its adjacent Steiner vertices, say s, and connect all other Steiner neighbors of r to s [5,7,8].

Drake and Hougardy [7] showed that approximating the non-metric version of the full Steiner tree problem is at least as hard as approximating the set cover problem. They showed that there is no polynomial time approximation algorithm for the full Steiner tree problem with performance ratio better than $(1 - o(1)) \ln n$ unless $NP = DTIME(n^{O(\log\log n)})$.

1.1. Our results

Let P denote a set of n points in the plane. The unit disk graph, $UDG(P)$, is defined to have P as its vertex set and there is an edge between two points p and q if their Euclidean distance is at most 1, i.e., $|pq| \leq 1$. Given a set of terminals $R \subset P$, we are interested in computing the minimum-weight full Steiner tree of $UDG(P)$; thus the Steiner vertices must be chosen from the set $P \setminus R$. We assume that the weight of an edge (p, q) is equal to the Euclidean distance between p and q; i.e., $w(p, q) = |pq|$. It is not known whether this problem can be solved in polynomial time.

The *aspect ratio* of an edge set E is the ratio of the length of a longest edge in E to the length of a shortest edge in E. The aspect ratio of a graph is defined as the aspect ratio of its edge set. For a constant $\lambda > 0$, a λ-*precise* (or λ-*precision*) UDG is a unit disk graph in which no two vertices are at distance smaller than λ; it is also known as λ-*civilized* UDG [6,12]. Thus, the aspect ratio of any λ-precise UDG is at most $\frac{1}{\lambda}$. Most often wireless devices in a wireless network cannot be too close, so it is reasonable to model a wireless ad-hoc network as a λ-precise UDG [6]. It can be seen that grid graphs are λ-precise UDG.

In this paper we present two polynomial-time approximation algorithms for the full Steiner tree problem in UDG and λ-precise UDG. Note that all previous results [4,5,7,8,11,14,16] are only applicable when the input graph G is metric. Thus, they cannot be applied to UDG, because it is not necessarily a complete graph. When the input graph G is a λ-precise UDG, we present a $(10 + \frac{1}{\lambda})$-approximation algorithm in Section 2. In Section 3 we extend our idea and present a 20-approximation algorithm for any unit disk graph. The combination of these two algorithms give us a full Steiner tree of approximation ratio $\min\{20, (10 + \frac{1}{\lambda})\}$ for UDG. Using the same technique as in Section 3, we can compute a full Steiner tree of approximation ratio 2Δ for general graphs, where Δ is the maximum vertex degree in the graph.

1.2. Preliminaries

Vazirani [20] showed that in polynomial time one can transform an instance of a Steiner tree problem into an equivalent instance of the metric Steiner tree problem; the transformation preserves the approximation factor. The transformation is as follows. For a given graph $G = (V, E)$, consider a complete graph G' with vertex set V. Define the weight of an edge (u, v) in G' as the weight of the shortest path between u and v in G. Next compute a minimum Steiner tree T' in G' and replace each edge (u, v) in T' by the corresponding shortest path in G. Finally compute a spanning tree T of the resulting subgraph. For any edge $(x, y) \in G$, its cost in G' is no more than its cost in G. Therefore, the cost of an optimal solution in G' is no more than the cost of an optimal solution in G [20].

Theorem. *(See Vazirani [20].) There is a polynomial-time approximation factor preserving reduction from the Steiner tree problem to the metric Steiner tree problem.*

Drake and Hougardy [7] showed that the similar transformation cannot be applied for the full Steiner tree problem. Fig. 2 which is borrowed from [7] shows that if $x > 2$ then the shortest path between Steiner vertices u and v passes through a terminal vertex. Finally when replacing (u, v) in T' with the original shortest path in G, the resulting Steiner tree T is not a full Steiner tree. As a result we have the following observation: