Computational Geometry 53 (2016) 21–26

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

A lower bound for computing geometric spanners

Mohammad Farshi *, Abolfazl Poureidi

Combinatorial and Geometric Algorithms Lab., Department of Computer Science, Yazd University, P.O. Box 89195-741, Yazd, Iran

ARTICLE INFO

Article history:
Received 20 June 2015
Accepted 18 December 2015
Available online 29 December 2015

Keywords:
Geometric network
Spanner
Lower bound
Steiner spanner

ABSTRACT

It is known that the problem of computing (Steiner) spanners on a set of n points has an $\Omega(n \log n)$ lower bound. However, the proof is based on an example of points on the real line. Therefore, if we assume that the points belong to the plane or higher dimensions, and moreover, they are in general position, then the lower bound example does not work.

In this paper, we show that the complexity of computing geometric spanners, possibly containing Steiner points, for a set of n points in d-dimensional Euclidean space ($\mathbb{R}^d$) that are in general position is $\Omega(n \log n)$, in the algebraic computation tree model. To this end, we reduce the spanner construction to a variant of the closest pair problem which has an $\Omega(n \log n)$ lower bound.

1. Introduction

Let $P \subset \mathbb{R}^d$ be a set of n points. Given a real number $t > 1$, we say that the Euclidean graph $G(P, E)$, i.e. an edge-weighted graph such that the weight of each edge is the Euclidean distance between its endpoints, is a geometric t-spanner of P if for each pair of points $u, v \in P$, there exists a path between u and v in G with length at most $t \times |uv|$. The length of a path is defined as the sum of the lengths of the edges on the path and $|uv|$ denotes the Euclidean distance between u and v. We call such a path, if it exists, a t-path between u and v. If the vertex set of the graph $G(V, E)$ is a superset of P and the spanner property holds between all pairs of points from P, then we call G a Steiner spanner of P. Note that when constructing Steiner t-spanners, we can add extra vertices to the graph, and we do not need any condition on path lengths between Steiner points and other (Steiner or non-Steiner) points, so the problem of constructing Steiner spanners is easier than the problem of constructing spanners.

Geometric spanners have several applications in theory and practice and have been the subject of several papers in the last decades [4]. Several algorithms are known that, given a set P of n points in $\mathbb{R}^d$, d is a constant, and a constant $t > 1$, construct a sparse t-spanner of P in $\mathcal{O}(n \log n)$ time. Several sparseness measures have been considered like linear number of edges, weight proportional to the weight of the minimum spanning tree and some other properties.

One of the major questions in the field was the possibility to improve the time complexity of previous spanner construction algorithms, i.e., constructing spanners in $o(n \log n)$ time or showing that the problem has an $\Omega(n \log n)$ time complexity.

In 2001, Chen, Das and Smid [3] gave a lower bound on the time complexity for constructing geometric Steiner spanners, for a set of n points in $\mathbb{R}^d$, see also [4, Section 3.4]. Chen, Das and Smid [3] showed that the time complexity of the problem in the algebraic computation tree model [1,4,5] is $\Omega(n \log n)$. Their lower bound proof is given in the case that points are from $\mathbb{R}$. In the case that the input points are not necessarily distinct, they reduced the element uniqueness problem to the problem of constructing Steiner spanners. Since the element uniqueness problem has $\Omega(n \log n)$ complexity, the same

* Corresponding author. Tel.: +98 35 3123 2689.
E-mail addresses: mfarshi@yazd.ac.ir (M. Farshi), poureidi@stu.yazd.ac.ir (A. Poureidi).

http://dx.doi.org/10.1016/j.comgeo.2015.12.004
0925-7721/

holds for computing Steiner spanners. When we know that the points in the input point set are pairwise distinct, the above argument does not work. So, in the same paper, Chen et al. also gave a direct proof of $\Omega(n \log n)$ lower bound in this case. The proof, again, was based on a point set from $\mathbb{R}$. Therefore, their argument does not work in higher dimensions, in the case one assumes that the points are in general position. This is one of the assumptions that we normally assume an input point set satisfies when we study a problem in computational geometry.

A set P of points in $\mathbb{R}^d$ is in general position, if no $d+1$ points of P lie on a $(d-1)$-dimensional subspace of $\mathbb{R}^d$. For $d=2$ it means that no three points lie on a line and for $d=3$, a set of points is in general position if no four points lie in a plane, i.e. 2-dimensional subspace of $\mathbb{R}^d$. We have the following, slightly different, definition of points in general position in the literature: a point set in $\mathbb{R}^d$ is in general position, if for each k between 3 and $d+1$, no k points on the point set are contained in a $(k-2)$-dimensional subspace of $\mathbb{R}^d$. These two definitions are equivalent because we can easily grow any set of k points ($k < d+1$) in $(k-2)$-dimensional subspace to a set of $d+1$ points in $(d-1)$-dimensional subspace of $\mathbb{R}^d$ by adding arbitrary points to the point set.

Since the proof of Chen et al. [3] does not work if one assumes that the input point set is from $\mathbb{R}^d$ ($d > 1$) and it is in general position, they posed the following conjecture.

Conjecture 1. *(See [3].) Let $d \geq 2$ be an integer constant. In the algebraic computation tree model, any algorithm that, given a set P of n points in $\mathbb{R}^d$ that are in general position, and a real number $t > 1$, constructs a Steiner t-spanner for P, takes $\Omega(n \log n)$ time in the worst case.*

In this paper, we propose a reduction which gives a positive answer to the conjecture. The organization of the paper is as follows: we start, in Section 2, with describing a variant of the element uniqueness problem, and we next reduce the problem of computing a geometric Steiner spanner on a set of points in $\mathbb{R}^d$ to this problem. At the end of Section 2, we introduce a variant of the closest pair problem which has an $\Omega(n \log n)$ lower bound. Then, in Section 3, we reduce the problem of computing a geometric Steiner spanner on a set of points in general position to this problem which completes the proof.

Throughout the paper, we assume that points come from $\mathbb{R}^d$, where d is a constant and also $t > 1$ is a constant.

2. A reduction from the extended element uniqueness problem

The (standard) element uniqueness problem is a decision problem that is defined as follows: Given n real numbers $x_1, x_2, \ldots, x_n$, decide whether they are pairwise distinct. It has a lower bound of $\Omega(n \log n)$ time complexity in the algebraic decision tree model [1,4,5]. We consider the element uniqueness problem in the plane (or higher dimensions), and we show that this problem has an $\Omega(n \log n)$ lower bound in the algebraic computation tree model, too. Then, we use this problem to give an $\Omega(n \log n)$ lower bound in the algebraic computation tree model for computing (Steiner) spanners on a set of n points (in general position) in $\mathbb{R}^d$.

Now, we show that the element uniqueness problem also has an $\Omega(n \log n)$ lower bound when the points come from d-dimensional Euclidean space. This can be done by a simple mapping which maps each real number to a point in d-dimensional space which keeps uniqueness. This show that any algorithm for solving the element uniqueness problem in $\mathbb{R}^d$ can be used to solve the (standard) element uniqueness problem on $\mathbb{R}$. The proof of the following theorem is straight-forward, but for the sake of completeness we give the proof, too.

Theorem 1. *The following problem has an $\Omega(n \log n)$ time complexity in the algebraic computation tree model:*

- **Extended element uniqueness problem:** *given a set of n points in d-dimensional Euclidean space ($\mathbb{R}^d$), decide whether they are pairwise distinct.*

Proof. Let $\mathcal{A}$ be an arbitrary algebraic computation tree algorithm that solves the extended element uniqueness problem, in $\mathbb{R}^d$, and let $T(n)$ be its time complexity. The following algebraic decision tree algorithm solves the element uniqueness problem on an input consisting of n real numbers $x_1, x_2, \ldots, x_n$. The algorithm $\mathcal{B}$ first maps each x_i to $\mathbb{R}^d$ using the following mapping: $x \longmapsto (x, x, \ldots, x)$. Hence, $\mathcal{B}$ maps each x_i to $p_i = (x_i, \ldots, x_i)$ (clearly, if $i \neq j$, then we have $x_i = x_j$ if and only if $p_i = p_j$). Then, $\mathcal{B}$ uses $\mathcal{A}$ to decide that the n points $p_1, \ldots, p_n$ are pairwise distinct.

Algorithm $\mathcal{B}$ has time complexity $T(n) + \mathcal{O}(nd)$ (d is a constant). On the other hand, by the time complexity of the element uniqueness problem, Algorithm $\mathcal{B}$ has an $\Omega(n \log n)$ lower bound. It follows that $T(n) = \Omega(n \log n)$. □

Let $\mathcal{A}$ be any algorithm that, given a set P of n points $p_1, \ldots, p_n$ in $\mathbb{R}^d$ and a real number $t > 1$, constructs a Steiner t-spanner G for P. We assume that the number of edges in the generated spanner is $o(n \log n)$ because any algorithm that computes a spanner with $\Omega(n \log n)$ edges clearly needs $\Omega(n \log n)$ time. Also, we assume that the algorithm attaches a label to each vertex of the generated graph such that we can distinguish between original points and Steiner points.

We reduce the problem of computing a Steiner t-spanner to the extended element uniqueness problem in the same way as in [3] (see also [4, Section 3.4]): First, we use $\mathcal{A}$ to construct a Steiner t-spanner $G(V, E)$ for P. Let $G'(V, E')$ be the