Approximating the Complete Euclidean Graph

J. Mark Keil

Department of Computational Science
University of Saskatchewan
Saskatoon, Canada S7N 0W0

1. Introduction

Let $S$ be a set of $N$ points in the plane. Then a Euclidean graph $G(S)$ on $S$ has the property that each vertex corresponds to a point in $S$ and the weight of an edge is equal to the Euclidean distance between the points it connects [3]. Let $d(p,q)$ be the Euclidean distance between points $p$ and $q$ and let $G(p,q)$ be the length of the shortest path in $G(S)$ between $p$ and $q$. Only if $G$ is the complete Euclidean graph can we guarantee that $G(p,q) = d(p,q)$.

We say a Euclidean graph $G(S)$ approximates the complete Euclidean graph if the maximum value of the ratio $\frac{G(p,q)}{d(p,q)}$ is bounded by a constant for any pair of points $p$ and $q$ in $S$. The problem is to identify classes of graphs which closely approximate the complete Euclidean graph yet contain only a linear number of edges.

The first result of this type concerns the graph of the Delaunay triangulation in the $L_1$ norm, $DT_1(S)$. Chew [1] has shown that the ratio of shortest distances in $DT_1(S)$ to the Euclidean distances is bounded above by $\sqrt{10}$ independent of $S$ and $N$. Along similar lines it has recently been shown [2] that if $DT(S)$ is the graph of the Delaunay triangulation with the Euclidean norm then the ratio of shortest distances in $DT(S)$ to the Euclidean distance is bounded above by $\frac{1+\sqrt{5}}{2}\pi$ independent of $S$ and $N$. Both of these types of graphs contain only a linear number of edges and can be computed from the point set in $O(N\log N)$ time.

Approximating the complete Euclidean graph has potential application in the development of approximation algorithms or heuristics for problems that involve finding shortest distances in the plane [1,2,3]. In some applications it may be that the Delaunay approximations are not sufficiently close. In this paper we describe a class of Euclidean graphs each type of which very closely approximates the complete Euclidean graph, yet contains only a linear number of edges and can be efficiently computed.
2. Fixed Angle Theta Graph

Consider the shortest path from \( p \) to \( q \) in a Euclidean graph \( G \). Let \( r \) be the intermediate vertex adjacent to \( p \) on this path. If the angle \( qpr \) is small we might suspect that \( G(p,q) \) is not much longer than \( d(p,q) \), whereas if angle \( qpr \) is large we might suspect a greater difference between \( G(p,q) \) and \( d(p,q) \). This intuition suggests that by defining a type of graph in which each point will be connected to a near neighbour in each of a variety of directions we may approximate the complete Euclidean graph.

Given a set \( S \) of points in the plane we define the \( \theta \)-graph, for \( \theta = \frac{2\pi}{k} \), \( k \) an integer constant such that \( k > 8 \), to be the Euclidean graph, \( \theta(S) \), whose edges are defined as follows. Each point will be the source of at most \( k \) edges. A point \( p \) is the source of a type 1 edge if there exists a point \( q \) such that if the origin is located at \( p \) then the angle \( \phi \) between the \( x \)-axis and the ray \( \overrightarrow{pq} \) is such that \( 0 \leq \phi < \frac{2\pi}{k} \). If there are several such points, the type 1 edge from \( p \) will go to the point of minimum \( x \)-coordinate. In general, a point \( p \) will have a type \( i \) edge, \( 1 \leq i \leq k \), if there exists a point \( q \) such that, if the origin is located at \( p \), the angle \( \phi \) between the \( x \)-axis and the ray \( \overrightarrow{pq} \) is such that \( \frac{2\pi(i-1)}{k} \leq \phi < \frac{2\pi i}{k} \). If there are several such points the type \( i \) edge from \( p \) will go to the point whose projection on the ray from \( p \) at angle \( \frac{2\pi(i-1)}{k} \) with the \( x \)-axis is closest to \( p \).

To construct the \( \theta \)-graph for a given point set we present a simple plane sweep algorithm. The algorithm makes use of a separate plane sweep for each of the \( k \) different types of edges.

Let us consider the plane sweep in which the type \( i \) edges are identified. During the performance of this plane sweep, three different orderings of the points are employed. In each of the orderings the points are ordered by the ordering of their projections onto the oriented line through the origin which forms an angle of \( \phi \) with the \( x \)-axis and which is oriented by the ray based at the origin which forms an angle of \( \phi \) with the \( x \)-axis. In the \( \alpha \) ordering \( \phi = \frac{2\pi(i-1)}{k} \), while in the \( \beta \) ordering \( \phi = \frac{2\pi(i-1)}{k} + \frac{\pi}{2} \), while in the \( \gamma \) ordering \( \phi = \frac{2\pi i}{k} + \frac{\pi}{2} \). Let \( \alpha(p) \) (\( \beta(p), \gamma(p) \)) be the rank of point \( p \) in the \( \alpha \) (\( \beta, \gamma \)) ordering. Note that if \( i = 1 \) the \( \alpha \) ordering is the ordering of the points by \( x \)-coordinate and the \( \beta \) ordering is the ordering of the points by \( y \)-coordinate.

The point set is swept in nonincreasing order of \( \beta \) rank. As the sweep progresses a table \( T \) of active points to which type \( i \) edges may be destined is maintained in \( \gamma \) order. When a point \( p \) is encountered on the sweep the following operations are performed.