Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Discrete Mathematics 32 (2009) 117–124
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

Collective Tree Spanners for Unit Disk Graphs with Applications

Feodor F. Dragan, Yang Xiang and Chenyu Yan

Algorithmic Research Laboratory, Department of Computer Science
Kent State University, Kent, Ohio, U.S.A.
{*cyan,yxiang,dragan*}*@cs.kent.edu*

Abstract

In this paper, we establish a novel *balanced separator* theorem for *Unit Disk Graphs (UDGs)*, which mimics the well-known Lipton and Tarjan's planar balanced shortest paths separator theorem. We prove that, in any n-vertex UDG G, one can find two hop-shortest paths $P(s,x)$ and $P(s,y)$ such that the removal of the 3-hop-neighborhood of these paths (i.e., $N_G^3[P(s,x) \cup P(s,y)]$) from G leaves no connected component with more than $2/3n$ vertices. This new *balanced shortest-paths—3-hop-neighborhood separator* theorem allows us to build, for any n-vertex UDG G, a system $\mathcal{T}(G)$ of at most $2\log_{\frac{3}{2}} n + 2$ spanning trees of G such that, for any two vertices x and y of G, there exists a tree T in $\mathcal{T}(G)$ with $d_T(x,y) \leq 3 \cdot d_G(x,y) + 12$. That is, the distances in any UDG can be approximately represented by the distances in at most $2\log_{\frac{3}{2}} n + 2$ of its spanning trees. Using these results, we propose a *new compact and low delay routing labeling scheme* for UDGs.

Keywords: unit disk graphs, collective tree spanners, routing and distance labeling schemes, balanced separators, efficient graph algorithms.

1571-0653/$ – see front matter, Published by Elsevier B.V.
doi:10.1016/j.endm.2009.02.016

1 Introduction

A common assumption for wireless ad hoc networks is that all nodes have the same maximum transmission range. By proper scaling, one can model these networks with *Unit Disk Graphs* (*UDGs*), which are defined as the intersection graphs of equal sized circles in the plane. In other words, there is an edge between two vertices in an UDG if and only if their Euclidean distance is no more than one.

In this paper, we propose a new compact and low delay routing labeling scheme for Unit Disk Graphs. We show that one can assign each vertex of an n-vertex UDG G a compact $O(\log^2 n)$-bit label such that, given the label of a source vertex and the label of a destination, it is possible to compute efficiently, based solely on these two labels, a neighbor of the source vertex that heads in the direction of the destination. We prove that this *routing labeling scheme* has a constant *hop route-stretch* (= *hop delay*), i.e., for each two vertices x and y of G, it produces a routing path with $h(x,y)$ hops such that $h(x,y) \leq 3 \cdot d_G(x,y) + 12$, where $d_G(x,y)$ is the hop distance between x and y in G. To the best of our knowledge, this is the first compact routing scheme for UDGs which not only guaranties delivery but has a low hop delay. It is easy to see that, for UDGs, a constant hop route-stretch implies a constant length route-stretch. Note also that, unlike geographic routing or any other routing strategies for UDGs (see [3,4,8] and papers cited therein), our routing scheme is *degree-independent*. The label assigned to a vertex in our scheme can be interpreted as its virtual coordinates. To assign those labels to vertices, we need to know only the topology of the input unit disk graph and relative Euclidean lengths of its edges.

To obtain our routing scheme, we establish a novel *balanced separator* theorem for UDGs, which mimics the well-known Lipton and Tarjan's planar balanced shortest paths separator theorem. We prove that, in any n-vertex UDG G, one can find two hop-shortest paths $P(s,x)$ and $P(s,y)$ such that the removal of the 3-hop-neighborhood of these paths (i.e., $N^3_G[P(s,x) \cup P(s,y)]$) from G leaves no connected component with more than $2/3n$ vertices. Our new *balanced shortest-paths—3-hop-neighborhood separator* theorem allows us to build, for any n-vertex UDG $G = (V,E)$, a system $\mathcal{T}(G)$ of at most $2\log_{\frac{3}{2}} n + 2$ spanning trees of G such that, for any two vertices x and y of G, there exists a tree T in $\mathcal{T}(G)$ with $d_T(x,y) \leq 3 \cdot d_G(x,y) + 12$. That is, the distances in any UDG can be approximately represented by the distances in at most $2\log_{\frac{3}{2}} n + 2$ of its spanning trees. Taking the union of all these spanning trees of G, we obtain a *hop* $(3,12)$*-spanner* H of G (i.e., a spanning

subgraph H of G with $d_H(x,y) \leq 3 \cdot d_G(x,y) + 12$ for any $x, y \in V$) with at most $O(n \log n)$ edges. There is a number of papers describing different types of *power-spanners*, *length-spanners* and *hop-spanners* for UDGs (see [2,5] and literature cited therein). Many of those spanners have nice properties of being planar or sparse, or having bounded maximum degree or bounded length (or hop) *spanner-stretch*, or having localized construction. Unfortunately, neither of those papers develops or discusses any routing schemes which could translate the constant spanner-stretch bounds into some constant route-stretch bounds.

2 Notions and notations

Let V be a set of $n = |V|$ nodes on the Euclidean plane and let $G = (V, E)$ be the unit disk graph (UDG) induced by those nodes. Let also $m = |E|$. For each edge (a, b) of G, by (a, b) we denote also the open straightline segment representing it, and by $|ab|$ the Euclidean length of the edge/segment (a, b). For simplicity, in what follows, we will assume that any two edges in G can intersect at no more than one point (i.e., no two intersecting edges are on the same straight line), and no three edges intersect at the same point.

For a path P of G, the *hop-count* of P is defined as the number of edges on P and the *length* of P is defined as the sum of the Euclidean length of its edges. For any two vertices x and y of G, we denote: by $d_G(x, y)$, the *hop-distance* (or simply *distance*) in G between x and y, i.e., the minimum hop-count of any path connecting x and y in G; by $l_G(x, y)$, the *length-distance* in G between x and y, i.e., the minimum length of any path connecting x and y in G.

Let $H = (V, E')$ be a spanning subgraph of a graph $G = (V, E)$. We say that H is: *hop* (α, β)*-spanner* of G if $d_H(x, y) \leq \alpha \cdot d_G(x, y) + \beta$, for any $x, y \in V$; *length* (α, β)*-spanner* of G if $l_H(x, y) \leq \alpha \cdot l_G(x, y) + \beta$, for any $x, y \in V$. It is said (see [1]) that a graph G admits a system of μ collective tree (α, β)-spanners if there is a system $\mathcal{T}(G)$ of at most μ spanning trees of G such that for any two vertices x,y of G a spanning tree $T \in \mathcal{T}(G)$ exists such that $d_T(x, y) \leq \alpha \cdot d_G(x, y) + \beta$.

For a vertex v of G, the kth *neighborhood* of v in G is the set $N_G^k[v] = \{u \in V : d_G(v, u) \leq k\}$. For a set $S \subseteq V$, by $N_G^k[S] = \bigcup_{v \in S} N_G^k[v]$ we denote the kth *neighborhood* of S in G.

A graph family Γ is said (see [7]) to have an $l(n)$ *bit* (s, r)*-approximate distance labeling scheme* if there is a function *Label* labeling the vertices of each n-vertex graph in Γ with distinct labels of up to $l(n)$ bits, and there exists an algorithm/function f, called *distance decoder*, that given two labels

$Label(v), Label(u)$ of two vertices v, u in a graph G from Γ, computes, in time polynomial in the length of the given labels, a value $f(Label(v), Label(u))$ such that $d_G(v,u) \leq f(Label(v), Label(u)) \leq s \cdot d_G(v,u) + r$. Note that the algorithm is not given any additional information, other that the two labels, regarding the graph from which the vertices were taken. Similarly, a family Γ of graphs is said (see [7]) to have an $l(n)$ *bit routing labeling scheme* if there exist a function $Label$, labeling the vertices of each n-vertex graph in Γ with distinct labels of up to $l(n)$ bits, and an efficient algorithm/function, called the *routing decision* or *routing protocol*, that given the label $Label(v)$ of a current vertex v and the label $Label(u)$ of the destination vertex u (the header of the packet), decides in time polynomial in the length of the given labels and using only those two labels, whether this packet has already reached its destination, and if not, to which neighbor of v to forward the packet.

Let $\mathcal{R}$ be a routing scheme and $R(x,y)$ be a route (path) produced by $\mathcal{R}$ for a pair of vertices x and y in a graph G. We say that $\mathcal{R}$ has: *hop* (α,β)-*route-stretch* if hop-count of $R(x,y)$ is at most $\alpha \cdot d_G(x,y) + \beta$, for any $x, y \in V$; *length* (α,β)-*route-stretch* if length of $R(x,y)$ is at most $\alpha \cdot l_G(x,y) + \beta$, for any $x, y \in V$.

3 Intersection lemmas and preliminaries

In this section we present few auxiliary lemmas. Their proofs are omitted in this version.

Lemma 3.1 *In an UDG $G = (V,E)$, if edges $(a,b),(c,d) \in E$ intersect, then G must have at least one of $(a,c),(b,d)$ and at least one of $(a,d),(c,b)$ in E.*

Let r be an arbitrary but fixed vertex of an UDG $G = (V,E)$, and $L_0, L_1, \ldots L_q$ be the *layering* of G with respect to r, where $L_i = \{u \in V : d_G(r,u) = i\}$. For G, using this layering, we construct a *layering tree* T_{orig} rooted at r as follows: each vertex $v \in L_i$ $(i \in \{1,\ldots,q\})$ chooses a neighbor u in L_{i-1} such that $|vu|$ is minimum (closest neighbor in L_{i-1}) to be its father in T_{orig} (breaking ties arbitrarily). Let $E(T_{orig})$ be the edge set of T_{orig}. This tree T_{orig} will help us to construct a balanced separator for G. It will be convenient, for each vertex $v \in V$, by $L(v)$ to denote the layer index of v, i.e., $L(v) = d_G(r,v)$.

Lemma 3.2 *In T_{orig}, no two edges (a,b) and (c,d) with $L(a) = L(c)$ and $L(b) = L(d)$ can cross.*

Lemma 3.3 *Let $(a,b),(c,d)$ be two edges in T_{orig} that intersect. If $L(a) = L(b) - 1$, $L(c) = L(d) - 1$ and $L(a) \leq L(c)$, then $L(a) = L(c) - 1$, $(a,d) \notin E$ and $(b,c) \in E$.*

For an UDG $G = (V, E)$, in what follows, by $G_p = (V_p, E_p)$ we denote the planar graph obtained from G by turning each edge intersection point in G into a vertex in G_p. The vertices of T_{orig} (i.e. vertices of G) will be called *real vertices*, to differentiate them from *imaginary* and *null* points that will be defined later. In the following, we will use the term "element" as a general name for real vertices, imaginary points and null points. Below, we will create an imaginary point (details will be given later) at the point where two edges (a,b) and (c,d) from T_{orig} intersect. By Lemma 3.3, we may assume that $L(a) = L(c) - 1$. Now, assuming that the imaginary point is m, we define $a(m) = a$, $b(m) = b$, $c(m) = c$ and $d(m) = d$.

4 Balanced separators for UDGs

In this short version of the paper, we demonstrate our idea of construction of a balanced separator on a simple case of special unit disk graphs, so-called simple-crossing UDGs. For much more complicated case, where we construct a balanced separator for an arbitrary UDG, we refer the reader to the full version of the paper. We define a *simple-crossing UDG* to be an UDG with each edge crossing at most one other edge.

In what follows, we will transform tree T_{orig} into a special spanning tree T for the planar graph G_p. Let $T = T_{orig}$ initially. For each two intersecting edges (a,b) and (c,d) of T_{orig} (by Lemma 3.3, we know $L(a) = L(c) - 1$), we do the following. Create a vertex $m_{a,b,c,d}$ at the point where (a,b) and (c,d) intersect. We call $m_{a,b,c,d}$ an *imaginary point*. Remove edges (a,b), (c,d) from T and add vertex $m_{a,b,c,d}$ and edges $(m_{a,b,c,d}, d)$, $(a, m_{a,b,c,d})$ and $(b, m_{a,b,c,d})$ into T. One can see that all the descendants of b and d in T find their way to the root via a. There are two other kinds of edge intersections in G. Assume a tree-edge (u,w) intersects a non-tree-edge (s,t). We create a new vertex, called a *null point*, say o, at the point where (u,w) and (s,t) intersect. We remove edge (u,w) from T and add vertex o and edges (u,o), (o,w) into T. Assume two non-tree-edges (a,b) and (c,d) intersect. We create a new vertex, called a *null point*, say o, at the point where (a,b) and (c,d) intersect. We add vertex o (as a pendant vertex) and edge (a,o) into T.

It is easy to see that T is a spanning tree for G_p. We will need the Lipton and Tarjan's planar separator theorem [6] in the following form.

Theorem 4.1 [6] *Let G be any planar graph with non-negative vertex weights and W be the total weight of G (which is the sum of the weights of its vertices). Let T be any spanning tree of G rooted at a vertex r. Then, there exist two vertices x and y in G such that if one removes from G the tree-paths connecting in T r with x and r with y, then each connected component of the resulting graph has total weight at most $2/3W$.*

We can apply Theorem 4.1 to T and G_p by letting the weight of each real vertex be 1 and the weight of each imaginary or null point be 0 in G_p. Then, there must exist in T two paths $P_1 = P_T(r, x)$ and $P_2 = P_T(r, y)$ such that removal of them from G_p leaves no connected component with more than $2/3n$ real vertices.

Using paths $P_1 = (x_0 = r, x_1, \ldots, x_{k-1}, x_k = x)$ and $P_2 = (y_0 = r, y_1, \ldots, y_{l-1}, y_l = y)$ of G_p (of T), we can create a balanced separator for G as follows:
(1) Skip all the null points in P_1 and P_2;
(2) Skip every imaginary point in P_i which is collinear with its two neighbors in P_i $(i = 1, 2)$;
(3) For any imaginary point $m_{a,b,c,d}$ in P_i $(i = 1, 2)$ which is not collinear with its two neighbors in P_i (the only possible case is when $L(a) = L(c) - 1$ and imaginary point $m_{a,b,c,d}$ connects a and d in P_i), replace the subpath $(a, m_{a,b,c,d}, d)$ by either (a, c, d) (if $(a, c) \in E$) or (a, b, d) (if $(b, d) \in E$). By Lemma 3.1, (a, c) or (b, d) is in E.

Let P_i' be the resulting path obtained from P_i $(i = 1, 2)$. It is easy to check that P_1' and P_2' are hop-shortest paths in G. We can also show that the union of $N_G^1[P_1']$ and $N_G^1[P_2']$ is a balanced separator for G, i.e., removal of $N_G^1[P_1'] \cup N_G^1[P_2']$ from G leaves no connected component with more that $2/3n$ vertices. Assume that removal of P_1 and P_2 from $G_p = (V_p, E_p)$ results in removing a set of edges E_p' from E_p, and removal of $N_G^1[P_1']$ and $N_G^1[P_2']$ from $G = (V, E)$ results in removing a set of edges E' from E. It is easy to check that, for any edge $e_p' \in E_p'$ there exists an edge $e' \in E'$ that covers e_p'. The latter implies that $N_G^1[P_1'] \cup N_G^1[P_2']$ is a balanced separator for G.

In an arbitrary unit disk graph G, an edge may cross any number of other edges. Our basic strategy for building a balanced separator for G is similar to one we used in the case of a simple-crossing UDG, but details are much more complicated. In this version of the paper, we report only the final result.

Theorem 4.2 *In any unit disk graph G, one can find two hop-shortest paths P_1 and P_2 such that the union of $N_G^3[P_1]$ and $N_G^3[P_2]$ is a balanced separator for G with $2/3$-split, i.e., removal of $N_G^3[P_1] \cup N_G^3[P_2]$ from G leaves no connected component with more than $2/3n$ vertices.*

5 Collective tree spanners for UDGs with applications

In this section, we show how one can use the above balanced separator theorem for UDGs to construct for them collective tree spanners with low stretch and to develop a compact and low delay routing labeling scheme. The details can be found in the full version of this paper. Here we list only the final results.

Theorem 5.1 *Any unit disk graph G with n vertices and m edges admits a system $\mathcal{T}(G)$ of at most $2\log_{3/2} n + 2$ collective tree $(3, 12)$-spanners, i.e., for any two vertices x and y in G, there exists a spanning tree $T \in \mathcal{T}(G)$ with $d_T(x, y) \leq 3d_G(x, y) + 12$. Moreover, such a system $\mathcal{T}(G)$ can be constructed in $O((C + m)\log n)$ time, where C is the number of crossings in G.*

Corollary 5.2 *Any unit disk graph G with n vertices admits a hop $(3, 12)$-spanner with at most $2(n - 1)(\log_{3/2} n + 1)$ edges.*

Theorem 5.3 *The family of n-vertex unit disk graphs admits an $O(\log^2 n)$ bit $(3, 12)$-approximate distance labeling scheme with $O(\log n)$ time distance decoder.*

Theorem 5.4 *The family of n-vertex unit disk graphs admits an $O(\log^2 n)$ bit routing labeling scheme. The scheme has hop $(3, 12)$-route-stretch. Once computed by the sender in $O(\log n)$ time, headers never change, and the routing decision is made in constant time per vertex.*

Theorem 5.5 *Any unit disk graph G with n vertices and m edges admits a system $\mathcal{T}(G)$ of at most $2\log_{3/2} n+2$ collective tree length $(5, 13)$-spanners, i.e., for any two vertices x and y in G, there exists a spanning tree $T \in \mathcal{T}(G)$ with $l_T(x, y) \leq 5l_G(x, y) + 13$. Moreover, such a system $\mathcal{T}(G)$ can be constructed in $O((C + m)\log n)$ time, where C is the number of crossings in G.*

Corollary 5.6 *Any unit disk graph G with n vertices admits a length $(5, 13)$-spanner with at most $2(n - 1)(\log_{3/2} n + 1)$ edges.*

Theorem 5.7 *The family of n-vertex unit disk graphs admits an $O(\log^2 n)$ bit $(5, 13)$-approximate length-distance labeling scheme with $O(\log n)$ time distance decoder.*

Theorem 5.8 *The family of n-vertex unit disk graphs admits an $O(\log^2 n)$ bit routing labeling scheme. The scheme has length $(5, 13)$-route-stretch. Once computed by the sender in $O(\log n)$ time, headers never change, and the routing decision is made in constant time per vertex.*

6 Conclusion

In this paper, we showed that every unit disk graph G has a balanced separator of form $N^3_G[P1] \cup N^3_G[P2]$, where $P1$ and $P2$ are hop-shortest paths of G. Using this separator theorem, we constructed for unit disk graphs collective tree spanners with low stretch and developed routing labeling schemes with $O(log^2 n)$ bit labels and hop (3,12)-route-stretch and length (5,13)-route-stretch. It is interesting to know if those stretch factors can be improved and if every unit disk graph G admits a balanced separator of form $N^1_G[P1] \cup N^1_G[P2]$, where $P1$ and $P2$ are (hop- or length-) shortest paths of G.

References

[1] Dragan, F.F., Yan, C., Lomonosov, I.: Collective tree spanners of graphs. SIAM J. Discrete Math 20, 241–260 (2006)

[2] Gao, J., Guibas, L.J., Hershberger, J., Zhang, L., Zhu, A.: Geometric spanner for routing in mobile networks. Proceedings of the 2nd ACM international symposium on mobile ad hoc networking & computing, October 04-05, 2001, Long Beach, CA, USA

[3] Giordano, S., Stojmenovic, I.: Position based routing algorithms for ad hoc networks: A taxonomy. In X. Cheng, X. Huang, and D. Du, editors, Ad Hoc Wireless Networking, pages 103–136. Kluwer (2004)

[4] Kuhn, F., Wattenhofer, R., Zhang, Y., Zollinger, A.: Geometric ad-hoc routing: of theory and practice. Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing, ACM Press, pp. 63–72 (2003)

[5] Li, X.-Y., Wang, Y.: Geometrical Spanner for Wireless Ad Hoc Networks. Handbook of Approximation Algorithms and Metaheuristics (Editor: Teofilo F. Gonzalez), Chapman & Hall/Crc (2006)

[6] Lipton, R.J., Tarjan, R.E.: A Separator Theorem for Planar Graphs. SIAM Journal on Applied Mathematics 36, 177-189, (1979).

[7] Peleg, D.: Distributed Computing: A Locality-Sensitive Approach. SIAM Monographs on Discrete Math. Appl., SIAM, Philadelphia (2000).

[8] Rao, A., Papadimitriou, C., Shenker, S., Stoica, I.: Geographical routing without location information. Proceedings of MobiCom 2003, pp. 96–108 (2003)