A Class of Orientation-Invariant Yao-type Subgraphs of a Unit Disk Graph *

T. Fevens
Dept. of Computer Science
and Software Engineering
Concordia University
Montral, Qubec, Canada
fevens@cs.concordia.ca

A.E. Abdallah
Dept. of Computer Science
and Software Engineering
Concordia University
Montral, Qubec, Canada
ae_abdal@cs.concordia.ca

T. El Salti
Dept. of Computer Science
and Software Engineering
Concordia University
Montral, Qubec, Canada
t_elsal@cs.concordia.ca

L. Harutyunyan
Dept. of Computer Science
and Software Engineering
Concordia University
Montral, Qubec, Canada
l_haruty@cs.concordia.ca

ABSTRACT

We introduce a generalization of the Yao graph where the cones used are adaptively centered on a set of nearest neighbors for each node, thus creating a directed or undirected spanning subgraph of a given unit disk graph (UDG). We also permit the apex of the cones to be positioned anywhere along the line segment between the node and its nearest neighbor, leading to a class of Yao-type subgraphs. We show that these locally constructed spanning subgraphs are strongly connected, have bounded out-degree, are t-spanners with bounded stretch factor, contain the Euclidean minimum spanning tree as a subgraph, and are orientation-invariant. Since a continuous set of cone angles are possible, these subgraphs also permit control over the degree of the graph. We demonstrate through simulations that these subgraphs of the UDG combines the desirable properties of the Yao and the Half Space Proximal subgraphs of the UDG.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network topology*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Theory, Design, Algorithms

Keywords

Ad hoc networks, geometric graph, spanning graph, unit disk graph, topology control

*The work in this paper was supported (in part) by NSERC.

Dial M-POMC '07 Portland, Oregon USA

1. INTRODUCTION

An *ad hoc network* is a system of wireless autonomous hosts that can communicate with each other without having any fixed infrastructure. Each host in the network can communicate with all other hosts within its transmission range [1, 2], which we will assume to be a fixed range r for all hosts. If two hosts are not able to communicate directly then a multi-hop routing protocol is needed for the hosts to send packets to each other. We will assume the position of the hosts can be obtained using GPS, for example, or some other technique. There are numerous ways to use position information in making routing decisions [13].

The set of N wireless hosts can be represented as a point set S in the Euclidean two-dimensional plane $\mathbb{R}^2$, each point possessing a geometric location. On S, a (Euclidean) graph can be modeled as a weighted (undirected or directed) graph $G = (S, E)$ where E is a subset of the pairs of nodes of S and the weight of an edge uv between nodes u and v is the Euclidean distance between the nodes which we denote as $|uv|$. The weight of a graph is the sum of its edge weights.

Further, in our wireless host model, two nodes are connected by an edge if the Euclidean distance between them is at most r, the transmission range of the nodes in S. The resulting graph is called a unit disk graph ($UDG(S)$). For node u, we denote the set of its neighbors by $N(u)$. Define a subgraph of G, $P(G)$, as a t-spanner of G if the length of the shortest path between any two nodes in $P(G)$ is not more than t times longer than the shortest path connecting them in G, where t is the stretch factor. The length of the path is the sum of the lengths of the edges along the path. A t-spanning path from a to b is *strong* if the length of every edge in the path is at most $|ab|$. The graph G is a strong t-spanner if there is a strong t-spanning path between every pair of vertices.

Many routing strategies use a spanning subgraph of the unit disk graph such that only the edges in the subgraph are used for routing. Therefore, much research effort has gone into the development of algorithms for topology control for ad hoc networks (see [9, 14] for surveys). Since the wireless hosts that we are modeling are commonly powered by a limited power supply like a battery as well as containing a limited amount of memory, may be mobile, and the topology of the whole network is usually not available and may be variable, localized algorithms (using information on neighboring nodes up to a fixed number of hops away) are typically preferred. These algorithms are designed to achieve various objectives such as bounding the node degree [17], planarity [3, 7, 16], low weight (close in weight to a minimal spanning tree) [12], power efficiency [11], bounding the stretch factor [10], or creating a bounded degree planar power spanners of a $UDG(S)$ with bounded power stretch factor (where the cost of a path is the sum of Euclidean lengths raised to some exponent p between 2 and 5 of all the edges in the path) [8, 15]. For a geometric graph G, an *Euclidean Minimum Spanning Tree* $EMST(G)$ is a minimum weight spanning tree of G. Several subgraphs also contain the $EMST(G)$ as a subgraph [6, 18]. In particular, we are interested in the Half Space Proximal subgraph [6] and Yao subgraph [18] derived from the unit disk graph.

For a geometric graph G, a *Yao Graph* (also called a *Theta Graph* [5]) $YG_k(G)$ with an integer parameter $k \geq 6$ is defined as follows [18]. First, we will define a directed Yao graph, $D\text{-}YG_k(G)$, for G. At each node u in G, k equally-separated rays originating at u define k cones. In each cone, only the directed edge (u, v) to the nearest neighbor v, if any, is part of $D\text{-}YG_k(G)$. Ties are broken arbitrarily. Let $YG_k(G)$ be the undirected graph obtained if the direction of each edge in $D\text{-}YG_k(G)$ is ignored, yielding a subgraph which may have crossing edges if $G = UDG$. The graph $YG_k(G)$ is a $1/(1 - 2\sin(\pi/k))$-spanner of G [11], has an out-degree of at most k, and contains the $EMST(G)$ as a subgraph [18]. One drawback of the $YG_k(G)$ graph is that it is not orientation-invariant. That is, if G is rotated by an arbitrary angle to give G' then the resulting $YG_k(G')$ subgraph is not necessarily a rotation of $YG_k(G)$.

For a geometric graph G, a *Half Space Proximal Graph* $HSP(G)$ is defined as follows [6]. As with the *Yao Graph*, first a directed $D\text{-}HSP(G)$ is defined. At each node u in G, the following iterative procedure is performed until all the neighbors of u are either discarded or are connected with an edge. A directed edge (u, v) is formed with the nearest neighbor v. An open half plane is defined by a line perpendicular to (u, v), intersecting (u, v) at its midway point, and containing v. All the nodes in this half plane are then discarded. The procedure then continues with the next nearest non-discarded neighbor and so on until all the nodes have been discarded. The selected directed edges determine the $D\text{-}HSP(G)$. The undirected $HSP(G)$ is obtained by ignoring the direction of the edges, yielding a subgraph that may still have crossing edges. Among the properties shown in [6] for the HSP subgraph that it is strongly connected, has an out-degree of at most six, has a stretch factor of at most $2\pi + 1$, contains the $EMST(G)$ as its subgraph, and is orientation-invariant. Bose *et al.* [4] show that this stretch factor of at most $2\pi + 1$ is incorrect and that no upper bound is known although the stretch factor is at least $3 - \epsilon$. Another drawback of the $HSP(G)$ graph is that, since the forbidden region is always defined by a straight line, there is no control over the degree of a node.

Recently, Bose *et al.* [4] introduced a family of directed geometric graphs, related to the HSP, that depend on two parameters λ and θ. For $0 \leq \theta < \pi/2$ and $1/2 < \lambda < 1$, their graph is a strong t-spanner, with $t = 1/((1-\lambda)\cos(\theta))$. The out-degree of a node at most $\lfloor (2\pi/(\min(\theta, \arccos(1/2\lambda)))\rfloor$.

In the next section, we generalize the definition of the Yao graph to define a class of orientation-invariant Yao-type graphs which includes the HSP graph as a special case. This class of Yao-type graphs combines the advantages of both the HSP subgraph and the Yao subgraph by permitting control over the degree of the subgraph while being orientation-invariant. For these graphs, in Section 3, we explore their properties experimentally.

2. DISPLACED APEX ADAPTIVE YAO GRAPHS

In this section, we give a formal definition of a class of Yao-type graphs and prove some basic properties of the graphs. Let S be a set of N points in the Euclidean two dimensional plane, each point possessing a geometric location. For the following, define the cone angle θ to be the half-angle of the cone's apex.

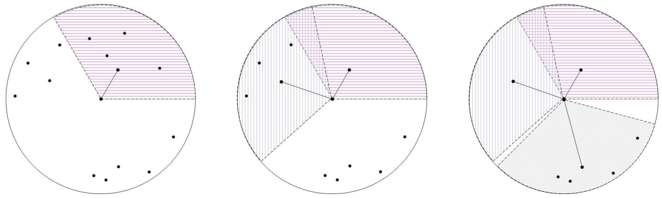
We will use the parameter s to parametrize the closed line segment between u and v: $(1 - s)u + sv$, $0 \leq s \leq 1$. Any particular choice of s represents the position of the apex of the cone. We will use a second parameter α, $0 \leq \alpha \leq 1$, to determine θ as a fraction of the maximum cone angle, $\theta_m(s, |uz|)$, which we define shortly, which is a function of s and the distance from the current node u to the nearest neighbor z for which the cone is determined.

Algorithm 1 Displaced Apex Adaptive Yao(G, α, s) graph algorithm

Input: *A graph G with the node set S, an angle parameter α, and a parameter s.*
Output: *A list of directed edges L for each node $u \in S$ which represent the Displaced Apex Adaptive Yao subgraph of G, $DAAY(G, \alpha, s)$.*

for all $u \in S$ **do**
  Create a list of neighbors of u: $LN(u) = N(u)$.
  repeat
    (a) Remove the nearest neighbor z node from $LN(u)$ and add the directed edge uz to L.
    (b) Determine $\theta_m(s, |uz|)$ such that $\theta = \alpha \cdot \theta_m(s, |uz|)$.
    (c) Let $r = (1-s)u + sz$ be a point on the line segment uz.
    (d) Consider the cone C with its apex at r with a cone angle θ and z in its interior, such that the line uz bisects the cone C into two equal halves (i.e., the segment uz lies in the center of the cone).
    (e) Scan the list $LN(u)$ and remove each node in the interior of C.
  until $LN(u)$ is empty
end for

Definition 1. Let G be a UDG with node set S. The directed Displaced Apex Adaptive Yao subgraph, $D\text{-}DAAY(G, \alpha, s)$, is defined to be the graph with node set S whose edges are obtained by applying the Displaced Apex Adaptive Yao(G, α, s) algorithm, Algorithm 1, on the graph G using cone angle $\theta = \alpha \cdot \theta_m(s, |uz|)$ and apex displacement parameter s. The undirected graph $DAAY(G, \alpha, s)$ is obtained by ignoring the direction of the edges in $D\text{-}DAAY(G, \alpha, s)$.



(a) (b) (c)

Figure 1: Applying the Displaced Apex Adaptive Yao($UDG, 1, 0$) graph algorithm on the node u of a UDG: (a) the nearest neighbor is first chosen; (b) the second nearest node out of the rest of the nodes is chosen. Note that its associated cone overlaps with the first cone; and (c) the third nearest neighbor is chosen from the list $LN(u)$.

When $s = 0$, we simply refer to the resultant graph as the *Adaptive Yao* graph. Note that the directions of the cones used in the Displaced Apex Adaptive Yao(G, α, s) algorithm only depend on the relative directions of the selected nearest neighbors. Therefore, the resultant subgraph is the same regardless of the orientation of the point set S. Hence the $DAAY(G, \alpha, s)$ is orientation-invariant. See Fig. 1. The running time of Algorithm 1 per node is $O(l^2)$ where l is the degree of the node and results in a subgraph that may still have crossing edges.

After determining some properties of cone angles, we define the maximum cone angle $\theta_m(s, |uz|)$.

Lemma 1. *Consider a node u and neighbor z of u. Consider an arbitrary point $k = (1-s)u+sz$ where the parameter s has a value in the range $[0, 1]$. Define L to be the line perpendicular to the line segment uz and that intersects uz at its midpoint m (corresponding to $s = 0.5$ in the above line equation). Define a cone with cone angle θ, with its apex at k, oriented such that z is in its interior and the segment uz lies in the center of the cone. See Fig. 2. Consider the boundary of this cone intersecting the line L at a point c. Then the cone angle θ satisfies*

$$\frac{\sin(\theta - \theta_0)}{\sin(\theta)} = \frac{s|uz|}{|uc|}, \qquad \textit{where } \cos(\theta_0) = \frac{1}{2}\frac{|uz|}{|uc|}.$$

Proof. Consider the triangle $\triangle uck$. Let θ_0 be the interior angle at u. Then the interior angle at c is $\theta - \theta_0$. The interior angle can be determined from the right triangle $\triangle muc$, $\cos(\theta_0) = \frac{1}{2}\frac{|uz|}{|uc|}$. Also, the interior angle at k is $\pi - \theta$. By the sine law,

$$\frac{\sin(\theta - \theta_0)}{\sin(\theta)} = \frac{s|uz|}{|uc|}.$$

□

Corollary 1. Using the same definitions as in Lemma 1, if $|uc| = |uz|$ then $\theta_0 = \pi/3$ and

$$\frac{\sin(\theta - \pi/3)}{\sin(\theta)} = \frac{s|uz|}{|uz|} = s.$$

Definition 2. Using the same definitions as in Lemma 1, define the maximum cone angle $\theta_m(s, |uz|)$ as follows, as a function of the parameter s and the distance $|uz|$:

$$\frac{\sin\left(\theta_m(s, |uz|) - \frac{\pi}{3}\right)}{\sin(\theta_m(s, |uz|))} = s \qquad \text{if } 0 \leq s < 0.5$$

$$\frac{\sin\left(\theta_m(s, |uz|) - \cos^{-1}\left(\frac{1}{2}\frac{|uz|}{r}\right)\right)}{\sin(\theta_m(s, |uz|))} = \frac{s|uz|}{r} \qquad \text{if } 0.5 \leq s \leq 1$$

Note that when $0 \leq s \leq 0.5$, then $\theta_m(s, |uz|)$ is only a function of s such that θ is a fixed angle for fixed values of s and α. When $s = 0.5$, $\theta_m(0.5, |uz|) = \pi/2$ and, if $\alpha = 1$ such that $\theta = \theta_m(0.5, |uz|)$, we obtain the Half Space Proximal subgraph [6].

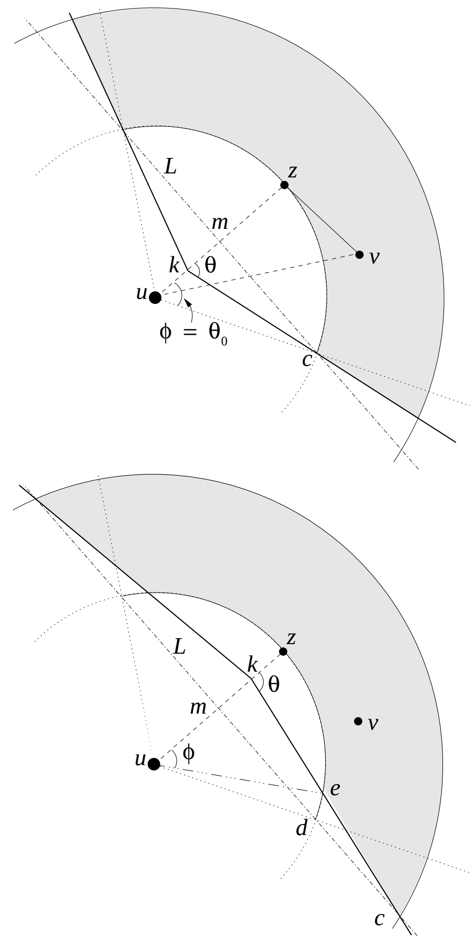


Figure 2: Figure for Theorem 1. The top diagram is for $0 \leq s < 0.5$. The bottom diagram is for $0.5 \leq s \leq 1$. In both diagrams, the dark shaded area is the forbidden region where any other neighboring nodes are excluded. The radius of the outer ring is the transmission range r.

Theorem 1. *Consider a node set S and $UDG(S)$ defined on S. If $UDG(S)$ is connected and the cone angle θ is less than or equal to $\theta_m(s, |uz|)$ then D-DAAY($UDG(S), \alpha, s$) is strongly connected.*

Proof. Consider a proof by contradiction. Assume that there is at least one edge $uv \in UDG(S)$ such that there is no directed path from u to v in $D\text{-}DAAY(UDG(S), \alpha, s)$.

Let uv be the shortest such edge in $UDG(S)$. This implies that there is an edge $uz \in$ D-DAAY$(UDG(S), \alpha, s)$ such that $|uz| < |uv|$, because the edge uv should be in the cone of uz selected by the Displaced Apex Adaptive Yao$(UDG(S), \alpha, s)$ algorithm.

Now consider the triangle $\triangle uzv$ in Fig. 2. The choice of maximum cone angles in Def. 2 is based on the idea that it ensures that v is contained in the open half-plane H containing z defined by the line perpendicular to the line segment uz in the middle of uz (the point corresponding to $s = 0.5$ in our parametrization of the uz; labeled as m in Fig. 2). Consider the two cases defined by the value of s. First, assume $0 \le s < 0.5$ (for example, the apex of the cone would be at the point labeled as k in the figure). Then to keep v in the interior of H, the maximum cone angle for θ would define a cone that intersects the boundary of H at a point at distance $|uz|$ from u (such a point is labeled as c in the figure). By Corollary 1, $\theta_m(s, |uz|)$ is as defined in Def. 2.

Now, assume $0.5 \le s \le 1$. To keep v in the interior of H, the maximum cone angle for θ would define a cone that intersects the boundary of H at a point at distance r from u (such a point is labeled as c in the Fig. 2). By Lemma 1 and noting that $|ud| = r$, $\theta_m(s, |uz|)$ is as defined in Def. 2.

In either case, since the cone angle is less than or equal to $\theta_m(s, |uz|)$, then any position of the node v inside the cone for z such that $|uz| \le |uv|$ would give $|zv|$ strictly less than $|uv|$. Since uv is an edge in $UDG(S)$, then zv is also an edge in $UDG(S)$. Therefore, there exists a directed path from z to v in D-DAAY$(UDG(S), \alpha, s)$, and so there is a directed path from u to v in D-DAAY$(UDG(S), \alpha, s)$. $\square$

Note that when $s > 0.5$, the further away the nearest neighbor z, the larger the cone angle limit $\theta_m(s, |uz|)$. If a fixed cone angle $\hat{\theta}$ was used, since as $|uz| \to 0$ the cone angle approaches $\pi/2$, it would have to be $\hat{\theta} \le \pi/2$ to ensure connectedness. Using $\hat{\theta} = \pi/2$ would then give a variation of the Half Space Proximal subgraph with the forbidden zone half-plane intersecting the line segment uz at the point corresponding to $s > 0.5$ rather than at the midway point. We will not consider this variation of the HSP graph.

THEOREM 2. *The out-degree of any node in D-DAAY$(UDG(S), \alpha, s)$, $\theta = \alpha \cdot \theta_m(s, |uz|)$, $0 \le \alpha \le 1$, is at most* $\left\lfloor \frac{2\pi}{\phi} \right\rfloor$ *where ϕ is defined by*

$$\frac{\sin(\theta - \phi)}{\sin(\theta)} = s. \quad (1)$$

PROOF. From the definition of Displaced Apex Adaptive Yao(G, α, s), the smallest angle between any two edges is θ because any nearest neighbor selected to form an edge will be outside, or on the boundary of, the cone for any other neighbor. Consider a node u. Let z be the nearest neighbor that defines a cone. For $0 \le s < 0.5$, the smallest angle between z and another nearest neighbor w is if w is placed at the intersection c of the cone boundary and the circle of radius uz centered on u (the point labeled as c in Fig. 2). By Corollary 1, the angle between c and z about u is defined by Eq. 1. If $\alpha = 1$ such that $\theta = \theta_m(s, |uz|)$, $\phi = \pi/3$. Similarly, for $0.5 \le s \le 1$, the smallest angle between z and another nearest neighbor w is if w is placed at the intersection e of the cone boundary and the circle of radius uz centered on u (the point labeled as e in Fig. 2). It is straightforward to show, using a proof similar to that for Lemma 1 and noting that $|ue| = |uz|$, that the angle ϕ between e and z about u is also defined by Eq. 1. Therefore, the angle between any two selected edges will be greater than or equal to ϕ, which is a function of s. So, the maximum out-degree for any node will be $\frac{2\pi}{\phi}$. Any fraction of a cone overlapping in the worst case will not add to the out-degree of the node, so the maximum integer out-degree of any node is be $\left\lfloor \frac{2\pi}{\phi} \right\rfloor$. $\square$

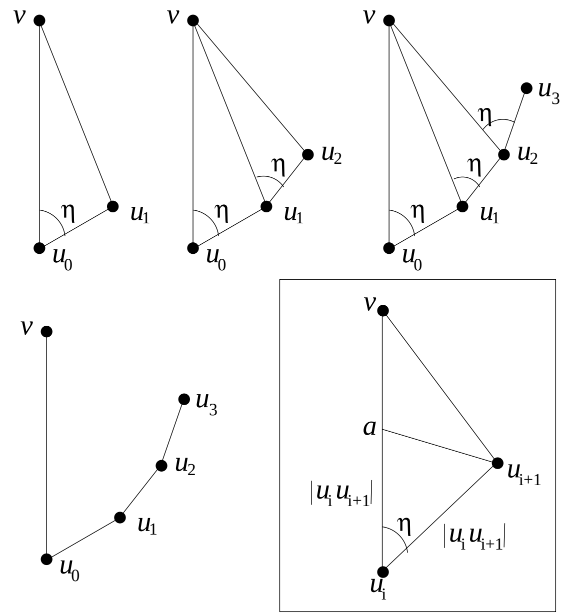


Figure 3: A scenario for the worst shortest path that could be selected in Displaced Apex Adaptive Yao(G, α, s). The edge (u_0, v) is not selected by the Displaced Apex Adaptive Yao(G, α, s) algorithm since (u_0, u_1) is shorter than (u_0, v), and (u_0, v) is inside the cone of (u_0, u_1). The same occurs for $(u_1, v), \ldots, (u_m, v)$. Box: One step of the iterative sequence of the path.

For example, if $s = 0$ and $\theta = \pi/3$, then $\phi = \theta$ and the maximum out-degree of any node is 6.

THEOREM 3. *Let $S \subseteq \mathbb{R}^2$ be a set of N points and let $\theta < \pi/3$ be the cone angle. Then DAAY$(UDG(S), \alpha, s)$ is a spanner with stretch factor* $\frac{1}{1 - 2\sin(\frac{\theta}{2})}$.

PROOF. Let uv be an edge in UDG that is not selected by Displaced Apex Adaptive Yao(G, α, s) algorithm. Since, by Theorem 1, DAAY$(UDG(S), \alpha, s)$ is connected, then there is a shortest path from u to v. Let a "worst" such path from u to v in DAAY$(UDG(S), \alpha, s)$ be $u_0 = u, u_1, u_2, \ldots, u_m = v$. See Fig. 3. By the Displaced Apex Adaptive Yao(G, α, s) algorithm, the angle $\angle u_{i+1} u_i v \le \eta$ (which we will determine) and $|u_i u_{i+1}| < |u_i v|$ since otherwise $u_i v$ would be part of DAAY$(UDG(S), \alpha, s)$ and part of the path. Also, by Theorem 1, $|u_{i+1} v| < |u_i v|$ since we can always decrease the distance to v from each u_i along the path.

Now consider the triangle $\triangle u_i u_{i+1} v$. See Fig. 3. Let a be the point on $u_i v$ such that $|u_i a| = |u_i u_{i+1}|$. By the triangular inequality $|u_{i+1}v| \leq |u_{i+1}a| + |av|$. Note that $|u_{i+1}a| = (2\sin\frac{\eta}{2})|u_iu_{i+1}|$, and $|av| = |u_iv| - |u_iu_{i+1}|$. Applying these two latter equations to the triangular inequality, we obtain

$$|u_{i+1}v| \leq |u_iv| - |u_iu_{i+1}|(1 - 2\sin\frac{\eta}{2}).$$

Applying the previous analysis iteratively on the entire path, we have
$\sum_{0\leq i<m} |u_{i+1}v| \leq \sum_{0\leq i<m} \left(|u_iv| - \left(|u_iu_{i+1}|(1 - 2\sin\frac{\eta}{2})\right)\right)$.
Therefore,

$$\sum_{0\leq i<m} (|u_iu_{i+1}|) \leq \left(\frac{1}{1 - 2\sin\frac{\eta}{2}}\right) \sum_{0\leq i<m} (|u_iv| - |u_{i+1}v|)$$

$$\leq \left(\frac{1}{1 - 2\sin\frac{\eta}{2}}\right) |u_0v|.$$

Note that for the stretch factor to be bounded by this inequality, then η must be restricted by $\eta < \pi/3$.

To determine the value of η, first consider the largest angle possible between u_i, v, and u_{i+1}. The larger the angle, the larger the stretch factor along the path. If u_{i+1} is a nearest neighbor of u_i defining a cone during the execution of the Displaced Apex Adaptive Yao(G, α, s) algorithm, then placing a node f at the intersection of the cone boundary and the boundary of the circle of radius r centered at u_i would give the largest angle (e.g., at position d in Fig. 2). Defining a triangle $\triangle u_ifu_{i+1}$ and using a similar analysis as used in the proof for Lemma 1, the internal angle η at u_i is

$$\frac{\sin(\theta - \eta)}{\sin(\theta)} = \frac{s|u_iu_{i+1}|}{|u_if|}.$$

Note that as $|u_iu_{i+1}| \rightarrow 0$, the angle η is maximized. For $0 \leq s < 0.5$, this maximum angle is θ, and for $0.5 \leq s \leq 1$, this maximum angle is $\pi/2$. To ensure that $\eta < \pi/3$, in both cases, we must restrict $\theta < \pi/3$ to bound the stretch factor. □

THEOREM 4. *Consider a node set S and $UDG(S)$ defined on S. Assume that $UDG(S)$ is connected. Then $DAAY(UDG(S), \alpha, s)$, $\theta = \alpha \cdot \theta_m(s, |uz|)$, $0 \leq \alpha \leq 1$, contains the Euclidean Minimum Spanning Tree $EMST(UDG(S))$ as a subgraph.*

PROOF. This proof is closely modeled on a similar proof for the $HSP(UDG(S))$ (see [6]) and is included for completeness. Let $EMST(UDG(S))$ be an Euclidean Minimum Spanning Tree of $UDG(S)$ that contains the maximum number of edges of $DAAY(UDG(S), \alpha, s)$. Consider a proof by contradiction. Assume there is an edge uv in $EMST(UDG(S))$ that is not in $DAAY(UDG(S), \alpha, s)$. This implies that there is an edge $uz \in DAAY(UDG(S), \alpha, s)$ such that $|uz| \leq |uv|$, because the edge uv should be in the cone of another shorter edge selected by the Displaced Apex Adaptive Yao$(UDG(S), \alpha, s)$ algorithm, and $|vz| < |uv|$ (otherwise, by Theorem 1, the $D\text{-}DAAY(UDG(S), \alpha, s)$ would not be strongly connected). Since $EMST(UDG(S))$ is a spanning tree, there is a path from v (or u) to z. If the path is from v to z, then removing uv from the graph and adding the edge uz we obtain a spanning tree with equal or less weight with an additional edge from $DAAY(UDG(S), \alpha, s)$, a contradiction. If the path is from u to z, then removing uv from the graph and adding the edge vz we obtain a spanning tree with less weight, again a contradiction. □

3. SIMULATION RESULTS

In our experiments we used randomly chosen connected unit disk graphs on an area of 100×100. We varied the number of nodes, N, between 65, 75, 85, 95, and 105 nodes. For all the results reported here (unless indicated), the results have been averaged over 23 graphs for each value of N. For all graphs, the transmission radius r used was 15 units.

In Fig. 4 we study the dependence of the Displaced Apex Adaptive Yao subgraphs on s and α. For all the plots, it is obvious that there is a stronger dependence on α than s. In addition, for the maximum node degree, average node degree, average number of edge crossings, and average weight, there is appears to be symmetry about the $s = 0.5$ value, which appears more pronounced for larger α values. For example, minimum for the average number of crossing edges occurs at $\alpha = 1$ and $s = 0.5$ with a total of two crossing edges over the 23 subgraphs. As is apparent from the isocontours, for these latter four graph properties, about the minimum at $s = 0.5$, for $\alpha = 1$, the values are larger as s goes to 0 as compared to the values as s goes to 1.

The dilation for a pair nodes u, v, $u \neq v$, is the ratio of the shortest length path between u and v in the subgraph over that for the original UDG. The path length is computed in terms of the number of hops along the path or the sum of the Euclidean lengths of the edges of the path. The maximum is taken over all distinct pairs u, v in the graph. The trends for the node degree, weight and number of edge crossings are reflected in the dilation plots in Fig. 4 except that for large α values, the dilation values are maximum at $s = 0.5$. In addition, for $\alpha = 1$, while the $s = 0$ Euclidean dilation values are larger than the $s = 1$ values (1.76 compared to 1.69), the situation is reversed for the hop number dilation (3.30 for $s = 0$ compared to 3.61 for $s = 1$). One primary observation that can be made from Fig. 4 is that, despite Theorem 3 stating that the stretch factor may be unbounded for $\theta \geq \pi/3$ (which occurs at $\alpha = 1$ for $s = 0$ and at $\alpha = 2/3$ for $s = 0.5$, for example), there appears to be no abrupt change in the dilation values as this α threshold is crossed.

For the rest of our simulations, for each UDG, an Adaptive Yao subgraph (equivalent to a Displaced Apex Adaptive Yao subgraph with $s = 0$), Displaced Apex Adaptive Yao subgraphs with $s = 0.125$ and $s = 0.25$, Half Space Proximal subgraph (equivalent to a Displaced Apex Adaptive Yao subgraph with $s = 0.5$), and Displaced Apex Adaptive Yao subgraphs with $s = 0.75$ and $s = 1.0$ are generated. For each Displaced Apex Adaptive Yao subgraph we used $\alpha = 1$ such that $\theta = \theta_m(s, |uz|)$ (recall, for $s > 0.5$, θ is a function of the distance to the chosen neighbor defining the cone). For comparison, we also generate the original Yao subgraph with $k = 6$ for each UDG.

The following tests were done on the undirected version of each of the graphs mentioned above: 1) Average Degree (Fig. 8); 2) Maximum path dilation, in terms of both hop number and Euclidean distance (Fig. 5); 3) Average path dilation, in terms of both hop number and Euclidean distance (Fig. 5); 4) Weight of each graph (Fig. 6); 5) Number of crossing edges of each graph (Fig. 6); and 6) Degree distribution of each graph (Fig. 7). Also, in Fig. 8, for the directed versions of the Yao subgraph with $k = 6$, the Adaptive Yao subgraph, Displaced Apex Adaptive Yao subgraphs with $s = 0.125$ and $s = 0.25$, Half Space Proximal subgraph, and Displaced Apex Adaptive Yao subgraphs with $s = 0.75$ and $s = 1.0$, we also measured 1) Maximum in-degree; 2)

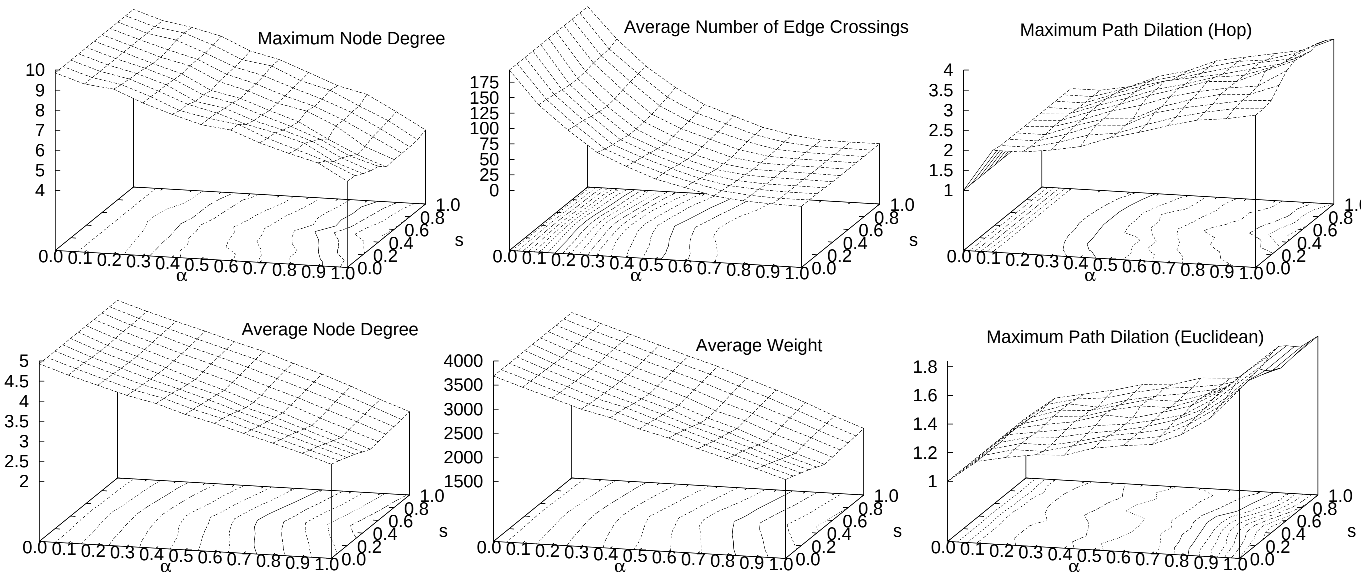


Figure 4: Plots of various Displaced Apex Adaptive Yao subgraph properties as a function of s and α. The values are averaged over 23 graphs with $N = 75$ nodes. The contours projected on the s-α plane are isocontours of evenly spaced values on the surfaces.

Average in-degree; 3) Maximum out-degree; and 4) Average out-degree.

As s approaches 0.5, from Fig. 8, the average and maximum node degrees monotonically decrease until $s = 0.5$ when we have the HSP graph. Then as s continues to increase to 1, the node degrees begin to increase again. This holds true across all values of N. We can see this trend reflected in the histograms of the node degrees in Fig. 7. Also, as we can see in Fig. 6, although the weights of the graphs follow the same trend as the node degrees, the number of crossing edges as s goes from 0.5 to 1 increases only slowly.

In terms of the dilations of the graphs, the Adaptive Yao graph ($s = 0$) has consistently the lowest maximum and average (hop number or Euclidean length) dilations. For our simulations, the dilation for the Adaptive Yao graph was about halfway between that of the HSP and the Yao graph with $k = 6$. As s increases to 0.5, the dilations increase to a maximum for $s = 0.5$. Then, mirroring the trends for node degree and the weights of graphs, the dilation again decreases as s approaches 1. In particular, for the average and maximum Euclidean dilations, the drop is more significant.

The reason for this behavior (also seen in the edge crossings plot in Fig. 4) is that for $s > 0.5$ we obtain a graph that maintains many of the properties of the HSP graph but with additional short edges. These edges are added since the inverted cone leaves a couple of gaps on either side of the directed edge to the chosen nearest neighbor (e.g., within the region defined by *cde* in Fig. 2) where additional close neighboring nodes may be selected. Although these additional edges are added and the node degrees of the graphs go up, the weights and the number of crossing edges increase more slowly.

4. CONCLUSION

In this paper, we present a class of Yao-type graphs that combine the advantages of both the HSP subgraph and the Yao subgraph by permitting control over the degree of the subgraph while being orientation-invariant. Indeed, the degree control is continuous since any cone angle less than $\theta_m(s, |uz|)$ can be used as differs from the Yao graph where only a discrete set of cone angles (π/k) are possible. Since the experimental path dilations appear to be bounded despite using $\alpha = 1$, it would appear that the actual stretch factor upper bounds are better behaved than indicated by Theorem 3 when $\theta \geq \pi/3$. In addition, unlike the Yao subgraph, the class of DAAY subgraphs easily extend to three dimensional UDGs. The class of graphs presented also preserve the common properties of the Yao and HSP graphs such as bounded out-degree, having the EMST as a subgraph, and being spanner graphs with bounded stretch factor.

5. ACKNOWLEDGMENTS

We are grateful to Evangelos Kranakis for suggesting the generalization of the Adaptive Yao graph given in Section 2.

6. REFERENCES

[1] L. Barrière, P. Fraigniaud, L. Narayanan, and J. Opatrny. Robust position-based routing in wireless ad hoc networks with irregular transmission ranges. *Wireless Communications and Mobile Computing Journal*, 3(2):141–153, 2003.

[2] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proc. of 4th ACM/IEEE*

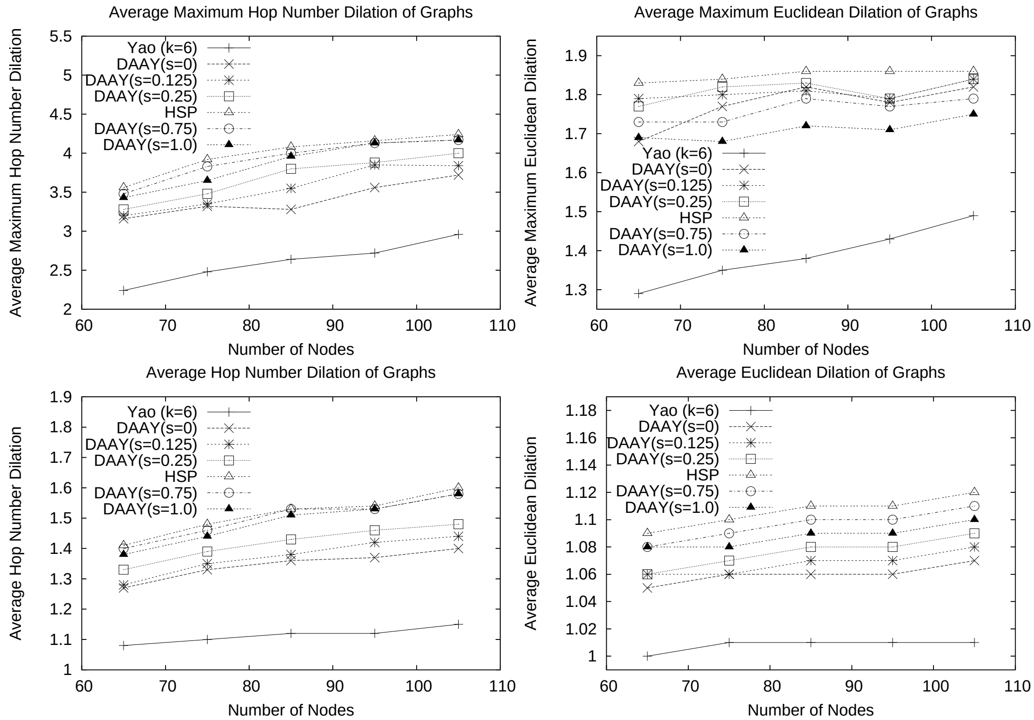


Figure 5: Top-Left: Average maximum hop number dilation for each graph with various number of nodes. Bottom-Left: Average hop number dilation for each graph with various number of nodes. Top-Right: Average maximum Euclidean dilation for each graph with various number of nodes. Bottom-Right: Average Euclidean dilation for each graph with various number of nodes. For the Displaced Apex Adaptive Yao subgraphs, α is set to 1.

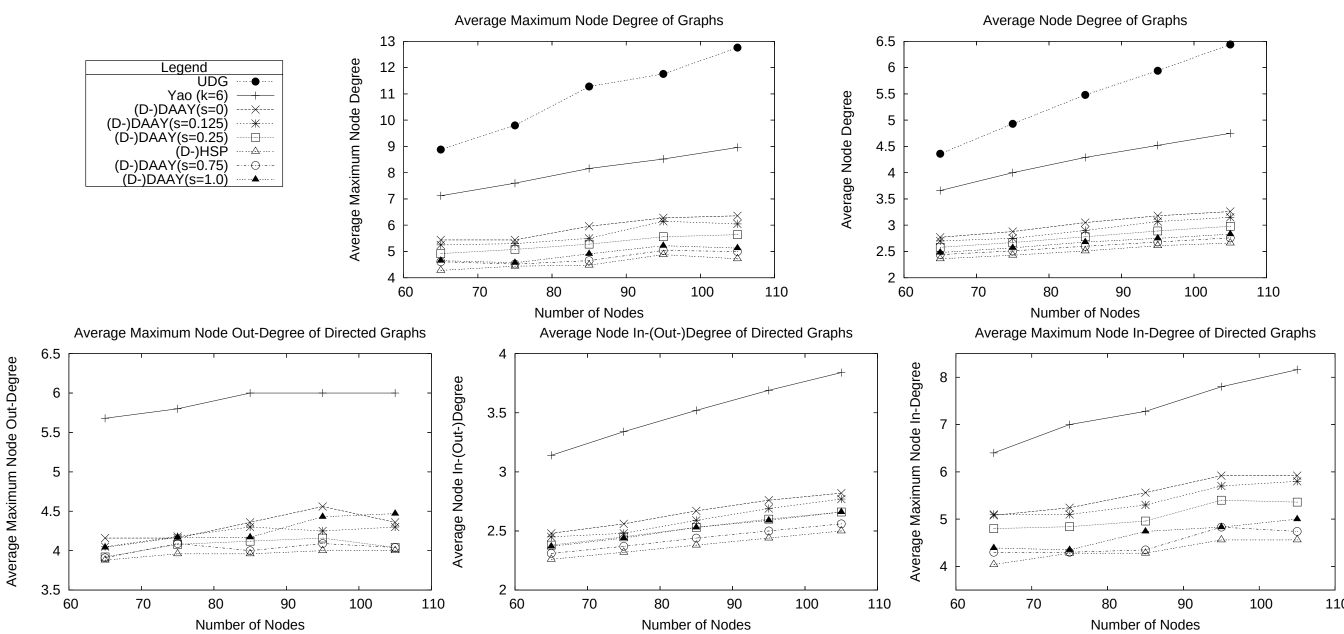


Figure 8: Same legend used for all plots. Top Left: Average maximum node degrees for each graph with various numbers of nodes. Top Right: Average node degrees. Bottom Left: Average maximum node in-degrees. Bottom Center: Average node in-degrees (out-degrees). Bottom Left: Average maximum node out-degrees. All the maximum and average degrees are averaged over 23 graphs for each N. For the Displaced Apex Adaptive Yao subgraphs, α is set to 1.

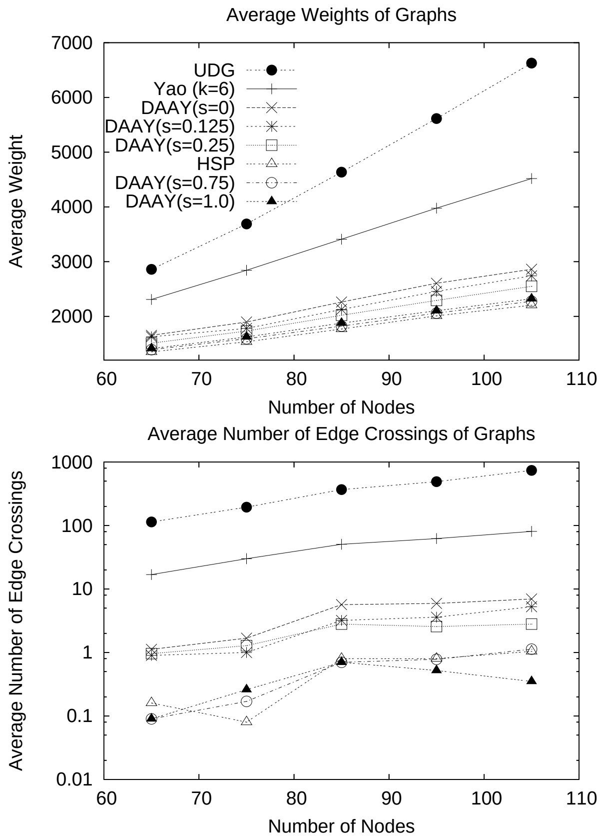


Figure 6: Same legend used for both plots. Top: Weights of graphs. Bottom: Number of crossing edges of graphs (log scale for number of crossing edges). All values are averaged over 23 graphs for each N. For the Displaced Apex Adaptive Yao subgraphs, α is set to 1.

Conference on Mobile Computing and Networking (Mobicom '98), pages 76–84, 1998.

[3] P. Boone, E. Chavez, L. Gleitzky, E. Kranakis, J. Opatrny, G. Salazar, and J. Urrutia. Morelia test: Improving the efficiency of the gabriel test and face routing in ad-hoc networks. In *Proc. of SIROCCO 2004, volume 3104 of LNCS*, pages 23–34, 2004.

[4] P. Bose, P. Carmi, M. Couture, M. Smid, and D. Xu. On a family of strong geometric spanners that admit local routing strategies. Technical Report TR-07-08, SCS, Carleton University, Ottawa, Canada, 2007.

[5] P. Bose, J. Gudmundsson, and P. Morin. Ordered theta graphs. *Computational Geometry: Theory and Applications*, 28(1):11–18, 2004.

[6] E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, H. Tejeda, and J. Urrutia. Half-space proximal: A new local test for extracting a bounded dilation spanner. In *Proc. of the International Conference On Principles of Distributed Systems (OPODIS 2005), volume 3974 of LNCS*, pages 235–245, Pisa, Italy, 2006.

[7] K. Gabriel and R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*,

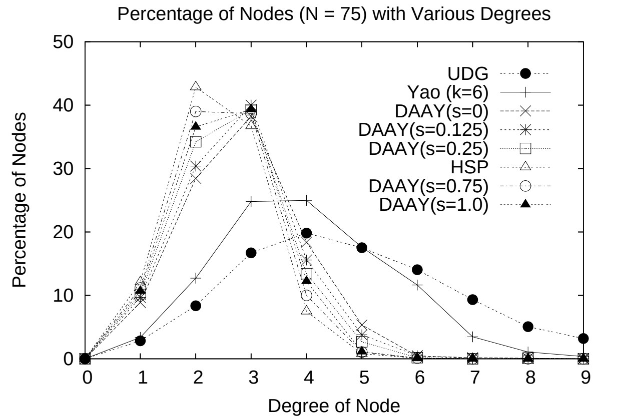


Figure 7: Histogram of degrees of nodes for each graph with 75 nodes. The percentages of each degree is averaged over 25 graphs. For the Displaced Apex Adaptive Yao subgraphs, α is set to 1.

18:259–278, 1969.

[8] I. Kanj and L. Perkovic. Improved stretch factor for bounded-degree planar power spanners of wireless ad-hoc networks. In S. Nikoletseas and J. D. P. Rolim, editors, *ALGOSENSORS*, volume 4240 of *LNCS*, pages 95–106. Springer, 2006.

[9] X. Li. *Topology Control in Wireless Ad Hoc Networks*. IEEE Press, 2003.

[10] X.-Y. Li, G. Călinescu, and P.-J. Wan. Distributed construction of planar spanner and routing for ad hoc wireless networks. In *Proc. of INFOCOM'2002*, 2002.

[11] X.-Y. Li, P.-J. Wan, and Y. Wang. Power efficient and sparse spanner for wireless ad hoc networks. In *Proc. of IEEE Int. Conf. on Computer Communications and Networks (ICCCN01)*, pages 564–567, 2002.

[12] X.-Y. Li, Y. Wang, P.-J. Wan, W.-Z. Song, and O. Frieder. Localized low-weight graph and its applications in wireless ad hoc networks. In *Proc. of Twenty-Third IEEE INFOCOM 2004*, Hong Kong, China, March 2004.

[13] M. Mauve, J. Widmer, and H. Hartenstein. A survey of position-based routing in mobile ad-hoc networks. *IEEE Network Magazine*, 15(6):30–39, 2001.

[14] R. Rajaraman. Topology control and routing in ad hoc networks: A survey. *SICACT News*, 33:60–73, 2002.

[15] W.-Z. Song, Y. Wang, X.-Y. Li, and O. Frieder. Localized algorithms for energy efficient topology in wireless ad hoc networks. *MONET*, 10(6):911–923, 2005.

[16] G. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.

[17] Y. Wang, X.-Y. Li, and O. Frieder. Distributed spannar with bounded degree for wireless ad hoc networks. *International Journal on Foundations of Computer Science*, 14(2):183–200, 2003.

[18] A.-C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Computing*, 11:721–736, 1982.