Yao Spanners for Wireless Ad Hoc Networks

A Thesis Presented to the Faculty of the
Department of Computing Sciences
Villanova University

In Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in
Computer Science

by
Nawar M. El Molla
December, 2009

Under the Direction of
Mirela Damian, PhD
Contents

1 Introduction .................................................. 2
  1.1 Our Contribution ........................................... 4

2 Background .................................................. 4
  2.1 The Relative Neighborhood Graph [5] .................... 4
  2.2 The Gabriel Graph [2] ...................................... 5
  2.3 The Voronoi Diagram [4] .................................... 6
  2.4 The Delaunay Triangulation [Boris Delaunay, 1934] .... 7
  2.5 The Yao Graph [10] ......................................... 8
  2.6 The Yao-Yao Graph [7] ...................................... 10

3 Related Work ................................................. 10

4 New Results on Yao-Yao Graphs ............... 12
  4.1 YY$_4$ is not a spanner ................................... 12
  4.2 YY$_6$ is not a spanner ................................... 14

5 New Results on Yao Graphs ....................... 16
  5.1 Y$_2$ is not a Spanner .................................... 17
  5.2 Y$_3$ is not a Spanner .................................... 18
  5.3 Y$_4$ is a Spanner for Convex Position ................. 19

6 Conclusion and Future Work ..................... 28

References ...................................................... 28
List of Tables

1  The Y4PATH algorithm. .......................... 23
List of Figures

1. The Relative Neighborhood Graph ........................................ 5
2. The Gabriel Graph .......................................................... 6
3. The Voronoi Diagram ....................................................... 7
4. Delaunay Triangulation .................................................... 8
5. The Relative Neighborhood Graph, Gabriel Graph, and Delaunay Triangulation applied on the same set of points ....................... 9
6. The Yao Graph ............................................................... 9
7. The Yao Graph ............................................................... 10
8. The Yao-Yao Graph ......................................................... 11
9. The Yao graph \( H \) induced by the point set \( V \) ...................... 12
10. The graph \( H \) is not a spanner: ........................................ 13
11. The graph steps of showing that \( YY_6 \) is not a spanner ............... 15
12. The \( Y_2 \) graph induced by the point set \( V \) ........................... 17
13. The \( Y_3 \) graph induced by the point set \( V \) ........................... 19
14. Constructing \( \mathcal{P}(u \rightsquigarrow v) \). (a) The outer square is \( SQ(u_i,v) = SQ(u_{i+1},v) \), and the inner square is \( SQ(u_{i+2},v) \); shaded area is \( Q(u_i,v) \); (b) \( \mathcal{P}(u \rightsquigarrow v) \) lies inside \( SQ(u,v) \) ................................................................. 20
15. Edges \( u_iu_j, u_ku_\ell \in \mathcal{P}(u \rightsquigarrow v) \) can not cross. ................. 22
16. \( \mathcal{P}(u \rightsquigarrow v) \) is arbitrarily long compared to \( |uv| \). ............. 23
17. \( \mathcal{P}(u_5 \rightsquigarrow u_2) \) lies in the shaded area. ....................... 24
18. If \( u_{i-1} \in Q_0 \), then \( u_{i+1} \notin Q(u_i,v) \). ................................. 25
19. The perimeter of \( T \) does not exceed \( (2 + \sqrt{2})\ell \). .................. 27
Abstract

Wireless nodes are often powered by batteries and have limited memory resources. These characteristics make it critical to compute and maintain, at each node, only a subset of neighbors that the node communicates with. These subsets of neighbors define a topology and the problem of choosing “appropriate” subsets of neighbors is called the topology control problem.

Highly relevant to the topology control problem is the class of Yao graphs, defined as follows. Let $P$ be a set of points in the plane. The Yao graph $Y_k(P)$, for $k > 2$, is defined as follows. At each point $u \in P$, any $k$ equal-separated rays originated at $u$ define $k$ cones; in each non-empty cone $C_u$, pick the shortest edge $\{u, v\}$ connecting $u$ to its nearest neighbor $v \in C_u$ (breaking ties arbitrarily), and add $\{u, v\}$ to $Y_k$.

For a fixed value $t \geq 1$, a graph $G$ embedded in the plane is a length $t$-spanner if, for all pairs of vertices $u, v \in P$, the shortest path in $G$ from $u$ to $v$ is no longer than $t \cdot |uv|$; here $|uv|$ denotes the Euclidean distance between $u$ and $v$. It is a standing open question to decide whether the Yao structure $Y_k$, for $k \leq 5$, is a length $t$-spanner or not, for some constant $t$. We make progress towards resolving this question by showing that $Y_4$ is a length spanner for sets of points in convex position. We prove that $Y_2$ and $Y_3$ are not length spanners. We show that a related structure, called Yao-Yao, is a length spanner for any set of points of bounded aspect ratio (defined as the ratio between the largest to the smallest interpoint distances).
1 Introduction

Wireless networks are one kind of computer networks that do not require the usage of physical connection (or wires). These networks share data through transmitting and receiving electromagnetic waves, mainly radio waves. There are many types of wireless networks that differ in the range of coverage and the purpose of use. These types are Wireless Personal Area Network (WPAN), Wireless Local Area Network (WLAN), Wireless Metropolitan area networks (WMAN), and mobile devices networks. The wireless networks technology has had a great impact on the world since the 1940s when it served as a way of communication in the Second World War. Since then, the uses of wireless networks have been vastly growing to include businesses, emergency services, and more. One substantial example of current businesses is the cellular phones industry which is deploying wireless network technology to serve billions of mobile devices holders worldwide.

Wireless ad hoc network is a collection of autonomous devices that can communicate with each other without the aid of any fixed infrastructure. This type of networks follow a decentralized structure of communication which means that each node is willing to forward data for other nodes within its transmission range. Pairs of wireless nodes not within transmission range of each other communicate through multi-hop wireless links by using intermediate nodes to relay their message. As a result, determination of which nodes forward data is made dynamically based on the network connectivity. To discover a path when a communication to another node is initiated, a node floods the network. Then with route discovery queries, it caches the path for later use. Nevertheless, there are two major problems to be addressed in this setting that do not arise in the wired case. One major problem is mobility, which renders cached routing paths invalid due to
the transient nature of the network topology. A second problem is interference between multiple users using a common communication medium. This problem is worsened by the limited battery and memory resources of each node. One way to avoid the expense of full flooding is for each node to communicate only with a selected subset of neighboring nodes. These subsets of neighbors define a topology and the problem of choosing "appropriate" subsets of neighbors is called the *Topology Control Problem*.

Topology control problem is one of the problems in computational geometry and graph theory. Wireless ad hoc networks are modeled as graphs in which nodes represent wireless devices and the connections between these nodes represent edges. We call this graph the network graph\(^1\). If the network is homogenous, we can normalize the transmission of the nodes to 1 and view the network graph as a *Unit Disk Graph*. The *Unit Disk Graph* is an Euclidean graph \(G = (V, E)\) where any two nodes are adjacent if and only if their Euclidean distance is at most 1. That is, for arbitrary \(u, v \in V\), it holds that \(u, v \in E \iff |u, v| \leq 1\). Communication over network graph edges is usually inefficient due to the existence of too many edges that induce large amount of interference. As discussed before flooding is also inefficient. A common solution to this problem is to extract a subgraph \(H = (V, E')\) of the network graph that is connected but is less prone to interference. However, as edges get eliminated, paths between remote nodes get larger. This renders necessary an additional restriction on \(H\) to contain short spanning paths. This property is known as the *Spanner Property*.

A graph is considered to be a geometric spanner or a k-spanner graph if every pair of points in the graph have a path between them no longer than \(t\) times the Euclidean distance separating these points for a fixed \(t\). The parameter \(t\) is called the *stretch factor* or *dilation factor* of the spanner. In other words \(t\) is defined as the largest ratio between

\(^1\)Throughout this thesis, we refer to nodes of \(G\) as points or vertices as well.
the length of a path and the direct Euclidean distance between its endpoints. Intuitively, a graph is a spanner if shortest paths in the graph closely approximate direct Euclidean distances.

The rest of the paper is organized as follows. Section two surveys five major graphs referencing their construction and their properties. Section three discusses the earlier work done on the previously described graphs. Section four presents the new found results on Yao Graph. Section five states new results on YaoYao Graph. Section six summaries the work done in this thesis and gives some future direction for research.

1.1 Our Contribution

In this thesis, we prove that, for any set of points $V$, $Y_4$, $Y_6$, $Y_2$, and $Y_3$ not to have the spanner property by means of three counterexamples. We, also, prove that for any set $V$ of points in convex position, the Yao graph $Y_4$ on $V$ is a $t$-spanner for $t = 4(2 + \sqrt{2})$. In addition, we conjecture that $Y_4$ is a spanner for arbitrary point sets.

2 Background

There are many types of spanners used as skeletons for routing topologies. In the following section, We give a summary on five major graphs; the Relative Neighborhood Graph, the Delaunay Triangulation, the Gabriel Graph, the Yao Graph, and the YaoYao Graph.

2.1 The Relative Neighborhood Graph [5]

The relative neighborhood graph $G = (V, E)$, denoted by RNG(G), comprises the set of all edges $uv \in E$ such that the lune formed by $D(u, |uv|) \cap D(v, |uv|)$ contains no other
nodes (We use $D(u, |uv|)$ to denote the disks of center $u$ and radius $uv$). In other words, there is no node $u \in V$ with $|uw| < |uv|$ and $|wv| < |uv|$ (see Figure 1). The relative neighborhood graph is known to be planar and to have a stretch factor of $O(n)$ [8]. It is also known to have a maximum node degree of $n - 1$ [7].

![Figure 1: The Relative Neighborhood Graph](image)

### 2.2 The Gabriel Graph [2]

The Gabriel Graph of a graph $G = (V, E)$ comprises the set of all edges $uv \in E$ such that there is no vertex or point $w$ that lies in the disk of diameter $uv$ (see Figure 2). In the context of wireless networks, Gabriel Graphs have two shortcomings. First, they might have a high degree as high as $n - 1$ [7]. Second is the large stretch factor, as large as $O(\sqrt{n})$ [8]. In 1980, Urquhart gave an $O(n \log n)$ time algorithm for constructing the Gabriel Graph by first finding the Delaunay Triangulation and Voronoi Diagram for the set of points. Then for each edge in the triangulation, if the edge intersects its Voronoi edge, it is added as an edge to the GG.
2.3 The Voronoi Diagram [4]

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of points in the two dimensional Euclidean plane. These are called sites. Partition the plane by assigning every point in the plane to its nearest site. All those points assigned to $p_i$ form the Voronoi region, denoted $V(p_i)$. $V(p_i)$ consists of all the points at least as close to $p_i$ as to any other site (see Figure 3):

$$V(p_i) = \{x : |p_i - x| \leq |p_j - x| \forall j \neq i\}$$

Voronoi diagrams have a surprising variety of uses [1]:

- Nearest neighbor search: For point $p_i$, finding its nearest neighbor from a fixed set of points $P$ is simply a matter of determining which cell in the Voronoi diagram of $P$ contains $p_i$.

- Facility location: Suppose that you need to locate a new restaurant in an area with several existing, competing restaurants. To minimize interference with existing restaurants, it should be located as far away from the closest restaurant as possible. This location is always at a vertex of the Voronoi diagram, and it can be found in a linear-time search through all the Voronoi vertices.
• Largest empty circle: Suppose you needed to obtain a large, contiguous, undeveloped piece of land on which to build a factory. The same condition used for picking restaurant’s locations is appropriate for other undesirable facilities, namely that it be as far as possible from any relevant sites of interest. A Voronoi vertex defines the center of the largest empty circle among the points.

• Path planning: If the sites of $P$ are the centers of obstacles we seek to avoid, the edges of the Voronoi diagram define the possible channels that maximize the distance to the obstacles. Thus in planning paths among the sites, it will be “safest” to stick to the edges of the Voronoi diagram.

2.4 The Delaunay Triangulation [Boris Delaunay, 1934]

A triangulation of a graph $G = (V, E)$ is a Delaunay triangulation, denoted by $\text{Del}(G)$, if the circumcircle of each of its triangles does not contain any other nodes of $G$ in its interior. A triangle is called the Delaunay triangle if its circumcircle is empty of nodes of $G$ (see Figure 4). If $G$ is a unit disk graph, the Delaunay triangulation for $G$, denoted
UDel(G), is obtained from Del(G) by deleting all edges that are longer than the unit (see Figure 5). Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation; they tend to avoid skinny triangles.

![Delaunay Triangulation](image)

Figure 4: Delaunay Triangulation

Figure 5 illustrates the relationship between the three graphs; RNG, GG, and DT. RNG is a subgraph of GG, which in turn is a subgraph of DT. In addition, the minimum spanning tree (MST), introduced by Kruskal [1956] and Prim [1957], is a subgraph of all three.

2.5 The Yao Graph [10]

For a set of points P in the plane representing wireless nodes, The Yao graph $Y_k(P)$, for $k \geq 2$, is defined as follows. At each point $u \in P$, any $k$ equal-separated rays originated at $u$ define $k$ cones; the Yao graph $Y_k$ contains edges connecting $u$ to a nearest neighbor in each nonempty cone, for a total of at most $k$ edges incident to $u$ (see Figure 6).

Yao graph has been proved to be a spanner for $k > 6$ with a stretch factor $\frac{1}{1-2\sin(\frac{\pi}{k})}$ [7]. One major benefit of the Yao graph is that it can be computed fast; all nodes pick
Figure 5: The Relative Neighborhood Graph, Gabriel Graph, and Delaunay Triangulation applied on the same set of points.

Figure 6: The Yao Graph.
their incident edges in parallel. The out-degree of the Yao graph is less than or equal to $k$ but the in-degree could be as high as $O(n)$. In a wireless network, if all neighbors try to send to a node, the result is high interference at the receiving node (see Figure 7).

![Figure 7: The Yao Graph](image)

2.6 The Yao-Yao Graph [7]

To reduce interference introduced by the Yao structure, Li and others added a refining step to the Yao graph. Amongst all incoming edges to a node, they eliminate all but the shortest incoming edge from each cone (see Figure 8). This step reduced the in-degree of each node to at most $k$, yielding a maximum degree of $2k$ for each node (incoming + outgoing). The new structure is called the Yao-Yao graph, denoted $YY_k$.

3 Related Work

For efficient communication, it is important for a graph to be a spanner. Accordingly, efforts were directed towards proving the spanner property of the previously mentioned
graphs. In 2002 both the Gabriel graph and the Relative Neighborhood Graph were proved to be spanners with $O(n)$ and $O(\sqrt{n})$ stretch factors respectively \[9\]. In the same year, Li and others proved that Yao graph is a spanner only for number of sectors greater than 6. They also calculated the stretch factor to be $\frac{1}{1-2\sin(\frac{\pi}{k})}$ or simply $O(1)$. The Delaunay Triangulation, being the super graph of both the Gabriel Graph and the Relative Neighborhood Graph, was later proved to be a $2.42$-spanner \[3\]. As new graphs like the Yao and the Yao-Yao were introduced, new open questions were put on the table asking whether the newly introduced graphs are suited for wireless network topology. And if they were, how efficient would the communication based on such a topology be. In the following chapters, we prove the spanner property of the Yao and the Yao-Yao graphs for some values of $k$. We also compute the stretch factors bounding these spanners if they exist.
4 New Results on Yao-Yao Graphs

Most of the results in the literature concern the classes of graphs $Y_k$ and $YY_k$, for $k > 6$. Motivated by recent advances in directional antennas, which are capable of focusing their energy in a small sector-shaped beam, we investigate the classes of Yao-based graphs for $k \leq 6$. Given that the number of antennas per node is usually small (3 to 6), it is likely that our results will prove useful to applied researchers in the wireless networking field.

4.1 $YY_4$ is not a spanner

We prove that $YY_4$ is not a spanner with the help of a simple counterexample. For a fixed small angular value $\theta > 0$, let $a_0b_0$ be an arbitrary line segment with slope $\theta$. Let

Figure 9: The Yao graph $H$ induced by the point set $V$. 
Let \( a_0, a_1, \ldots, a_k \) be points equally distributed along a line of slope \( \pi/2 - \theta \), at intervals equal to \( |a_0b_0| \) (so \( |b_i b_{i+1}| = |a_0b_0| \), for each \( i = 1, \ldots, k - 1 \)). See Figure 9. Let \( a_0, a_1, \ldots, a_k \) be points equally distributed along a line of slope \( \pi/2 + \theta \), such that \( a_i b_i \) and \( a_j b_j \) are parallel, for each \( 0 \leq i, j \leq k \).

Note that, for fixed \( \theta, a_0 \) and \( b_0 \), the point set \( V = \{a_0, a_1, \ldots, a_k\} \cup \{b_0, b_1, \ldots, b_k\} \), is uniquely defined. For a fixed \( k \), we select the angle \( \theta \) small enough so that two lines including \( a_0 a_1 \) and \( b_0 b_1 \) are almost vertical.

Now note that in the Yao step, each node \( a_i \) selects the edges \( \overrightarrow{a_i b_i}, \overrightarrow{a_i a_{i+1}} \) (if \( i < k \)) and \( \overrightarrow{a_i a_{i-1}} \) (if \( i > 0 \)). Similarly, each node \( b_i \) selects \( \overrightarrow{b_i b_{i+1}} \) and \( \overrightarrow{b_i a_{i+1}} \) (if \( i < k \)), \( \overrightarrow{b_i b_{i-1}} \) (if \( i > 0 \)), and \( \overrightarrow{b_i a_i} \) (if \( i = 0 \)). The result is the Yao graph \( H \) shown in Figure 9. In

![Figure 10: The graph \( H \) is not a spanner:](image-url)
the reverse Yao step, since edges $\overrightarrow{a_i a_{i+1}}$ and $\overrightarrow{b_i a_{i+1}}$ lie in the same quadrant for $a_{i+1}$ (for $i < k$), and since $|a_i a_{i+1}| < |b_i a_{i+1}|$, the node $a_{i+1}$ will eliminate $\overrightarrow{b_i a_{i+1}}$ from $H$. Similarly, since $\overrightarrow{a_i b_i}$ and $\overrightarrow{b_{i-1} b_i}$ lie in the same quadrant for $b_i$, and since $|b_{i-1} b_i| < |a_i b_i|$, the node $b_i$ will eliminate $\overrightarrow{a_i b_i}$ from $H$. The result is the YaoYao graph $H$ illustrated in Figure 10.

For arbitrarily small $\theta$, we have that $|a_k b_k| \approx |a_0 b_0|$. However, a shortest path between $a_k$ and $b_k$ in the YaoYao graph $H$ has length

$$\delta_H(a_k, b_k) = \sum_{i=1}^{k} |a_i a_{i-1}| + |a_0 b_0| + \sum_{i=0}^{k-1} |b_i b_{i+1}|$$

$$> n \cdot |a_0 b_0|$$

It follows that $H$ is not a spanner for the graph induced by the point set $V$.

4.2 \textit{YY} is not a spanner

\textbf{Lemma 1} \textit{YY} is not a Spanner.

\textbf{Proof:} We prove that \textit{YY} is not a spanner with the help of a counterexample. For a fixed small angular value $\theta > 0$, let $a_1 b_1$ be an arbitrary line segment with slope $\theta$. From $a_1$ and $b_1$, draw two parallel vertical lines. Let $b_1, \ldots , b_k$ be points equally distributed on the line passing through $b_1$ and $a_1, \ldots , a_k$ be points equally distributed along a line passing through $a_1$ such that $a_i b_i a_{i+1}$ is an equilateral triangle.

Next, rotate the $a$-line counter-clockwise by an angle $\alpha$ around $a_1$. Similarly, rotate the $b$-line clockwise by $\alpha$ around $b_1$ (see Figure 11(a)). As a result and by applying the law of cosines, we have $\frac{a_i a_{i+1}}{\cos \angle a_i b_i a_{i+1}} = \frac{b_i a_{i+1}}{\cos \angle b_i b_{i+1} a_{i+1}}$. Since $\cos \angle a_i b_i a_{i+1} = 60^\circ$ and $\cos \angle b_i b_{i+1} a_{i+1} > 60^\circ$ then $|a_i a_{i+1}| < |b_i a_{i+1}|$. 

14
Figure 11: The graph steps of showing that $YY_6$ is not a spanner
Next, we slightly slide the points along the two lines such that

- (a) the $60^\circ$ cones at each node are disjoint (see Figure 11(b)).
- (b) the inequality $|a_ia_{i+1}| < |b_ia_{i+1}|$ is preserved.

Now note that in the Yao step, each node $a_i$ selects the edges $\overrightarrow{a_ib_i}, \overrightarrow{a_ia_{i+1}}$ (if $i < k$) and $\overrightarrow{a_ia_{i-1}}$ (if $i > 0$). Similarly, each node $b_i$ selects $\overrightarrow{b_ib_{i+1}}$ and $\overrightarrow{b_ia_{i+1}}$ (if $i < k$), $\overrightarrow{b_ib_{i-1}}$ (if $i > 0$), and $\overrightarrow{b_ia_i}$ (if $i = 0$). The result is the Yao graph shown in Figure 11(c).

In the reverse Yao step, since edges $\overrightarrow{a_ia_{i+1}}$ and $\overrightarrow{b_ia_{i+1}}$ lie in the same sector for $a_{i+1}$ (for $i < k$), and since $|a_{i+1}a_{i+1}| < |b_ia_{i+1}|$, the node $a_{i+1}$ will eliminate $\overrightarrow{b_ia_{i+1}}$ from $H$. Similarly, since $\overrightarrow{a_ib_i}$ and $\overrightarrow{b_{i-1}b_i}$ lie in the same sector for $b_i$, and since $|b_{i-1}b_i| < |a_ib_i|$, the node $b_i$ will eliminate $\overrightarrow{a_ib_i}$ from $H$. The result is the YaoYao graph illustrated in Figure 11(d).

For arbitrarily small $\alpha$, we have that $|a_kb_k| \approx |a_1b_1|$. However, a shortest path between $a_k$ and $b_k$ in the YaoYao graph has length

$$
\delta_H(a_k, b_k) = \sum_{i=1}^{k} |a_ia_{i+1}| + |a_1b_1| + \sum_{i=0}^{k-1} |b_ib_{i+1}| > n \cdot |a_1b_1|
$$

It follows that the $YY_6$ is not a spanner for the graph in Figure 11.

5 New Results on Yao Graphs

We start with two negative results showing that $Y_2$ and $Y_3$ are not spanners, then proceed to proving that $Y_4$ is a spanner for points in convex position. We conjecture that $Y_4$ is a spanner for points in general position, and leave the problem open for further investigations.
5.1 $Y_2$ is not a Spanner

**Lemma 2** $Y_2$ is not a Spanner.

**Proof:** We prove that $Y_2$ is not a spanner with the help of a counterexample. Let $a_1b_1$ be a line segment perpendicular to the $y$-axis where $b_1$ is symmetric to $a_1$ with respect to the $y$-axis. From the origin, draw two lines; $a$-line passes through $a_1$ and $b$-line passes through $b_1$. Each line forms an angle $\alpha$ with the $y$-axis (see Figure 12(a)). Let $b_1, \ldots, b_k$ be points equally distributed on the $b$-line and $a_1, \ldots, a_k$ be points equally distributed along the $a$-line such that $|a_ia_{i+1}| = |b_ib_{i+1}|$ and $|b_1b_2| < |a_2b_2|$. The first condition can be done by symmetrically adding the vertices along both lines. The second condition, though, is achieved by assigning an appropriate value for angle $\alpha$. We choose this value by applying the sine rule. As a result we get:

![Figure 12: The $Y_2$ graph induced by the point set $V$](image)
\[
\sin \alpha = \frac{a_2 b_2}{b_0 b_2} = \frac{a_2 b_2}{4b_1 b_2}
\]

\[
\Rightarrow \frac{a_2 b_2}{b_1 b_2} = 4 \sin \alpha > 1
\]

\[
\Rightarrow \sin \alpha > \frac{1}{4}
\]

\[
\Rightarrow \alpha > 14.5^\circ
\]

In the Yao step, each node \(a_i\) selects the edges \(a_i \rightarrow a_{i+1}\) (if \(i < k\)) and \(a_i \rightarrow a_{i-1}\) (if \(i > 0\)) since \(|a_i a_{i+1}| < |a_i b_i|\). Similarly, each node \(b_i\) selects \(b_i \rightarrow b_{i+1}\) (if \(i < k\)), \(b_i \rightarrow b_{i-1}\) (if \(i > 0\)) since \(|b_i b_{i+1}| < |a_i b_i|\). The result is the Yao graph shown in Figure 12(b).

For the given \(\alpha\), we have that \(|a_k b_k| \approx |a_1 b_1|\). However, a shortest path between \(a_k\) and \(b_k\) in the Yao graph of Figure 12 has length

\[
\delta_H(a_k, b_k) = \sum_{i=1}^{k} |a_i a_{i-1}| + |a_1 b_1| + \sum_{i=0}^{k-1} |b_i b_{i+1}|
\]

\[
> n \cdot |a_1 b_1|
\]

It follows that the \(Y_2\) is not a spanner for the graph in Figure 12.

\[\square\]

5.2 \(Y_3\) is not a Spanner

**Lemma 3** \(Y_3\) is not a Spanner.

**Proof:** We prove that \(Y_3\) is not a spanner with the help of a counterexample. We follow the same steps we used to construct the \(Y_2\) counterexample. We, then, perform the Yao step where each node \(a_i\) selects the edges \(a_i \rightarrow a_{i+1}\) (if \(i < k\)) and \(a_i \rightarrow a_{i-1}\) (if \(i > 0\)) since \(|a_i a_{i+1}| < |a_i b_i|\). Similarly, each node \(b_i\) selects \(b_i \rightarrow b_{i+1}\) (if \(i < k\)), \(b_i \rightarrow b_{i-1}\) (if \(i > 0\)) since \(|b_i b_{i+1}| < |a_i b_i|\). The result is the Yao graph shown in Figure 13(b).
For the given value of $\alpha$, we have that $|a_k b_k| \approx |a_1 b_1|$. However, a shortest path between $a_k$ and $b_k$ in the Yao graph of Figure 13 has length

$$
\delta_H(a_k, b_k) = \sum_{i=1}^{k} |a_i a_{i-1}| + |a_1 b_1| + \sum_{i=0}^{k-1} |b_i b_{i+1}|
> n \cdot |a_1 b_1|
$$

It follows that the $Y_3$ is not a spanner for the graph in Figure 13.

\[\square\]

5.3 \textbf{$Y_4$ is a Spanner for Convex Position}

We conjecture that $Y_4$ is a spanner for arbitrary point sets, however we only prove this conjecture for sets of points in convex position.
Let \( u, v \in V \) be arbitrary. Let \( Q(u, v) \) denote the cone (quadrant) with origin at \( u \) that contains \( v \), and let \( SQ(u, v) \) denote a smallest square centered at \( v \) whose boundary contains \( u \). Note that \( Q(u, v) \) and \( SQ(u, v) \) are uniquely defined for a given pair of vertices \( u \) and \( v \) (see Figure 14a). Let \( R(u, v) \) denote the rectangle with diagonal \( uv \).

![Figure 14: Constructing \( \mathcal{P}(u \leadsto v) \). (a) The outer square is \( SQ(u_i, v) = SQ(u_{i+1}, v) \), and the inner square is \( SQ(u_{i+2}, v) \); shaded area is \( Q(u_i, v) \); (b) \( \mathcal{P}(u \leadsto v) \) lies inside \( SQ(u, v) \).](image)

For a fixed pair of vertices \( u, v \in V \), we define recursively a directed path \( \mathcal{P}(u \leadsto v) \) from \( u \) to \( v \) in \( Y_4 \) as follows. Let \( u_0 = u \), and let \( u_iu_{i+1} \in Y_4 \) be the edge that lies in \( Q(u_i, v) \), for each \( i \). Then

\[
\mathcal{P}(u_i \leadsto v) = \begin{cases} 
\bot, & \text{if } u_i = v \\
\overrightarrow{u_ii_{i+1}} \oplus \mathcal{P}(u_{i+1} \leadsto v), & \text{otherwise.}
\end{cases}
\]

(1)

Here \( \bot \) represents the null path and \( \oplus \) represents the concatenation operator. This definition is illustrated in Figure 14a. Fischer et al. [6] show the following properties of
Property 4 Let $P(u \leadsto v) = u_0, u_1, \ldots$ be as defined in (1). Then, for each $i \geq 0$,

(a) $SQ(u_{i+1}, v) \subseteq SQ(u_i, v)$.

(b) $SQ(u_{i+2}, v) \subset SQ(u_i, v)$.

(c) $P(u_i \leadsto v)$ lies entirely inside $SQ(u_i, v)$.

In the example from Figure 14a for instance, $SQ(u_{i+1}, v) = SQ(u_i, v)$, however $SQ(u_{i+2}, v) \subset SQ(u_i, v)$. This latter property is critical to ensure that progress is made towards $v$ and that the path $P(u \leadsto v)$ is well defined. We now prove another important property of $P(u \leadsto v)$.

Lemma 5 $P(u \leadsto v)$ does not self-intersect\(^2\).

Proof: The proof is by contradiction. Assume to the contrary that $P_{uv} = P(u \leadsto v)$ is a self-intersecting path. Let $(u = u_0), u_1, u_2, \ldots$ be the vertices of $P_{uv}$ in the order in which they appear from $u$ to $v$ along $P_{uv}$. Let $k > 0$ be the smallest index such $P_{uv}[u, u_k]$ does not self-intersect, but $P_{uv}[u, u_{k+1}]$ self-intersects (so $P_{uv}[u, u_{k+1}]$ contains at least three edges, meaning that $k > 1$). Then it must be that $u_k u_{k+1}$ intersects one of the edges on the subpath $P_{uv}[u, u_k]$. Let $i < k$ be such that $u_i u_{i+1}$ intersects $u_k u_{k+1}$ (see Figure 15). Since $u_i u_{i+1}$ and $u_k u_{k+1}$ are non-adjacent, it must be that $k > i + 1$. This along with Property 4(b) implies that $SQ(u_{k+1}, v) \subset SQ(u_{i+1}, v)$. By Property 4(a), $SQ(u_{i+1}, v) \subseteq SQ(u_i, v)$. These together show that $u_{k+1}$ cannot coincide with $u_i$ or $u_{i+1}$, and therefore $u_k u_{k+1}$ and $u_i u_{i+1}$ do not share an endpoint. Thus, if $u_i u_{i+1}$ and $u_k u_{k+1}$ intersect, they must share a point interior to each of them. This along with the

\(^2\)A path self-intersects if two non-adjacent edges share a point.
Fact that $u_{k+1}$ lies strictly inside $SQ(u_{i+1}, v)$ implies that $u_{k+1} \in R(u_i, u_{i+1})$. This in turn implies that $u_{k+1} \in Q(u_i, v)$ and $|u_iu_{k+1}| < |u_iu_{i+1}|$, contradicting the fact that $u_iu_{i+1}$ is a shortest edge in $Q(u_i, v)$ (i.e., a Yao edge). This completes the proof. \(\square\)

An immediate consequence of Lemma 5 is the following:

**Corollary 6** Let $P(u \leadsto v) = (u = u_0), u_1, \ldots, u_k = v)$. Then for any $j$, $0 \leq j < k$, $u_ju_{j+1}$ is on the convex hull of $u_j, u_{j+1}, \ldots, u_{k-1}, u_k$.

The path $P(u \leadsto v)$ can have an interesting oscillating behavior, as shown in Figure 16. This is due to the fact that, at each intermediate point $u_i$, the path continues towards $v$ along the Yao edge incident to $u_i$ that lies in $Q(u_i, v)$, ignoring the incident (and potentially shorter) edges from adjacent quadrants. As a result, a vertex $u_j \in P(u \leadsto v)$ that does not lie in $Q(u_i, v)$, but lies very close to $u_i$, might end up separated from $u_i$ by a long subpath in $P(u \leadsto v)$ (see $P_{uv}[u_i, u_j]$ from Figure 16). We compensate for this shortcoming by refining $P(u \leadsto v)$ in a second step, as described in the Y4Path.
Figure 16: \( \mathcal{P}(u \leadsto v) \) is arbitrarily long compared to \(|uv|\).

algorithm below. The refinement step replaces longer subpaths of \( P_{uv} \) by shorter paths from \( Y_4 \), if possible.

For the example shown in Figure 17 for instance, the Y4Path algorithm would replace the subpath \((u_2u_3, u_3u_4, u_4u_5)\) by \( \mathcal{P}(u_5 \leadsto u_2) \). We will show that \( \mathcal{P}(u_5 \leadsto u_2) \) has a nice behavior and does not interfere with the existing path \( P_{uv} \). In particular, \( \mathcal{P}(u_5 \leadsto u_2) \) lies exterior to \( \mathcal{H}(P_{uv}) \), in the region shaded in Figure 17. This property is shown formally in the following lemma.
Lemma 7 Let \( u_i u_j \) be an edge processed in Step 2 of the Y4Path algorithm, with \( i > 0 \). Then \( P(u_j \leadsto u_i) \) lies in the closed rectangle \( R = R(u_j, u_i) \).

**Proof:** Let \( Q_0 = Q(u_i, u_{i+1}) \), and assume without loss of generality that \( Q_1 = Q(u_i, u_j) \) lies counterclockwise from \( Q_0 \) (see Figure 18). Let \( Q_2 \) and \( Q_3 \) be the other two quadrants with origin \( u_i \), in counterclockwise order around \( u_i \).

The proof is by contradiction. Let \( \overrightarrow{w_kw_{k+1}} \in Q(w_k, u_i) \) be the first edge along \( P(u_j \leadsto u_i) \) that crosses the boundary of \( R \). Thus \( w_k \in R \) and \( w_{k+1} \notin R \). Since \( w_k \in R \), we have that

\[
|w_k u_i| < |u_j u_i| \quad (2)
\]

Since \( w_{k+1} \in Q(w_k, u_i) \) and \( \overrightarrow{w_kw_{k+1}} \) is a Yao edge, it must be that

\[
|w_k w_{k+1}| \leq |w_k u_i| \quad (3)
\]

This implies that \( w_{k+1} \notin Q_3 \). Substituting (3) in the triangle inequality

\[
|w_k w_{k+1}| <\]
\[ |u_i w_k| + |w_k w_{k+1}| \text{ yields } |u_i w_{k+1}| < 2|w_k u_i|. \] Substituting (2) in this latter inequality yields

\[ |u_i w_{k+1}| < 2|u_j u_i| \tag{4} \]

Recall that Step 2 of the Y4Path algorithm processes edges \( u_i u_j \) of length \( |u_i u_j| < |u_i u_{i+1}|/2 \). This together with (4) yields

\[ |u_i w_{k+1}| < |u_i u_{i+1}| \tag{5} \]

Inequality (5) along with the fact that \( \overrightarrow{u_i u_{i+1}} \) is a Yao edge, implies that \( w_{k+1} \not\in Q_0 \). It also holds that \( w_{k+1} \not\in Q_1 \), since \( Q_1 \cap Q(w_k, u_i) \subseteq R \) and \( w_{k+1} \in Q(w_k, u_i) \setminus R \).

Figure 18: If \( u_{i-1} \in Q_0 \), then \( u_{i+1} \not\in Q(u_i, v) \).

It remains to discuss the case \( w_{k+1} \in Q_2 \) (see Figure 18). To derive a contradiction,
we consider possible positions of $u_{i-1}$ with respect to $u_i$ (since $i > 0$, $u_{i-1}$ exists). Recall that $u_iu_j \in \partial H(P_{uv})$. This implies that $u_{i-1}$ and $u_{i+1}$ are on the same side of the line $u_iu_j$, and so $u_{i-1} \notin Q_2$. By Cor. 6, $u_{i-1}u_i$ is a hull edge of $P_{uv}[u_{i-1}, v]$, so $u_{i+1}$ and $u_j$ are on the same side of the line $u_{i-1}u_i$. This implies that

(a) $u_{i-1} \notin Q_1$, since $u_iu_j$ is also a hull edge of $P_{uv}$.

(b) If $u_{i-1} \in Q_0$, then $u_{i-1}$ must lie outside the convex angle $\angle u_ju_iu_{i+1}$ (as in Figure 18).

In this latter case, note that $v$ must lie in $Q(u_{i-1}, u_i) \setminus R(u_{i-1}, u_i)$ (since $\overrightarrow{u_{i-1}u_i}$ is a Yao edge). But this would contradict the fact that $u_{i+1} \in Q(u_i, v)$, so $u_{i-1} \notin Q_0$.

Finally, in the case $u_{i-1} \in Q_3$, the quadrilateral $u_{i-1}u_{i+1}u_jw_{k+1}$ would contain $u_i$ in its interior, contradicting the fact that $V$ is in convex position. These arguments together establish that $w_{k+1} \notin Q_2$.

Having exhausted all cases, we conclude that $w_kw_{k+1}$ cannot cross the boundary of $R$. \hfill \Box

The following corollary follows immediately from Lemma 7 and the fact that $V$ is in convex position.

**Corollary 8** Let $u_iu_j$ be an edge processed in Step 2 of the Y4Path algorithm, with $i > 0$. Then $P(u_i \leadsto u_j) \subset \partial H(P_{uv})$.

**Lemma 9** If $P$ is a convex polygon containing $u$ and $v$ that lies inside $SQ(u, v)$, then the perimeter of $P$ is no more than $2(2 + \sqrt{2}) \cdot |uv|$.

**Proof:** Let $ab$ be a side of $P$ containing $v$. Extend $ab$ beyond each endpoint until it meets the boundary of $SQ(u, v)$ at two points, call them $w_1$ and $w_2$ (see Figure 19.
Then $w_1w_2$ splits $SQ(u,v)$ into two congruent trapezoids (when $w_1w_2$ is a diagonal of $SQ(u,v)$, the two trapezoids degenerate into two right triangles). Since $P$ is convex, all points of $P$ are on the same side of $w_1w_2$, so $P$ has to lie inside one of these trapezoids – call it $T$. The fact that $P \subseteq T$ along with the convexity property of $P$ implies that the perimeter of $P$ does not exceed the perimeter of $T$.

![Figure 19: The perimeter of $T$ does not exceed $(2 + \sqrt{2})\ell$.](image)

Next we calculate the perimeter of $T$. Let $\ell$ be the side length of $SQ(u,v)$. Since $w_1$ and $w_2$ are symmetric with respect to $v$, the lengths of the two bases of $T$ sum up to $\ell$. The third side of $T$ other than $w_1w_2$ also has length $\ell$. So the perimeter of $T$ is

$$2\ell + |w_1w_2| \leq (2 + \sqrt{2})\ell$$

Since $u$ is on the boundary of $SQ(u,v)$, $|uv| \geq \ell/2$. This along with the inequality above shows that the perimeter of $P$ is no more than $2(2 + \sqrt{2})|uv|$.

\[\square\]

**Theorem 10** For any set $V$ of points in convex position, the Yao graph $Y_4$ on $V$ is a $t$-spanner, for $t = 4(2 + \sqrt{2})$. 

27
Proof: Cor. 8 implies that the only edges on $P_{uv}$ crossing the interior of $\mathcal{H}(P_{uv})$ are those edges not processed in Step 2 of the Y4Path algorithm. These are diagonals $u_iu_{i+1}$ of $\mathcal{H}(P_{uv})$ adjacent to $u_iu_j \in \partial\mathcal{H}(P_{uv}) \setminus P_{uv}$, with the property that $|u_iu_{i+1}| \leq 2|u_iu_j|$. It follows that $P_{uv}$ is no longer than twice the perimeter of $\partial\mathcal{H}(P_{uv})$. This along with Lem. 9 concludes the proof.

Concluding Remarks. Proving that $Y_4$ is a spanner for arbitrary point sets remains open.

6 Conclusion and Future Work

In this thesis, we solved five open problem question and gave detailed proofs of the results. Our work has significant impact in the wireless networking community, due to the fact that the Yao structure can be computed in a distributed fashion with constant communication among nodes. Throughout the course of this thesis, we proved $YY_4$, $YY_6$, $Y_2$, $Y_3$ not to have the spanner property by means of four counterexamples. Then, we showed that $Y_4$ is a $4(2 + \sqrt{2})$-spanner for points in convex position. Our analysis provides much insight into structure of $Y_4$, leading us to believe that similar techniques can be used to prove that $Y_4$ is a spanner for arbitrary sets of points. We would like to further investigate the possibility of $Y_4$ being a $t$-spanner for points in arbitrary position. In addition, we would like to tackle another interesting open problem that is whether $YY_k$ can be a spanner for $k > 6$. Currently we are looking into the possibility of $YY_8$ being a length spanner.

References


