Partial Delaunay Triangulation-Based Asynchronous Planarization of Quasi Unit Disk Graphs

Florentin Neumann and Hannes Frey

March 11, 2015 © NetSys 2015
Problem statement

Distributed (local) planarization of wireless network graphs

Definition (Spanning ratio)

Spanning ratio (stretch factor) of subgraph \( H \subseteq G \) defined by

\[
\max_{u,v \in V(G)} \left\{ \frac{d_H(u,v)}{d_G(u,v)} \right\}.
\]

Today: \( d_H(u,v) \) Euclidean shortest path distance in graph \( H \)
Problem statement

Distributed (local) planarization of wireless network graphs

Definition (Spanning ratio)

Spanning ratio (stretch factor) of subgraph \( H \subseteq G \) defined by

\[
\max_{u,v \in V(G)} \left\{ \frac{d_H(u, v)}{d_G(u, v)} \right\}.
\]

Today: \( d_H(u, v) \) Euclidean shortest path distance in graph \( H \)
Standard techniques and models

Unit Disk Graph (UDG)

- Spanning ratio of Gabriel Graph (GG): $\Theta(\sqrt{n})$  
  [Bose et al. 2006]

- Spanning ratio of Partial Delaunay Triangulation (PDT): $O(1)$  
  [Neumann & Frey 2012]
Standard techniques and models

Unit Disk Graph (UDG)

- **Spanning ratio of Gabriel Graph (GG):** $\Theta(\sqrt{n})$
  [Bose et al. 2006]

- **Spanning ratio of Partial Delaunay Triangulation (PDT):** $O(1)$
  [Neumann & Frey 2012]
Standard techniques and models

Unit Disk Graph (UDG)

- Spanning ratio of Gabriel Graph (GG): $\Theta(\sqrt{n})$  
  [Bose et al. 2006]

- Spanning ratio of Partial Delaunay Triangulation (PDT): $O(1)$  
  [Neumann & Frey 2012]
Standard techniques and models

Unit Disk Graph (UDG)

- Spanning ratio of **Gabriel Graph (GG)**: $\Theta(\sqrt{n})$  
  [Bose et al. 2006]

- Spanning ratio of **Partial Delaunay Triangulation (PDT)**: $O(1)$  
  [Neumann & Frey 2012]
Standard techniques and models

Unit Disk Graph (UDG)

Quasi Unit Disk Graph (QUDG)

- Spanning ratio of **Gabriel Graph (GG)**: \( \Theta(\sqrt{n}) \)  
  [Bose et al. 2006]

- Spanning ratio of **Partial Delaunay Triangulation (PDT)**: \( O(1) \)  
  [Neumann & Frey 2012]
Standard techniques and models

Unit Disk Graph (UDG)

Quasi Unit Disk Graph (QUDG)

- Spanning ratio of Gabriel Graph (GG): $\Theta(\sqrt{n})$
  [Bose et al. 2006]

- Spanning ratio of Partial Delaunay Triangulation (PDT): $O(1)$
  [Neumann & Frey 2012]
## Related work and the level of synchrony

<table>
<thead>
<tr>
<th>Reference</th>
<th>Localized</th>
<th>Arb.</th>
<th>QUDG</th>
<th>Planar</th>
<th>E. spanner</th>
<th>H. spanner</th>
<th>ASYNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barrière et al. 2003</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Moaveninejad et al. 2005</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Chávez et al. 2006</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Funke et al. 2007</td>
<td>✓</td>
<td>✓/X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>Kuhn et al. 2008</td>
<td>✓</td>
<td>✓</td>
<td>✓/X</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lillis et al. 2008</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Guan 2009</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>●</td>
</tr>
<tr>
<td>Damian et al. 2009</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chen et al. 2010</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>●</td>
</tr>
</tbody>
</table>

The level of synchrony in Distributed Computing [Peleg 2000]

<table>
<thead>
<tr>
<th>Synchronous</th>
<th>Asynchronous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded link delay</td>
<td>Unbounded (finite) link delay</td>
</tr>
<tr>
<td>Global clock ticks</td>
<td>No global clock</td>
</tr>
<tr>
<td>Round-based algorithms</td>
<td>Event-driven algorithms</td>
</tr>
</tbody>
</table>

Class of LOCAL Algorithms

Class of ASYNC Algorithms
Related work and the level of synchrony

<table>
<thead>
<tr>
<th>Reference</th>
<th>Localized</th>
<th>Arb.</th>
<th>QUDG</th>
<th>Planar</th>
<th>E. spanner</th>
<th>H. spanner</th>
<th>(ASYNC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barrière et al.</td>
<td>2003</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Moaveninejad et al.</td>
<td>2005</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Chávez et al.</td>
<td>2006</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Funke et al.</td>
<td>2007</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Kuhn et al.</td>
<td>2008</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Lillis et al.</td>
<td>2008</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Guan</td>
<td>2009</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Damian et al.</td>
<td>2009</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Chen et al.</td>
<td>2010</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
</tr>
</tbody>
</table>

The level of synchrony in Distributed Computing [Peleg 2000]

**Synchronous**
- Bounded link delay
- Global clock ticks
- Round-based algorithms

Class of **LOCAL** Algorithms

**Asynchronous**
- Unbounded (finite) link delay
- No global clock
- Event-driven algorithms

Class of **ASYNC** Algorithms
Related work and the level of synchrony

<table>
<thead>
<tr>
<th>Reference</th>
<th>Localized</th>
<th>Arb. QUDG</th>
<th>Planar</th>
<th>E. spanner</th>
<th>H. spanner</th>
<th>ASYNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barrière et al. 2003</td>
<td>(✓)</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Moaveninejad et al. 2005</td>
<td>(✓)</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
</tbody>
</table>

The level of synchrony in Distributed Computing [Peleg 2000]

- Synchronous
  - Bounded link delay
  - Global clock ticks
  - Round-based algorithms
  - Class of LOCAL Algorithms

- Asynchronous
  - Unbounded (finite) link delay
  - No global clock
  - Event-driven algorithms
  - Class of ASYNC Algorithms
Contribution

Algorithm AsyncPDT (extension of Barrière et al. 2003)

- Input: QUDG with $\frac{r_{\text{max}}}{r_{\text{min}}} \leq \sqrt{2}$
- Output: Planar connected PDT-based overlay graph

Model and assumptions:

- No global clock, no clock synchronization
- Message transmissions are reliable but take unpredictable (finite) time
- Nodes know their positions in the Euclidean plane
- No four nodes are cocircular (Delaunay Triangulations)
General idea

Input QUDG $G$
General idea

Obtain supergraph $S(G)$ after Completion Phase
General idea

Result $PDT(S(G))$ after Extraction Phase
Control flow

Initalization

Completion Phase

$S(G)$

Extraction Phase

$PDT(S(G))$

Routing Phase

Message of type:

- new neighbor
- new witness

$G$

$S(G)$

$PDT(S(G))$
Initialization

Execution by node \( u \)

- \( \mathcal{L}(u) = {} \)
- \( \mathcal{W}(u) = {} \)
- Local broadcast: request for neighbors
- Start Completion Phase
Completion Phase

Execution by node $u$, processing of (virtual) neighbor $v$

$$\forall w \in \mathcal{L}(u) : r_{\text{min}} < ||v-w||_2 \leq r_{\text{max}}$$

Case: $w \in \text{Disk}(uv)$

- Send ‘(new, $w$)’ to $v$
- Send ‘(new, $v$)’ to $w$
Completion Phase

Execution by node $u$, processing of (virtual) neighbor $v$

$$\forall w \in \mathcal{L}(u) : r_{\text{min}} < \|vw\|_2 \leq r_{\text{max}}$$

Case: $w \in \text{Disk}(uv)$

- Send ‘(new, $w$)’ to $v$
- Send ‘(new, $v$)’ to $w$
Completion Phase

Execution by node $u$, processing of (virtual) neighbor $v$

\[ \forall w \in \mathcal{L}(u) : r_{\text{min}} < \|vw\|_2 \leq r_{\text{max}} \]

Case: $w \in \text{Disk}(uv)$
- Send ‘(new, $w$)’ to $v$
- Send ‘(new, $v$)’ to $w$

Case: $w \notin \text{Disk}(uv)$
- Send ‘(witness, $w$)’ to $v$
- Send ‘(witness, $v$)’ to $w$
Completion Phase

Execution by node $u$, processing of (virtual) neighbor $v$

$$\forall w \in \mathcal{L}(u) : r_{\min} < ||vw||_2 \leq r_{\max}$$

Case: $w \in \text{Disk}(uv)$
- Send ‘(new, $w$)’ to $v$
- Send ‘(new, $v$)’ to $w$

Case: $w \notin \text{Disk}(uv)$
- Send ‘(witness, $w$)’ to $v$
- Send ‘(witness, $v$)’ to $w$
Completion Phase (cont’d)

Execution by node $u$, processing of witness $w$

$$\forall v \in \mathcal{L}(u) : r_{\text{min}} < \|vw\|_2 \leq r_{\max}$$

Send ‘(witness, $w$)’ to $v$
Completion Phase (cont’d)

Execution by node \( u \), processing of witness \( w \)

\[ \forall v \in \mathcal{L}(u) : r_{\text{min}} < \|vw\|_2 \leq r_{\text{max}} \]

Send ‘(witness, \( w \))’ to \( v \)
Extraction Phase

Execution by node $u$

For all (virtual) edges $uv$, check if $uv \in PDT(S(G))$ w.r.t. $\mathcal{L}(u) \cup \mathcal{W}(u)$
Extraction Phase

Execution by node \( u \)

For all (virtual) edges \( uv \), check if \( uv \in PDT(S(G)) \) w.r.t. \( \mathcal{L}(u) \cup \mathcal{W}(u) \)
Extraction Phase

Execution by node $u$

For all (virtual) edges $uv$, check if $uv \in PDT(S(G))$ w.r.t. $L(u) \cup W(u)$
Extraction Phase

Execution by node $u$

For all (virtual) edges $uv$, check if $uv \in PDT(S(G))$ w.r.t. $\mathcal{L}(u) \cup \mathcal{W}(u)$
Extraction Phase

Execution by node $u$

For all (virtual) edges $uv$, check if $uv \in PDT(S(G))$ w.r.t. $\mathcal{L}(u) \cup \mathcal{W}(u)$
Extraction Phase

Execution by node $u$

For all (virtual) edges $uv$, check if $uv \in PDT(S(G))$ w.r.t. $\mathcal{L}(u) \cup \mathcal{W}(u)$

$PDT(S(G))[u]$

$GG(S(G))[u]$
Termination and symmetry

**Lemma (Barrière et al. 2003)**

*Completion Phase and Extraction Phase terminate.*

**Lemma**

$PDT(S(G))$ is symmetric.
Termination and symmetry

Lemma (Barrière et al. 2003)

Completion Phase and Extraction Phase terminate.

Lemma

\( \text{PDT}(S(G)) \) is symmetric.
Connectivity, planarity, and message size

**Theorem**
If $G$ is connected, then $PDT(S(G))$ is connected.

**Theorem**
$PDT(S(G))$ is a planar graph.

**Observation**
Message size is $O(P_{\text{max}})$. 
Discussion of locality

Lemma (Barrière et al. 2003)

For any $k > 0$, there is a configuration s.t. the path corresponding to a virtual edge has hop-length $> k$.

Lemma (Kuhn et al. 2008)

If $G$ is civilized for $\lambda > 0$, and $r_{\text{max}} = 1$, any virtual edge in $G\hat{G}(S(G))$ has hop-length at most

$$1 + \frac{1}{2\lambda^2}.$$
Discussion of locality

Lemma (Barrière et al. 2003)

For any $k > 0$, there is a configuration s.t. the path corresponding to a virtual edge has hop-length $> k$.

Lemma (Kuhn et al. 2008)

If $G$ is civilized for $\lambda > 0$, and $r_{\text{max}} = 1$, any virtual edge in $GG(S(G))$ has hop-length at most

$$1 + \frac{1}{2\lambda^2}.$$
Discussion of locality (cont’d)

Lemma
If $G$ is civilized for $\lambda > 0$, then the decision by $u$ if (virtual) edge $uv \in PDT(S(G))$ depends on at most the $c \cdot k$-neighborhood of $u$, where

$$c \cdot k < 9 \cdot \left[ 1 + 1/(2\lambda^2) \right]$$

Corollary
If $G$ is civilized, to compute $u$’s adjacency in $PDT(S(G))$ it suffices to execute AsyncPDT in $u$’s $c \cdot k$-neighborhood.
Discussion of locality (cont’d)

Lemma
If \( G \) is civilized for \( \lambda > 0 \), then the decision by \( u \) if (virtual) edge \( uv \in PDT(S(G)) \) depends on at most the \( c \cdot k \)-neighborhood of \( u \), where

\[
c \cdot k < 9 \cdot \left[ 1 + 1/(2\lambda^2) \right]
\]

Corollary
If \( G \) is civilized, to compute \( u \)'s adjacency in \( PDT(S(G)) \) it suffices to execute AsyncPDT in \( u \)'s \( c \cdot k \)-neighborhood.
Simulation Setup

- **Simulator for Network Algorithms** (Sinalgo, DCG @ ETH Zurich)
- Area $500 \times 500$, $r_{\text{min}} = 36$, $r_{\text{max}} = 50$, $\frac{r_{\text{max}}}{r_{\text{min}}} \approx \sqrt{2}$
- Node density $\delta \in [5..16]$
- 500 random graphs per density

Comparison of Barrière et al. 2003 and AsyncPDT

- Euclidean spanning ratio
- Message complexity
Spanning Ratio

The “good news”

![Graph showing the comparison of Barriere et al. and AsyncPDT in terms of Avg. Eucl. spanning ratio and Avg. improvement (%)]
Message Complexity

The “bad news”
Conclusive remarks

• AsyncPDT
  • provably correct
  • local
  • good spanners
  • high msg. complexity

• Can reduce msg. complexity using idea from Moaveninejad et al. 2005

• Conjecture: $PDT(S(G))$ has constant Euclidean spanning ratio