Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension

Part I:
Problem Definition

Alexander Wolff
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.
Let $V' \subseteq V$ and $H = G[V']$
Let $\Gamma_H$ be a representation of $H$. 
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

Let $\Gamma_H$ be a representation of $H$.

Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$.

Let $\Gamma_H$ be a representation of $H$.

Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$. 
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.
Let $V' \subseteq V$ and $H = G[V']$
Let $\Gamma_H$ be a representation of $H$.
Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

Let $\Gamma_H$ be a representation of $H$.

Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.
Let $V' \subseteq V$ and $H = G[V']$
Let $\Gamma_H$ be a representation of $H$.
Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$

Polytime for:
- (unit) interval graphs
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.
Let $V' \subseteq V$ and $H = G[V']$
Let $\Gamma_H$ be a representation of $H$.
Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$

Polytime for:
- (unit) interval graphs
- permutation graphs
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.
Let $V' \subseteq V$ and $H = G[V']$
Let $\Gamma_H$ be a representation of $H$.
Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$

Polytime for:
- (unit) interval graphs
- permutation graphs
- circle graphs
Partial Representation Extension Problem

Let \( G = (V, E) \) be a graph.
Let \( V' \subseteq V \) and \( H = G[V'] \)
Let \( \Gamma_H \) be a representation of \( H \).
Find a representation \( \Gamma_G \) of \( G \) that extends \( \Gamma_H \)

Polytime for:
- (unit) interval graphs
- permutation graphs
- circle graphs

NP-hard for:
- planar straight-line drawings
- contacts of
- disks
- triangles
- orthogonal segments
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$.

Let $\Gamma_H$ be a representation of $H$.

Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$.

Polytime for:
- (unit) interval graphs
- permutation graphs
- circle graphs

NP-hard for:
- planar straight-line drawings
- contacts of disks
- triangles
- orthogonal segments
- planar straight-line drawings
- contacts of disks
- triangles
- orthogonal segments
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

Let $\Gamma_H$ be a representation of $H$.

Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$.

NP-hard for:
- planar straight-line drawings
- contacts of disks
- triangles
- orthogonal segments

Polytime for:
- (unit) interval graphs
- permutation graphs
- circle graphs
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.
Let $V' \subseteq V$ and $H = G[V']$
Let $\Gamma_H$ be a representation of $H$.
Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$

Polytime for:
- (unit) interval graphs
- permutation graphs
- circle graphs

NP-hard for:
- planar straight-line drawings
- contacts of disks
- disks
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.

Let $V' \subseteq V$ and $H = G[V']$

Let $\Gamma_H$ be a representation of $H$.

Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$

---

NP-hard for:
- planar straight-line drawings
- contacts of
- disks
- triangles

Polytime for:
- (unit) interval graphs
- permutation graphs
- circle graphs
Partial Representation Extension Problem

Let $G = (V, E)$ be a graph.
Let $V' \subseteq V$ and $H = G[V']$
Let $\Gamma_H$ be a representation of $H$.
Find a representation $\Gamma_G$ of $G$ that extends $\Gamma_H$

Polytime for:
- (unit) interval graphs
- permutation graphs
- circle graphs

NP-hard for:
- planar straight-line drawings
- contacts of
  - disks
  - triangles
  - orthogonal segments
Bar Visibility Representation
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called bars.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called \textit{bars}.
- \textbf{Edges} correspond to unobstructed vertical lines of sight.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called bars.
- Edges correspond to unobstructed vertical lines of sight.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?

Models.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?

Models.

- **Strong:**
  
  Edge $uv \iff$ unobstructed **0-width** vertical lines of sight.
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?

**Models.**

- **Strong:**
  Edge $uv \iff$ unobstructed **0-width** vertical lines of sight.
- **Epsilon:**
  Edge $uv \iff \varepsilon$-wide vertical lines of sight for $\varepsilon > 0$. 
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?

**Models.**

- **Strong:**
  Edge $uv \iff$ unobstructed **0-width** vertical lines of sight.
- **Epsilon:**
  Edge $uv \iff \epsilon$-wide vertical lines of sight for $\epsilon > 0$. 
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?

**Models.**

- **Strong:**
  Edge $uv \iff$ unobstructed **0-width** vertical lines of sight.
- **Epsilon:**
  Edge $uv \iff \epsilon$-wide vertical lines of sight for $\epsilon > 0$. 
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called **bars**.
- **Edges** correspond to unobstructed vertical lines of sight.
- What about unobstructed **0-width** vertical lines of sight? Do all visibilities induce edges?

**Models.**

- **Strong:**
  Edge $uv \iff$ unobstructed **0-width** vertical lines of sight.

- **Epsilon:**
  Edge $uv \iff$ $\epsilon$-wide vertical lines of sight for $\epsilon > 0$.

- **Weak:**
  Edge $uv \Rightarrow$ unobstructed vertical sightlines exists, i.e., any subset of **visible** pairs
Bar Visibility Representation

- Vertices correspond to horizontal open line segments called bars.
- Edges correspond to unobstructed vertical lines of sight.
- What about unobstructed 0-width vertical lines of sight? Do all visibilities induce edges?

Models.
- Strong:
  Edge $uv \iff$ unobstructed 0-width vertical lines of sight.
- Epsilon:
  Edge $uv \iff \epsilon$-wide vertical lines of sight for $\epsilon > 0$.
- Weak:
  Edge $uv \Rightarrow$ unobstructed vertical sightlines exists, i.e., any subset of visible pairs
Problems
Problems

weak

strong
Problems

weak

strong

epsilon
Problems

weak

strong

epsilon

a

b

c

da

d

b

c

b

c

da

d

b

c

b

c

a

d

a

d

Problems

weak

strong

epsilon
Recognition Problem.
Given a graph $G$, decide whether there exists a weak/strong/ε bar visibility representation $\psi$ of $G$. 
Recognition Problem.
Given a graph $G$, decide whether there exists a weak/strong/ε bar visibility representation $\psi$ of $G$.

Construction Problem.
Given a graph $G$, construct a weak/strong/ε bar visibility representation $\psi$ of $G$ – if one exists.
Problems

Recognition Problem.
Given a graph $G$, decide whether there exists a weak/strong/$\epsilon$ bar visibility representation $\psi$ of $G$.

Construction Problem.
Given a graph $G$, construct a weak/strong/$\epsilon$ bar visibility representation $\psi$ of $G$ – if one exists.

Partial Representation Extension Problem.
Given a graph $G$ and a set of bars $\psi'$ of $V' \subset V(G)$, decide whether there exists a weak/strong/$\epsilon$ bar visibility representation $\psi$ of $G$ where $\psi|_{V'} = \psi'$ (and construct $\psi$ if a representation exists).
Background

weak

strong

epsilon
Weak Bar Visibility.

- All planar graphs. [Tamassia & Tollis '86; Wismath '85]
- Linear time recognition and construction [T&T '86]
- Representation Extension is NP-complete [Chaplick et al. '14]
Background

Weak Bar Visibility.

- All planar graphs. [Tamassia & Tollis ’86; Wismath ’85]
Background

Weak Bar Visibility.

- All planar graphs. [Tamassia & Tollis ’86; Wismath ’85]
- Linear time recognition and construction [T&T ’86]
Background

Weak Bar Visibility.
- All planar graphs. [Tamassia & Tollis ’86; Wismath ’85]
- Linear time recognition and construction [T&T ’86]
- Representation Extension is NP-complete [Chaplick et al. ’14]
Background

Weak Bar Visibility.

- All planar graphs. [Tamassia & Tollis '86; Wismath '85]
- Linear time recognition and construction [T&T '86]
- Representation Extension is NP-complete [Chaplick et al. '14]

Strong Bar Visibility.
Weak Bar Visibility.
- All planar graphs. [Tamassia & Tollis ’86; Wismath ’85]
- Linear time recognition and construction [T&T ’86]
- Representation Extension is NP-complete [Chaplick et al. ’14]

Strong Bar Visibility.
- NP-complete to recognize [Andreae ’92]
**Background**

- **ε-Bar Visibility.**

- Planar graphs that can be embedded with all cut vertices on the outerface. [T&T '86, Wismath '85]
- Linear-time recognition and construction [T&T '86]
- Representation extension?
**Background**

**ε-Bar Visibility.**

- Planar graphs that can be embedded with all cut vertices on the outerface. [T&T ’86, Wismath ’85]
Background

ε-Bar Visibility.

- Planar graphs that can be embedded with all cut vertices on the outerface. [T&T '86, Wismath '85]
- Linear-time recognition and construction [T&T '86]
Background

- Bar Visibility.
  - Planar graphs that can be embedded with all cut vertices on the outerface. [T&T ’86, Wismath ’85]
  - Linear-time recognition and construction [T&T ’86]
  - Representation extension?
**Background**

- **ε-Bar Visibility.**
  - Planar graphs that can be embedded with all **cut vertices** on the outerface. [T&T ’86, Wismath ’85]
  - Linear-time recognition and construction [T&T ’86]
  - Representation extension? **This Lecture!**
Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension

Part II:
Recognition & Construction

 Alexander Wolff
ε-bar Visibility and $st$-Graphs

Recall that an $st$-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$. 
$\varepsilon$-bar Visibility and $st$-Graphs

Recall that an $st$-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$. 
$\varepsilon$-bar Visibility and $st$-Graphs

Recall that an $st$-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$. 
$\varepsilon$-bar Visibility and $st$-Graphs

Recall that an $st$-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

**Observation.**

$st$-orientations correspond to $\varepsilon$-bar visibility representations.
ε-bar Visibility and \textit{st}-Graphs

Recall that an \textit{st-graph} is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

\textbf{Observation.}
\textit{st}-orientations correspond to ε-bar visibility representations.
\(\varepsilon\)-bar Visibility and \(st\)-Graphs

Recall that an \(st\)-graph is a planar digraph \(G\) with exactly one source \(s\) and one sink \(t\) where \(s\) and \(t\) occur on the outer face of an embedding of \(G\).

**Observation.**
\(st\)-orientations correspond to \(\varepsilon\)-bar visibility representations.
ε-bar Visibility and $st$-Graphs

Recall that an $st$-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

**Observation.**

$st$-orientations correspond to ε-bar visibility representations.
\(\varepsilon\)-bar Visibility and \(st\)-Graphs

Recall that an \(st\)-graph is a planar digraph \(G\) with exactly one source \(s\) and one sink \(t\) where \(s\) and \(t\) occur on the outer face of an embedding of \(G\).

**Observation.**

\(st\)-orientations correspond to \(\varepsilon\)-bar visibility representations.
\( \varepsilon \)-bar Visibility and \( st \)-Graphs

Recall that an \( st \)-graph is a planar digraph \( G \) with exactly one source \( s \) and one sink \( t \) where \( s \) and \( t \) occur on the outer face of an embedding of \( G \).

**Observation.**

\( st \)-orientations correspond to \( \varepsilon \)-bar visibility representations.
**ε-bar Visibility and \textit{st}-Graphs**

Recall that an \textit{st}-graph is a planar digraph $G$ with exactly one \textbf{source} $s$ and one \textbf{sink} $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

\textbf{Observation.} \textit{st}-orientations correspond to ε-bar visibility representations.
ε-bar Visibility and \textit{st}-Graphs

Recall that an \textit{st}-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

\textbf{Observation.}

\textit{st}-orientations correspond to ε-bar visibility representations.
**ε-bar Visibility and \textit{st}-Graphs**

Recall that an \textit{st}-graph is a planar digraph $G$ with exactly one \textbf{source} $s$ and one \textbf{sink} $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

**Observation.**

\textit{st}-orientations correspond to \textbf{ε-bar} visibility representations.
$\varepsilon$-bar Visibility and $st$-Graphs

Recall that an $st$-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

**Observation.**

$st$-orientations correspond to $\varepsilon$-bar visibility representations.
ε-bar Visibility and \textit{st}-Graphs

Recall that an \textit{st}-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

\textbf{Observation.} \textit{st}-orientations correspond to ε-bar visibility representations.
ε-bar Visibility and \textit{st}-Graphs

Recall that an \textit{st}-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

\textbf{Observation.} \textit{st}-orientations correspond to ε-bar visibility representations.
Recall that an \textit{st-graph} is a planar digraph $G$ with exactly one \textit{source} $s$ and one \textit{sink} $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

\textbf{Observation.} \textit{st}-orientations correspond to $\varepsilon$-bar visibility representations.
\( \varepsilon \)-bar Visibility and \( st \)-Graphs

Recall that an \( st \)-graph is a planar digraph \( G \) with exactly one source \( s \) and one sink \( t \) where \( s \) and \( t \) occur on the outer face of an embedding of \( G \).

**Observation.**
\( st \)-orientations correspond to \( \varepsilon \)-bar visibility representations.
$\varepsilon$-bar Visibility and $st$-Graphs

Recall that an $st$-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

**Observation.**

$st$-orientations correspond to $\varepsilon$-bar visibility representations.
\( \varepsilon \)-bar Visibility and \( st \)-Graphs

Recall that an \( st \)-graph is a planar digraph \( G \) with exactly one source \( s \) and one sink \( t \) where \( s \) and \( t \) occur on the outer face of an embedding of \( G \).

**Observation.**

\( st \)-orientations correspond to \( \varepsilon \)-bar visibility representations.
**ε-bar Visibility and st-Graphs**

Recall that an *st-graph* is a planar digraph $G$ with exactly one *source* $s$ and one *sink* $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

Testing whether an acyclic planar digraph has a weak bar visibility representation is NP-complete.

**Observation.**

*st*-orientations correspond to ε-bar visibility representations.
Recall that an \textit{st-graph} is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

Testing whether an acyclic planar digraph has a weak bar visibility representation is NP-complete.

\begin{itemize}
  \item This is upward planarity testing! [Garg & Tamassia '01]
\end{itemize}
**ε-bar Visibility and *st*-Graphs**

Recall that an *st-graph* is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

- *ε-bar* visibility testing is easily done via *st*-graph recognition.

**Observation.**

*st*-orientations correspond to *ε-bar* visibility representations.
$\varepsilon$-bar Visibility and $st$-Graphs

Recall that an $st$-graph is a planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the outer face of an embedding of $G$.

- $\varepsilon$-bar visibility testing is easily done via $st$-graph recognition.
- Strong bar visibility recognition... open!

**Observation.**

$tst$-orientations correspond to $\varepsilon$-bar visibility representations.
**ε-bar Visibility and \( st \)-Graphs**

Recall that an \( st \)-graph is a planar digraph \( G \) with exactly one source \( s \) and one sink \( t \) where \( s \) and \( t \) occur on the outer face of an embedding of \( G \).

- **ε-bar visibility testing** is easily done via \( st \)-graph recognition.
- Strong bar visibility recognition... open!
- In a **rectangular** bar visibility representation \( \psi(s) \) and \( \psi(t) \) span an enclosing rectangle.

**Observation.**

\( st \)-orientations correspond to \( ε \)-bar visibility representations.
Theorem 1. [Chaplick et al. ’18]
Rectangular ε-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for $st$-graphs.
Results and Outline

Theorem 1. [Chaplick et al. ’18]
**Rectangular ε-Bar Visibility Representation Extension** can be solved in $O(n \log^2 n)$ time for \textit{st}-graphs.

- Dynamic program via SPQR-trees
Results and Outline

<table>
<thead>
<tr>
<th>Theorem 1. [Chaplick et al. ’18]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for $st$-graphs.</td>
</tr>
</tbody>
</table>

- Dynamic program via SPQR-trees
- Easier version: $O(n^2)$
Results and Outline

**Theorem 1.** [Chaplick et al. ’18]
**Rectangular** $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for $st$-graphs.
- Dynamic program via SPQR-trees
- Easier version: $O(n^2)$

**Theorem 2.** [Chaplick et al. ’18]
$\varepsilon$-Bar Visibility Representation Extension is NP-complete.
Results and Outline

**Theorem 1.** [Chaplick et al. ’18]

Rectangular $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for $st$-graphs.

- Dynamic program via SPQR-trees
- Easier version: $O(n^2)$

**Theorem 2.** [Chaplick et al. ’18]

$\varepsilon$-Bar Visibility Representation Extension is NP-complete.

- Reduction from Planar Monotone 3-SAT
Results and Outline

**Theorem 1.** [Chaplick et al. ’18]
**Rectangular** \(\varepsilon\)-Bar Visibility Representation Extension can be solved in \(O(n \log^2 n)\) time for \(st\)-graphs.
- Dynamic program via SPQR-trees
- Easier version: \(O(n^2)\)

**Theorem 2.** [Chaplick et al. ’18]
\(\varepsilon\)-Bar Visibility Representation Extension is NP-complete.
- Reduction from Planar Monotone 3-SAT

**Theorem 3.** [Chaplick et al. ’18]
\(\varepsilon\)-Bar Visibility Representation Extension is NP-complete for (series-parallel) \(st\)-graphs when restricted to the integer grid (or if any fixed \(\varepsilon > 0\) is specified).
Results and Outline

Theorem 1. [Chaplick et al. ’18]
Rectangular $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for st-graphs.

- Dynamic program via SPQR-trees
- Easier version: $O(n^2)$

Theorem 2. [Chaplick et al. ’18]
$\varepsilon$-Bar Visibility Representation Extension is NP-complete.

- Reduction from Planar Monotone 3-SAT

Theorem 3. [Chaplick et al. ’18]
$\varepsilon$-Bar Visibility Representation Extension is NP-complete for (series-parallel) st-graphs when restricted to the integer grid (or if any fixed $\varepsilon > 0$ is specified).

- Reduction from 3-Partition
Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension

Part III:
SPQR-Trees

Alexander Wolff
SPQR-Tree

- An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by **separation pairs**.
An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by *separation pairs*.
An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by **separation pairs**.
SPQR-Tree

- An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by **separation pairs**.
- The nodes of $T$ are of four types:
  - $S$-nodes represent a series composition
  - $P$-nodes represent a parallel composition
  - $Q$-nodes represent a single edge
  - $R$-nodes represent 3-connected (rigid) subgraphs
SPQR-Tree

- An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by separation pairs.

- The nodes of $T$ are of four types:
  - $S$-nodes represent a series composition
  - $P$-nodes represent a parallel composition
  - $Q$-nodes represent a single edge
  - $R$-nodes represent 3-connected (rigid) subgraphs
SPQR-Tree

- An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by separation pairs.

- The nodes of $T$ are of four types:
  - **S**-nodes represent a series composition
  - **P**-nodes represent a parallel composition
  - **Q**-nodes represent a single edge
  - **R**-nodes represent 3-connected (rigid) subgraphs
SPQR-Tree

- An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by **separation pairs**.

- The nodes of $T$ are of four types:
  - **S**-nodes represent a series composition
  - **P**-nodes represent a parallel composition
  - **Q**-nodes represent a single edge
  - **R**-nodes represent 3-connected (rigid) subgraphs
An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by **separation pairs**.

The nodes of $T$ are of four types:

- **S**-nodes represent a series composition
- **P**-nodes represent a parallel composition
- **Q**-nodes represent a single edge
- **R**-nodes represent 3-connected (rigid) subgraphs
An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by *separation pairs*.

The nodes of $T$ are of four types:
- **S**-nodes represent a series composition
- **P**-nodes represent a parallel composition
- **Q**-nodes represent a single edge
- **R**-nodes represent 3-connected (*rigid*) subgraphs

A decomposition tree of a series-parallel graph is an SPQR-tree without **R**-nodes.
SPQR-Tree

An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by **separation pairs**.

The nodes of $T$ are of four types:
- **S**-nodes represent a series composition
- **P**-nodes represent a parallel composition
- **Q**-nodes represent a single edge
- **R**-nodes represent 3-connected (rigid) subgraphs

A decomposition tree of a series-parallel graph is an SPQR-tree without **R**-nodes.

$T$ represents all planar embeddings of $G$. 
An **SPQR-tree** $T$ is a decomposition of a planar graph $G$ by **separation pairs**.

The nodes of $T$ are of four types:
- **S**-nodes represent a series composition
- **P**-nodes represent a parallel composition
- **Q**-nodes represent a single edge
- **R**-nodes represent 3-connected (**rigid**) subgraphs

A decomposition tree of a series-parallel graph is an SPQR-tree without **R**-nodes.

$T$ represents all planar embeddings of $G$.

$T$ can be computed in $O(n)$ time. [Gutwenger, Mutzel ’01]
SPQR-Tree Example
SPQR-Tree Example

$G$

root

reference edge

Q

14

1

14

1
SPQR-Tree Example

G

P

Q
SPQR-Tree Example
SPQR-Tree Example
SPQR-Tree Example
SPQR-Tree Example
SPQR-Tree Example
SPQR-Tree Example
SPQR-Tree Example
SPQR-Tree Example
SPQR-Tree Example
SPQR-Tree Example
SPQR-Tree Example
Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension

Part IV:
Rectangular Representation Extension

Alexander Wolff
Theorem 1'.
Rectangular $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n^2)$ time for $st$-graphs.
Theorem 1'. 
**Rectangular** $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n^2)$ time for $st$-graphs.
Theorem 1'. Rectangular $\varepsilon$-Bar Visibility Representation Extension can be solved in $\mathcal{O}(n^2)$ time for $st$-graphs.
Theorem 1’.
Rectangular $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n^2)$ time for $st$-graphs.
Theorem 1'.

Rectangular $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n^2)$ time for $st$-graphs.

- Simplify with assumption on y-coordinates

Representation Extension for st-Graphs
Theorem 1'.

Rectangular ε-Bar Visibility Representation Extension can be solved in $O(n^2)$ time for st-graphs.

- Simplify with assumption on y-coordinates
- Look at connection to SPQR-trees – tiling
Representation Extension for st-Graphs

Theorem 1'. Rectangular $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n^2)$ time for $st$-graphs.

- Simplify with assumption on $y$-coordinates
- Look at connection to SPQR-trees – tiling
- Solve problems for $S$-, $P$-, and $R$-nodes
Theorem 1'.
Rectangular $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n^2)$ time for $st$-graphs.

- Simplify with assumption on $y$-coordinates
- Look at connection to SPQR-trees – tiling
- Solve problems for $S$-, $P$-, and $R$-nodes
- Dynamic program via SPQR-tree
y-Coordinate Invariant

Let $G = (V, E)$ be an $st$-graph, and let $\psi'$ be a representation of $V' \subseteq V$. 
y-Coordinate Invariant

- Let $G = (V, E)$ be an $st$-graph, and let $\psi'$ be a representation of $V' \subseteq V$.
- Let $y: V \to \mathbb{R}$ such that

  - for each $v \in V'$, $y(v) = \text{the y-coordinate of } \psi'(v)$.
  - for each edge $(u, v)$, $y(u) < y(v)$.
y-Coordinate Invariant

- Let $G = (V, E)$ be an st-graph, and let $\psi'$ be a representation of $V' \subseteq V$.
- Let $y : V \to \mathbb{R}$ such that
  - for each $v \in V'$, $y(v) =$ the y-coordinate of $\psi'(v)$.
y-Coordinate Invariant

- Let $G = (V, E)$ be an $st$-graph, and let $\psi'$ be a representation of $V' \subseteq V$.
- Let $y: V \to \mathbb{R}$ such that
  - for each $v \in V'$, $y(v) = \text{the y-coordinate of } \psi'(v)$.
  - for each edge $(u, v)$, $y(u) < y(v)$.

Let $G = (V, E)$ be an $st$-graph, and let $\psi'$ be a representation of $V' \subseteq V$.
y-Coordinate Invariant

- Let $G = (V, E)$ be an $st$-graph, and let $\psi'$ be a representation of $V' \subseteq V$.
- Let $y: V \to \mathbb{R}$ such that
  - for each $v \in V'$, $y(v) =$ the y-coordinate of $\psi'(v)$.
  - for each edge $(u, v)$, $y(u) < y(v)$.

**Lemma 1.**

$G$ has a representation extending $\psi'$ $\iff$ $G$ has a representation extending $\psi'$ where the y-coordinates of the bars are as in $y$. 
Let $G = (V, E)$ be an $st$-graph, and let $\psi'$ be a representation of $V' \subseteq V$.

Let $y : V \rightarrow \mathbb{R}$ such that
- for each $v \in V'$, $y(v)$ = the $y$-coordinate of $\psi'(v)$.
- for each edge $(u, v)$, $y(u) < y(v)$.

Lemma 1.
$G$ has a representation extending $\psi'$ $\iff$ $G$ has a representation extending $\psi'$
where the $y$-coordinates of the bars are as in $y$.

Proof idea. The relative positions of adjacent bars must match the order given by $y$.
So, we can adjust the $y$-coordinates of any solution to be as in $y$ by sweeping from bottom to top.
y-Coordinate Invariant

- Let $G = (V, E)$ be an $st$-graph, and let $\psi'$ be a representation of $V' \subseteq V$.
- Let $y: V \rightarrow \mathbb{R}$ such that
  - for each $v \in V'$, $y(v) =$ the y-coordinate of $\psi'(v)$.
  - for each edge $(u, v)$, $y(u) < y(v)$.

Lemma 1.
$G$ has a representation extending $\psi'$ $\iff$
$G$ has a representation extending $\psi'$
where the y-coordinates of the bars are as in $y$.

Proof idea. The relative positions of adjacent bars must match the order given by $y$.
So, we can adjust the y-coordinates of any solution to be as in $y$ by sweeping from bottom to top.
But why do SPQR-Trees help?
But why do SPQR-Trees help?
But why do SPQR-Trees help?
But why do SPQR-Trees help?
But why do SPQR-Trees help?

Lemma 2.
The SPQR-tree of an \(st\)-graph \(G\) induces a recursive tiling of any \(\varepsilon\)-bar visibility representation of \(G\).
But why do SPQR-Trees help?

**Lemma 2.**
The SPQR-tree of an $st$-graph $G$ induces a recursive tiling of any $\varepsilon$-bar visibility representation of $G$. 

Solve tiles bottom-up
Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension

Part V:
Dynamic Program

Alexander Wolff
Tiles

**Convention. Orange** bars are from the partial representation

\[ \psi(t) \]

\[ \psi(s) \]
Tiles

Convention. Orange bars are from the partial representation

\[ \psi(t) \]

\[ \psi(s) \]
Tiles

Con**vention.** Orange bars are from the partial representation

\[ \psi(t) \]

\[ \psi(s) \]

**Observation.**
The bounding box (tile) of any solution \( \psi \) contains the bounding box of the partial representation.
Tiles

Convention. **Orange** bars are from the partial representation

\[ \psi(t) \]

\[ \psi(s) \]

**Observation.**
The bounding box (tile) of any solution \( \psi \) **contains** the bounding box of the partial representation.

How many different **types** of tiles are there?
Types of Tiles

- Right **Fixed** – due to the orange bar
- Left **Loose** – due to the orange bar
Types of Tiles

- **Right Fixed** – due to the orange bar
- **Left Loose** – due to the orange bar

- **Left Fixed** – due to the orange bar
- **Right Loose** – due to the orange bar
Types of Tiles

- Right *Fixed* – due to the orange bar
- Left *Loose* – due to the orange bar

- Left *Fixed* – due to the orange bar
- Right *Loose* – due to the orange bar
Types of Tiles

- Right Fixed – due to the orange bar
- Left Loose – due to the orange bar

- Left Fixed – due to the orange bar
- Right Loose – due to the orange bar
Types of Tiles

- Right Fixed – due to the orange bar
- Left Loose – due to the orange bar
- Left Fixed – due to the orange bar
- Right Loose – due to the orange bar

Four different types: FF, FL, LF, LL
P-Nodes

\[ \psi(t) \]

\[ \psi(s) \]
P-Nodes

\[ \psi(t) \]

\[ \psi(s) \]
$P$-Nodes

$\psi(t)$

$\psi(s)$
P-Nodes

\[ \psi(s) \]

\[ \psi(t) \]
P-Nodes

\[ \psi(t) \]

\[ \psi(s) \]
**P-Nodes**

Children of P-node with **prescribed bars** occur in given left-to-right order
\textbf{P-Nodes}

- Children of \textbf{P}-node with \textit{prescribed bars} occur in given left-to-right order
- But there might be some \textit{gaps}...
**P-Nodes**

- Children of P-node with prescribed bars occur in given left-to-right order.
- But there might be some gaps...

**Idea.**

Greedily fill the gaps by preferring to “stretch” the children with prescribed bars.
Children of $P$-node with prescribed bars occur in given left-to-right order.

But there might be some gaps.

**Idea.**
Greedily fill the gaps by preferring to “stretch” the children with prescribed bars.

**Outcome.**
After processing, we must know the valid types for the corresponding subgraphs.
$S$-Nodes

$\psi(t)$

$\psi(s)$
S-Nodes

\[ \psi(t) \]

\[ \psi(s) \]

This fixed vertex means we can only make a Fixed-Fixed representation!
This fixed vertex means we can only make a Fixed-Fixed representation!
This fixed vertex means we can only make a Fixed-Fixed representation!
S-Nodes

This fixed vertex means we can only make a Fixed-Fixed representation!

Here we have a chance to make all (LL, FL, LF, FF) types.

\[ \psi(s) \] \[ \psi(t) \]
R-Nodes
R-Nodes

$\psi(t)$

$\psi(14)$

$\psi(13)$

$\psi(5)$

$\psi(10)$

$\psi(s)$

$\psi(13)$

$\psi(10)$

$\psi(5)$

$\psi(t)$

$\psi(s)$

$\psi(14)$

$\psi(13)$

$\psi(10)$

$\psi(5)$

$\psi(t)$

$\psi(14)$

$\psi(13)$

$\psi(10)$

$\psi(5)$

$\psi(t)$

$\psi(14)$

$\psi(13)$

$\psi(10)$

$\psi(5)$

$\psi(t)$

$\psi(14)$

$\psi(13)$

$\psi(10)$

$\psi(5)$
R-Nodes
R-Nodes
**R-Nodes**

\[ \psi(t) \]
\[ \psi(14) \]
\[ \psi(5) \]
\[ \psi(10) \]
\[ \psi(s) \]

\[ \psi(13) \]
$\mathcal{R}$-Nodes

\[
\begin{align*}
\psi(s) & \rightarrow \psi(5) & \psi(10) & \rightarrow \psi(13) & \rightarrow \psi(t) \\
& & & & \\
& & & & \\
& & & & \\
& & & & \\
\end{align*}
\]
R-Nodes

\[ \psi(t) \]

\[ \psi(14) \]

\[ \psi(5) \]

\[ \psi(10) \]

\[ \psi(13) \]

\[ \psi(s) \]
R-Nodes

\[ \psi(s) \rightarrow \psi(5) \rightarrow \psi(10) \rightarrow \psi(13) \rightarrow \psi(t) \]
R-Nodes

\[ \psi(t) \]

\[ \psi(14) \]

\[ \psi(5) \]

\[ \psi(10) \]

\[ \psi(13) \]

\[ \psi(s) \]
R-Nodes

ψ(5)  ψ(10)  ψ(13)  ψ(t)

ψ(s)
R-Nodes

ψ(t)ψ(14)

ψ(5)ψ(10)ψ(13)

ψ(s)
R-Nodes

\[ \psi(t) \]

\[ \psi(14) \]

\[ \psi(5) \]

\[ \psi(10) \]

\[ \psi(13) \]

\[ \psi(s) \]
R-Nodes

\[ \psi(s) \]
\[ \psi(5) \]
\[ \psi(10) \]
\[ \psi(13) \]
\[ \psi(14) \]
\[ \psi(t) \]

separation pair!
R-Nodes

ψ(t)
ψ(14)
ψ(5)
ψ(10)
ψ(13)
ψ(s)

separation pair!
**R-Nodes**

- for each child (edge) $e$:

![Diagram showing separation pair!](image-url)
R-Nodes

- for each child (edge) $e$:
  - find all types of $\{\text{FF, FL, LF, LL}\}$ that admit a drawing
  - consistency clauses

\[\psi(s)\]

\[\psi(5)\]

\[\psi(10)\]

\[\psi(13)\]

\[\psi(14)\]

\[\psi(t)\]

separation pair!
R-Nodes with 2-SAT Formulation

- for each child (edge) \( e \):
  - find all types of \{FF, FL, LF, LL\} that admit a drawing
  - 2 variables \( l_e, r_e \) encoding fixed/loose type of its tile
  - consistency clauses
R-Nodes with 2-SAT Formulation

- for each child (edge) \( e \):
  - find all types of \( \{ FF, FL, LF, LL \} \) that admit a drawing
  - 2 variables \( l_e, r_e \) encoding fixed/loose type of its tile

\[ \psi(s) \rightarrow \psi(5) \rightarrow \psi(10) \rightarrow \psi(13) \rightarrow \psi(t) \]

separation pair!
R-Nodes with 2-SAT Formulation

- for each child (edge) $e$:
  - find all types of \{FF, FL, LF, LL\} that admit a drawing
  - 2 variables $l_e, r_e$ encoding fixed/loose type of its tile

\[\psi(s)\]
\[\psi(t)\]

separation pair!
**R-Nodes with 2-SAT Formulation**

- for each child (edge) $e$:
  - find all types of \{FF, FL, LF, LL\} that admit a drawing
  - 2 variables $l_e, r_e$ encoding fixed/loose type of its tile

---

![Graph diagram with nodes and edges labeled with variables and types.](image-url)
R-Nodes with 2-SAT Formulation

- for each child (edge) $e$:
  - find all types of \{FF, FL, LF, LL\} that admit a drawing
  - 2 variables $l_e, r_e$ encoding fixed/loose type of its tile
  - consistency clauses

```plaintext
ψ(s) ψ(5) FL

ψ(10) FF

ψ(13) LL

ψ(14) LL

ψ(t) FL
```

separation pair!
**R-Nodes with 2-SAT Formulation**

- for each child (edge) $e$:
  - find all types of $\{\text{FF, FL, LF, LL}\}$ that admit a drawing
  - 2 variables $l_e, r_e$ encoding fixed/loose type of its tile
  - consistency clauses
R-Nodes with 2-SAT Formulation

- for each child (edge) $e$:
  - find all types of $\{\text{FF, FL, LF, LL}\}$ that admit a drawing
  - 2 variables $l_e, r_e$ encoding fixed/loose type of its tile
  - consistency clauses

separation pair!
**R-Nodes with 2-SAT Formulation**

- for each child (edge) $e$:
  - find all types of $\{\text{FF, FL, LF, LL}\}$ that admit a drawing
  - 2 variables $l_e, r_e$ encoding fixed/loose type of its tile
  - consistency clauses \(-O(n^2)\) many,
**R-Nodes with 2-SAT Formulation**

- for each child (edge) \( e \):
  - find all types of \( \{\text{FF,FL,LF,LL}\} \) that admit a drawing
  - 2 variables \( l_e, r_e \) encoding fixed/loose type of its tile
  - consistency clauses \( - O(n^2) \) many, but can be reduced to \( O(n \log^2 n) \)
Visualization of Graphs

Lecture 9:
Partial Visibility Representation Extension

Part VI:
NP-Hardness
of the General Case

Alexander Wolff
NP-Hardness of RepExt in the General Case

Theorem 2.
\( \varepsilon \)-Bar Visibility Representation Extension is NP-complete.

- Reduction from Planar Monotone 3-SAT
NP-Hardness of RepExt in the General Case

Theorem 2.
\( \varepsilon \)-Bar Visibility Representation Extension is NP-complete.

Reduction from Planar Monotone 3-SAT
NP-Hardness of RepExt in the General Case

**Theorem 2.**
$\varepsilon$-Bar Visibility Representation Extension is NP-complete.

- Reduction from Planar Monotone 3-SAT

Diagram:

```
x1 \lor x3 \lor x6
```

- NP-complete
  [Berg & Khosravi ’10]
NP-Hardness of RepExt in the General Case

Theorem 2.
\( \varepsilon \)-Bar Visibility Representation Extension is NP-complete.

- Reduction from Planar Monotone 3-SAT

NP-complete

[Berg & Khosravi '10]
Variable Gadget
Variable Gadget
Variable Gadget
Variable Gadget
Variable Gadget
Variable Gadget
Variable Gadget
Variable Gadget
Variable Gadget
Variable Gadget

$x = \text{FALSE}$

$x = \text{TRUE}$
Clause Gadget

\[ x \lor y \lor z \]
Clause Gadget

\[ x \lor y \lor z \]
Clause Gadget

$x \lor y \lor z$
Clause Gadget

\[ x \lor y \lor z \]
Clause Gadget

\[ x \lor y \lor z \]
Clause Gadget

\[ x \lor y \lor z \]

\[ x \lor y = \text{TRUE} \]
Clause Gadget

\[ x \lor y \lor z \]

\begin{align*}
\text{OR} & \quad x \lor y = \text{TRUE} \\
& \quad x \lor y = \text{FALSE}
\end{align*}
Clause Gadget

\[ x \lor y \lor z \]

\[ x \lor y = \text{TRUE} \]

\[ x \lor y = \text{FALSE} \]
Clause Gadget

\[ x \lor y \lor z \]

\[ x \lor y = \text{False} \]
\[ x \lor y = \text{True} \]
\[ x \lor y \lor z = \text{True} \]
Clause Gadget

$x \lor y \lor z$

$x \lor y = \text{True}$

$x \lor y = \text{False}$

$x \lor y \lor z = \text{True}$

$x \lor y \lor z = \text{False}$
Clause Gadget

\[ x \lor y \lor z \]

\[ x \lor y = \text{True} \]

\[ x \lor y = \text{False} \]

\[ x \lor y \lor z = \text{True} \]

\[ x \lor y \lor z = \text{False} \]
Clause Gadget

$x \lor y \lor z$

$x \lor y = \text{TRUE}$

$x \lor y = \text{FALSE}$

$x \lor y \lor z = \text{TRUE}$

$x \lor y \lor z = \text{FALSE}$
Clause Gadget

\[ x \lor y \lor z \]

\[ x \lor y = \text{True} \]

\[ x \lor y = \text{False} \]

\[ x \lor y = \text{or True} \]

or \[ \text{True} \]

\[ x \lor y \lor z = \text{True} \]

\[ x \lor y \lor z = \text{False} \]
Clause Gadget

\[ x \lor y \lor z \]

\[ x \lor y = \text{TRUE} \]

\[ x \lor y = \text{FALSE} \] or TRUE

\[ x \lor y \lor z = \text{TRUE} \] or TRUE

\[ x \lor y \lor z = \text{FALSE} \] or TRUE
OR’ Gadget

\[ x \]

\[ y \]
OR’ Gadget

\begin{figure}
\begin{center}
\begin{tikzpicture}

% Nodes
\node[shape=circle,fill=orange!20] (x) at (0,0) {$x$};
\node[shape=circle,fill=orange!20] (y) at (0,-2) {$y$};
\node[shape=rectangle,fill=blue!20] (x_bar) at (3,0) {$x$};
\node[shape=rectangle,fill=blue!20] (y_bar) at (3,-2) {$y$};

% Connections
\draw (x) -- (y);
\draw (x) -- (x_bar);
\draw (y) -- (y_bar);
\draw (x) -- (y_bar);
\draw (y) -- (x_bar);
\end{tikzpicture}
\end{center}
\end{figure}
**OR' Gadget**

The diagram illustrates an OR' gadget with input variables $x$ and $y$. The gadget is composed of a network of interconnections, with $x$ and $y$ as inputs, leading to the final output in a cascade manner. The structure ensures the correct functionality of the OR' operation, as depicted by the flow of connections and the labeled nodes.
OR’ Gadget

\[ x \]

\[ y \]
OR' Gadget
OR’ Gadget
OR’ Gadget

$x$

$y$

---

$x$

$y$
OR’ Gadget

\[ x \quad y \]
OR’ Gadget

\[ x \quad y \]

\[ (\quad ) \quad (\quad ) \quad (\quad ) \]

\[ (\quad ) \quad (\quad ) \quad (\quad ) \]

\[ (\quad ) \quad (\quad ) \quad (\quad ) \]

\[ (\quad ) \quad (\quad ) \quad (\quad ) \]

\[ (\quad ) \quad (\quad ) \quad (\quad ) \]
OR’ Gadget
OR’ Gadget

Diagram showing the connections and nodes labeled as ‘x’ and ‘y’.
OR’ Gadget
OR’ Gadget

\[ x \]

\[ y \]
OR’ Gadget
OR’ Gadget
Discussion

- Rectangular $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for $st$-graphs.
Discussion

- *Rectangular* $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for $st$-graphs.

- $\varepsilon$-Bar Visibility Representation Extension is NP-complete.
Discussion

- *Rectangular* $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for $st$-graphs.

- $\varepsilon$-Bar Visibility Representation Extension is NP-complete.

- $\varepsilon$-Bar Visibility Representation Extension is NP-complete for (series-parallel) $st$-graphs when restricted to the *Integer Grid* (or if any fixed $\varepsilon > 0$ is specified).
Discussion

- **Rectangular ε-Bar Visibility Representation Extension** can be solved in $O(n \log^2 n)$ time for st-graphs.

- ε-Bar Visibility Representation Extension is NP-complete.

- ε-Bar Visibility Representation Extension is NP-complete for (series-parallel) st-graphs when restricted to the Integer Grid (or if any fixed $\varepsilon > 0$ is specified).

Open Problems:

- Can **rectangular** ε-Bar Visibility Representation Extension be solved in polynomial time for st-graphs?
Discussion

- *Rectangular* $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for $st$-graphs.

- $\varepsilon$-Bar Visibility Representation Extension is NP-complete.

- $\varepsilon$-Bar Visibility Representation Extension is NP-complete for (series-parallel) $st$-graphs when restricted to the *Integer Grid* (or if any fixed $\varepsilon > 0$ is specified).

Open Problems:

- Can *rectangular* $\varepsilon$-Bar Visibility Representation Extension be solved in polynomial time for $st$-graphs? For DAGs?
Discussion

- **Rectangular** $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for $st$-graphs.

- $\varepsilon$-Bar Visibility Representation Extension is NP-complete.

- $\varepsilon$-Bar Visibility Representation Extension is NP-complete for (series-parallel) $st$-graphs when restricted to the *Integer Grid* (or if any fixed $\varepsilon > 0$ is specified).

Open Problems:

- Can **rectangular** $\varepsilon$-Bar Visibility Representation Extension be solved in polynomial time for $st$-graphs? For DAGs?

- Can **Strong** Bar Visibility Recognition / Representation Extension can be solved in polynomial time for $st$-graphs?
Literature

Main source:
- [Chaplick, Guśpiel, Gutowski, Krawczyk, Liotta ’18]
  The Partial Visibility Representation Extension Problem

Referenced papers:
- [Gutwenger, Mutzel ’01] A Linear Time Implementation of SPQR-Trees
- [Wismath ’85] Characterizing bar line-of-sight graphs
- [Tamassia, Tollis ’86] Algorithms for visibility representations of planar graphs
- [Andreae ’92] Some results on visibility graphs
- [Chaplick, Dorbec, Kratchovíl, Montassier, Stacho ’14]
  Contact representations of planar graphs: Extending a partial representation is hard