Algorithms for Graph Visualization

Summer Semester 2019
Lecture #11

Partial Visibility Representation Extension

see also: https://arxiv.org/abs/1512.00174
Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)
Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)
Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)

Edges correspond to vertical unobstructed vertical sightlines
Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)

Edges correspond to vertical unobstructed vertical sightlines
Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)
Edges correspond to vertical unobstructed vertical sightlines

Models:
Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)

Edges correspond to vertical unobstructed vertical sightlines

Models:

**Strong:** edge $uv \iff$ unobstructed (0-width) vertical sightline
Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)
Edges correspond to vertical unobstructed vertical sightlines

Models:

Strong: edge $uv \iff$ unobstructed (0-width) vertical sightline

$\varepsilon$: edge $uv \iff \varepsilon$ wide sight-line for $\varepsilon > 0$
Bar Visibility Representations

Vertices correspond to horizontal open line segments (bars)
Edges correspond to vertical unobstructed vertical sightlines

Models:
Strong: edge $uv \iff$ unobstructed (0-width) vertical sightline
$\varepsilon$: edge $uv \iff \varepsilon$ wide sight-line for $\varepsilon > 0$
Weak: edge $uv \Rightarrow$ unobstructed sightline
i.e., any subset of visible pairs
Problems

Recognition:
Given a graph $G$, **decide** if there exists a weak/strong/ε bar visibility representation $\psi$ of $G$. 
Problems

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

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

Recognition:
Given a graph $G$, decide if there exists a weak/strong/$\varepsilon$ bar visibility representation $\psi$ of $G$.

Construction:
Given a graph $G$, construct a weak/strong/$\varepsilon$ bar visibility representation $\psi$ of $G$ when one exists.

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

\[ a \rightarrow b \quad c \rightarrow b \quad d \rightarrow d \]

\[ b \rightarrow c \quad a \rightarrow a \]

\[ d \rightarrow b \quad c \rightarrow a \]

\[ \varepsilon \]

Weak

Strong
Weak Bar Visibility

- All planar graphs. [Tammasia & Tollis 1986; Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- Representation Extension is NP-complete [C., Dorbec, Kratochvíl, Montassier, Stacho 2014]
Background

Weak Bar Visibility
- All planar graphs. [Tammasia & Tollis 1986; Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- Representation Extension is NP-complete [C., Dorbec, Kratochvíl, Montassier, Stacho 2014]

Strong Bar Visibility
- NP-complete to recognize [Andreae 1992]
Background

Weak Bar Visibility
- All planar graphs. [Tammasia & Tollis 1986; Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- Representation Extension is NP-complete [C., Dorbec, Kratochvíl, Montassier, Stacho 2014]

Strong Bar Visibility
- NP-complete to recognize [Andreae 1992]

ε-Bar Visibility
- Planar graphs that can be embedded with all cut vertices on the outerface. [T&T 1986, Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- What about Representation Extension?
Background

Weak Bar Visibility
- All planar graphs. [Tammasia & Tollis 1986; Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- Representation Extension is NP-complete [C., Dorbec, Kratochvıl, Montassier, Stacho 2014]

Strong Bar Visibility
- NP-complete to recognize [Andreae 1992]

$\varepsilon$-Bar Visibility
- Planar graphs that can be embedded with all cut vertices on the outerface. [T&T 1986, Wismath 1985]
- Linear time recognition and construction [T&T 1986]
- What about Representation Extension? Let’s see!

Let's see!
First another definition ..... planar digraphs

**Planar st-graphs**: planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the same face (i.e., the outerface) of an embedding of $G$. 
First another definition ..... planar digraphs

**Planar $st$-graphs**: planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the same face (i.e., the outerface) of an embedding of $G$. 
First another definition ..... planar digraphs

**Planar st-graphs**: planar digraph $G$ with exactly one **source** $s$ and one **sink** $t$ where $s$ and $t$ occur on the same **face** (i.e., the **outerface**) of an embedding of $G$.

**$st$-orientations correspond to $\varepsilon$-Bar Visibility Representations**
First another definition ..... planar digraphs

Planar \textit{st}-graphs: planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the same face (i.e., the outerface) of an embedding of $G$.

\textit{st}-orientations correspond to $\varepsilon$-Bar Visibility Representations
First another definition ..... planar digraphs

**Planar st-graphs:** planar digraph $G$ with exactly one source $s$ and one sink $t$ where $s$ and $t$ occur on the same face (i.e., the outerface) of an embedding of $G$.

**st-orientations correspond to ε-Bar Visibility Representations**

Note: unlike for undirected planar graphs, testing whether a directed acyclic planar graph has a **Weak Bar Visibility representation** is NP-complete → this is **upward planarity testing** [Garg & Tamassia 2001].
First another definition ..... planar digraphs

**Planar \(st\)-graphs**: planar digraph \(G\) with exactly one source \(s\) and one sink \(t\) where \(s\) and \(t\) occur on the same face (i.e., the outerface) of an embedding of \(G\).

**\(st\)-orientations correspond to \(\varepsilon\)-Bar Visibility Representations**

Note: unlike for undirected planar graphs, testing whether a directed acyclic planar graph has a **Weak Bar Visibility representation** is NP-complete → this is **upward planarity testing** [Garg & Tamassia 2001].

\(\varepsilon\)-Bar Visibility testing is easily done via \(st\)-graph recognition.
First another definition ..... planar digraphs

**Planar \(st\)-graphs**: planar digraph \(G\) with exactly one **source** \(s\) and one **sink** \(t\) where \(s\) and \(t\) occur on the same **face** (i.e., the **outerface**) of an embedding of \(G\).

**\(st\)-orientations correspond to \(\varepsilon\)-Bar Visibility Representations**

Note: unlike for undirected planar graphs, testing whether a directed acyclic planar graph has a **Weak Bar Visibility representation** is NP-complete → this is **upward planarity testing** ([Garg & Tamassia 2001]).

\(\varepsilon\)-Bar Visability testing is easily done via \(st\)-graph recognition.

Strong Bar Visibility recognition... open?
Results

**rectangular** $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for $st$-graphs via **Dynamic Program: via SPQR-trees**
**Results**

\textit{rectangular} $\varepsilon$-Bar Visibility Representation Extension can be solved in $O(n \log^2 n)$ time for \textit{st}-graphs

- Dynamic Program: via \textit{SPQR-trees}

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

- Reduction: \textit{planar monotone 3-SAT}
**Results**

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

- Dynamic Program: via *SPQR-trees*

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

- Reduction: *planar monotone 3-SAT*

$\varepsilon$-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: *3-partition*
Results

\textit{rectangular} \(\varepsilon\)-Bar Visibility Representation Extension can be solved in \(O(n \log^2 n)\) time for \(st\)-graphs

- Dynamic Program: via \textit{SPQR-trees}
  - (easier version: \(O(n^2)\))

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

- Reduction: \textit{planar monotone 3-SAT}

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

- Reduction: \textit{3-partition}
Results

ε-Bar Visibility Representation Extension is NP-complete

Reduction: planar monotone 3-SAT

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

Dynamic Program: via $SPQR$-trees

(easier version: $O(n^2)$)

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).

Reduction: 3-partition
$st$-graphs : $\varepsilon$-Bar Visibility Representation Extension

Simplifying our life a little: $y$-coordinate invariant
st-graphs: $\varepsilon$-Bar Visibility Representation Extension

Simplifying our life a little: \textit{y-coordinate invariant}

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

- for each $v \in V'$, $y_v =$ the $y$-coordinate of $\psi'(v)$.
- for each $u \rightarrow v$, $y_u < y_v$.


Simplifying our life a little: \textit{y-coordinate invariant}

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

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

\textbf{Lemma:} $G$ has a representation extending $\psi'$ if and only if $G$ has a representation $\psi$ extending $\psi'$ where the $y$-coordinates of the bars are as in $y$. 
Simplifying our life a little: \textit{y-coordinate invariant}

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

- for each $v \in V'$, $y_v$ = the \textit{y}-coordinate of $\psi'(v)$.
- for each $u \rightarrow v$, $y_u < y_v$.

**Lemma:** $G$ has a representation extending $\psi'$ if and only if $G$ has a representation $\psi$ extending $\psi'$ where the \textit{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 \textit{y}-coordinates of any solution to be as in $y$ by sweeping from bottom-to-top. ■
**st-graphs: ε-Bar Visibility Representation Extension**

We can now assume all y-coordinates are given!

Simplifying our life a little: **y-coordinate invariant**

Let $G$ be an st-graph, and $\psi'$ be a representation of $V' \subseteq V(G)$.

Let $y_v : V(G) \rightarrow \mathbb{R}$ such that

- for each $v \in V'$, $y_v$ = the y-coordinate of $\psi'(v)$.
- for each $u \rightarrow v$, $y_u < y_v$.

**Lemma:** $G$ has a representation extending $\psi'$ if and only if $G$ has a representation $\psi$ 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. ■

Simplifying our life a little:

We can now assume all y-coordinates are given!
SPQR-trees

SPQR-tree: decomposition of a planar graph by *separation pairs*.
SPQR-trees

SPQR-tree: decomposition of a planar graph by *separation pairs*.
SPQR-tree: decomposition of a planar graph by *separation pairs*.
SPQR-trees

SPQR-tree: decomposition of a planar graph by *separation pairs*. 
SPQR-trees

SPQR-tree: decomposition of a planar graph by separation pairs.
SPQR-trees

SPQR-tree: decomposition of a planar graph by separation pairs.
But why do SQPR-trees help?
But why do SQPR-trees help?
But why do SQPR-trees help?
But why do SQPR-trees help?

**Lemma:** The SPQR-tree of an \( st \)-graph \( G \) induces a recursive \textit{tiling} of any \( \varepsilon \)-Bar Visibility Representation of \( G \).
Tiles

Note: orange bars are from the partial representation
Tiles

Note: orange bars are from the partial representation

$\psi(t)$

$\psi(s)$

**Obs:** the bounding box (tile) of any solution $\psi$, contains the bounding box of the partial rep.

How many **different** tiles can we really have?
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

\[ \psi(s) \]

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

\[ \psi(t) \]

\[ \psi(s) \]
Children with prescribed bars occur in given left-to-right order. But there will be some gaps.
Children with prescribed bars occur in given left-to-right order.
But there will be some gaps.

Idea: greedily fill the gaps by preferring to “stretch” the children with prescribed bars.
P-nodes

Children with prescribed bars occur in given left-to-right order. But there will 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 subgraph.
S-nodes

\[ \psi(t) \]

\[ \psi(s) \]
This fixed vertex means we can only make a Fixed-Fixed representation!
S-nodes

\[ \psi(t) \]

\[ \psi(s) \]

This fixed vertex means we can only make a Fixed-Fixed representation!
Now, we have a chance to make all $(LL, FL, LF, FF)$ types
S-nodes

Now, we have a chance to make all (LL, FL, LF, FF) types. How does this work?
R-nodes
R-nodes

2-SAT formulation:
R-nodes 2-SAT formulation:

- 2 variables for each child: encoding fixed/loose state of its tile.
- 2 variables for each face: encoding position of the splitting line.
R-nodes

2-SAT formulation:

- 2 variables for each child: encoding fixed/loose state of its tile.
- 2 variables for each face: encoding position of the splitting line.
- Restriction clauses for each child to subset of \{FF, FL, LF, LL\}
R-nodes

2-SAT formulation:

- 2 variables for each child: encoding fixed/loose state of its tile.
- 2 variables for each face: encoding position of the splitting line.
- restriction clauses for each child to subset of \{FF,FL,LF,LL\}
- consistency clauses for each face.

\[
\begin{align*}
\psi(t) & \quad \chi(f_5) \\
\psi(14) & \quad \chi(f_3) \\
\psi(13) & \quad \chi(f_4) \\
\psi(s) & \quad \chi(t^*)
\end{align*}
\]
R-nodes

2-SAT formulation:

- 2 variables for each child: encoding fixed/loose state of its tile.
- 2 variables for each face: encoding position of the splitting line.
- restriction clauses for each child to subset of \{FF, FL, LF, LL\}
- consistency clauses for each face.
- ordering clauses.
### R-nodes

#### 2-SAT formulation:

- 2 variables for each child: encoding fixed/loose state of its tile.
- 2 variables for each face: encoding position of the splitting line.
- Restriction clauses for each child to subset of \{FF, FL, LF, LL\}
- Consistency clauses for each face.
- Ordering clauses. \textbf{Quadratically many}

\[
\begin{align*}
\psi(t) & \quad \psi(14) & \quad \chi(f_5) \\
\chi(f_1) & \quad \psi(13) & \quad \psi(10) \\
\chi(f_3) & \quad \chi(f_4) & \quad \chi(t^*) \\
\chi(s^*) & \quad \psi(s) & \quad \psi(s) \\
\end{align*}
\]
R-nodes

2-SAT formulation:

- 2 variables for each child: encoding fixed/loose state of its tile.
- 2 variables for each face: encoding position of the splitting line.
- restriction clauses for each child to subset of \{FF,FL,LF,LL\}
- consistency clauses for each face.
- ordering clauses.

Quadratically many

tricky part: use only \(O(n \log^2 n)\) clauses
Hardness Results

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

\[ \text{Reduction: planar monotone 3-SAT} \]

\( \varepsilon \)-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).

\[ \text{Reduction: 3-partition} \]
NP-hardness of Representation Extension

Planar Monotone 3-SAT

NP-complete [Berg and Khosravi 2010]
NP-hardness of Representation Extension
NP-hardness of Representation Extension

Wire Transmission

\[
\begin{align*}
\overline{x_1} \lor \overline{x_4} \lor \overline{x_6} \\
\overline{x_1} \lor \overline{x_2} \lor \overline{x_3} \\
\overline{x_2} \lor \overline{x_3} \lor \overline{x_4} \\
\overline{x_4} \lor \overline{x_5} \lor \overline{x_6} \\
\overline{x_5} \lor \overline{x_6} \\
\overline{x_6} \\
\end{align*}
\]
NP-hardness of Representation Extension

Note: the following details omit the copying gadgets used for multiple occurrences of the variables
NP-hardness of Representation Extension

NOT gate
NP-hardness of Representation Extension

Note: the bars of $x$ and $y$ cannot occur between $a$ and $b$ since $a$ and $b$ are not supposed to be adjacent either of $\bot$ and $\top$
NP-hardness of Representation Extension

OR gate
NP-hardness of Representation Extension

Subtle point: only need to guarantee that “false” values transmit

OR gate
NP-hardness of Representation Extension

subtle point: only need to guarantee that “false” values transmit

OR gate
NP-hardness of Representation Extension
NP-hardness on the Integer Grid (or fixed $\varepsilon$)

**Problem:** Representation extension in the Integer Grid.
NP-hardness on the Integer Grid (or fixed $\varepsilon$)

Problem: Representation extension in the Integer Grid.

3-Partition

Input: A set of positive integers $w, a_1, a_2, \ldots, a_{3m}$ such that for each $i = 1, \ldots, 3m$, we have $\frac{w}{4} < a_i < \frac{w}{2}$.

Question: Can \{$a_1, \ldots, a_{3m}$\} be partitioned into $m$ triples, such that the total sum of each triple is exactly $w$?

Strongly NP-complete [Garey Johnson 1979]
Problem: Representation extension in the Integer Grid.

3-Partition

Input: A set of positive integers $w, a_1, a_2, \ldots, a_{3m}$ such that for each $i = 1, \ldots, 3m$, we have $\frac{w}{4} < a_i < \frac{w}{2}$.

Question: Can $\{a_1, \ldots, a_{3m}\}$ be partitioned into $m$ triples, such that the total sum of each triple is exactly $w$?

\[ a_i \rightarrow \]
NP-hardness on the Integer Grid (or fixed $\varepsilon$)

**Problem:** Representation extension in the Integer Grid.

**3-Partition**

**Input:** A set of positive integers $w, a_1, a_2, \ldots, a_{3m}$ such that for each $i = 1, \ldots, 3m$, we have $\frac{w}{4} < a_i < \frac{w}{2}$.

**Question:** Can \{a_1, \ldots, a_{3m}\} be partitioned into $m$ triples, such that the total sum of each triple is exactly $w$?

\[ a_i \rightarrow \psi(s_i) \rightarrow a_i \rightarrow \psi(t_i) \rightarrow a_i \rightarrow H_i \]
NP-hardness on the Integer Grid (or fixed $\varepsilon$)

**Problem:** Representation extension in the Integer Grid.

**3-Partition**

**Input:** A set of positive integers $w, a_1, a_2, \ldots, a_{3m}$ such that for each $i = 1, \ldots, 3m$, we have $\frac{w}{4} < a_i < \frac{w}{2}$.

**Question:** Can $\{a_1, \ldots, a_{3m}\}$ be partitioned into $m$ triples, such that the total sum of each triple is exactly $w$?

$$a_i \rightarrow H_i$$

$$u_0 \rightarrow u_1 \rightarrow \cdots \rightarrow u_m$$

$$s \rightarrow s_1 \rightarrow \cdots \rightarrow s_{3m}$$

$$t \rightarrow t_1 \rightarrow t_2 \rightarrow \cdots \rightarrow t_{3m}$$
Conclusion

• *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).
Conclusion

- **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 can be solved in polynomial time on $st$-graphs? DAGs?

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