Bitonic *st*-orderings for Upward Planar Graphs

Martin Gronemann

University of Cologne, Germany

September 20, 2016

Introduction

Graph Drawing 2014

- **Bitonic st-orderings** of biconnected planar graphs.
- A special *st*-ordering that is similar to canonical orderings.
- Works with the algorithm of de Fraysseix, Pach and Pollack.

Introduction

Graph Drawing 2014

- **Bitonic st-orderings** of biconnected planar graphs.
- A special *st*-ordering that is similar to canonical orderings.
- Works with the algorithm of de Fraysseix, Pach and Pollack.

Here today

- Apply the idea to directed graphs, esp. **planar st-graphs**.
- Use it to create **upward planar** straight-line drawings.

Introduction

Planar *st*-graphs

- Planar directed acyclic graph $G = (V, E)$ with
- a single source $s \in V$ and single sink $t \in V$.
- G has a fixed embedding with s and t on the outer face.

Introduction

Planar *st*-graphs

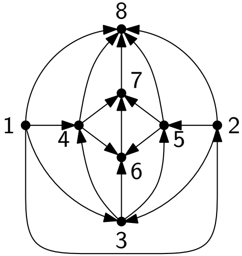
- Planar directed acyclic graph $G = (V, E)$ with
- a single source $s \in V$ and single sink $t \in V$.
- G has a fixed embedding with s and t on the outer face.

Important results on upward planar graphs

- Every upward planar graph is a spanning subgraph of a planar *st*-graph (Di Battista & Tamassia).
  $\Rightarrow$ We focus on planar *st*-graphs.
- Every upward planar graph admits an upward planar straight-line drawing (Di Battista & Tamassia).
- Some require exponential area (Di Battista, Tamassia, Tollis).

Idea

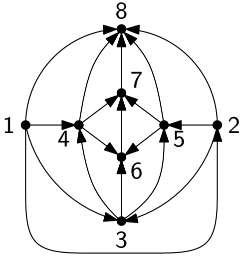
Use the algorithm of de Fraysseix, Pach and Pollack to obtain upward planar straight-line drawings for planar *st*-graphs.

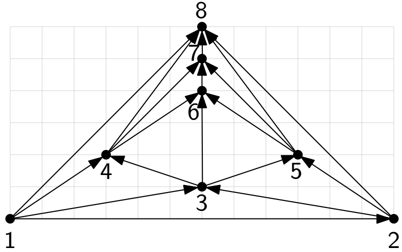


Incremental upward planar straight-line drawings

Idea

Use the algorithm of de Fraysseix, Pach and Pollack to obtain upward planar straight-line drawings for planar *st*-graphs.

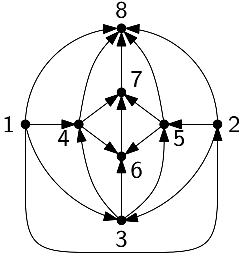


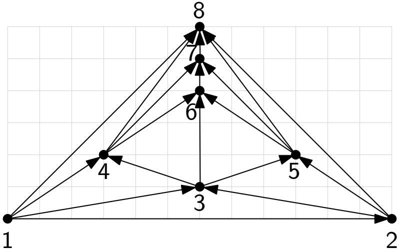


Incremental upward planar straight-line drawings

Idea

Use the algorithm of de Fraysseix, Pach and Pollack to obtain upward planar straight-line drawings for planar *st*-graphs.



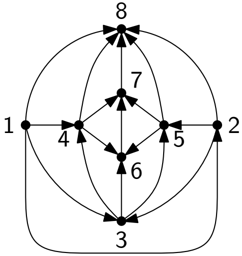


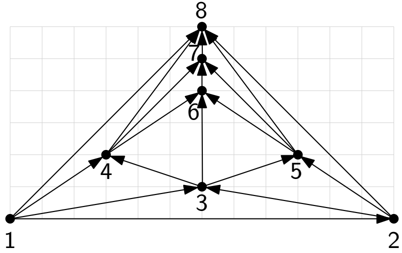
- If the (canonical) ordering complies with the orientation of the edges, the result is upward planar straight-line.

Incremental upward planar straight-line drawings

Idea

Use the algorithm of de Fraysseix, Pach and Pollack to obtain upward planar straight-line drawings for planar *st*-graphs.



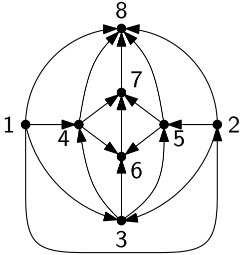


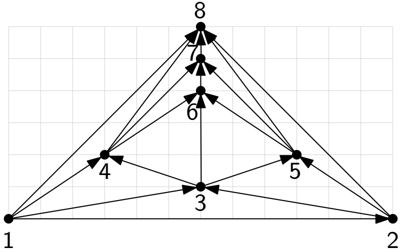
- If the (canonical) ordering complies with the orientation of the edges, the result is upward planar straight-line.
- Canonical orderings do not extend to directed graphs, . . .

Incremental upward planar straight-line drawings

Idea

Use the algorithm of de Fraysseix, Pach and Pollack to obtain upward planar straight-line drawings for planar *st*-graphs.



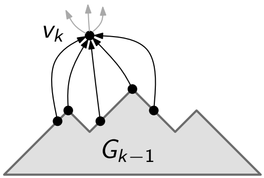


- If the (canonical) ordering complies with the orientation of the edges, the result is upward planar straight-line.
- Canonical orderings do not extend to directed graphs, ...
- ... but *st*-orderings do $\Rightarrow$ We use an *st*-ordering instead!

Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.

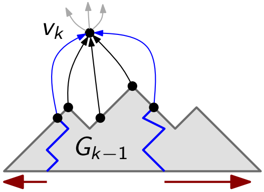


≥ 2 predecessors

Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.

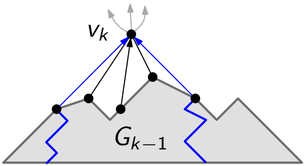


≥ 2 predecessors

Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.

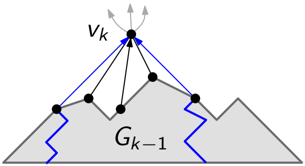


≥ 2 predecessors

Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.



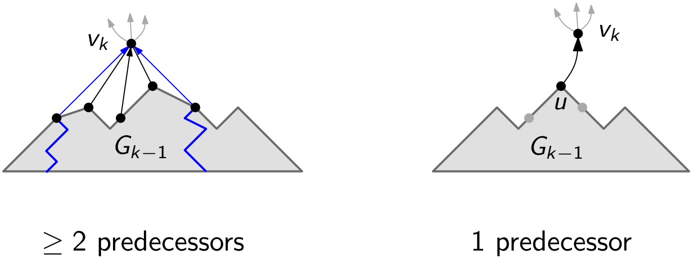
≥ 2 predecessors

- Multiple-predecessor case works like in the original algorithm.

Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.

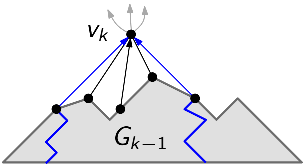


- Multiple-predecessor case works like in the original algorithm.

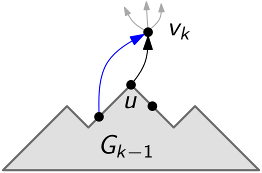
Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.



≥ 2 predecessors



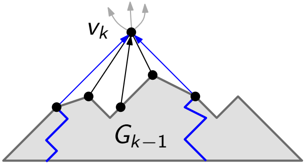
1 predecessor

- Multiple-predecessor case works like in the original algorithm.

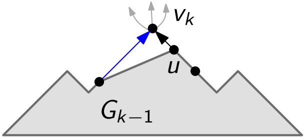
Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.



≥ 2 predecessors



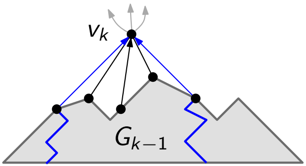
1 predecessor

- Multiple-predecessor case works like in the original algorithm.

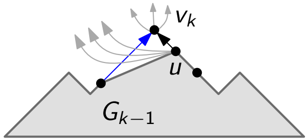
Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.



≥ 2 predecessors



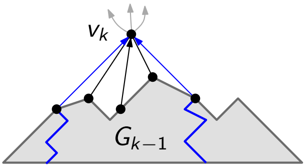
1 predecessor

- Multiple-predecessor case works like in the original algorithm.

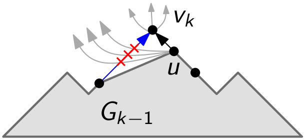
Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.



≥ 2 predecessors



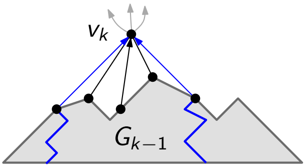
1 predecessor

- Multiple-predecessor case works like in the original algorithm.

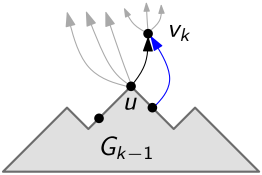
Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.



≥ 2 predecessors



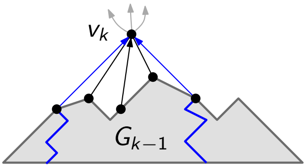
1 predecessor

- Multiple-predecessor case works like in the original algorithm.

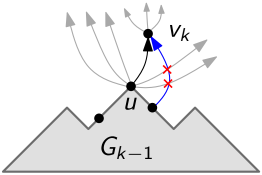
Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.



≥ 2 predecessors



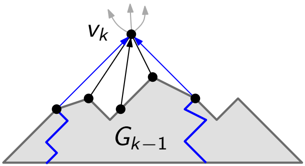
1 predecessor

- Multiple-predecessor case works like in the original algorithm.

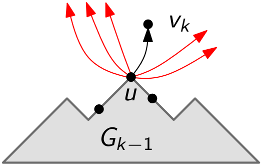
Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.



≥ 2 predecessors



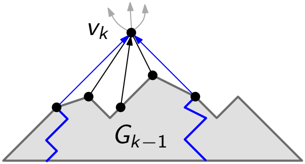
1 predecessor

- Multiple-predecessor case works like in the original algorithm.

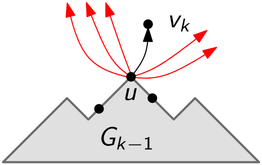
Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.



≥ 2 predecessors



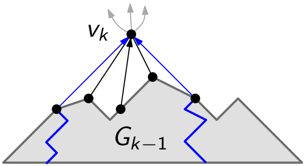
1 predecessor

- Multiple-predecessor case works like in the original algorithm.
- One-predecessor case works only when the unattached edges are all on one side (either left or right of (u, v_k)).

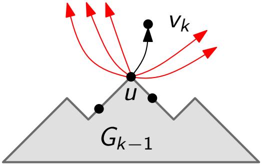
Incremental upward planar straight-line drawings

Problem

Running the FPP-algorithm with an *st*-ordering does not work.



≥ 2 predecessors



1 predecessor

- Multiple-predecessor case works like in the original algorithm.
- One-predecessor case works only when the unattached edges are all on one side (either left or right of (u, v_k)).
- They are **consecutive** $\Rightarrow$ they are **all on one side**

Canonical vs. *st*-orderings

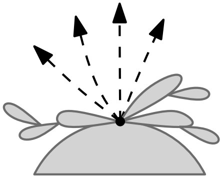
Intuition

At any time, all outgoing edges that are **not yet present** in the current drawing, appear **consecutively** in the embedding.

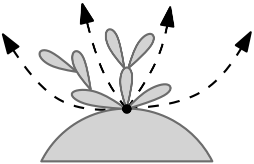
Canonical vs. *st*-orderings

Intuition

At any time, all outgoing edges that are **not yet present** in the current drawing, appear **consecutively** in the embedding.



canonical ordering

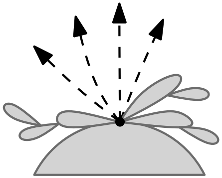


st-ordering

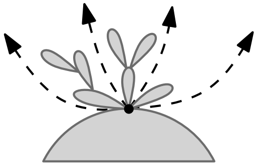
Canonical vs. *st*-orderings

Intuition

At any time, all outgoing edges that are **not yet present** in the current drawing, appear **consecutively** in the embedding.



canonical ordering



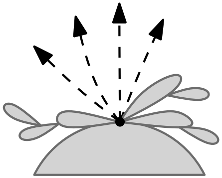
st-ordering

- We need a special *st*-ordering in which they are consecutive like in a canonical ordering!

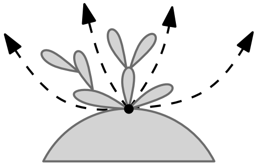
Canonical vs. *st*-orderings

Intuition

At any time, all outgoing edges that are **not yet present** in the current drawing, appear **consecutively** in the embedding.



canonical ordering



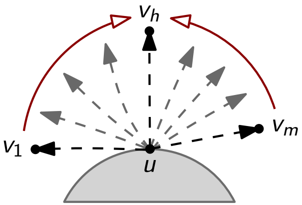
st-ordering

- We need a special *st*-ordering in which they are consecutive like in a canonical ordering! $\rightarrow$ **bitonic st-ordering**

Bitonic *st*-orderings

Definition

Given an *st*-ordering $\pi : V \mapsto \{1, \ldots, |V|\}$ with $\pi(v)$ being the rank of $v \in V$ in the ordering.



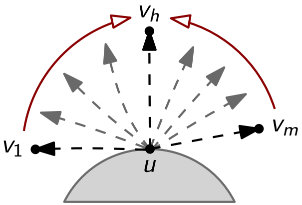
$$S(u) = \{v_1, \ldots, v_m\}$$

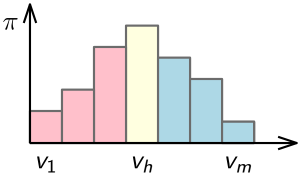
Successors of u ordered as in the embedding

Bitonic *st*-orderings

Definition

Given an *st*-ordering $\pi : V \mapsto \{1, \dots, |V|\}$ with $\pi(v)$ being the rank of $v \in V$ in the ordering.





$S(u) = \{v_1, \dots, v_m\}$

Successors of u ordered as in the embedding

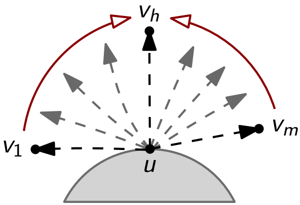
$\pi(v_1) < \dots < \pi(v_h) > \dots > \pi(v_m)$

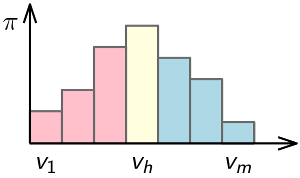
An increasing and then decreasing sequence ⇒ **bitonic**

Bitonic *st*-orderings

Definition

Given an *st*-ordering $\pi : V \mapsto \{1, \ldots, |V|\}$ with $\pi(v)$ being the rank of $v \in V$ in the ordering.





$$S(u) = \{v_1, \ldots, v_m\}$$

Successors of u ordered as in the embedding

$$\pi(v_1) < \cdots < \pi(v_h) > \cdots > \pi(v_m)$$

An increasing and then decreasing sequence $\Rightarrow$ **bitonic**

$\forall u \in V : S(u)$ is bitonic w.r.t. $\pi \Rightarrow \pi$ is a **bitonic** *st*-ordering

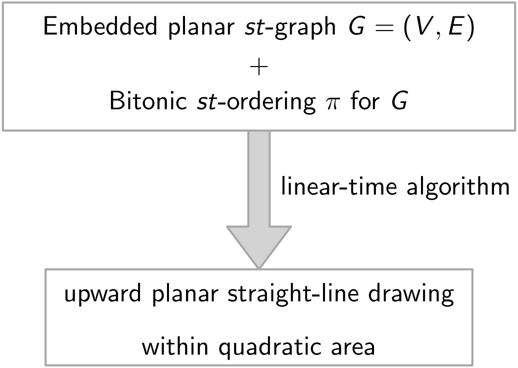
Bitonic *st*-orderings - Results

Embedded planar *st*-graph $G = (V, E)$

$+$

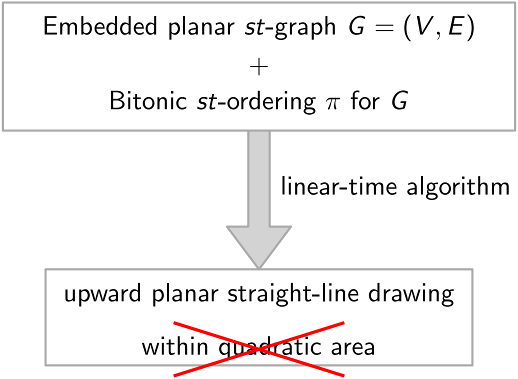
Bitonic *st*-ordering π for G

Bitonic *st*-orderings - Results



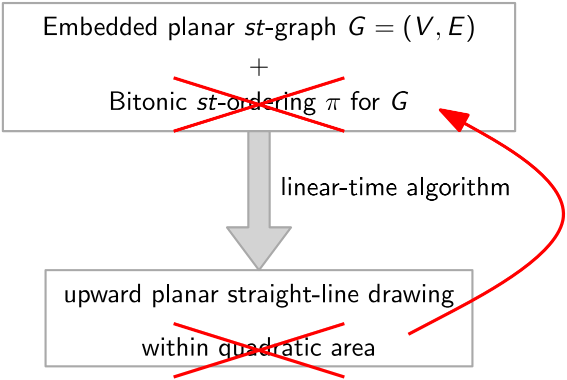
- The arXiv-version contains a full description and a listing.

Bitonic *st*-orderings - Results



- The arXiv-version contains a full description and a listing.
- Some planar *st*-graphs require exponential area!

Bitonic *st*-orderings - Results

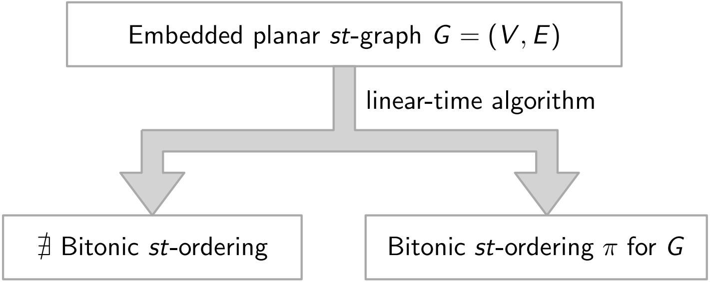


- The arXiv-version contains a full description and a listing.
- Some planar *st*-graphs require exponential area!
- Not every planar *st*-graph admits a bitonic st-ordering.

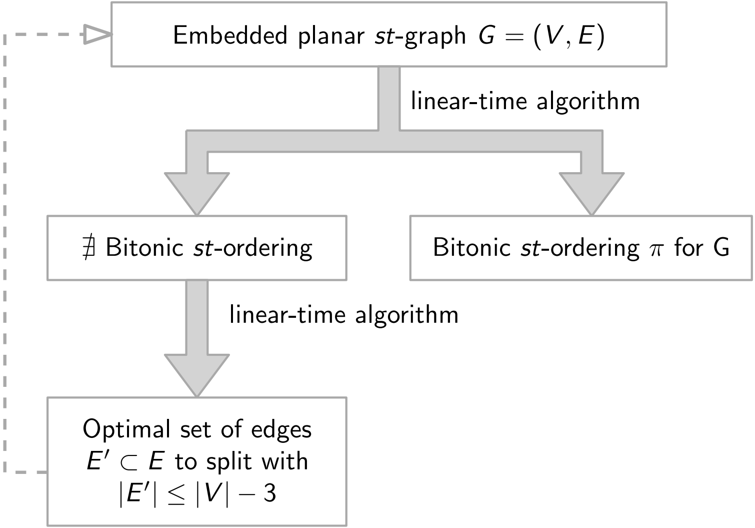
Bitonic *st*-orderings - More results

Embedded planar *st*-graph $G = (V, E)$

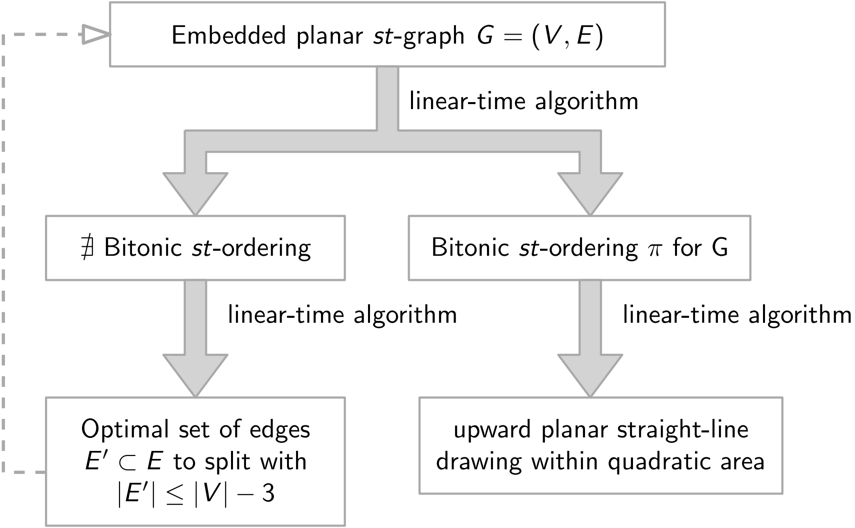
Bitonic *st*-orderings - More results



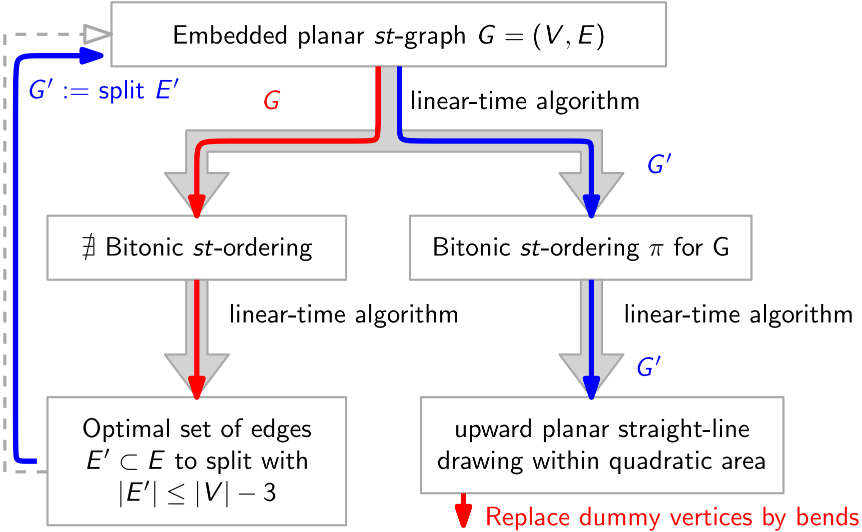
Bitonic *st*-orderings - More results



Bitonic *st*-orderings - More results



Bitonic *st*-orderings - More results



Bitonic *st*-orderings - More results

Embedded planar *st*-graph $G = (V, E)$

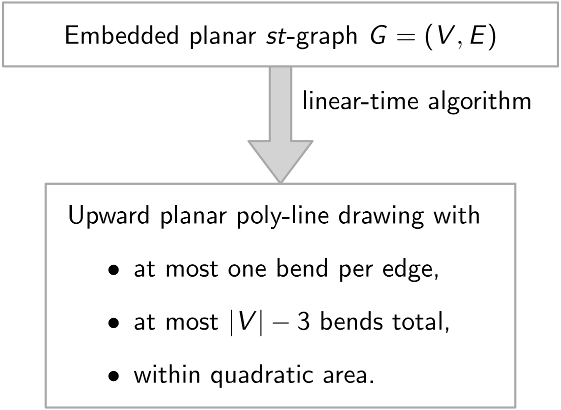
↓ linear-time algorithm

Upward planar poly-line drawing with

- at most one bend per edge,
- at most $|V| - 3$ bends total,
- within quadratic area.

- Best upper bound so far (Di Battista et al.): $2|V| - 5$.

Bitonic *st*-orderings - More results



- Best upper bound so far (Di Battista et al.): $2|V| - 5$.
- Drawing properties extend to all upward planar graphs.

Conclusion

Summary

- Classic incremental planar graph drawing for planar *st*-graphs.
- Does not work on all planar *st*-graphs, but we can split edges.
- The algorithms are simple, fast and easy to implement.

Conclusion

Summary

- Classic incremental planar graph drawing for planar *st*-graphs.
- Does not work on all planar *st*-graphs, but we can split edges.
- The algorithms are simple, fast and easy to implement.

Thank you for your attention!