

Applying Crossing Reduction Strategies to Layered Compound Graphs

Michael Forster
`forster@fmi.uni-passau.de`

University of Passau, Germany

Overview

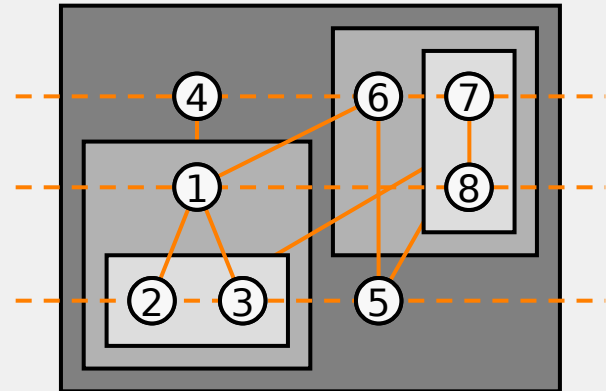
- Layered Compound Graphs
 - Definition
 - Drawing Conventions
- Simple Layout Algorithms
- Advanced Crossing Reduction
 - Arising Problems
 - Our solutions
- Summary

1. Layered Compound Graphs

Example, Drawing Conventions, Definition

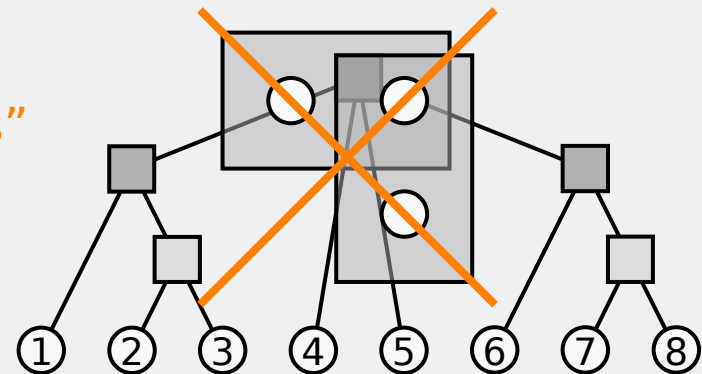
Example / Drawing Conventions

- Base nodes
- Compound nodes
 - Contain base nodes
 - Or other compound nodes
 - Inclusion hierarchy is a tree
 - Node intersection is forbidden



- Edges
 - Connect base nodes
 - Or compound nodes
 - Or each other
- yesterday:
"cross edges"

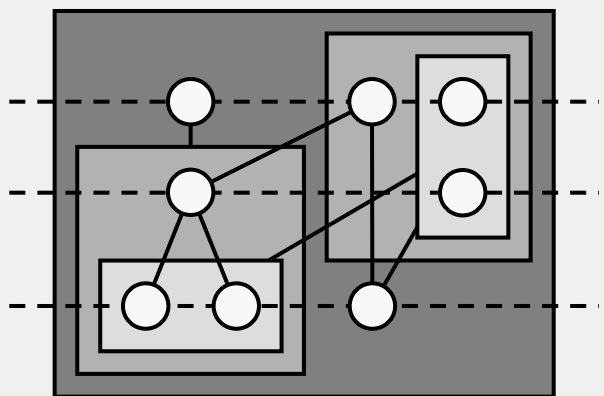
- Layers
 - Horizontal lines for the base nodes
 - Bends are only allowed on layers



Global Layers vs. Local Layers

Global Layers

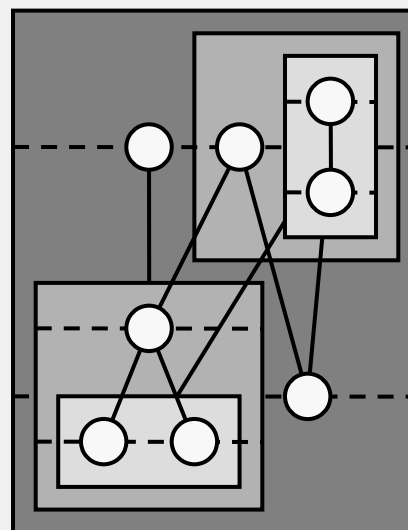
- Single set of layers for all nodes
- Compound nodes can span multiple layers



- More compact layout
- Algorithm by Sander, 1996

Local Layers

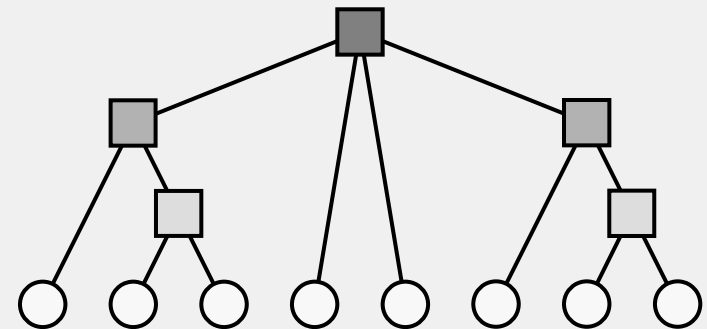
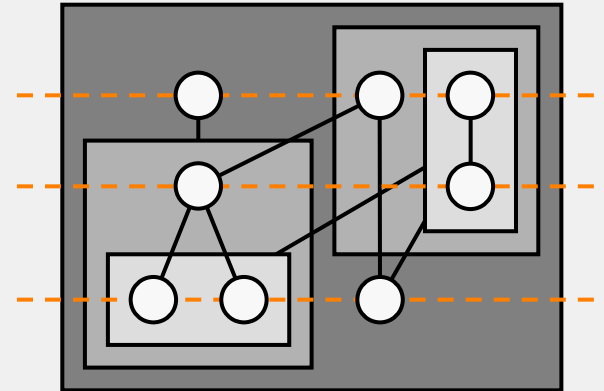
- One set of layers per compound node
- Compound Nodes are not allowed to span layers



- Fewer layers
- Algorithm by Sugiyama / Misue, 1991

Definitions

- Compound graph: $G = (V, E, H)$
 - Nodes: V
 - Adjacency edges: $E \subseteq V \times V$
 - Hierarchy edges: $H \subseteq V \times V$
 - Hierarchy tree: $T = (V, H)$
 - Base nodes: $B = \text{leaves}(T)$
 - Compound nodes: $C = V \setminus B$
 - Base graph: $G|_B$
- Layered compound graph: $G = (V, E, H, L)$
 - Layer assignment: $L: B \rightarrow \mathbb{Z}$
- Clustered graph: $G = (V, E, H)$
[Yesterday]
 - Adjacency edges: $E \subseteq B \times B$
- Layered clustered graph: $G = (V, E, H, L)$
 - analogous

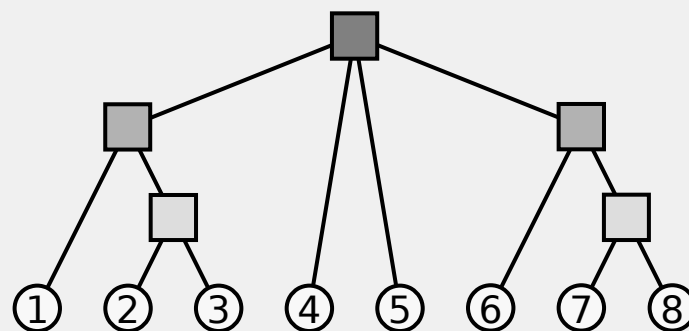
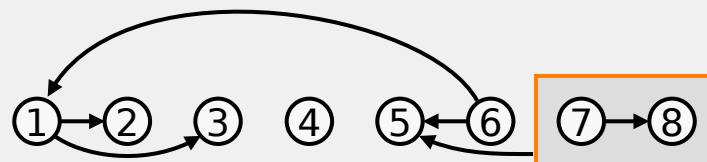


2. Simple Algorithms

... and why they are not optimal

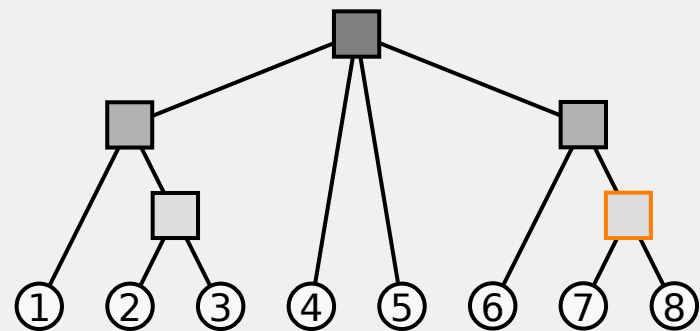
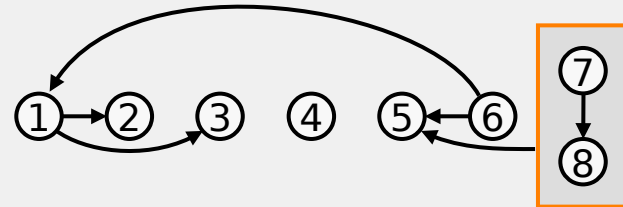
Bottom Up

- Start with base graph
- For each compound node



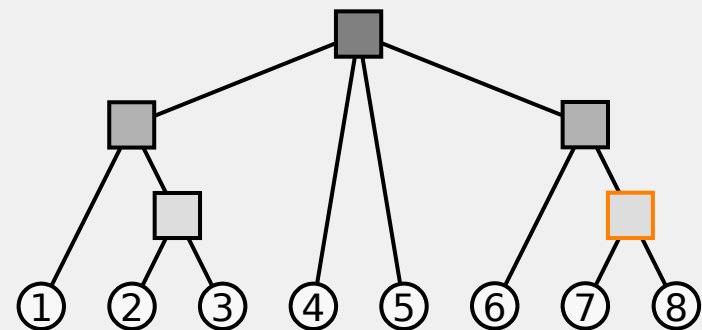
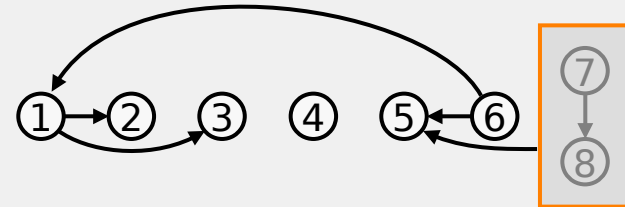
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents



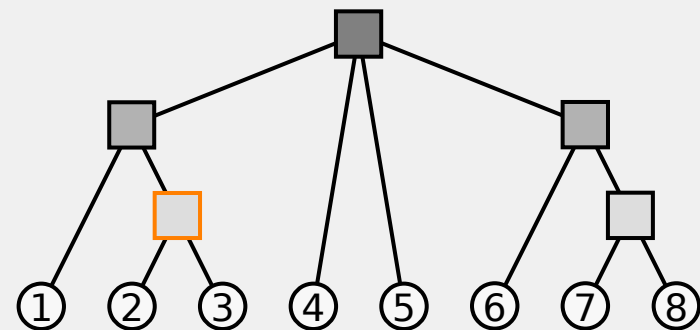
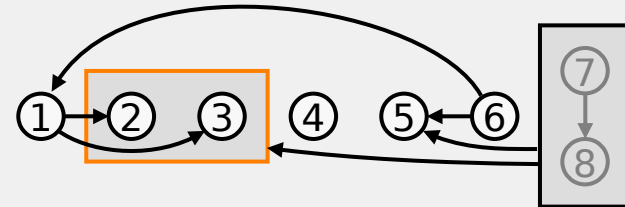
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



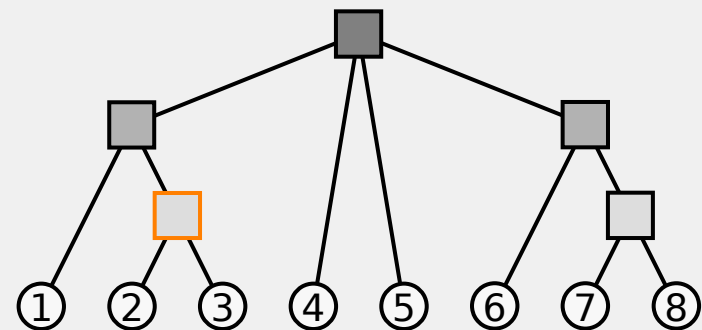
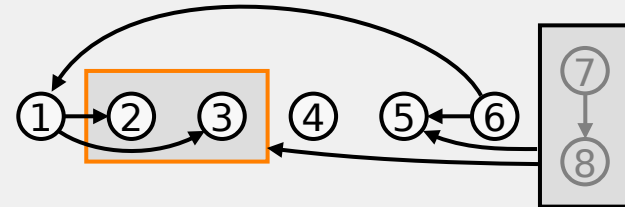
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



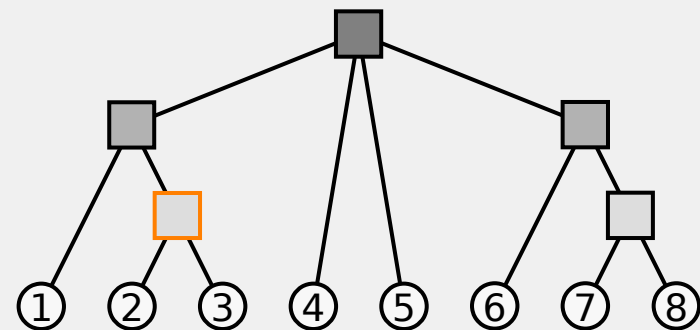
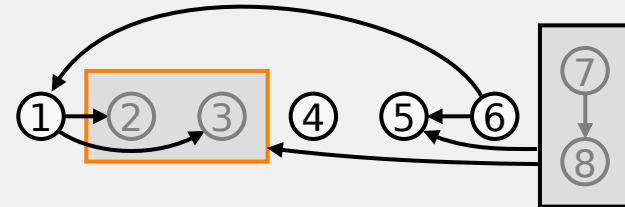
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



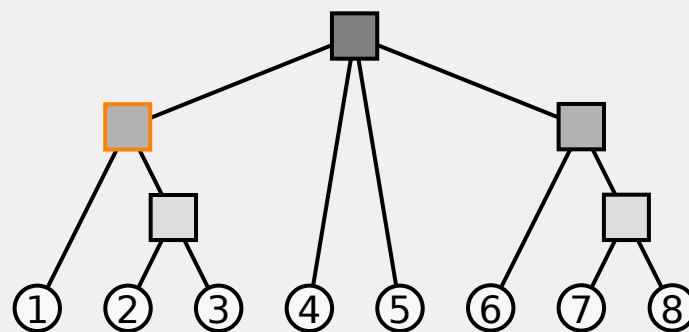
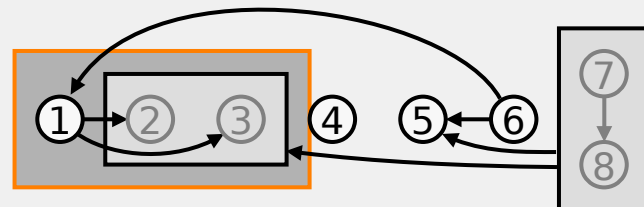
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



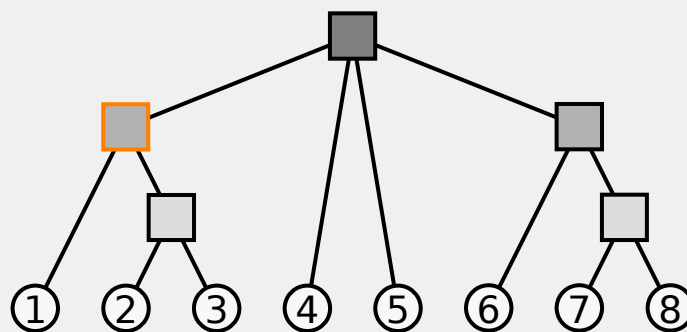
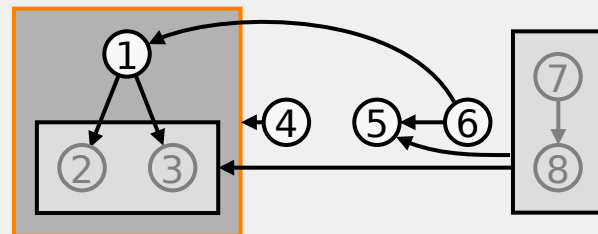
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



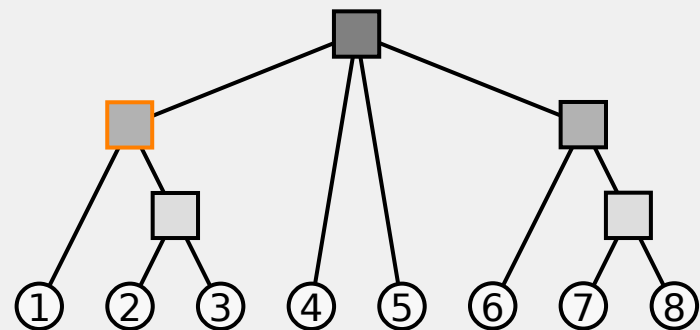
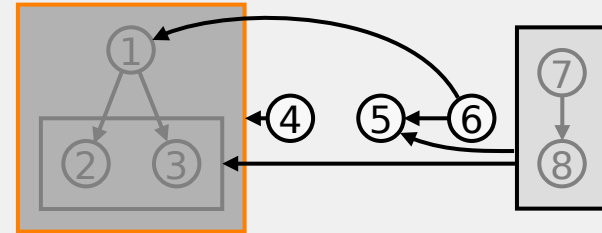
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



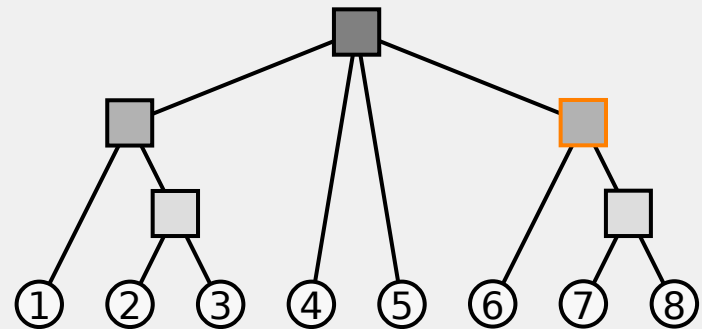
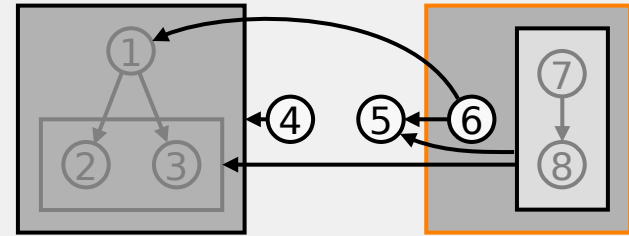
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



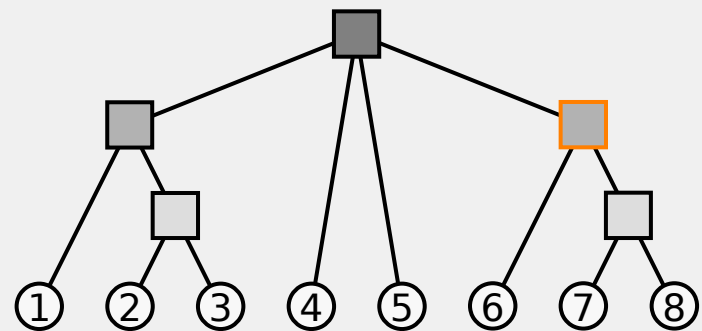
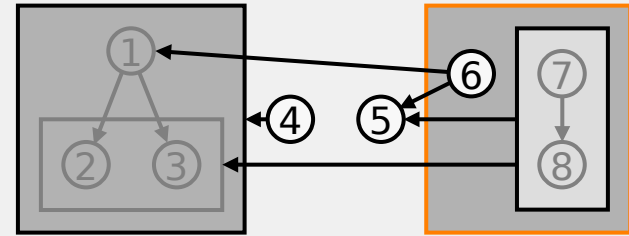
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



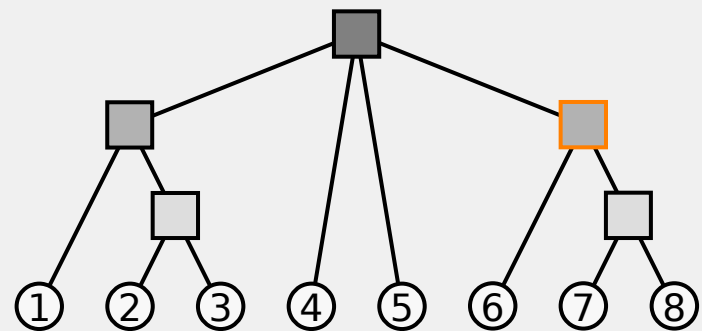
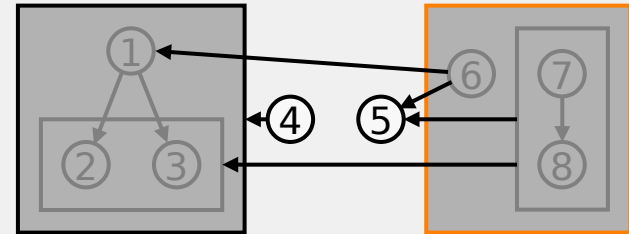
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



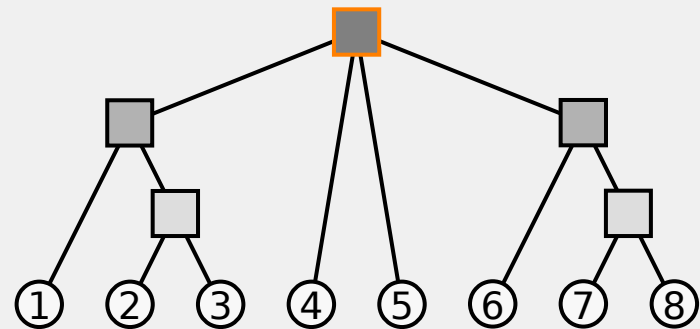
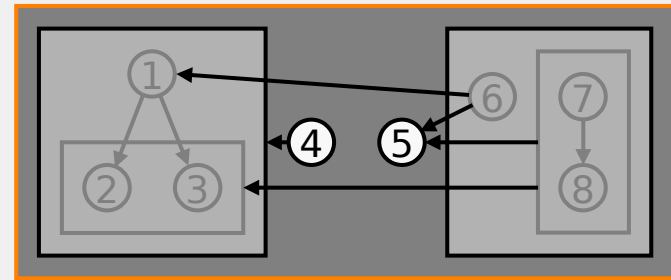
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



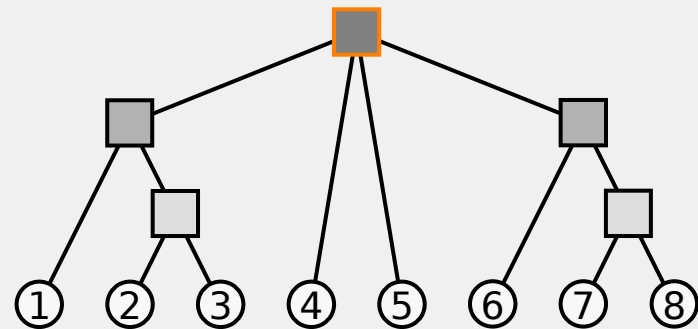
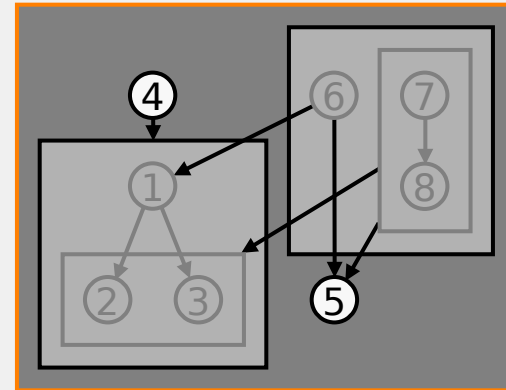
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



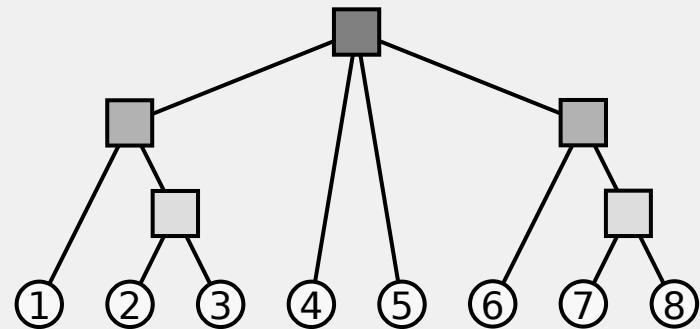
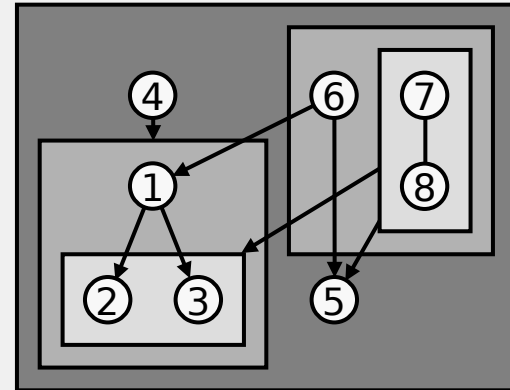
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents



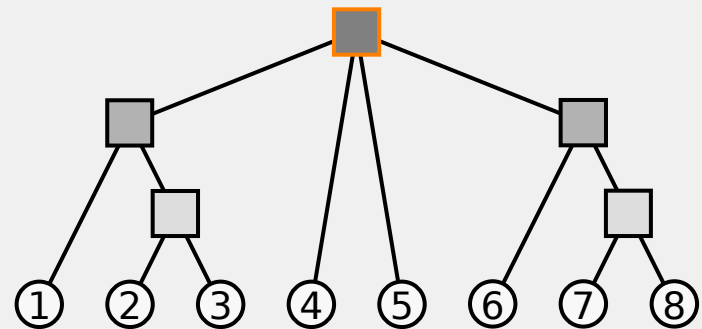
Bottom Up

- Start with base graph
- For each compound node
 - Layout contents
 - Hide contents
- Finished



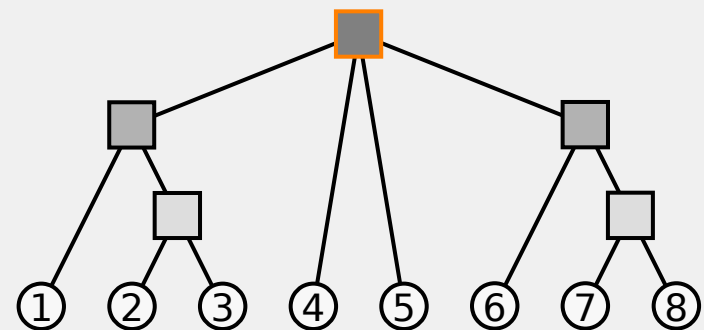
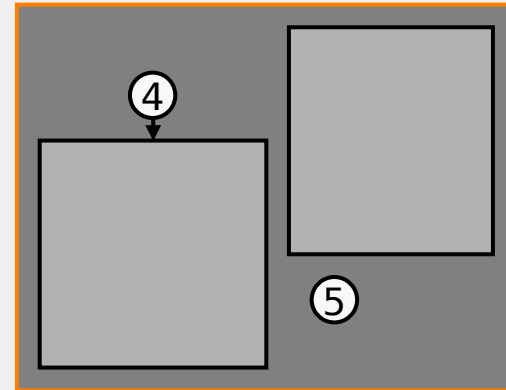
Top Down

- Start with hierarchy root
- For each compound node



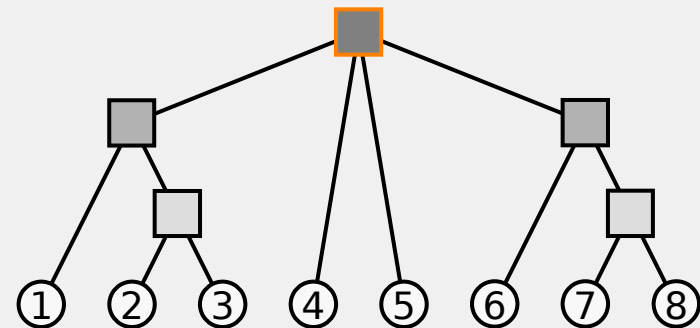
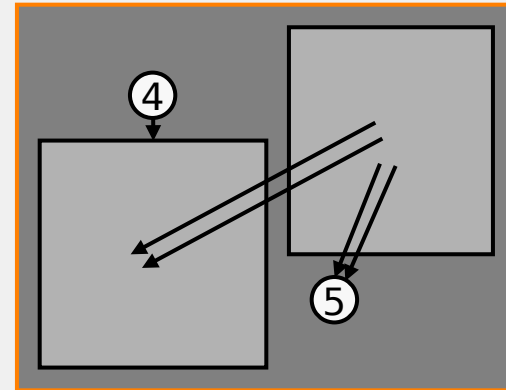
Top Down

- Start with hierarchy root
- For each compound node
 - Layout contents



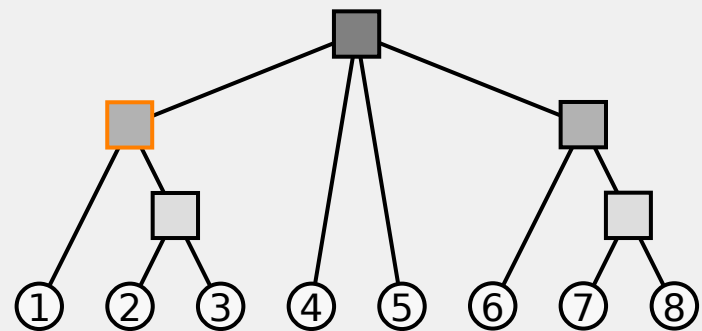
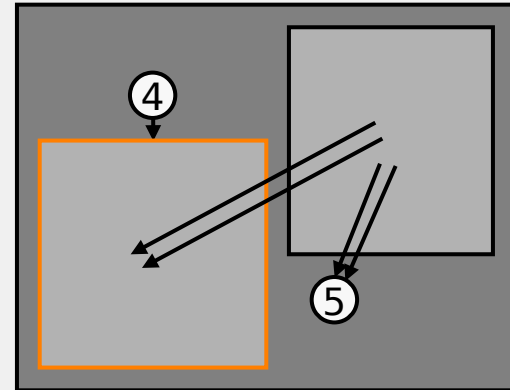
Top Down

- Start with hierarchy root
- For each compound node
 - Layout contents



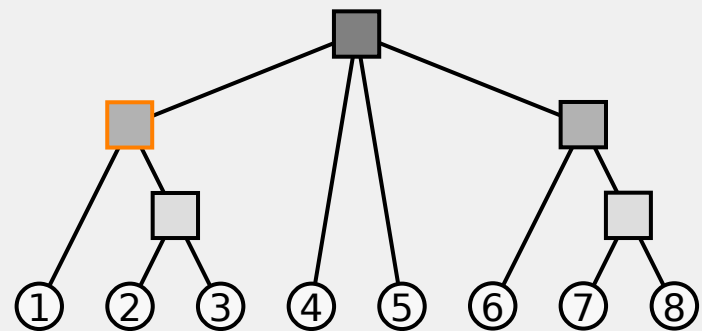
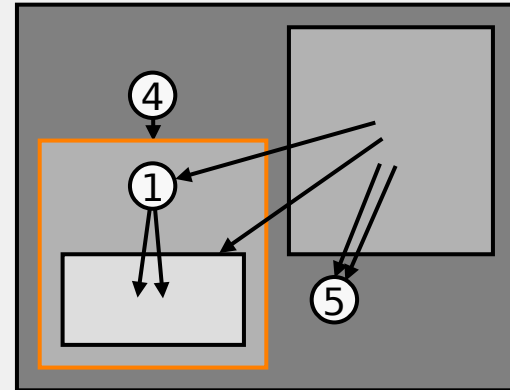
Top Down

- Start with hierarchy root
- For each compound node
 - Layout contents



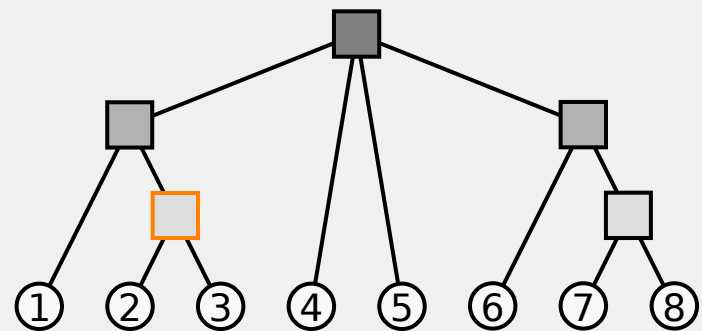
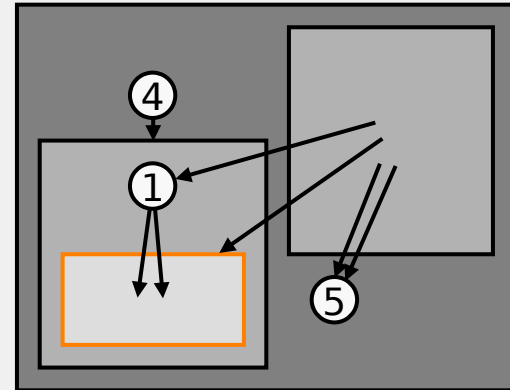
Top Down

- Start with hierarchy root
- For each compound node
 - Layout contents



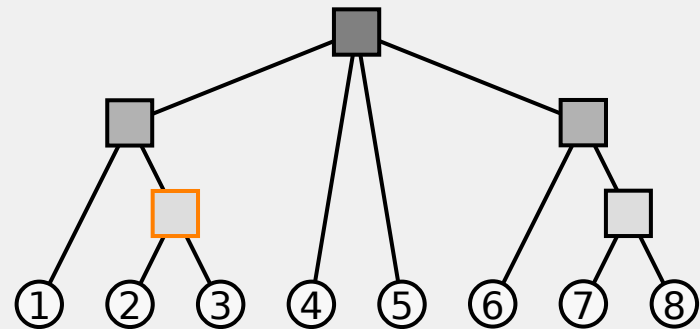
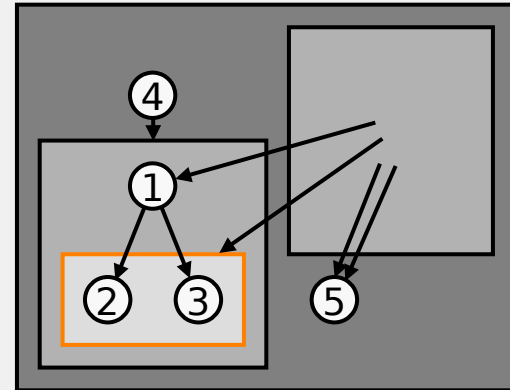
Top Down

- Start with hierarchy root
- For each compound node
 - Layout contents



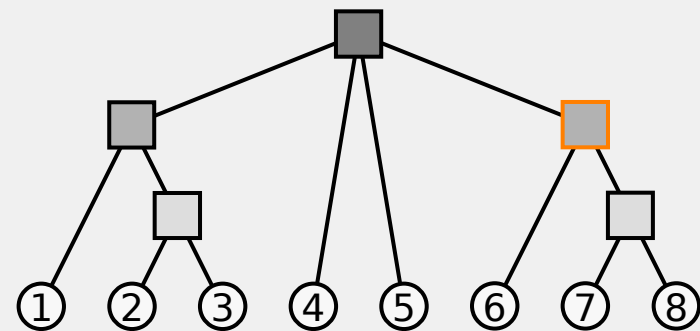
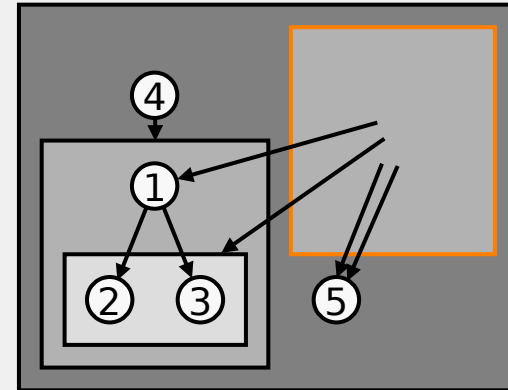
Top Down

- Start with hierarchy root
- For each compound node
 - Layout contents



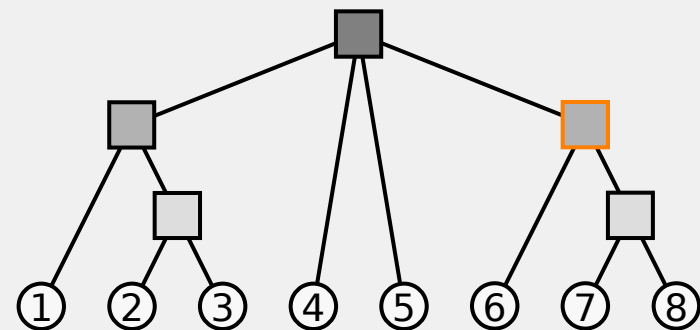
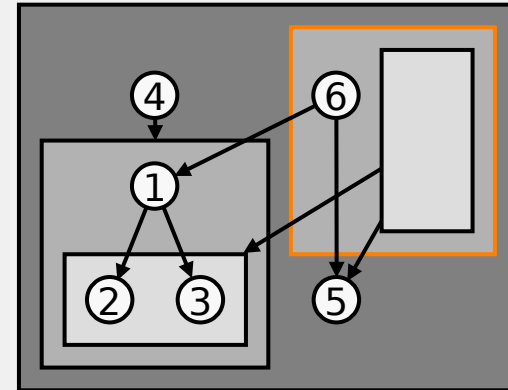
Top Down

- Start with hierarchy root
- For each compound node
 - Layout contents



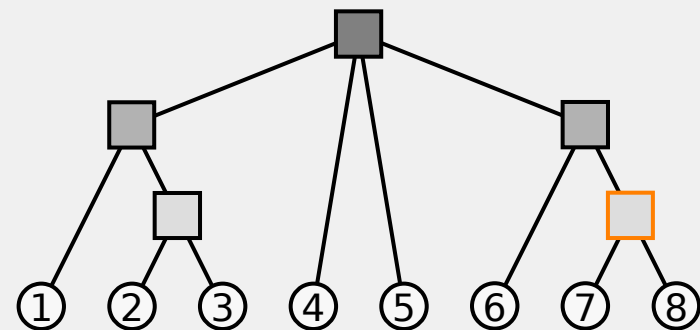
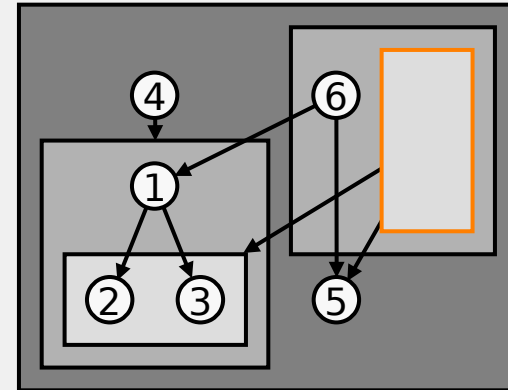
Top Down

- Start with hierarchy root
- For each compound node
 - Layout contents



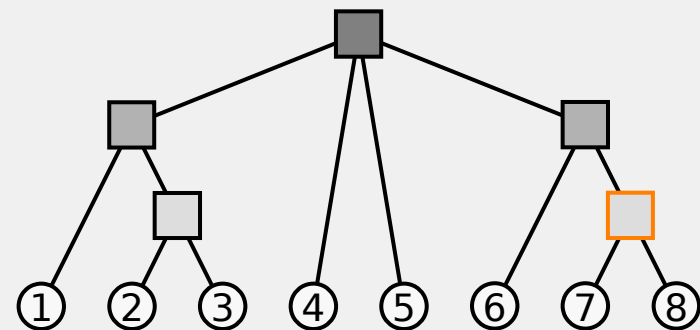
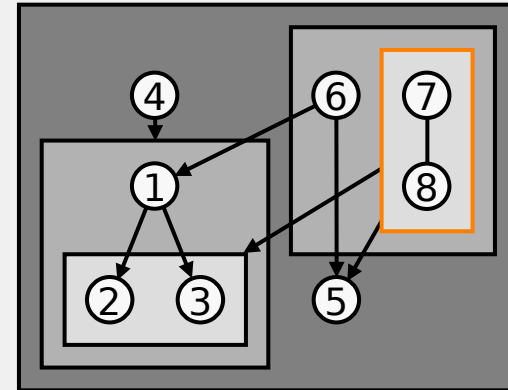
Top Down

- Start with hierarchy root
- For each compound node
 - Layout contents



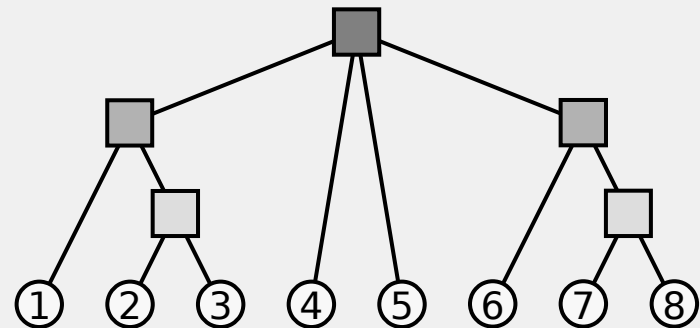
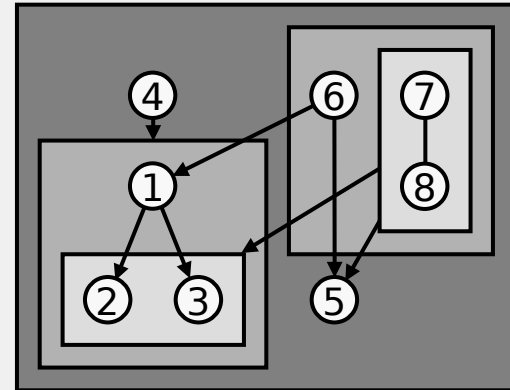
Top Down

- Start with hierarchy root
- For each compound node
 - **Layout contents**



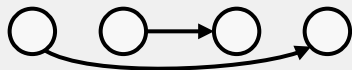
Top Down

- Start with hierarchy root
- For each compound node
 - Layout contents
- Finished
- Remarks
 - Height of Compound Nodes must be known for layering
 - Preprocessing
 - Width can be computed afterwards



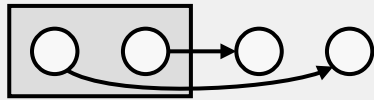
Unnecessary Crossings

Bottom Up



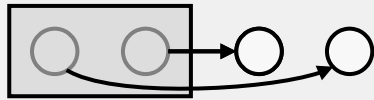
Unnecessary Crossings

Bottom Up



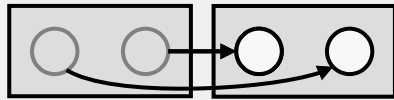
Unnecessary Crossings

Bottom Up



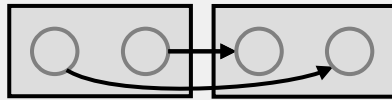
Unnecessary Crossings

Bottom Up



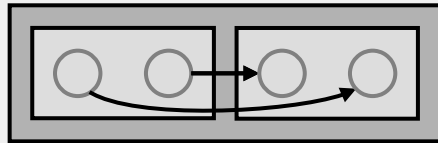
Unnecessary Crossings

Bottom Up



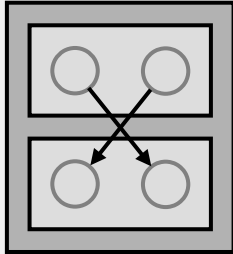
Unnecessary Crossings

Bottom Up



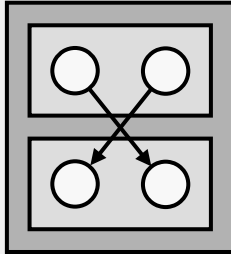
Unnecessary Crossings

Bottom Up

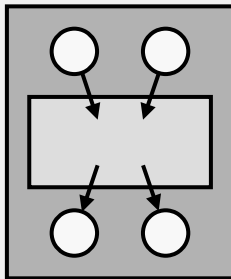


Unnecessary Crossings

Bottom Up

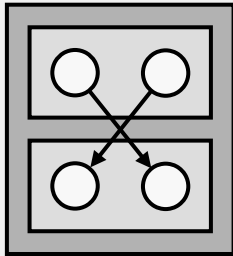


Top Down

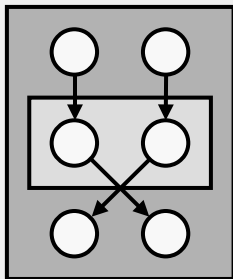


Unnecessary Crossings

Bottom Up



Top Down



Remarks

- Trying to respect connectivity of border nodes helps, but is not optimal
- Crossings appear even when using optimal crossing reduction strategy in each step
- Revise **application** of crossing reduction strategy

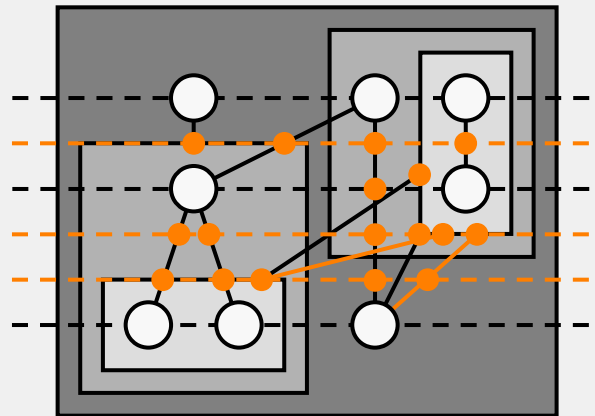
That is, what this talk is about!

3. Advanced Crossing Reduction

Sander's Approach and Our Improvements

Layout Algorithm

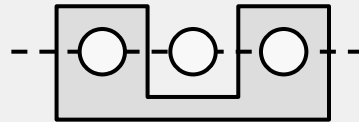
- Overview
 - Convert compound graph to clustered graph
 - Use Sugiyama algorithm for base graph
 - Additionally respect compound nodes at each step
- Crossing Reduction
 - Start with layered clustered graph
 - Insert dummy nodes for long span edges
 - Permute base node orders respecting compound nodes



Respecting the compound nodes

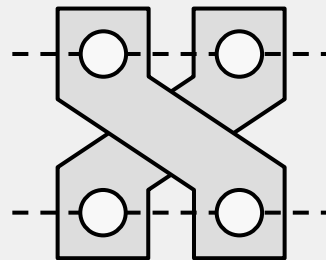
Single Layer Restriction:

- Children of a compound node must be placed next to each other with no other nodes between them
- Forbidden:



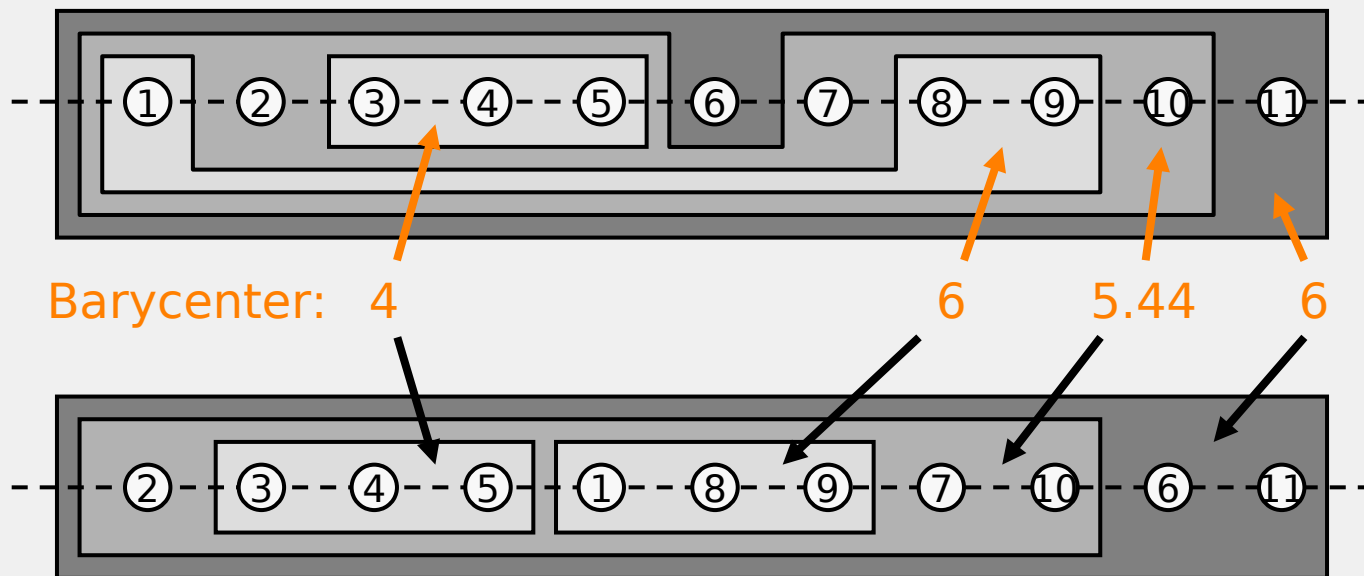
Multiple Layer Restriction

- The relative position of compound nodes must be the same on all layers
- Forbidden:



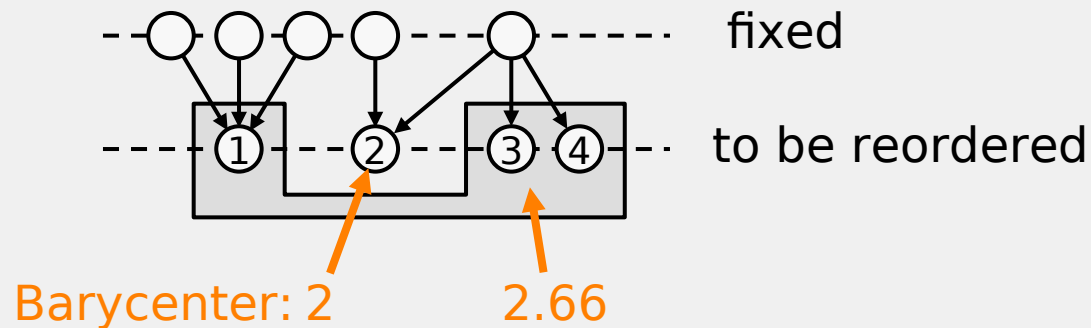
Sander's Approach

- Apply conventional DAG crossing reduction method
 - Ignore compound nodes
 - Violations of the restrictions may occur
- Resolve violations afterwards:



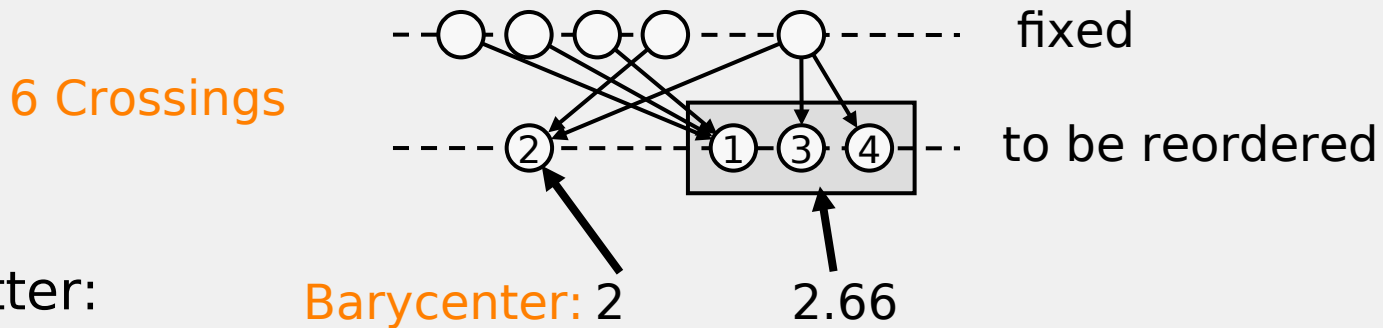
Sander's Approach: Evaluation

- Obviously not optimal
 - Only intermediate node order is considered
 - Adjacency edges are ignored in second step
- Unnecessary Crossings:

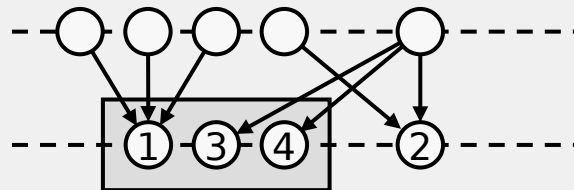


Sander's Approach: Evaluation

- Obviously not optimal
 - Only intermediate node order is considered
 - Adjacency edges are ignored in second step
- Unnecessary Crossings:



- Better:
- 2 Crossings

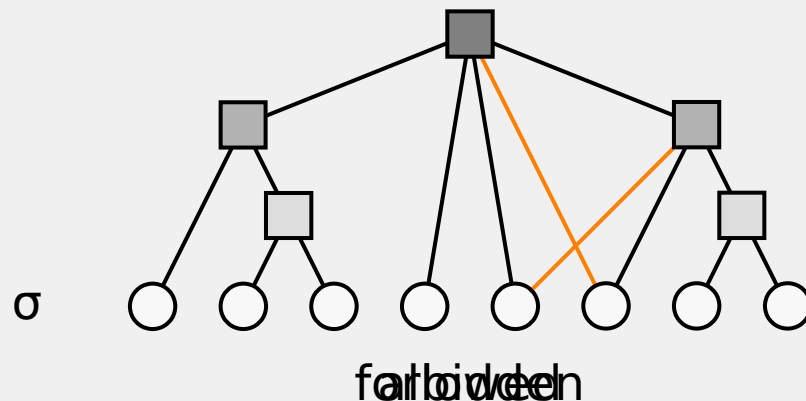


Our Crossing Reduction Method

- Basically use Sander's algorithm
- But: respect compound nodes right from the start
- Outline of the rest of the talk
 - Respect the single layer restriction
 - Extend this to also respect the multiple layer restriction

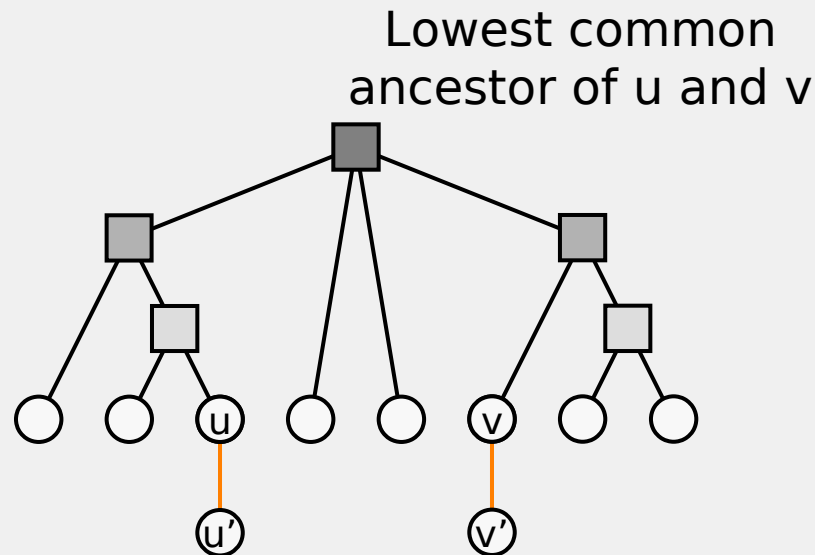
Single Layer Restriction

- What base node orders σ are allowed?
- **Lemma 1:** Equivalent:
 - σ is allowed
 - The layer hierarchy tree T has no crossings
 - There exists a child order of T such that σ can be obtained by a pre-/postorder traversal
- Consequence: optimize hierarchy tree child order



Single Layer Restriction

- When do adjacency edges cross?
- **Lemma 2:** Equivalent:
 - Two adjacency edges (u, u') , (v, v') cross
 - Corresponding children of lowest common ancestor of u and v have different relative order than u' and v'
- Consequence: Assign each crossing to the lowest common ancestor

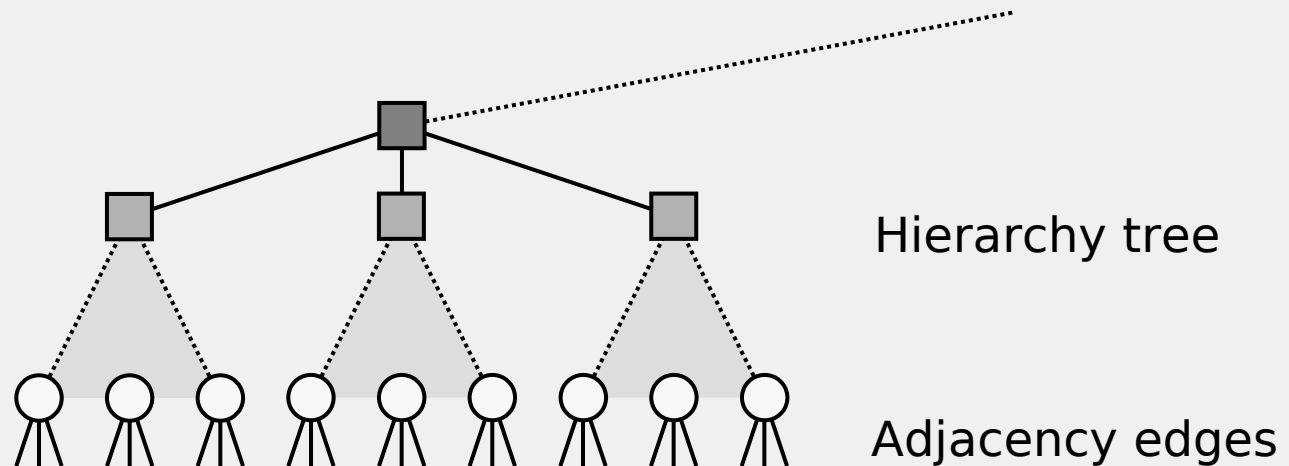


Single Layer Restriction

- What are good hierarchy tree child orders?
- **Lemma 3:** The number of crossings is the sum of the crossings associated to each compound node.
- **Theorem:** Equivalent:
 - An order of the base nodes has a minimal number of crossings
 - The corresponding hierarchy tree has a minimal number of crossings associated with each compound node
- Consequence: Number of crossings can be minimized for each compound node independently

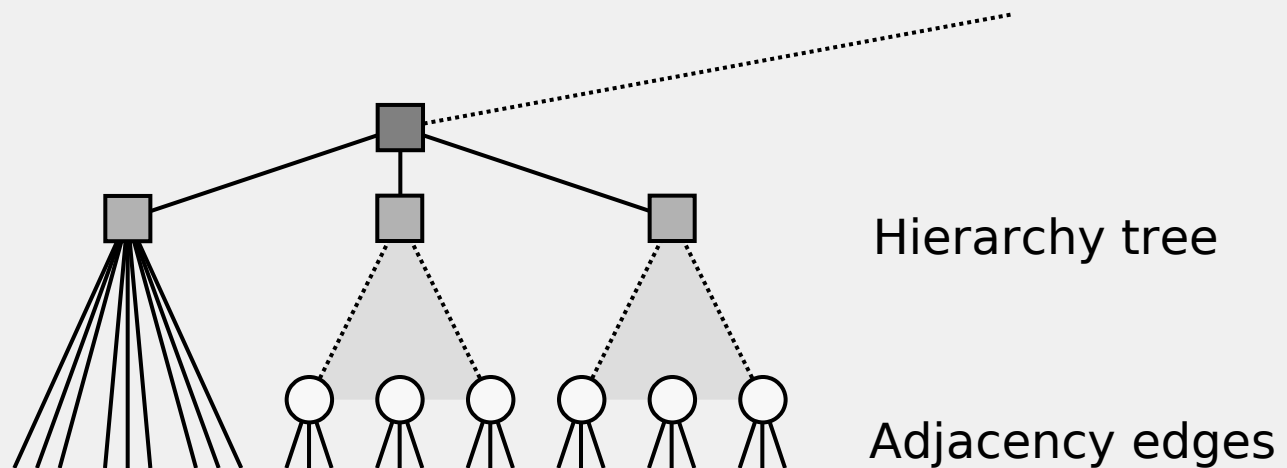
Single Layer Restriction

- Construct crossing reduction graph
 - “Pull up” adjacency edges



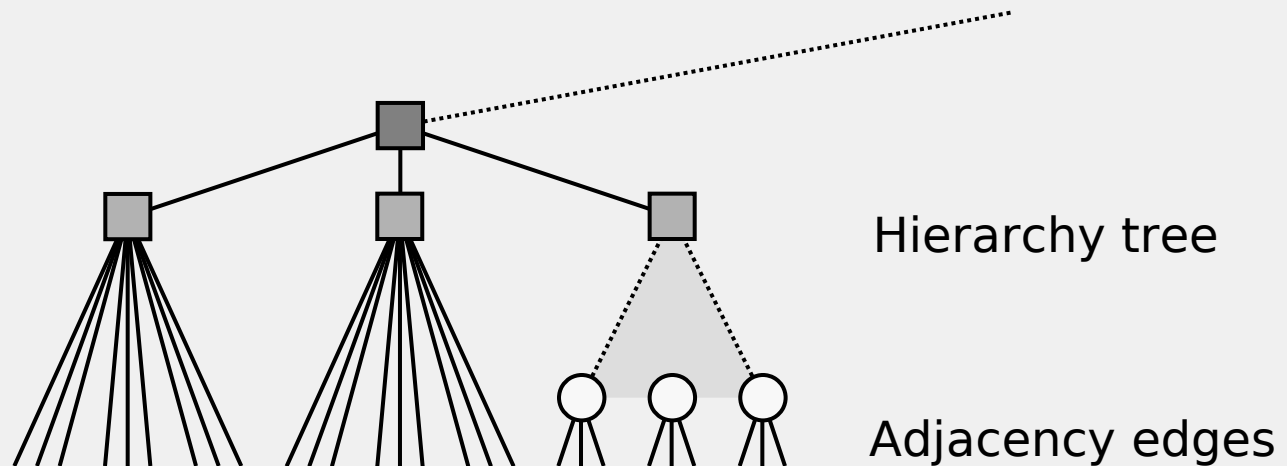
Single Layer Restriction

- Construct crossing reduction graph
 - “Pull up” adjacency edges



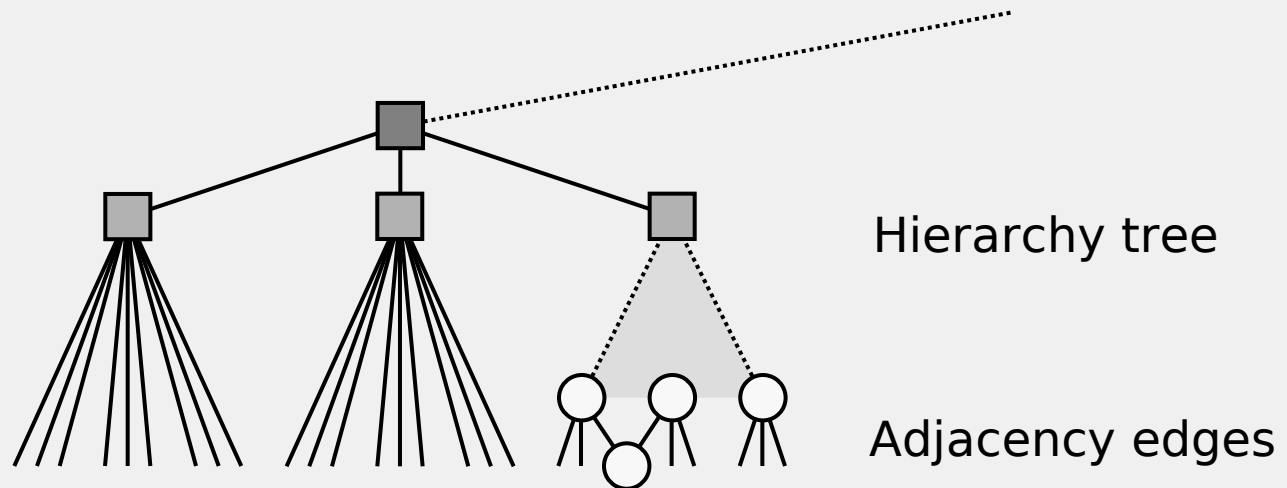
Single Layer Restriction

- Construct crossing reduction graph
 - “Pull up” adjacency edges



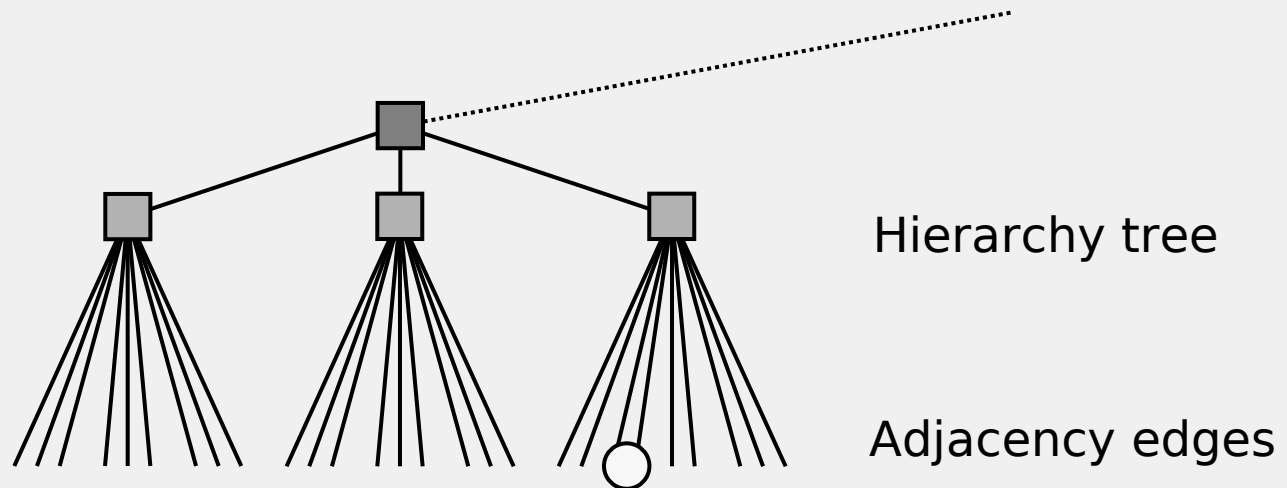
Single Layer Restriction

- Construct crossing reduction graph
 - “Pull up” adjacency edges



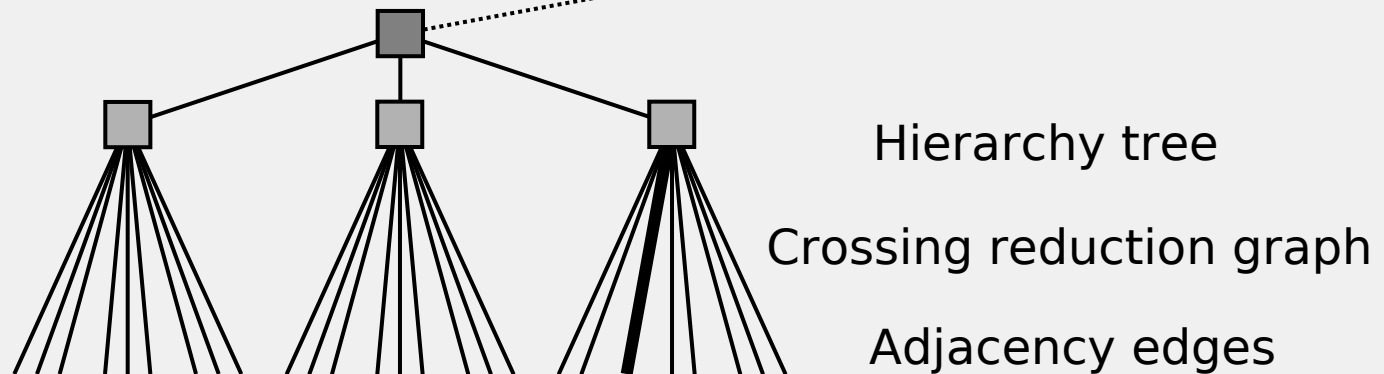
Single Layer Restriction

- Construct crossing reduction graph
 - “Pull up” adjacency edges

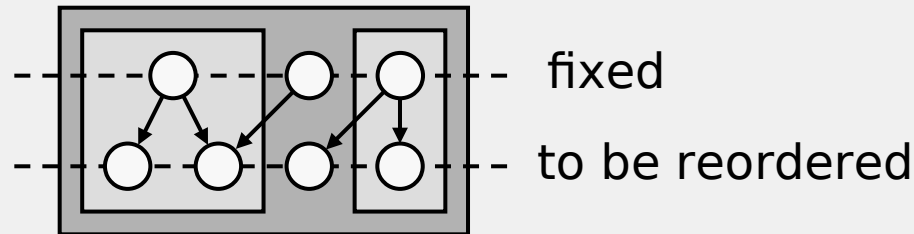


Single Layer Restriction

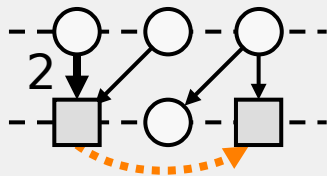
- Construct crossing reduction graph
 - “Pull up” adjacency edges
 - Use weights for multiple edges (if appropriate)
- Apply any conventional 1-sided 2-layer crossing reduction method
- Repeat for all compound nodes
- Remarks
 - Order of application does not matter
 - Result is optimal if 2-layer crossing reduction method is optimal
 - Crossing reduction graphs can be computed in a preprocessing step



Multiple Layer Restriction

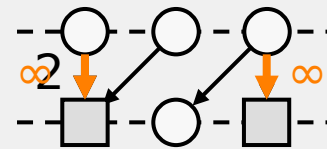


Constraint method



Heavy Edge method

crossing
reduction graph



Evaluation

- Guarantees compliance
- Depends on a 2-layer crossing reduction strategy, that supports constraints

Evaluation

- Cannot guarantee compliance
- Works with any (weighted) crossing reduction strategy
- Has side effects (experimental results needed)

Summary

Past and Future Work

Summary

- New crossing reduction method for layered compound graphs
 - Does not introduce unnecessary crossings
 - Optimal if used with optimal 2-layer crossing reduction strategy
- Implementation in progress
 - Constraint crossing reduction strategy by Schreiber, 2001
 - Preliminary results are promising
 - Too early to present numbers
- Future plans
 - Finish implementation
 - Apply directly to compound graphs ?
 - Improve Methods for respecting multiple layer restriction ?