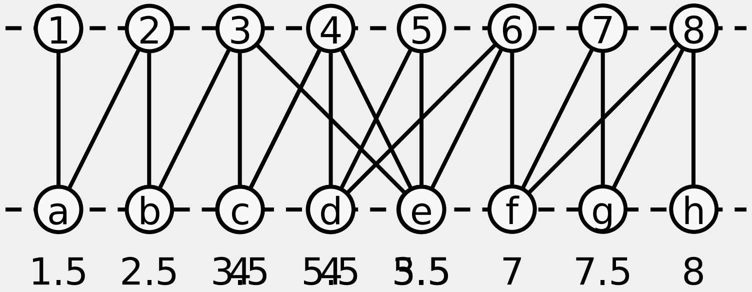
A Fast and Simple Heuristic for Constrained Two-Level Crossing Reduction

Michael Forster
University of Passau, Germany

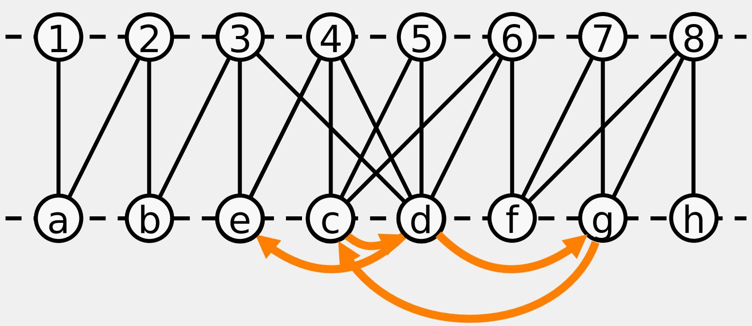
One-Sided Two-Level Crossing Reduction

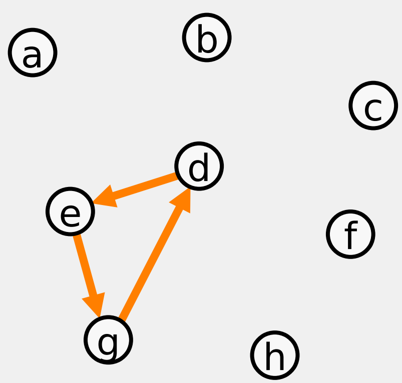
- Important and well-known problem in hierarchical graph drawing
  - Two-level graph, permutation of the first level is fixed
  - Wanted: Permutation of the second level with few crossings
- NP-hard → heuristics
- Barycenter heuristic: Sort second level by barycenter values



Constrained Crossing Reduction

- In addition: Constraints
  - Predetermined vertex order
  - Violated / satisfied
- Applications
  - Given by the user
  - Big vertices / clusters
  - Preserving mental map
- Objective
  - Satisfy all constraints
  - Few crossings
- Constraint graph
  - Must be acyclic
  - Important special case: Single path + isolated vertices





Overview

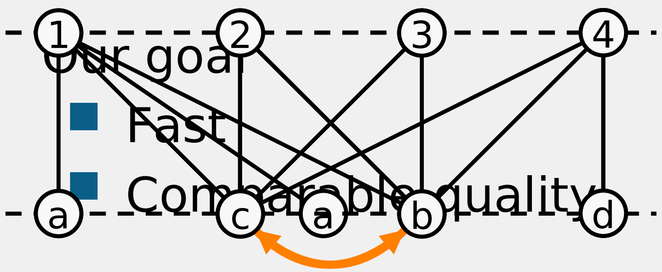
- Introduction
- Previous work
  - Barycenter heuristic extensions
  - Penalty graph method
- New algorithm
  - Idea
  - Details
- Experimental results
  - Comparison to the penalty graph method
- Summary

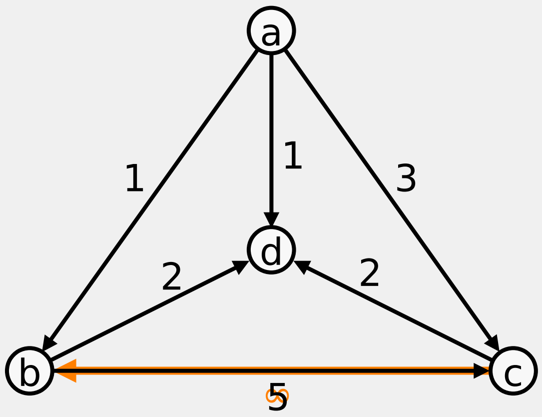
Previous Work: Barycenter Extensions

- Sander [1996]
  - Extends iterative two-level crossing reduction algorithms
  - Start with arbitrary admissible permutation
  - Execute updates only if no violations
- Waddle [2000]
  - Compute barycenter values
  - Violated constraint: Change barycenter value of source vertex
- Evaluation
  - Very fast
  - Medium quality
  - Especially bad for many constraints

Previous Work: Penalty Graph Method

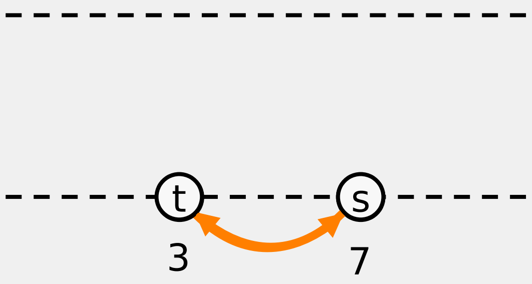
- Schreiber [2001], Finocchi [2001]
  - Compute the penalty graph
  - Insert constraints as ∞-edges
  - Find a feedback arc set with small weight (NP-hard)
  - Sort vertices topologically
- Evaluation (Experimental results of Schreiber)
  - High quality (up to 15% less crossings)
  - Significantly slower
- Our goal
  - Fast
  - Comparable quality





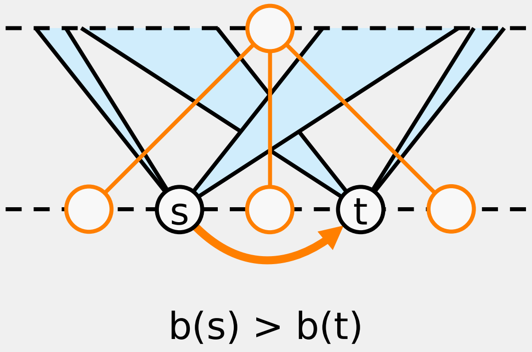
Idea

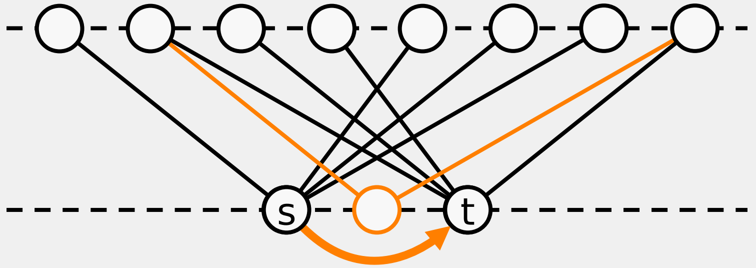
- New extension of barycenter heuristic
- First step: compute barycenter values b(v) for all vertices
- Consider a constraint (s, t)
  - b(s) > b(t) → violated by sorting
  - b(s) < b(t) → satisfied by sorting
- Violation must be prohibited
  - Sander: Prohibit swapping of s and t
  - Waddle: Increase barycenter value of s



Idea

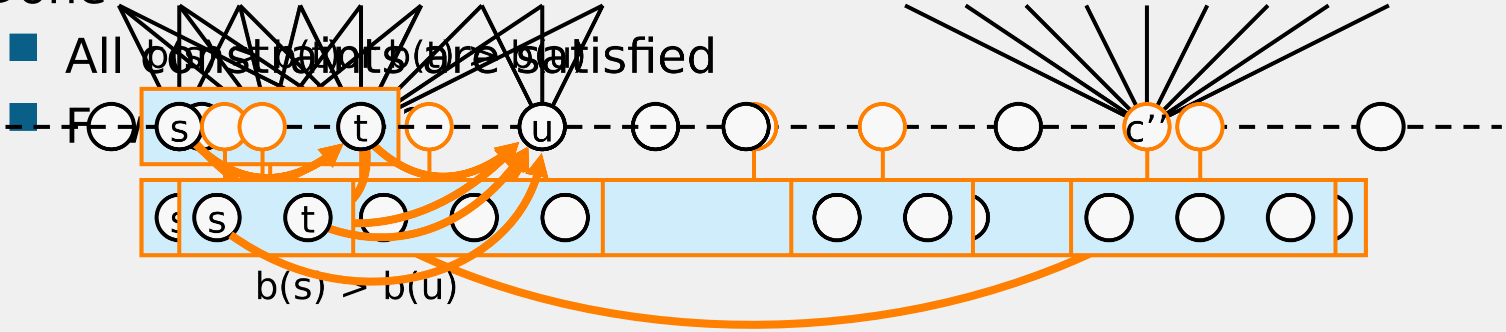
- Our Assumption
  - Violated constraint → No other vertices between the end vertices
- Valid?
  - True for some special cases
  - False in general
- But: Justified by good experimental results (see later)



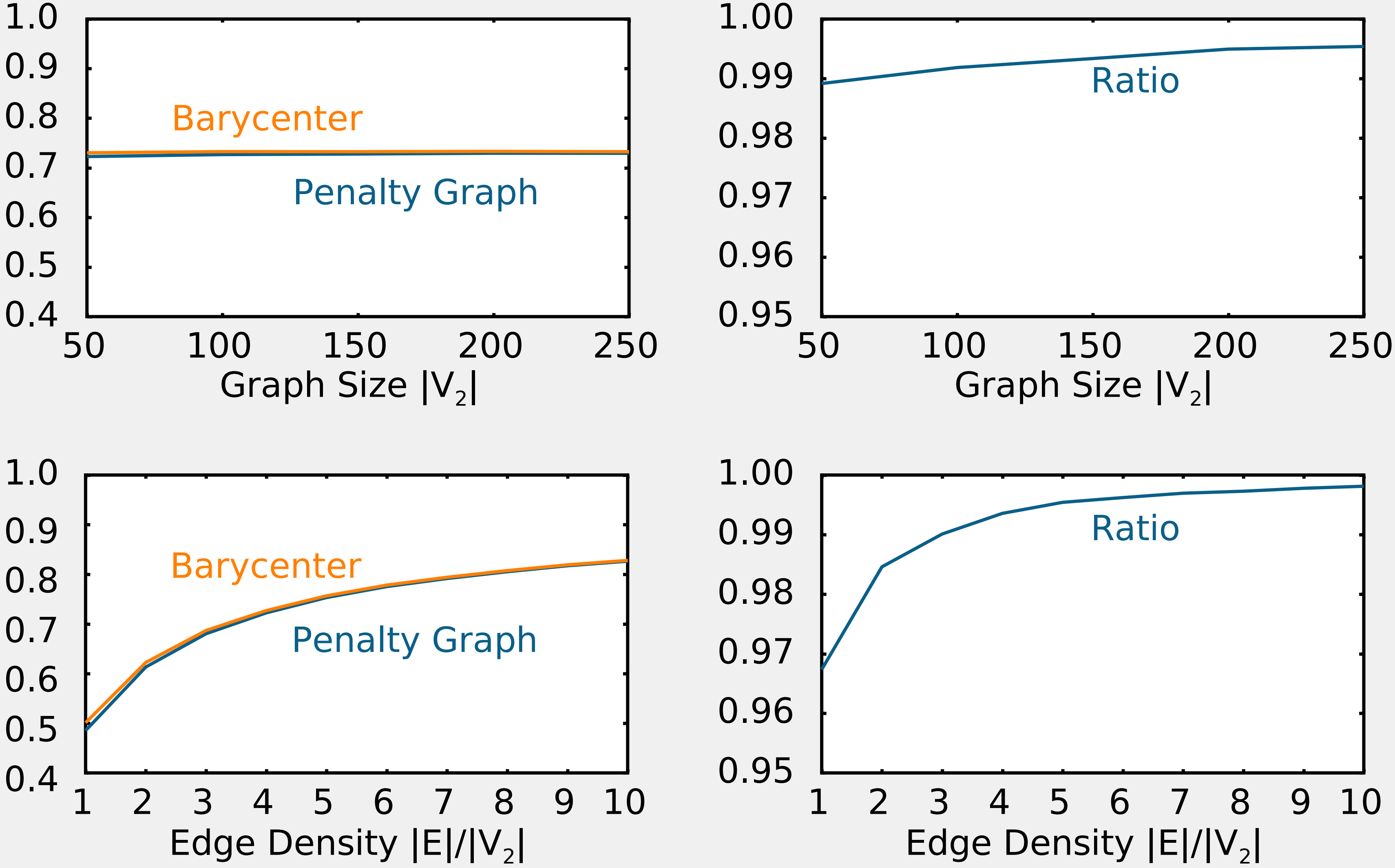


Algorithm

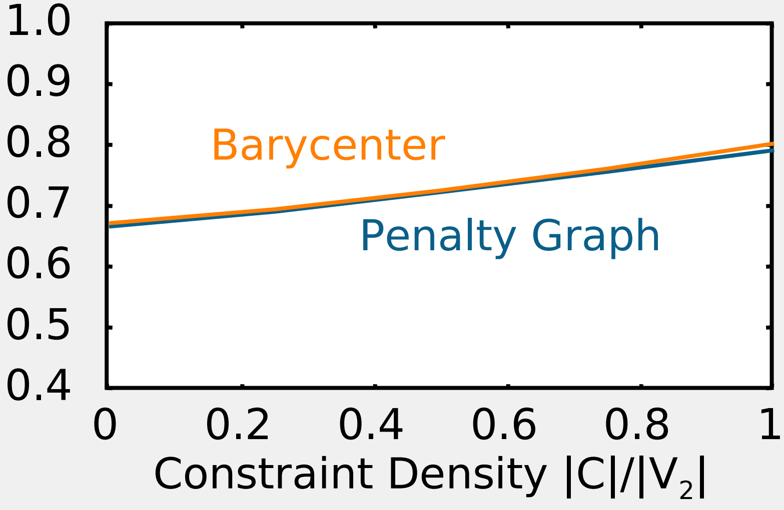
- For every violated constraint
  - Replace end vertices with a single vertex
  - Compute barycenter value in constant time: $b(c) = \frac{b(s) \cdot deg(s) + b(t) \cdot deg(t)}{deg(s) + deg(t)}$
- No constraint cycles must be introduced
  - Constraints must be processed in correct order
  - Modified topological sorting
- When no violated constraints are left
  - Sort remaining vertices by barycenter values
  - Re-insert removed vertices
- Done
  - All constraints are satisfied
  - F

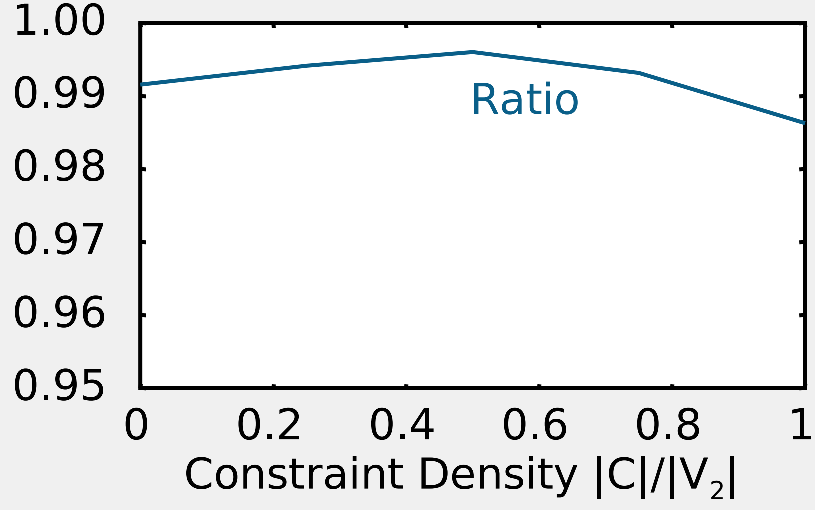


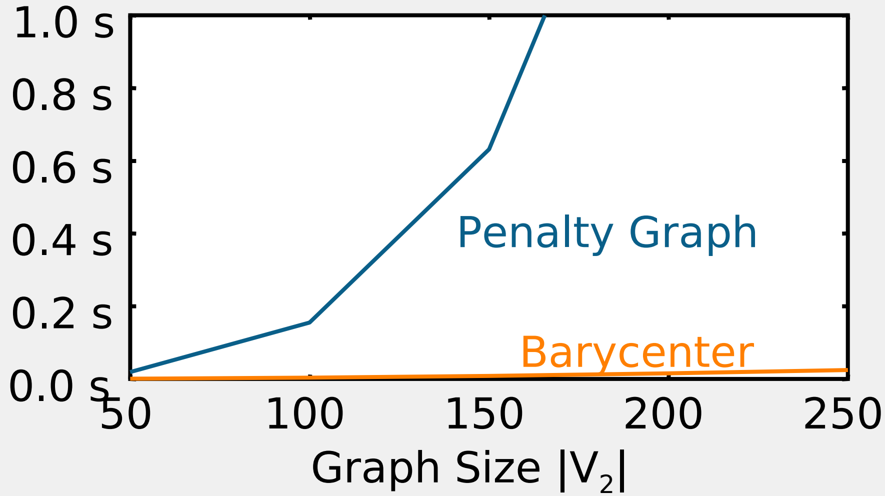
Experimental Results



Experimental Results







Summary

- Constrained one-sided two-level crossing minimization
  - New heuristic
  - Based on barycenter heuristic
- Quality
  - Better than other simple extensions
  - Smaller dependency on the number of constraints
  - Comparable to penalty graph approach
- Running time
  - $O(|V_2| \log |V_2| + |E| + |C|^2)$
  - Significantly faster than penalty graph approach
- Easy to implement (~ 100 LOC)
- Future Research
  - Improve complexity
  - Investigate linear constraint graphs (sifting?)

The End

Thank you for your attention!