Star-Struck by Fixed Embeddings
Modern Crossing Number Heuristics

GD ’21

Joint work with Markus Chimani and Tilo Wiedera

Max Ilsen, University of Osnabrück

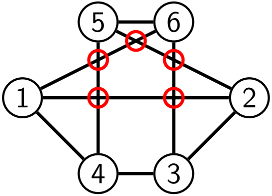
max.ilsen@uos.de

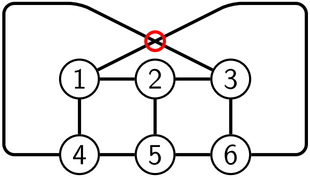
September 15, 2021

Crossing Number Problem

Given: Graph G

Task: Find minimum number of edge crossings in any drawing of G

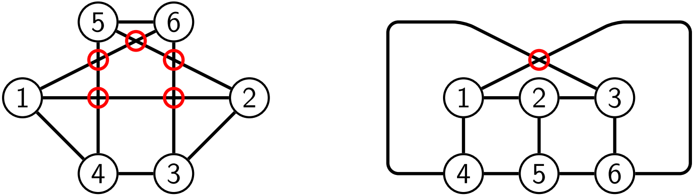




Crossing Number Problem

Given: Graph *G*

Task: Find minimum number of edge crossings in any drawing of *G*



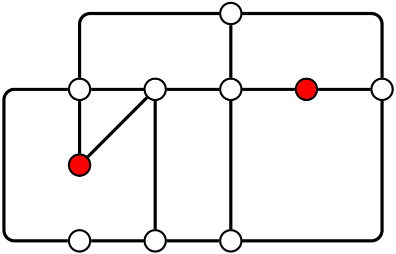
NP-hard even in restricted settings

→ Let's evaluate some heuristics!

Planarization Method

Heuristic #1

→ compute planar subgraph, insert remaining edges iteratively

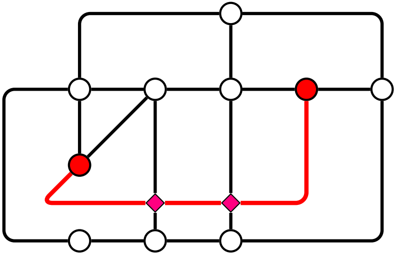


Batini, Talamo, and Tamassia 1984

Planarization Method

Heuristic #1

→ compute planar subgraph, insert remaining edges iteratively

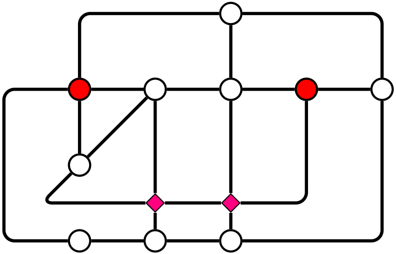


Batini, Talamo, and Tamassia 1984

Planarization Method

Heuristic #1

→ compute planar subgraph, insert remaining edges iteratively

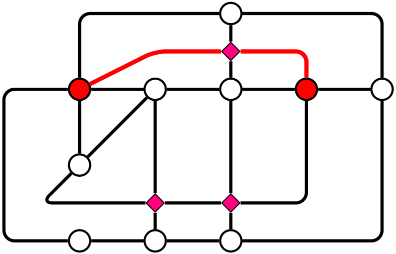


Batini, Talamo, and Tamassia 1984

Planarization Method

Heuristic #1

→ compute planar subgraph, insert remaining edges iteratively

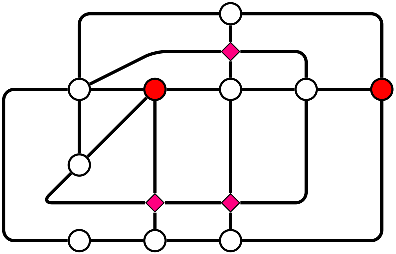


Batini, Talamo, and Tamassia 1984

Planarization Method

Heuristic #1

→ compute planar subgraph, insert remaining edges iteratively

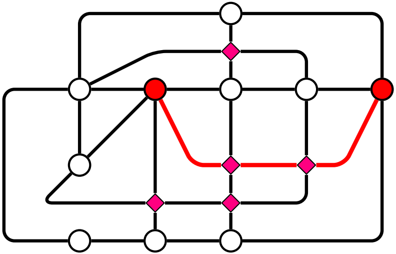


Batini, Talamo, and Tamassia 1984

Planarization Method

Heuristic #1

→ compute planar subgraph, insert remaining edges iteratively

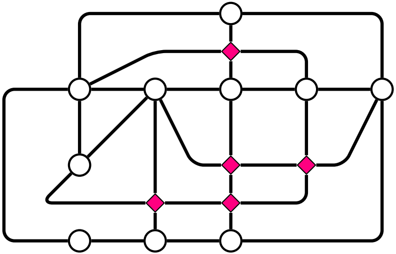


Batini, Talamo, and Tamassia 1984

Planarization Method

Heuristic #1

→ compute planar subgraph, insert remaining edges iteratively



Batini, Talamo, and Tamassia 1984

Planarization Method
Variants

Edge Insertion:

- **fix**ed embedding (Batini, Talamo, and Tamassia 1984)
- **var**iable embedding (Gutwenger, Mutzel, and Weiskircher 2005)
- **multi**ple edge insertion (Chimani and Hlinený 2011)

Planarization Method

Variants

Edge Insertion:

- **fix**ed embedding (Batini, Talamo, and Tamassia 1984)
- **var**iable embedding (Gutwenger, Mutzel, and Weiskircher 2005)
- **multi**ple edge insertion (Chimani and Hlinený 2011)

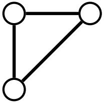
Post-Processing:

- **all**: remove and reinsert every edge at the end (Gutwenger and Mutzel 2003)
- **inc**: perform **all** after each edge insertion (Chimani and Gutwenger 2012)

Chordless Cycle Method

Heuristic #2

→ compute chordless cycle, insert partial stars connected to it

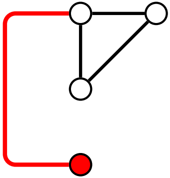


[1] Clancy, Haythorpe, and Newcombe 2019

Chordless Cycle Method

Heuristic #2

→ compute chordless cycle, insert partial stars connected to it

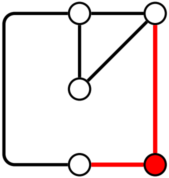


[1] Clancy, Haythorpe, and Newcombe 2019

Chordless Cycle Method

Heuristic #2

→ compute chordless cycle, insert partial stars connected to it

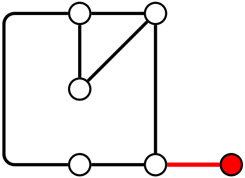


Clancy, Haythorpe, and Newcombe 2019

Chordless Cycle Method

Heuristic #2

→ compute chordless cycle, insert partial stars connected to it

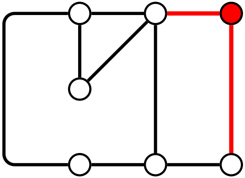


Clancy, Haythorpe, and Newcombe 2019

Chordless Cycle Method

Heuristic #2

→ compute chordless cycle, insert partial stars connected to it

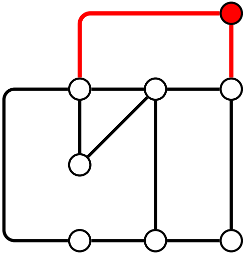


Clancy, Haythorpe, and Newcombe 2019

Chordless Cycle Method

Heuristic #2

→ compute chordless cycle, insert partial stars connected to it

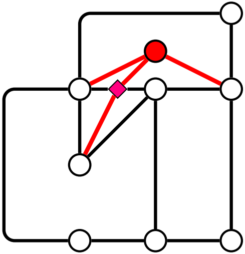


Clancy, Haythorpe, and Newcombe 2019

Chordless Cycle Method

Heuristic #2

→ compute chordless cycle, insert partial stars connected to it

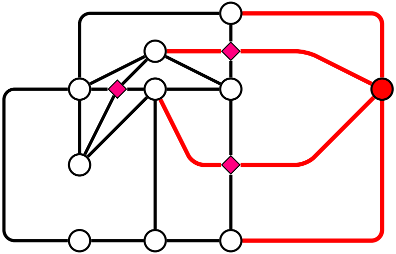


Clancy, Haythorpe, and Newcombe 2019

Chordless Cycle Method

Heuristic #2

→ compute chordless cycle, insert partial stars connected to it

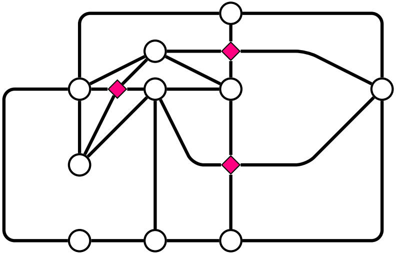


Clancy, Haythorpe, and Newcombe 2019

Chordless Cycle Method

Heuristic #2

→ compute chordless cycle, insert partial stars connected to it

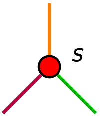


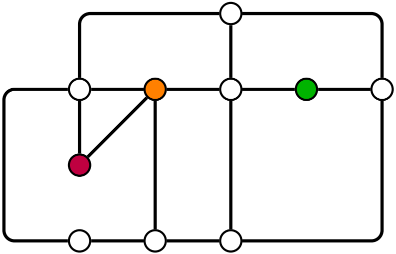
Clancy, Haythorpe, and Newcombe 2019

Star Insertion (Fixed Embedding)

Given: Planar graph G, embedding Π of G, star s not yet in G

Task: Insert s into Π s.t. the number of crossings in Π is minimized



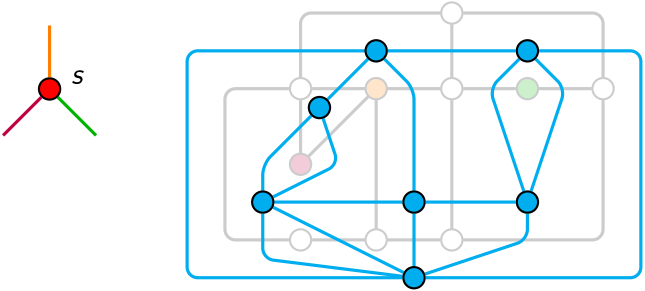


Chimani, Gutwenger, et al. 2009

Star Insertion (Fixed Embedding)

Given: Planar graph G, embedding Π of G, star s not yet in G
Task: Insert s into Π s.t. the number of crossings in Π is minimized

→ add up distance values from breadth-first-searches in dual graph

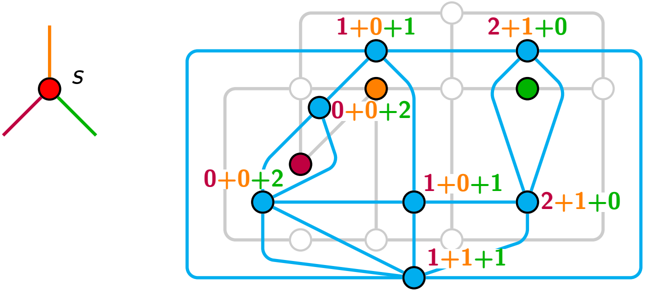


Chimani, Gutwenger, et al. 2009

Star Insertion (Fixed Embedding)

Given: Planar graph G, embedding Π of G, star s not yet in G
Task: Insert s into Π s.t. the number of crossings in Π is minimized

$\rightarrow$ add up distance values from breadth-first-searches in dual graph

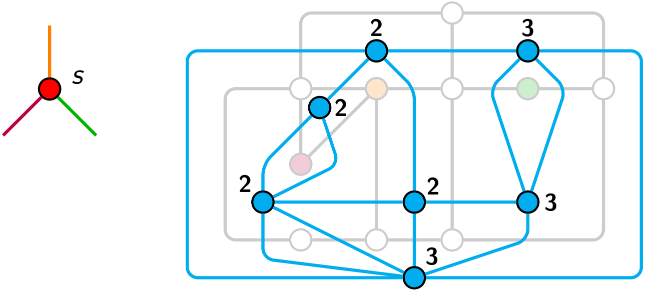


Chimani, Gutwenger, et al. 2009

Star Insertion (Fixed Embedding)

Given: Planar graph G, embedding Π of G, star s not yet in G
Task: Insert s into Π s.t. the number of crossings in Π is minimized

→ add up distance values from breadth-first-searches in dual graph



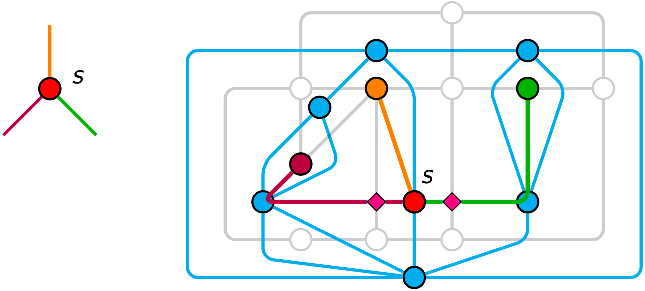
Chimani, Gutwenger, et al. 2009

Star Insertion (Fixed Embedding)

Given: Planar graph G, embedding Π of G, star s not yet in G

Task: Insert s into Π s.t. the number of crossings in Π is minimized

→ add up distance values from breadth-first-searches in dual graph

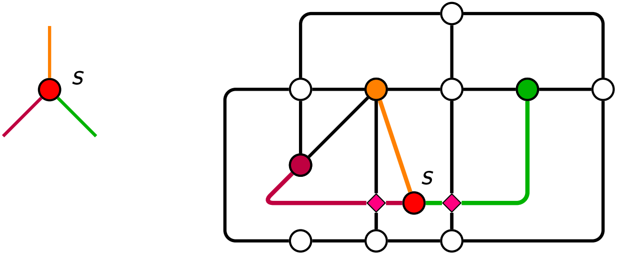


Chimani, Gutwenger, et al. 2009

Star Insertion (Fixed Embedding)

Given: Planar graph G, embedding Π of G, star s not yet in G
Task: Insert s into Π s.t. the number of crossings in Π is minimized

→ add up distance values from breadth-first-searches in dual graph

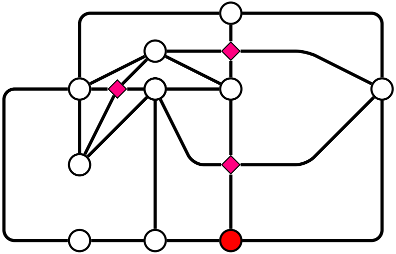


Chimani, Gutwenger, et al. 2009

Star Reinsertion Method

Post-Processing

→ create initial planarization, remove and reinsert stars until no reinsertion can decrease number of crossings

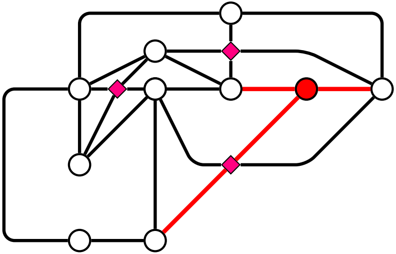


Clancy, Haythorpe, and Newcombe 2019

Star Reinsertion Method

Post-Processing

→ create initial planarization, remove and reinsert stars until no reinsertion can decrease number of crossings

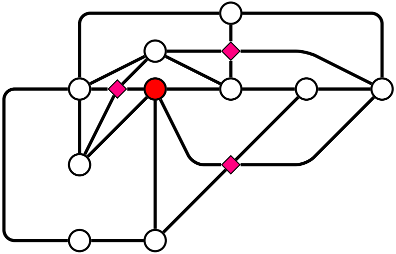


Clancy, Haythorpe, and Newcombe 2019

Star Reinsertion Method

Post-Processing

→ create initial planarization, remove and reinsert stars until no reinsertion can decrease number of crossings

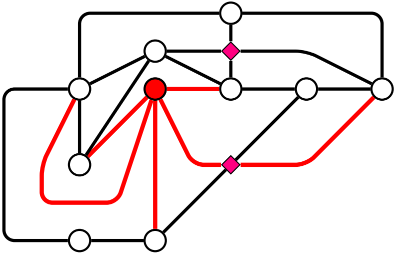


Clancy, Haythorpe, and Newcombe 2019

Star Reinsertion Method

Post-Processing

→ create initial planarization, remove and reinsert stars until no reinsertion can decrease number of crossings

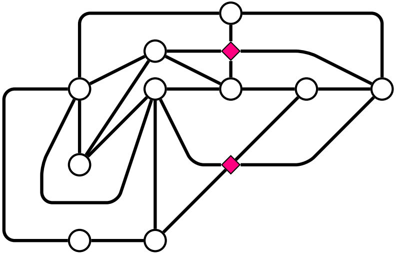


Clancy, Haythorpe, and Newcombe 2019

Star Reinsertion Method

Post-Processing

→ create initial planarization, remove and reinsert stars until no reinsertion can decrease number of crossings

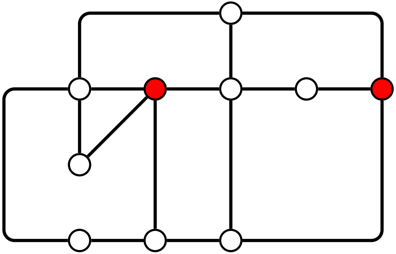


Clancy, Haythorpe, and Newcombe 2019

Mixed Insertion Method

Heuristic #3: Our Own Approach

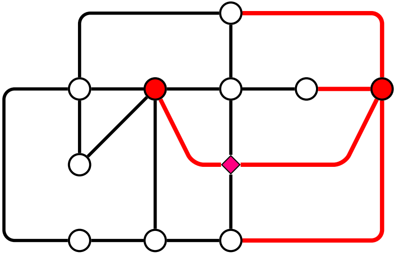
→ compute planar subgraph, insert remaining edges via star insertion of their endpoints (if possible)



Mixed Insertion Method

Heuristic #3: Our Own Approach

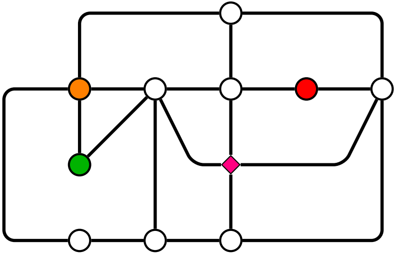
→ compute planar subgraph, insert remaining edges via star insertion of their endpoints (if possible)



Mixed Insertion Method

Heuristic #3: Our Own Approach

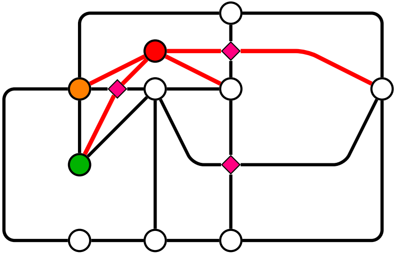
→ compute planar subgraph, insert remaining edges
via star insertion of their endpoints (if possible)



Mixed Insertion Method

Heuristic #3: Our Own Approach

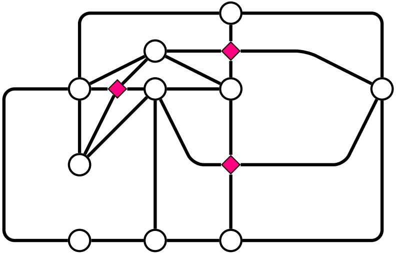
→ compute planar subgraph, insert remaining edges
via star insertion of their endpoints (if possible)



Mixed Insertion Method

Heuristic #3: Our Own Approach

→ compute planar subgraph, insert remaining edges via star insertion of their endpoints (if possible)



Evaluation

Setup:

- C++ (GCC 8.3.0), Open Graph Drawing Framework (OGDF)
- single physical processor of a Xeon Gold 6134 CPU (3.2 GHz)
- memory limit of 4 GB, no time limit

[1] Chimani and Gutwenger 2009

Evaluation

Setup:

- C++ (GCC 8.3.0), Open Graph Drawing Framework (OGDF)
- single physical processor of a Xeon Gold 6134 CPU (3.2 GHz)
- memory limit of 4 GB, no time limit

Pre-Processing:

- non-planar core reduction[1]
- precomputed planar subgraph and chordless cycle

[1] Chimani and Gutwenger 2009

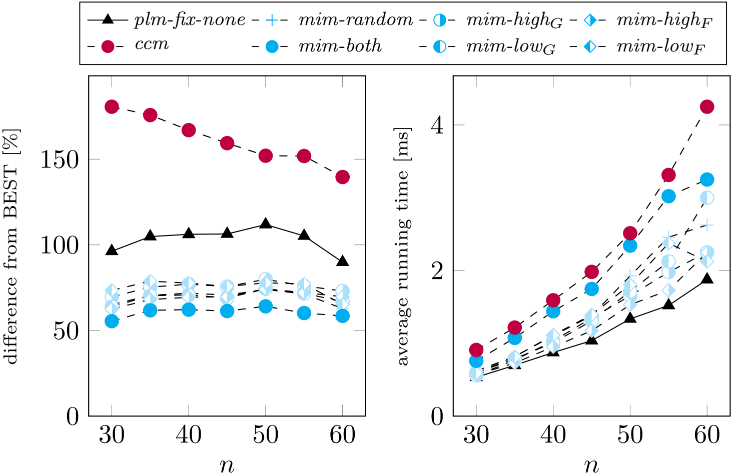
Instances

	$\geq$ 25 vertices in NPC	
Name	#	$\lvert V(G) \rvert$
Rome	3668	25–58
North	106	25–64
Webcompute	75	25–112
Expanders	240	30–100
Circuit-Based	45	26–3045
ISCAS-85	9	180–3045
ISCAS-89	24	60–584
ITC-99	12	26–980

	all instances	
Name	#	$\lvert V(G) \rvert$
Complete	46	5–50
Complete-Bip.	666	10–80
KnownCR	1946	9–250
$C \,\square\, C$	251	9–250
$G \,\square\, P$	893	15–245
$G \,\square\, C$	624	15–250
$P(\_, \_)$	178	10–250

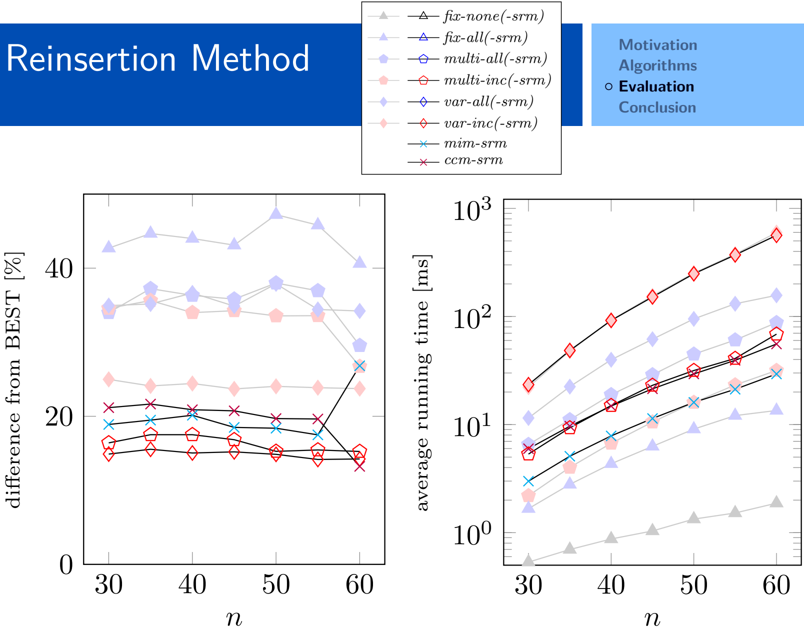
Fast Heuristics

Rome



Star Reinsertion Method

Rome



Conclusion

- **fixed star reinsertion** beats variable edge insertion
- **mixed insertion method:** best quality among fast heuristics
- ***srm*-post-processing** improves all heuristics

Conclusion

- **fixed star reinsertion** beats variable edge insertion
- **mixed insertion method:** best quality among fast heuristics
- ***srm*-post-processing** improves all heuristics

- **permutations** useful for heuristics with *srm*-post-processing
- easy improvements via removal of **non-simple crossings**

Conclusion

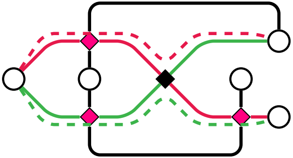
- **fixed star reinsertion** beats variable edge insertion
- **mixed insertion method:** best quality among fast heuristics
- ***srm*-post-processing** improves all heuristics

- **permutations** useful for heuristics with *srm*-post-processing
- easy improvements via removal of **non-simple crossings**

Thank you! Any questions?

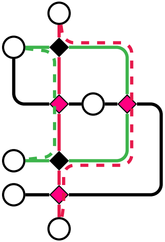
Non-simple Crossings

Removal



α-**crossing**

adjacent edges

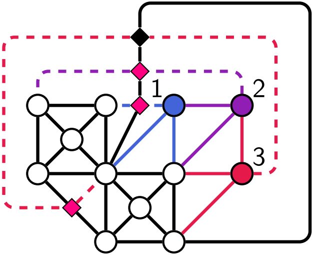


β-**crossing**

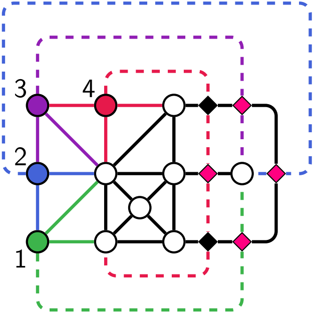
two edges, multiple crossings

Non-simple Crossings

Creation



α-**crossing**

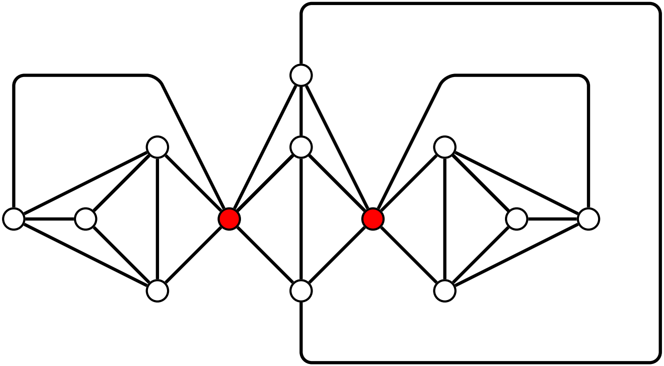


β-**crossing**

Mixed Insertion Method

Heuristic #3: Our Own Approach

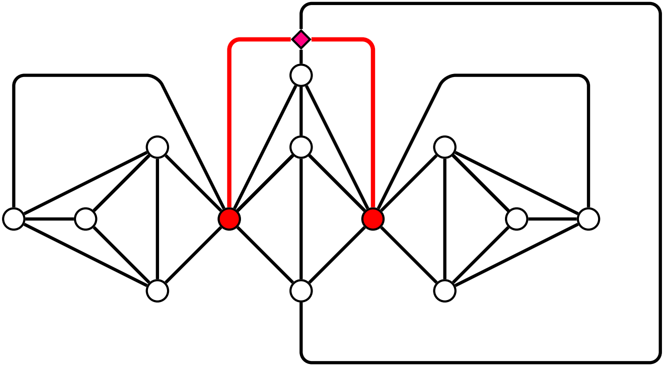
→ compute planar subgraph, insert remaining edges
via star insertion of their endpoints (if possible)



Mixed Insertion Method

Heuristic #3: Our Own Approach

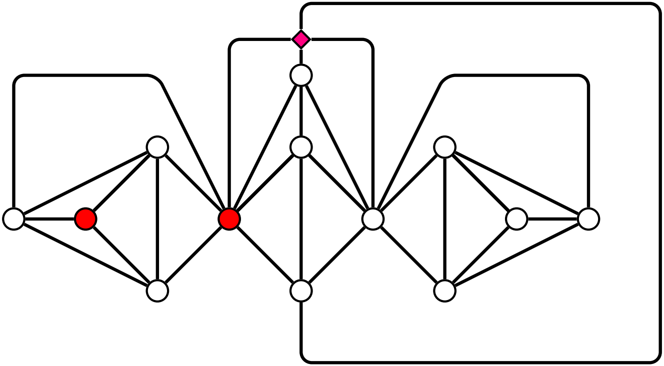
→ compute planar subgraph, insert remaining edges via star insertion of their endpoints (if possible)



Mixed Insertion Method

Heuristic #3: Our Own Approach

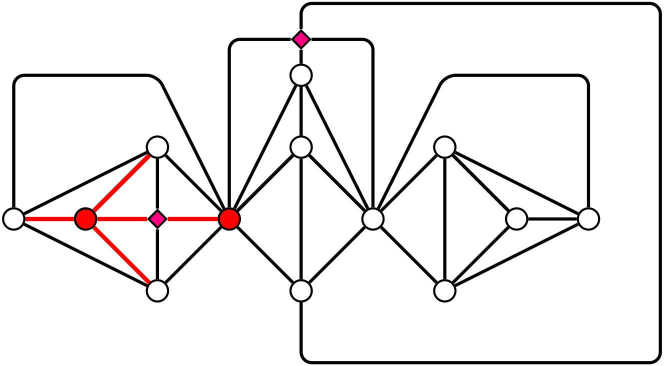
→ compute planar subgraph, insert remaining edges via star insertion of their endpoints (if possible)



Mixed Insertion Method

Heuristic #3: Our Own Approach

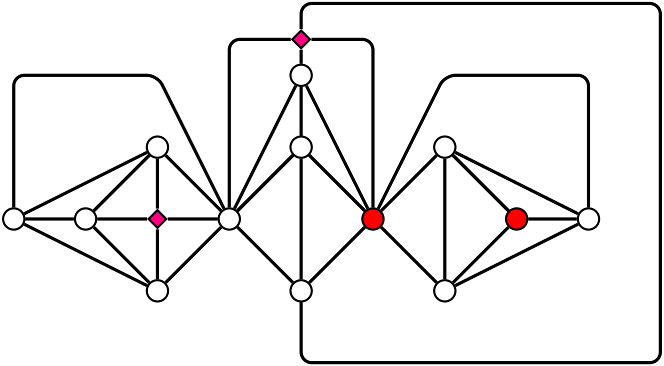
→ compute planar subgraph, insert remaining edges
via star insertion of their endpoints (if possible)



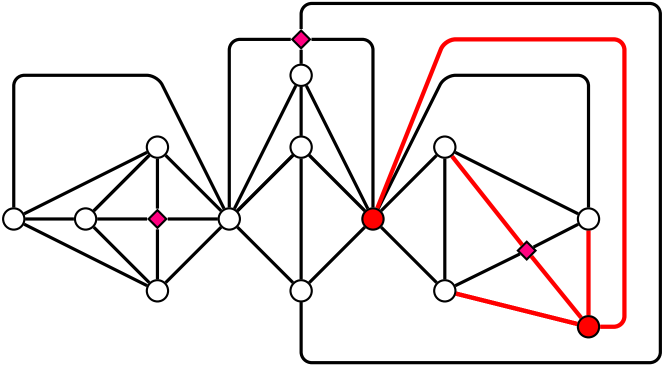
Mixed Insertion Method

Heuristic #3: Our Own Approach

→ compute planar subgraph, insert remaining edges via star insertion of their endpoints (if possible)



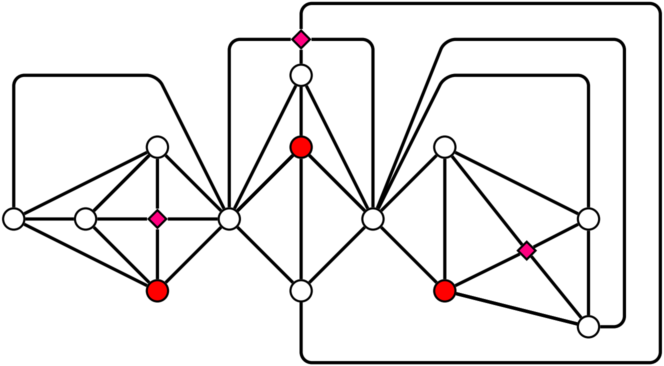
→ compute planar subgraph, insert remaining edges via star insertion of their endpoints (if possible)



Mixed Insertion Method

Heuristic #3: Our Own Approach

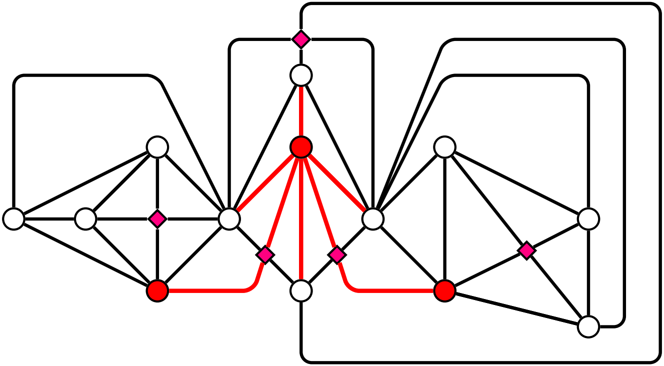
→ compute planar subgraph, insert remaining edges
via star insertion of their endpoints (if possible)



Mixed Insertion Method

Heuristic #3: Our Own Approach

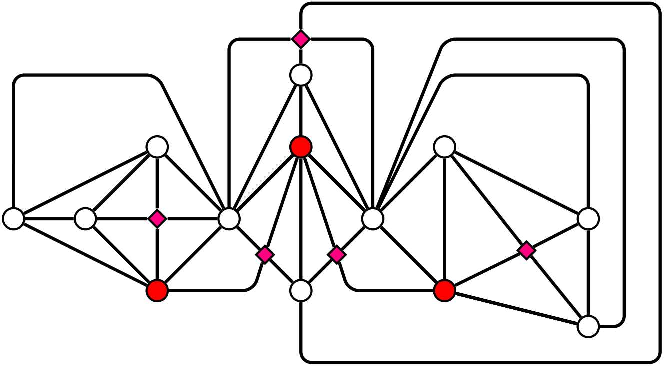
→ compute planar subgraph, insert remaining edges via star insertion of their endpoints (if possible)



Mixed Insertion Method

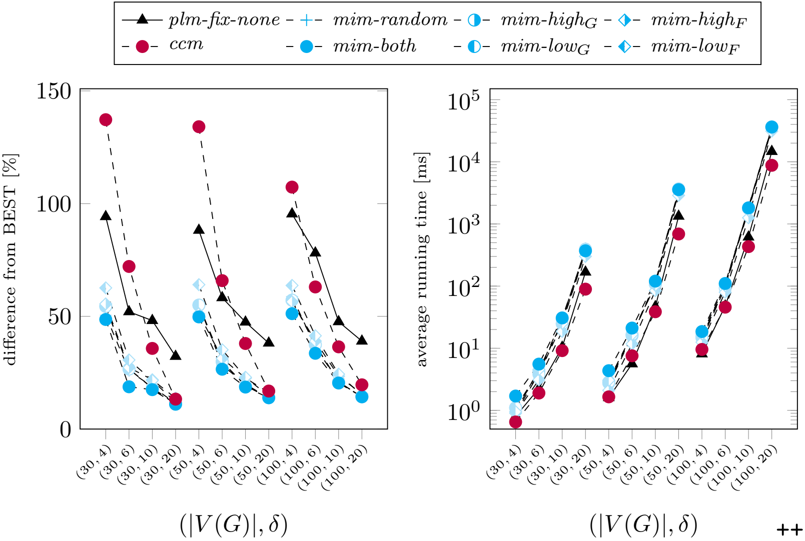
Heuristic #3: Our Own Approach

→ compute planar subgraph, insert remaining edges via star insertion of their endpoints (if possible)



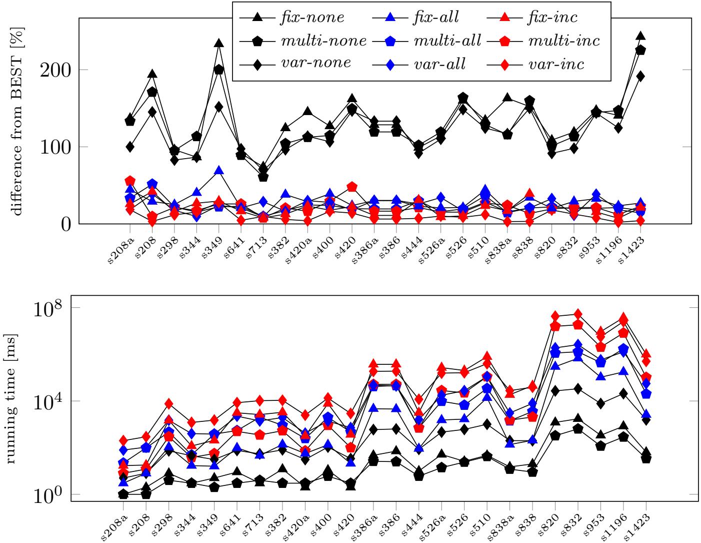
Fast Heuristics

Expanders



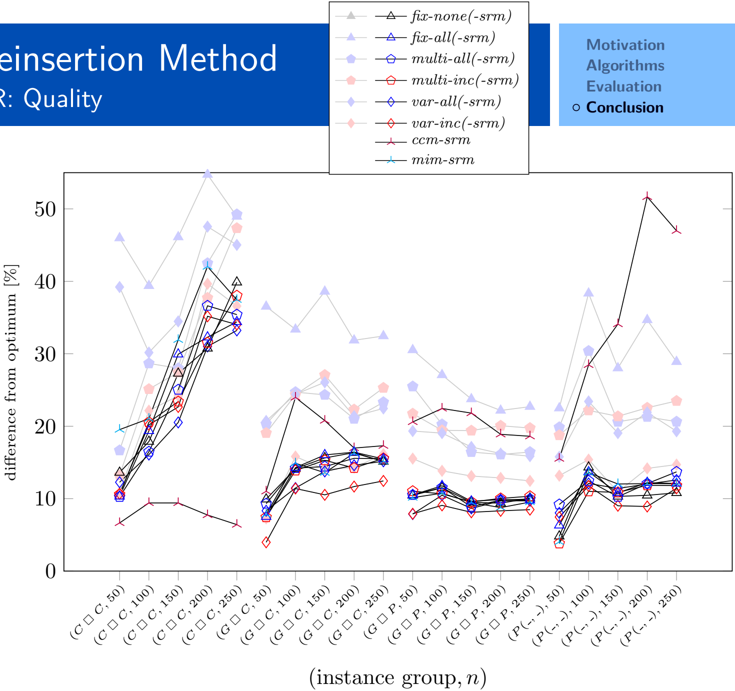
Planarization Method

ISCAS-89



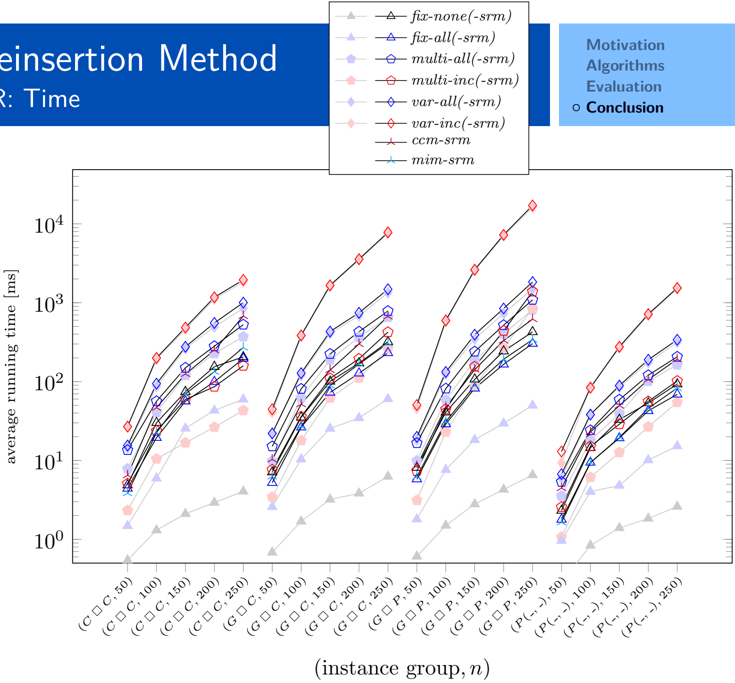
Star Reinsertion Method

KnownCR: Quality



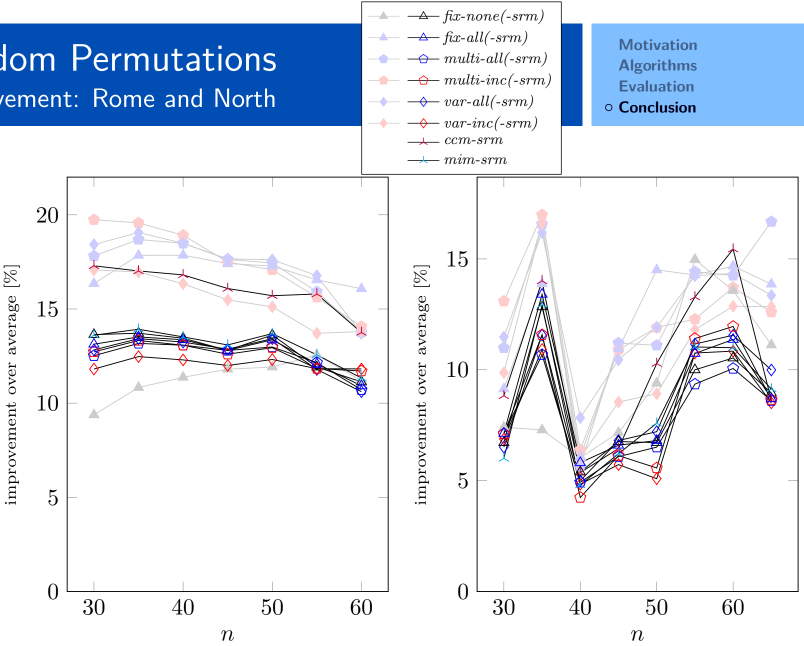
Star Reinsertion Method

KnownCR: Time



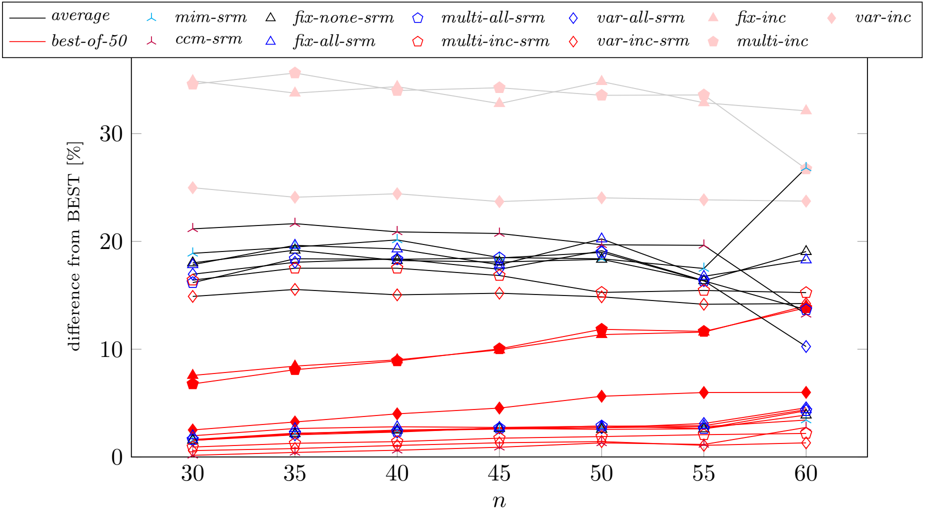
Random Permutations

Improvement: Rome and North



Random Permutations

Rome



Literature I

Batini, C., M. Talamo, and R. Tamassia (1984). "Computer aided layout of entity relationship diagrams". In: *J. Syst. Softw.* 4.2-3, pp. 163–173. URL: https://doi.org/10.1016/0164-1212(84)90006-2.

Chimani, M. and C. Gutwenger (2009). "Non-planar core reduction of graphs". In: *Discret. Math.* 309.7, pp. 1838–1855. URL: https://doi.org/10.1016/j.disc.2007.12.078.

— (2012). "Advances in the Planarization Method: Effective Multiple Edge Insertions". In: *J. Graph Algorithms Appl.* 16.3, pp. 729–757. URL: https://doi.org/10.7155/jgaa.00264.

Chimani, M., C. Gutwenger, et al. (2009). "Inserting a vertex into a planar graph". In: *Proc. SODA 2009*. SIAM, pp. 375–383. URL: http://dl.acm.org/citation.cfm?id=1496770.1496812.

Chimani, M. and P. Hlinený (2011). "A tighter insertion-based approximation of the crossing number". In: *Proc. ICALP 2011*. Vol. 6755. LNCS. Springer, pp. 122–134. URL: https://doi.org/10.1007/978-3-642-22006-7%5C_11.

Literature II

Clancy, K., M. Haythorpe, and A. Newcombe (2019). "An effective crossing minimisation heuristic based on star insertion". In: *J. Graph Algorithms Appl.* 23.2, pp. 135–166. URL: https://doi.org/10.7155/jgaa.00487.

Gutwenger, C. and P. Mutzel (2003). "An Experimental Study of Crossing Minimization Heuristics". In: *GD 2003: Revised Papers*. Vol. 2912. LNCS. Springer, pp. 13–24. URL: https://doi.org/10.1007/978-3-540-24595-7_2.

Gutwenger, C., P. Mutzel, and R. Weiskircher (2005). "Inserting an Edge into a Planar Graph". In: *Algorithmica* 41.4, pp. 289–308. URL: https://doi.org/10.1007/s00453-004-1128-8.