ImPrEd

An Improved Force-Directed Algorithm that Prevents Nodes from Crossing Edges

Authors:

Paolo Simonetto
Daniel Archambault
David Auber
Romain Bourqui

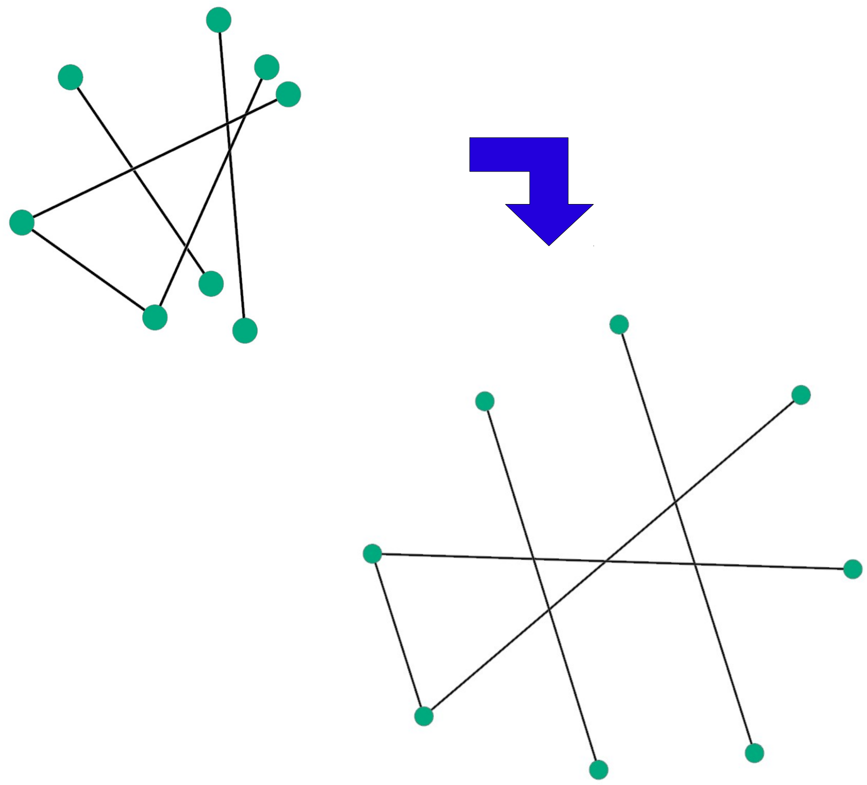
LaBRI, Université Bordeaux 1

INRIA Bordeaux Sud-Ouest

University College Dublin

PrEd - Introduction

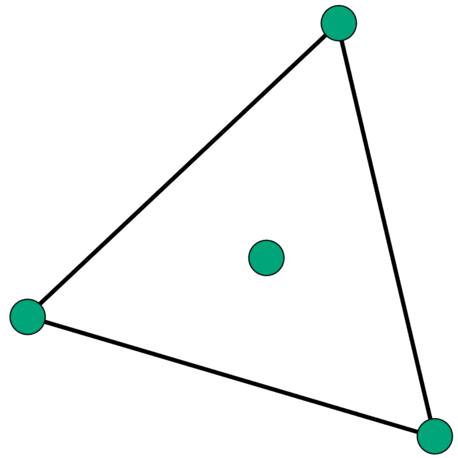
- PrEd is a force-directed algorithm that optimises an existing layout.
- PrEd preserves edge crossing properties:
  - no new crossings created,
  - no old crossings undone.
- PrEd can be used for:
  - planar graph drawing,
  - interactive layout.

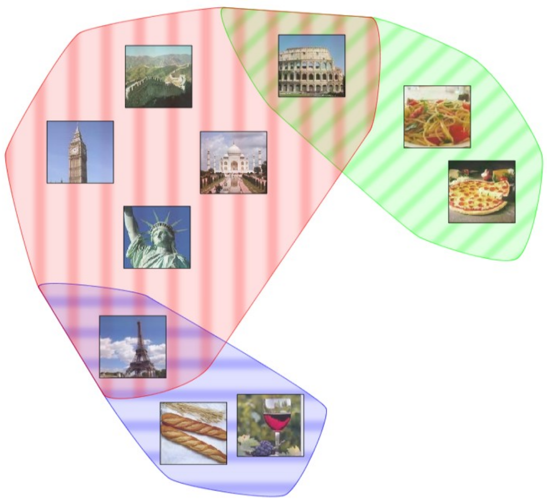


François Bertault. ‘A Force-Directed Algorithm that Preserves Edge Crossing Properties’. In: *Information Processing Letters* 74.1–2 (Apr. 2000), pp. 7–13.

PrEd – The principle

- When moving, nodes must not cross edges:
  - edge crossings properties preserved,
  - node position can be constrained using edges.



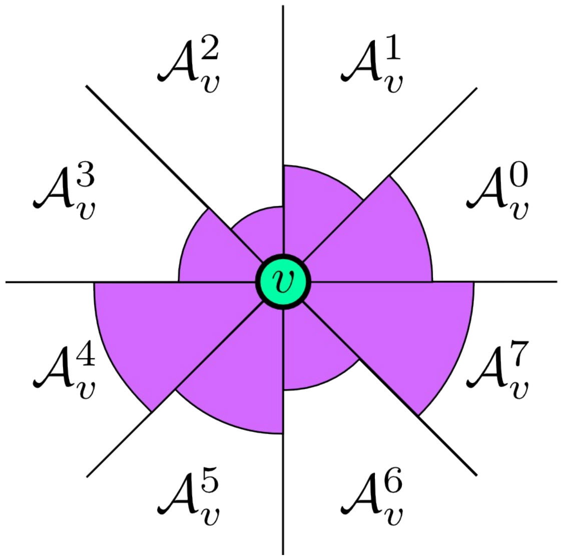


- PrEd can be used to generate Euler diagrams.

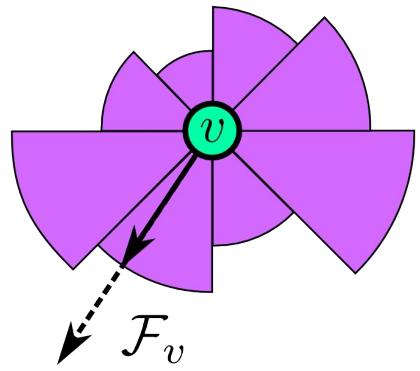
  Paolo Simonetto, David Auber and Daniel Archambault. 'Fully Automatic Visualisation of Overlapping Sets'. In: *Computer Graphics Forum* (EuroVis09) 28.3 (June 2009), pp. 967–974.

PrEd – The Algorithm

- At each iteration:
  - Max movement *M* computation:
    - According to 8 sectors.
  - Force *F* computation:
    - node-node repulsion,
    - adjacent node attraction,
    - edge-node repulsion.
  - Node displacement:
    - along vector *F*,
    - max distance according to *M*.



$$\angle \mathcal{F}_v \in \mathcal{A}_v^5 \rightarrow ||\mathcal{F}_v|| < \mathcal{M}_v^5$$



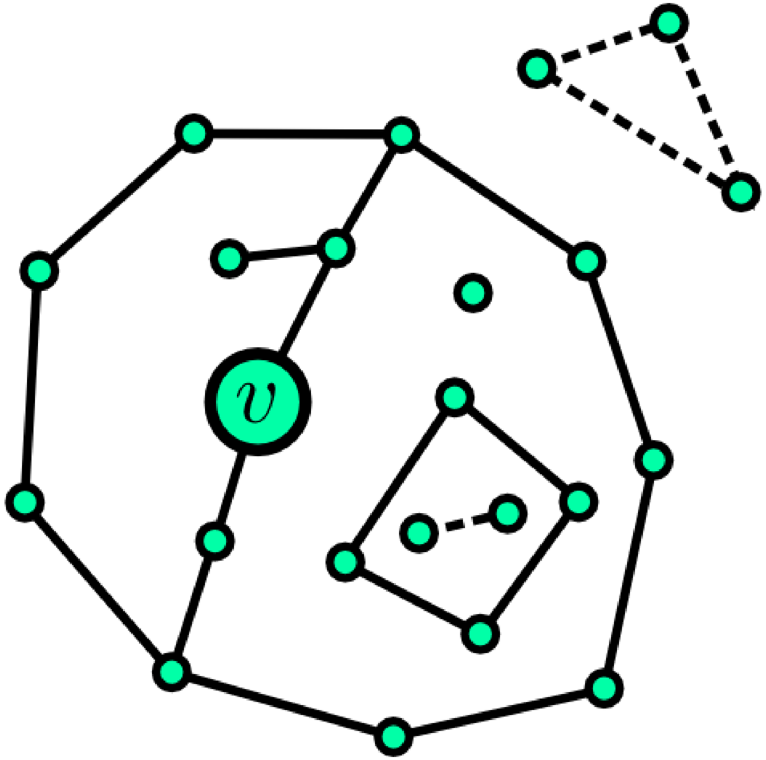
Improving PrEd

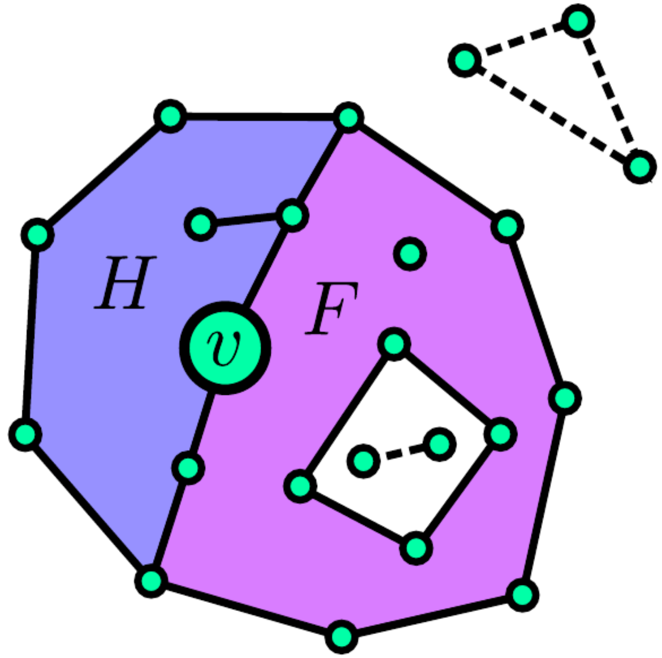
- PrEd limitations:
  1. computationally expensive,
  2. overestrictive in terms of node movements,
  3. sub-optimal in terms of layout quality,
  4. not very flexible.

- ImPrEd mainly features:
  - surrounding edges (1),
  - application of QuadTrees (as in previous work) (1),
  - new max movement rules (2),
  - force system cooling (3),
  - crossable and flexible edges (3-4).

ImPrEd – Surrounding Edges

Max movement needs to be computed for every node/edge?





$$\mathcal{S}_v = B^e(F) \cup B^e(H)$$

In the paper: how to identify a face with disconnected boundaries and crossings.

ImPrEd – Surrounding Edges

- What:
  - improve the running time.
- How:
  - compute for each node its surrounding edges.
- Where:
  - computation as a pre-processing step,
  - used in max movement and in node-edge repulsion.

ImPrEd – Application of QuadTrees

- What:
  - improve the running time.
- How:
  - efficiently detect the nearby elements of a node,
  - distant elements are ignored.
- Where:
  - computation at the beginning of each iteration,
  - used in node-node and node-edge repulsion,
  - used in max movement, because of global max movement.

ImPrEd – Surrounding Edges and QuadTrees

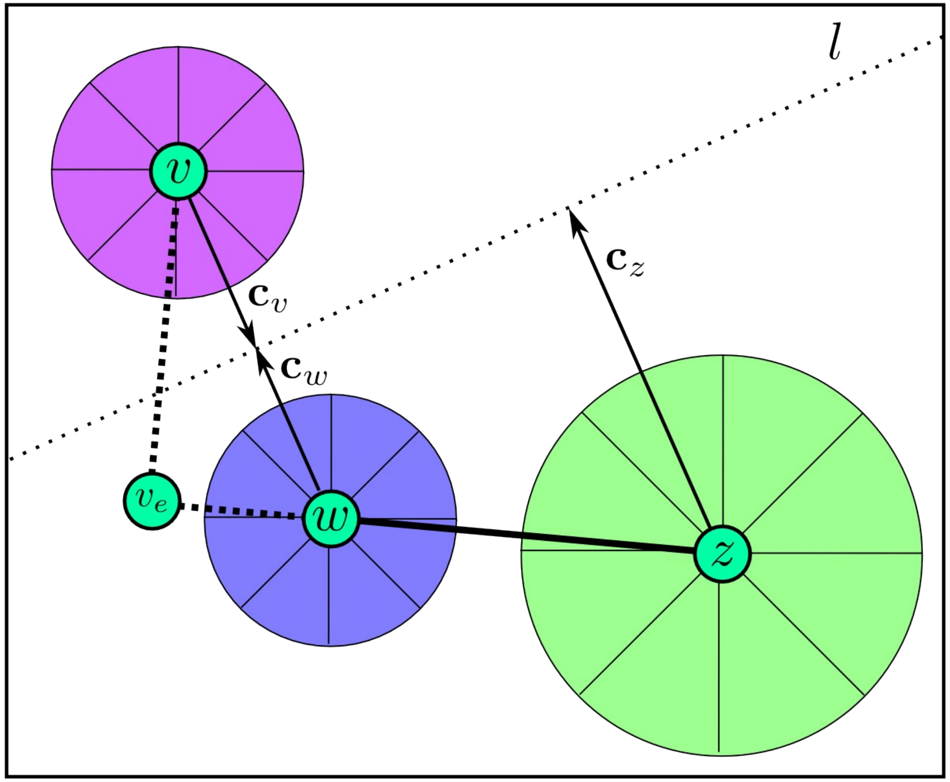
- What if we have a tree?
  - One face → Surrounding edges useless,
  - Nodes lightly constrained → QuadTrees are efficient.

- What if the graph is very intricate?
  - Many faces → Surrounding edges efficient,
  - Nodes heavily constrained → QuadTrees lose efficiency.

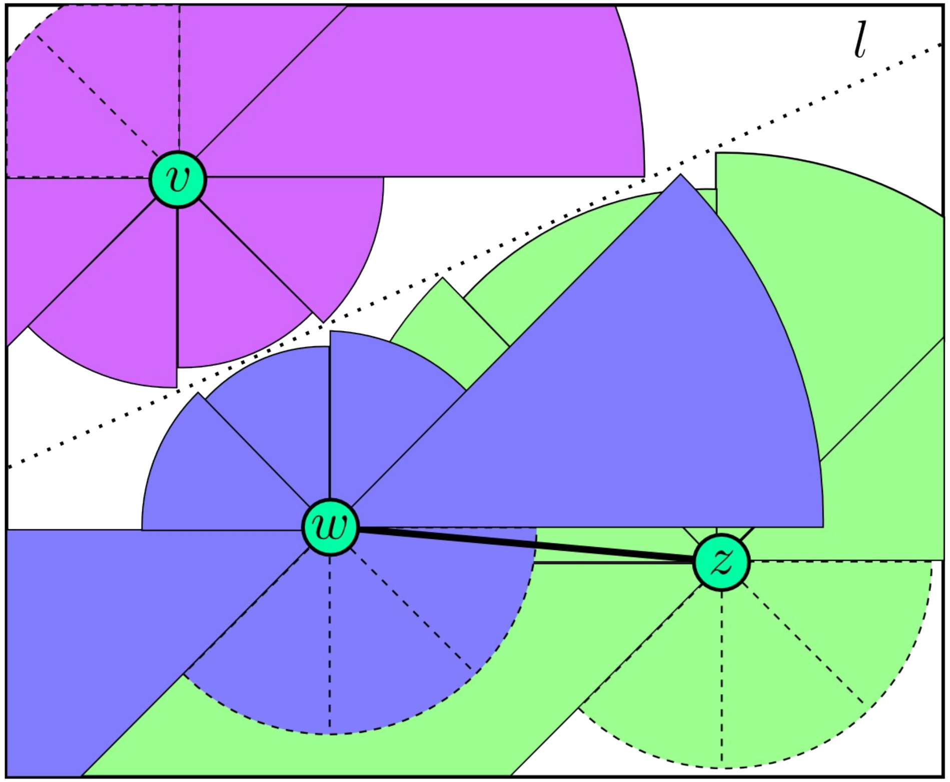
- Surrounding edges and QuadTrees combines well:
  - Sparse graphs: SE low, QT high
  - Dense graphs: QT low, SE high

ImPrEd – New Rules for Max Movement

ImPrEd maximises the sectors when computing the max movement.



PrEd



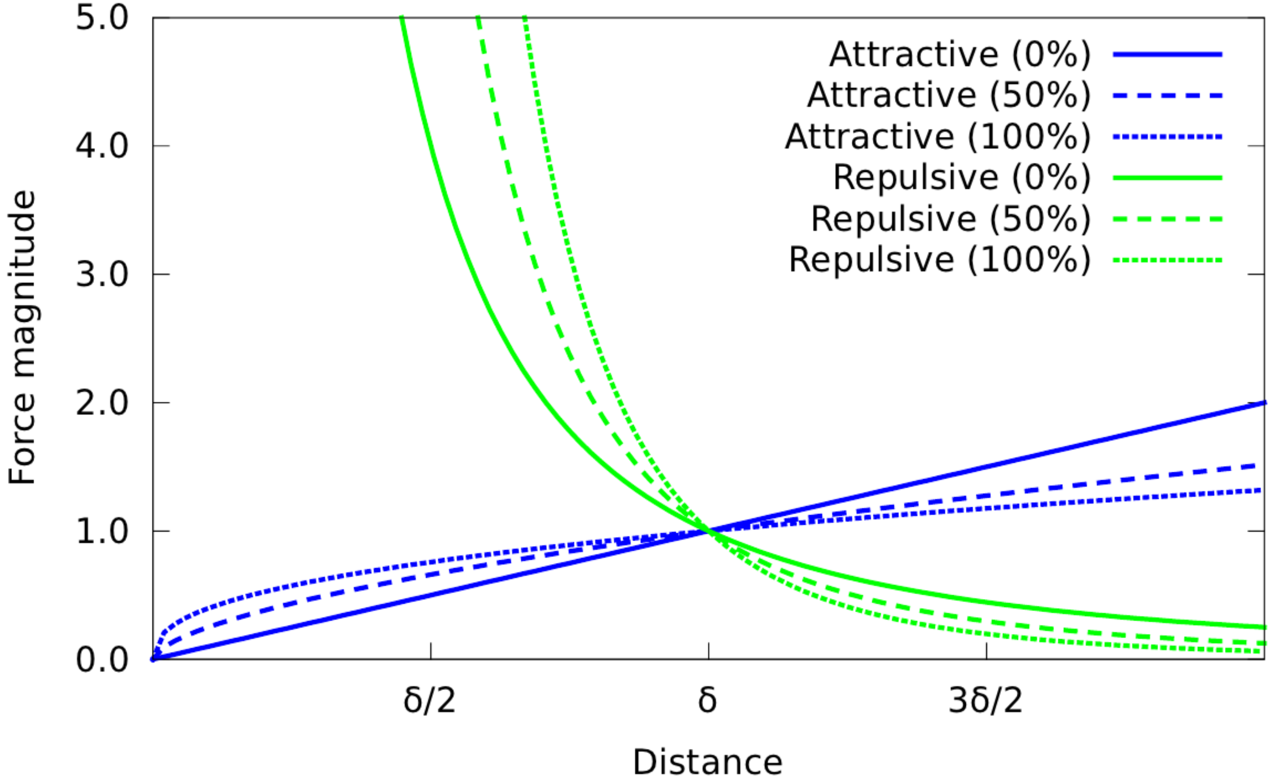
ImPrEd

ImPrEd – New Rules for Max Movement

- What:
  - improves the running time / drawing quality.
- How:
  - increase the node mobility, accelerating convergence.
- Where:
  - changes the max movement computation rules.

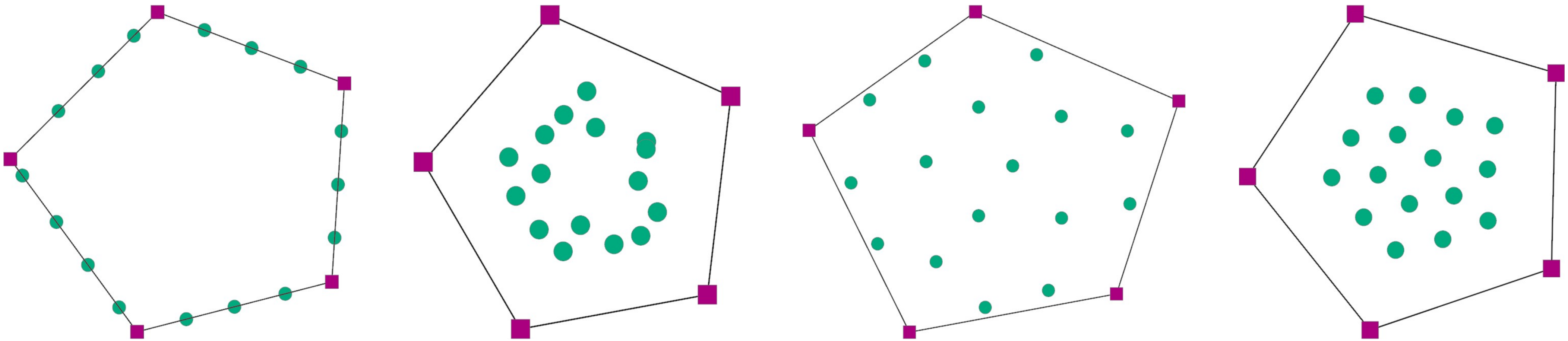
ImPrEd – Force System Cooling

- We gradually adapt the forces through the iterations:
  - increase the exponent of repulsive forces,
  - decrease the exponent of attractive forces.



ImPrEd – Force System Cooling

- δ is the optimal edge length, Υ is the optimal node-edge distance.



$\delta = 5$	$\delta = 2$	$\delta = 5$	$\delta = 2$
$\Upsilon = 2$	$\Upsilon = 5$	$\Upsilon = 2$	$\Upsilon = 5$

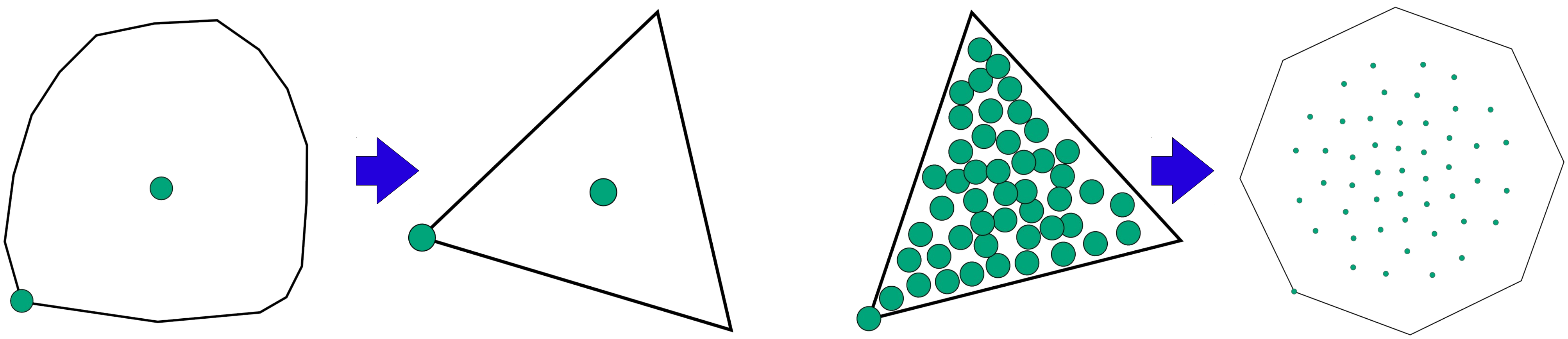
PrEd ImPrEd

ImPrEd – Force System Cooling

- What:
  - improves the quality of the drawing,
  - improves the reliability of the parameters.
- How:
  - gradually shifts the effect of the forces from global to local.
- Where:
  - exponents at force computation.

ImPrEd – Crossable and Flexible Edges

- Input class expanded to polyline multigraphs.
- Each edge can be labelled as:
  - **Uncrossable/Crossable**: crossable edges influence forces but not max movement,
  - **Rigid/Flexible**: flexible edges can expand and contract (change the number of bends) according to their stress.



ImPrEd – Crossable and Flexible Edges

- What:
  - improve the quality of the drawing,
  - improve the flexibility of the algorithm.
- How:
  - allow polyline edges,
  - define uncrossable/crossable and rigid/flexible edges.
- Where:
  - pre and post processing to handle polyline edges,
  - every *x* iterations, flexible edges are updated.

Results – Speed-Up

We executed PrEd and ImPrEd on:

- 10 planar + 10 non-planar graphs of 50, 70, 100, 150 and 200 nodes randomly generated,
- 2 intersection and grid graphs of real Euler diagrams,

with 100, 250 and 500 iterations.

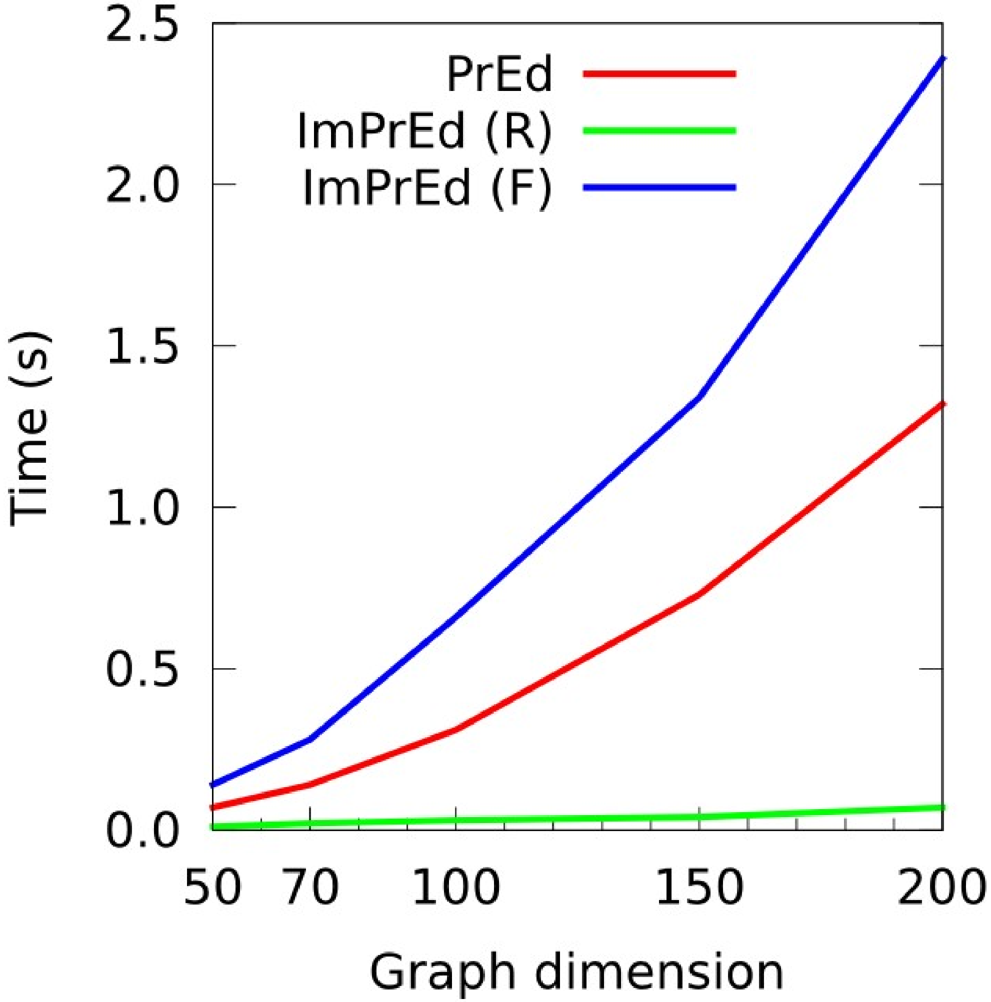
We obtained that:

- ImPrEd performed consistently faster,
- flexible edges introduce a quality/time tradeoff,
- the overhead introduced is negligible even for small graphs.

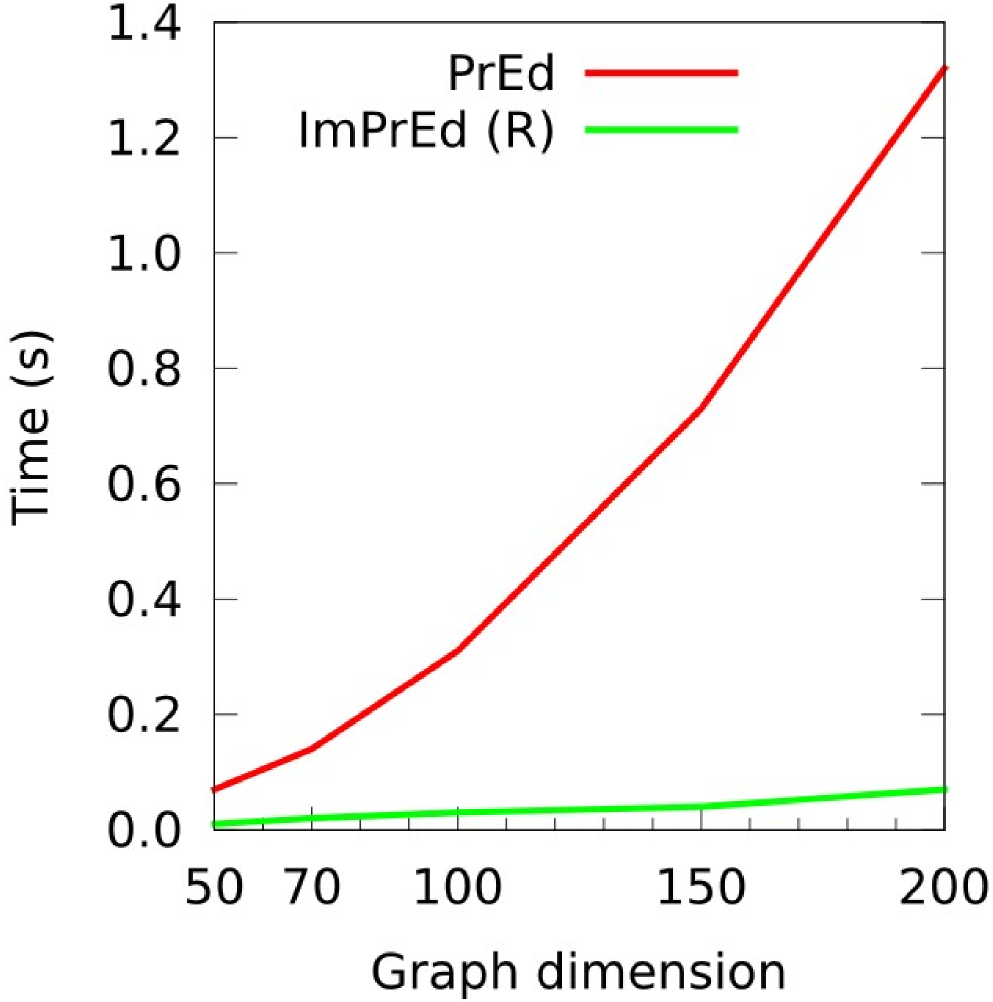
The worst case complexity is unchanged.

Results – Speed-Up

Random planar



Random non-planar

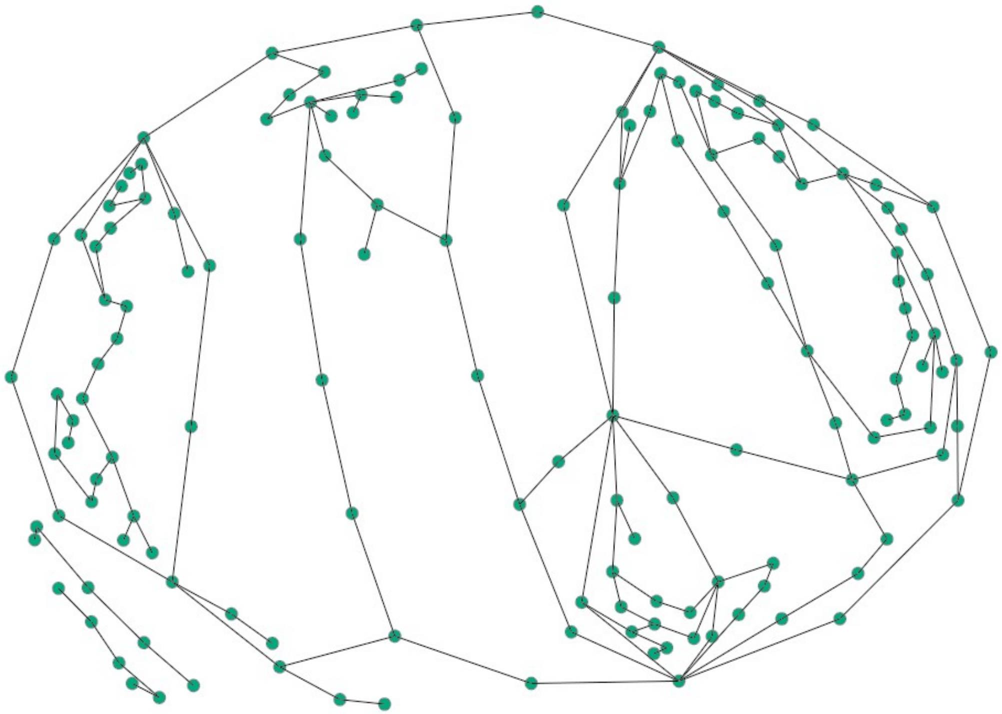


Results – Speed-Up

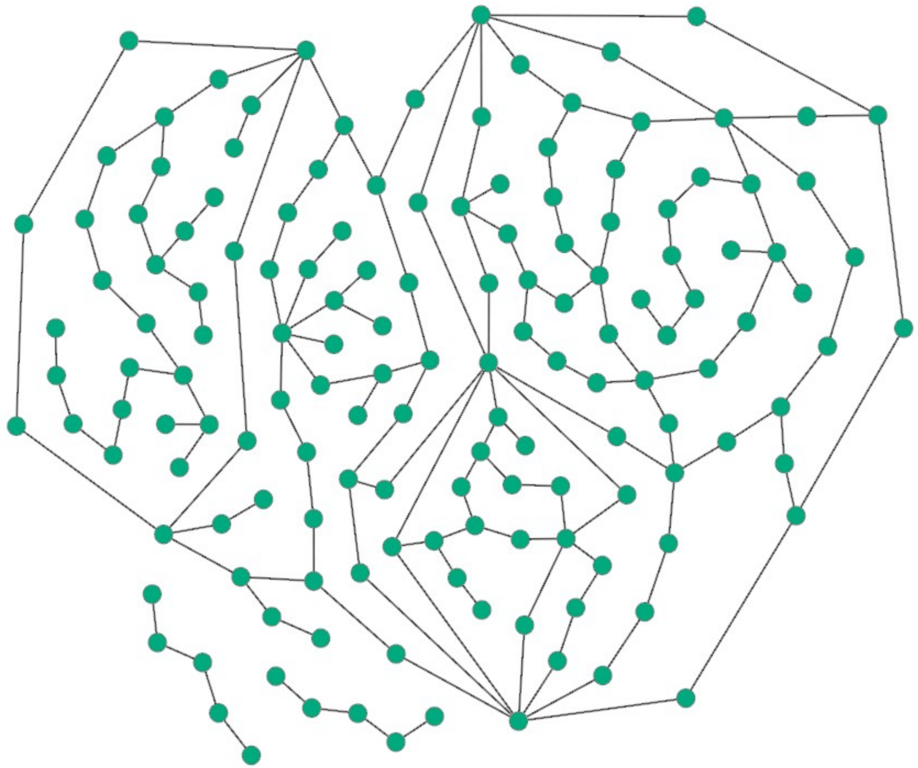
Graph		iGraphA	gGraphA	iGraphB	gGraphB
Nodes		149	2601	23	220
Edges		169	507	22	362
PrEd	100	17.50	2721.79	0.51	—
	250	40.91	6792.53	1.28	—
	500	80.46	13 585.36	2.53	—
ImPrEd (R)	100	4.90	131.55	0.30	15.65
	250	11.81	308.83	0.73	33.49
	500	23.34	597.55	1.42	65.34
ImPrEd (F)	100	6.68	97.22	0.47	9.25
	250	22.02	215.62	1.73	18.07
	500	35.29	409.18	3.79	30.81
ImPrEd (PP)	—	0.13	5.64	0.02	0.14
ImPrEd (R)	Gain	3.49	21.81	1.75	—
ImPrEd (F)	Gain	2.25	30.90	0.73	—

Results – Drawing Quality – iGraph

ImPrEd generates a more precise and regular node spacing.



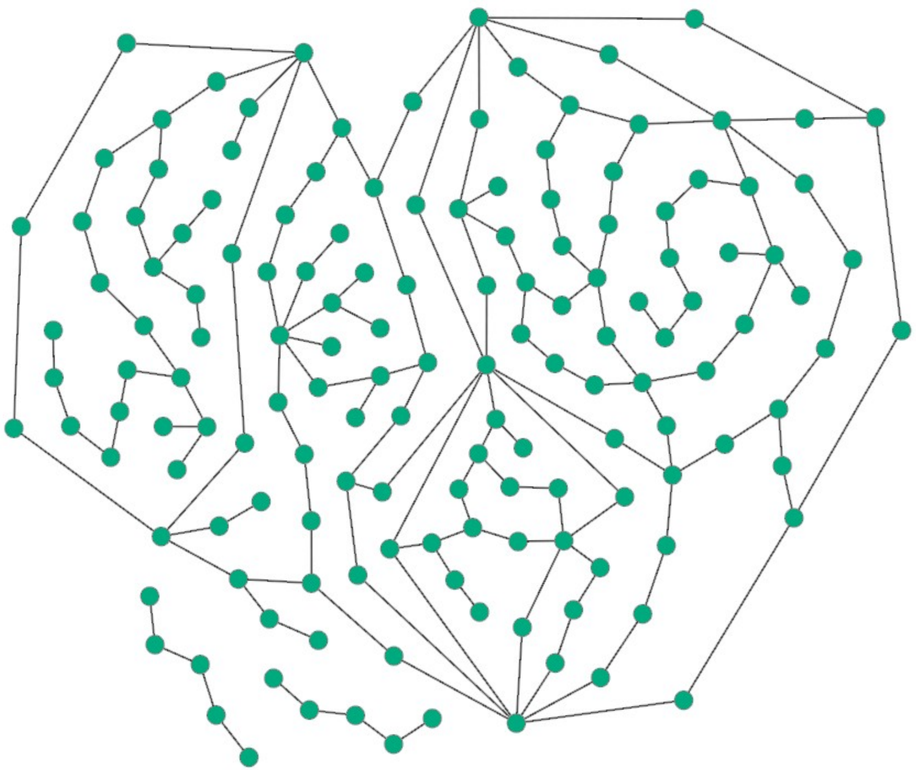
PrEd



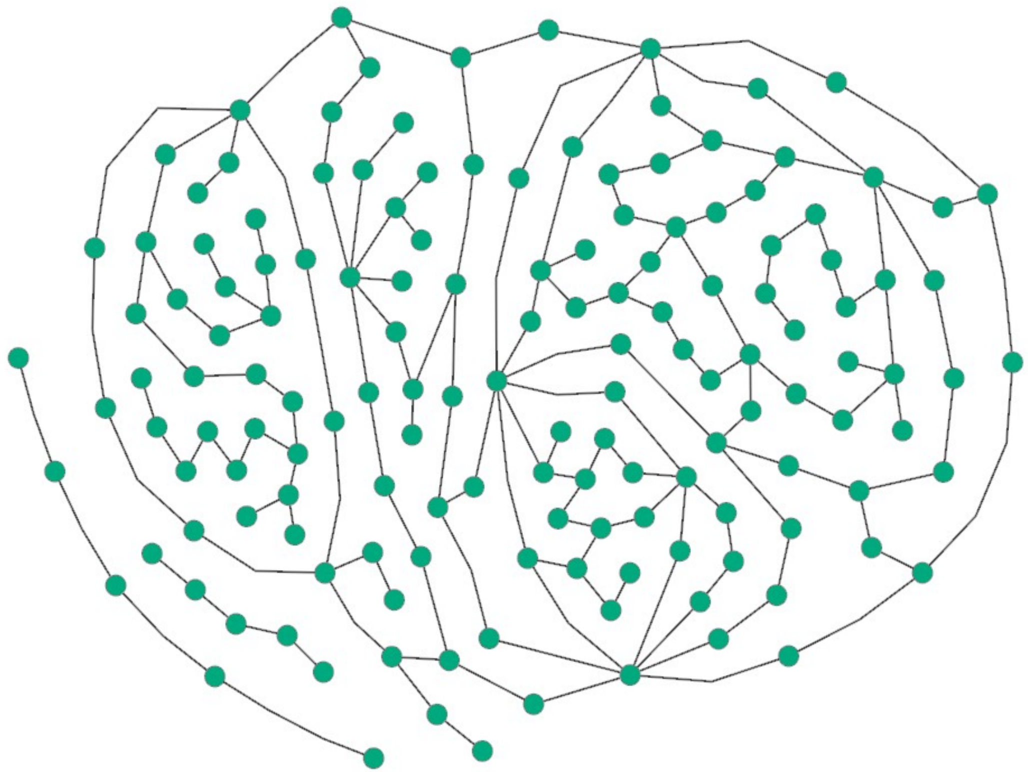
ImPrEd (R)

Results – Drawing Quality – iGraph

Flexible edges help improving the angular resolution.

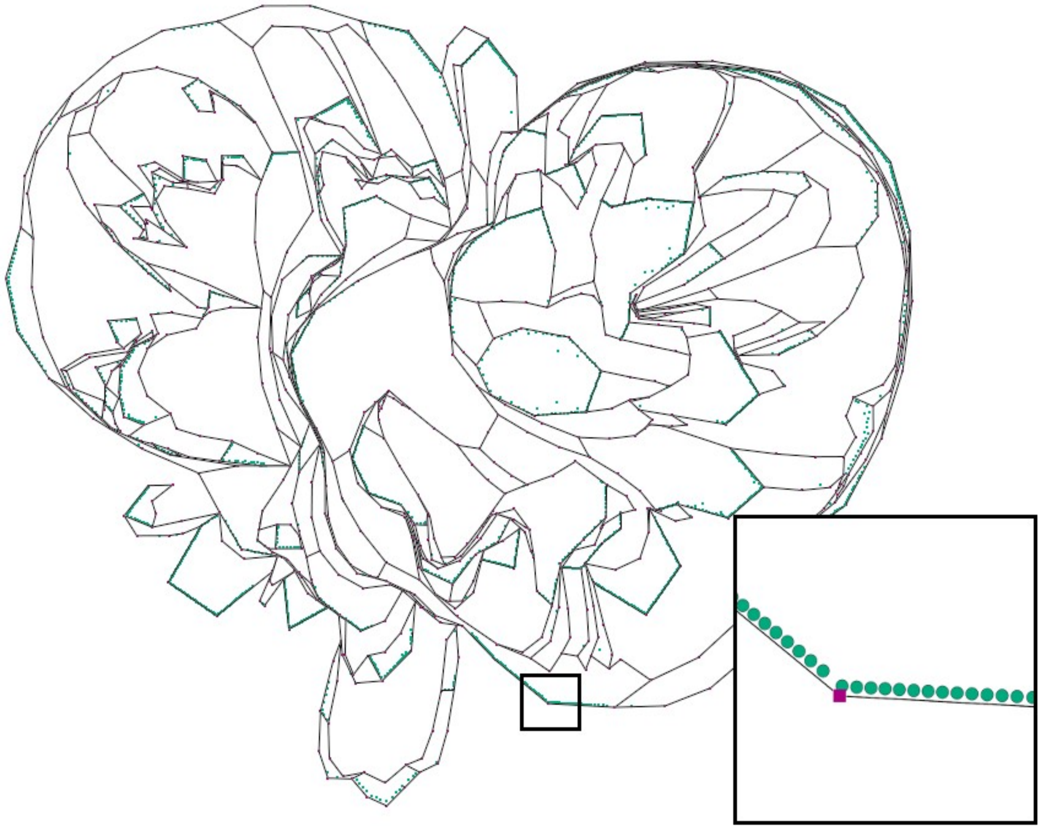


ImPrEd (R)



ImPrEd (F)

Results – Drawing Quality – gGraph

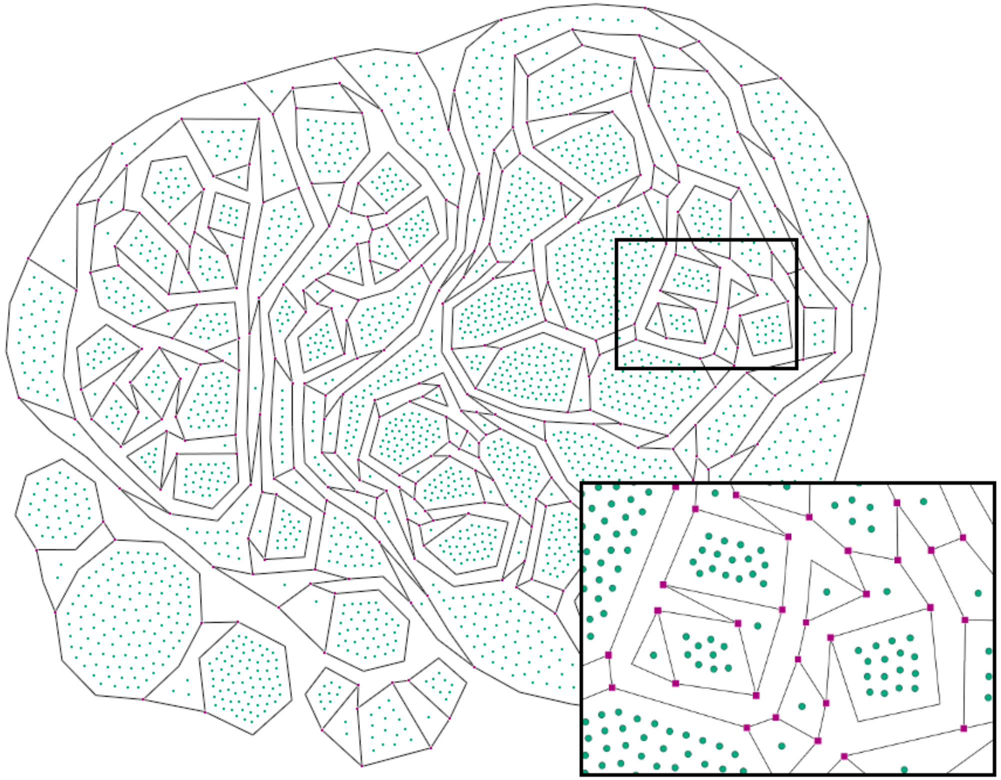


PrEd

ImPrEd (R)

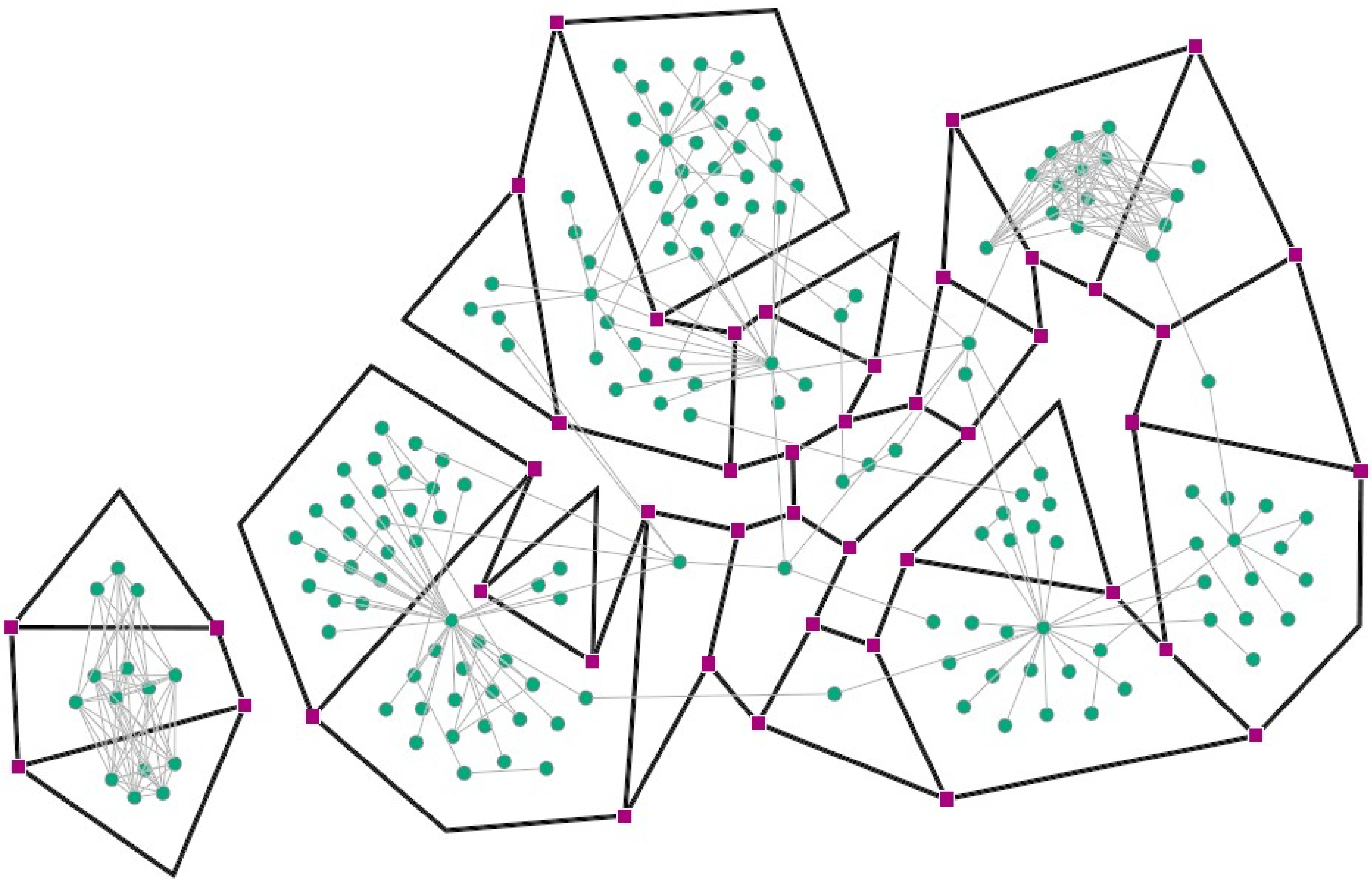
Results – Drawing Quality – gGraph

ImPrEd (R)



ImPrEd (F)

Results – Drawing Quality and Flexibility

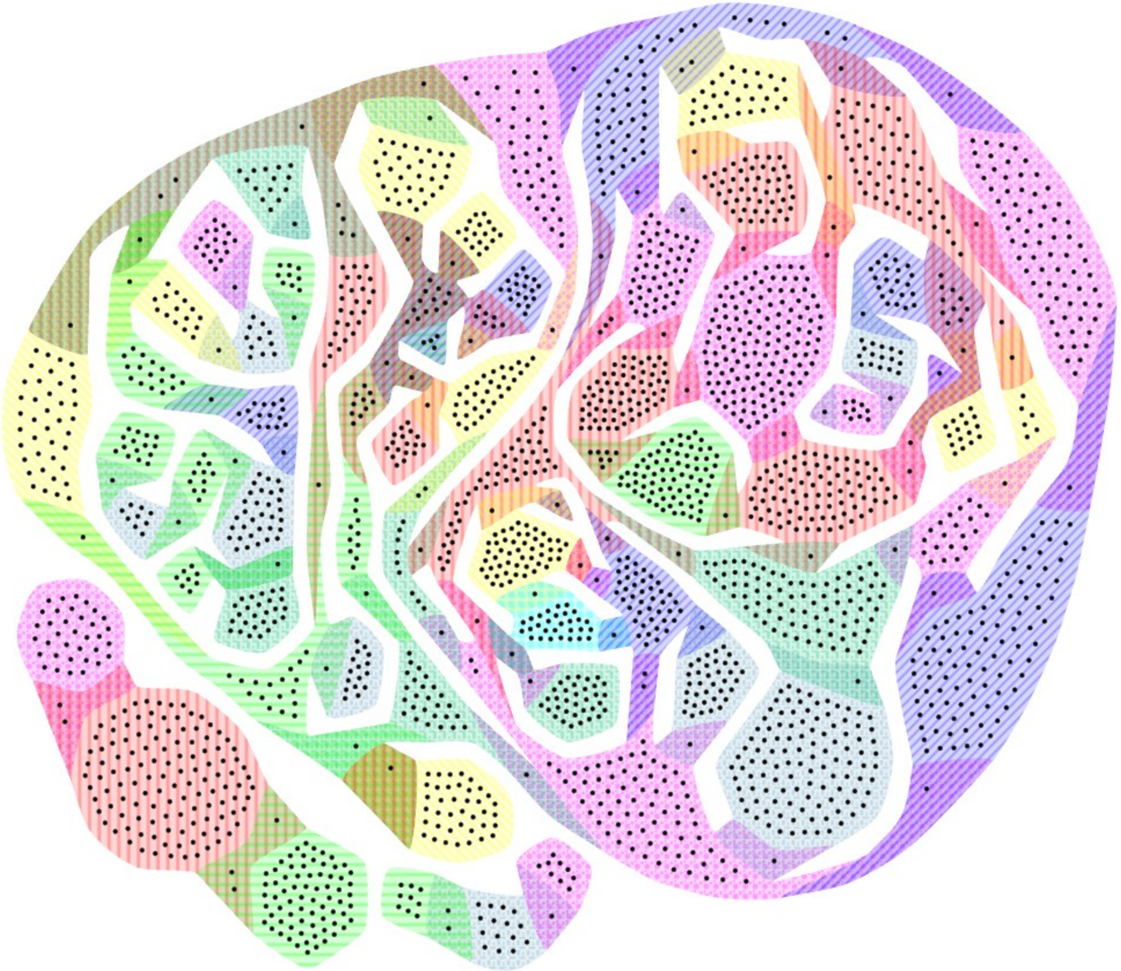


Conclusions

- We designed ImPrEd to improve PrEd in terms of:
  - running time (surrounding edges and QuadTrees),
  - drawing quality (force cooling and new max move),
  - flexibility (polyline, crossable and flexible edges).

- We presented evidences of improvements based on:
  - 100 randomly generated planar and non-planar graphs,
  - 4 Euler diagram graphs from real world data,
  - over 900 algorithm executions.

- Improve the complexity to scale to larger graphs.



Thank you for your attention.