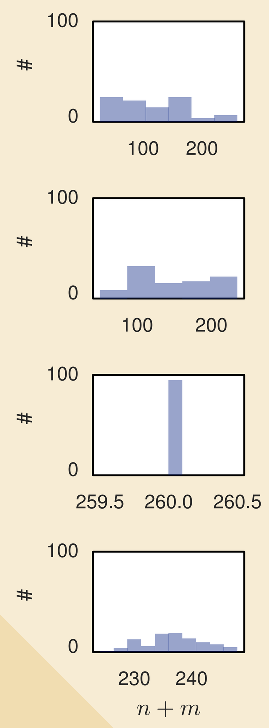
A Geometric Heuristic for Rectilinear Crossing Minimization

Marcel Radermacher • Klara Reichard • Ignaz Rutter • Dorothea Wagner

Problem Statement

Crossing minimization is an active field of research. While there is a lot of work on heuristics for topological drawings, these techniques are typically not transferable to the **rectilinear** setting. We introduce and evaluate three heuristics for rectilinear crossing minimization. The approaches are based on the primitive operation of **moving a single vertex** to its **crossing-minimal position** in the current drawing Γ. In an experimental evaluation, we demonstrate that our algorithms compute straight-line drawings with significantly fewer crossings than energy-based algorithms, though at the cost of a higher running time.

Evaluation



North

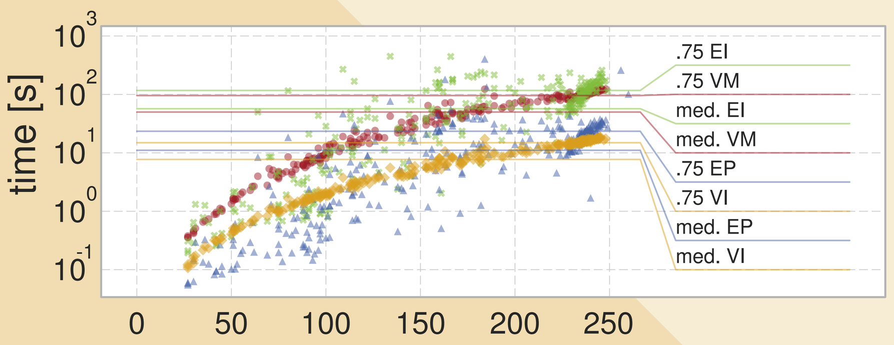
Rome

Triangulation + X
- + 10 random edges

Community
- Resembles community structure

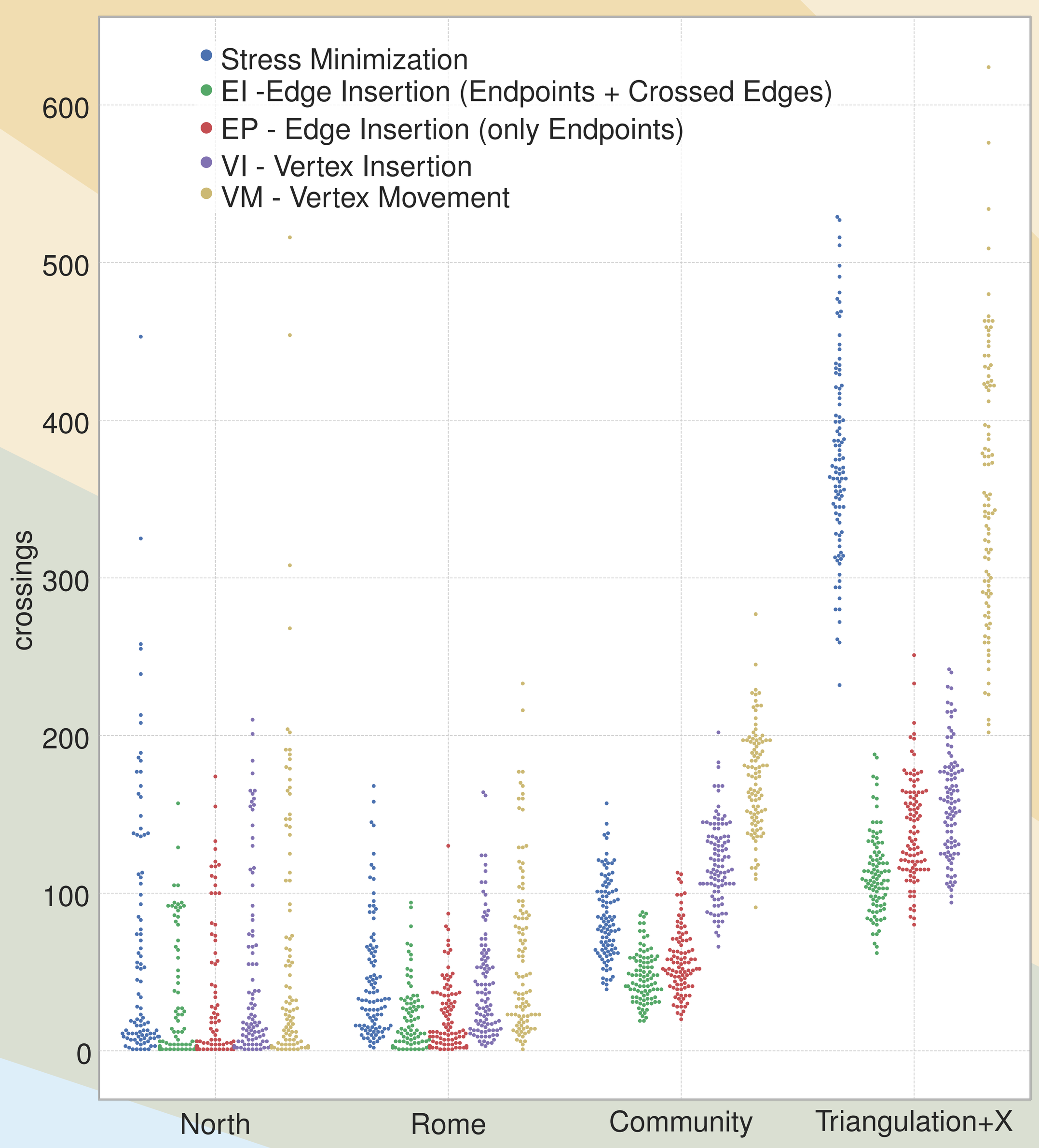
100 randomly selected graphs per class.

Running Time



The high running time is due to precise geometric operations.

Number of Crossings

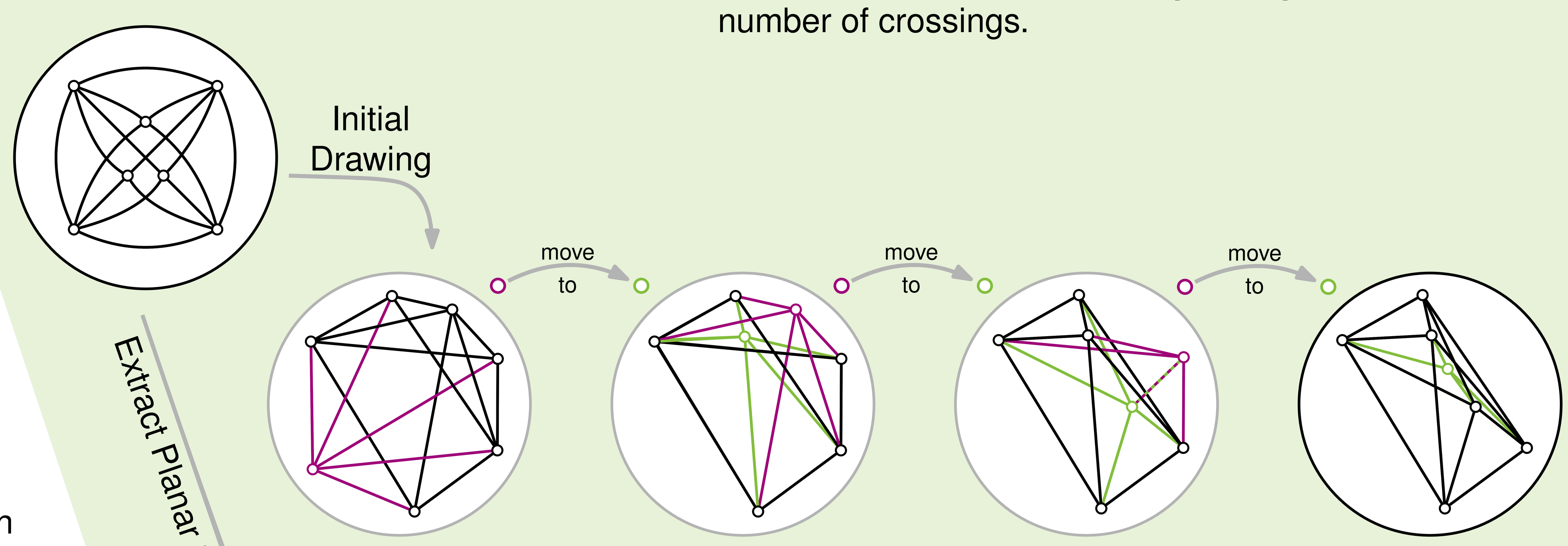


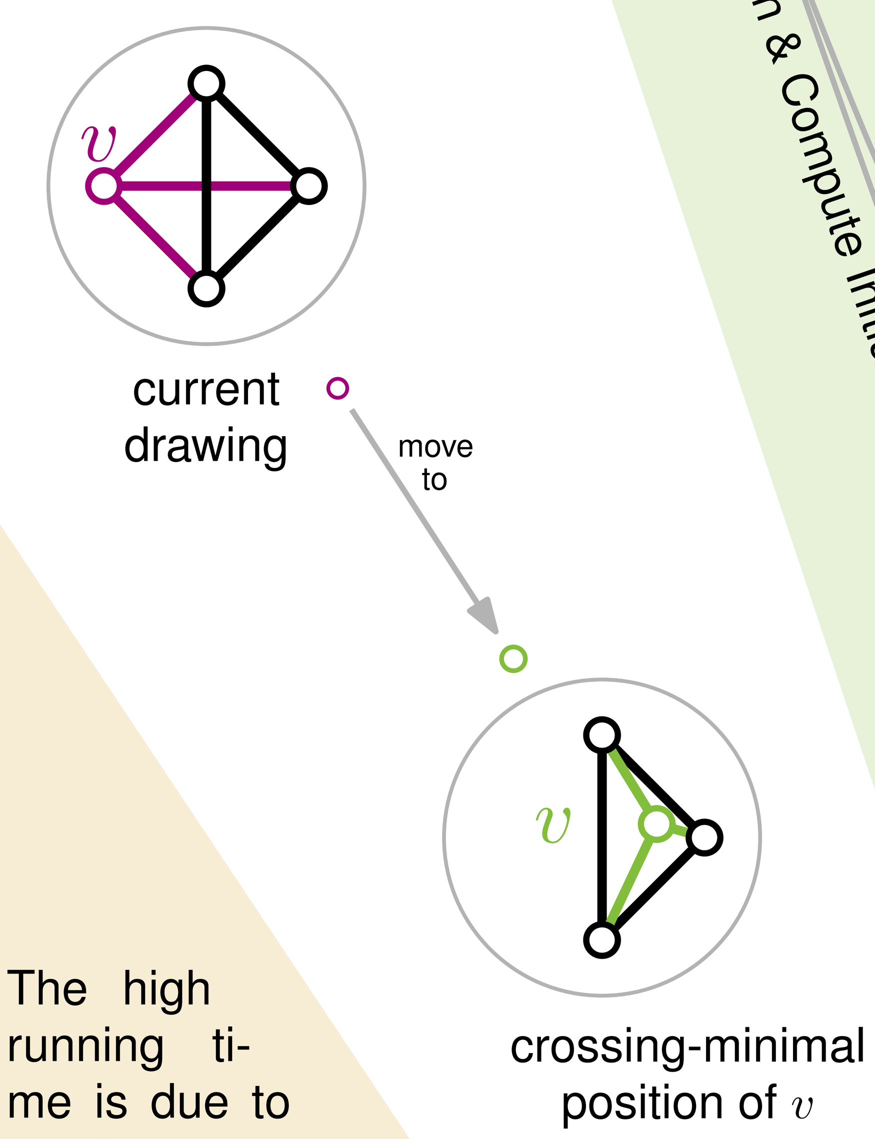
A point corresponds to the number of crossing of one drawing computed by the algorithm indicated by the color. The catagorization by graph classes shows that stress minimization has particular problems in optimizing the number of crossings in the graph class Triangulation+X. The plot suggests that edge insertion computes drawings with the smallest number of crossings. Our statistical test indeed confirms this observation with a significance level of $\alpha = 0.05$.

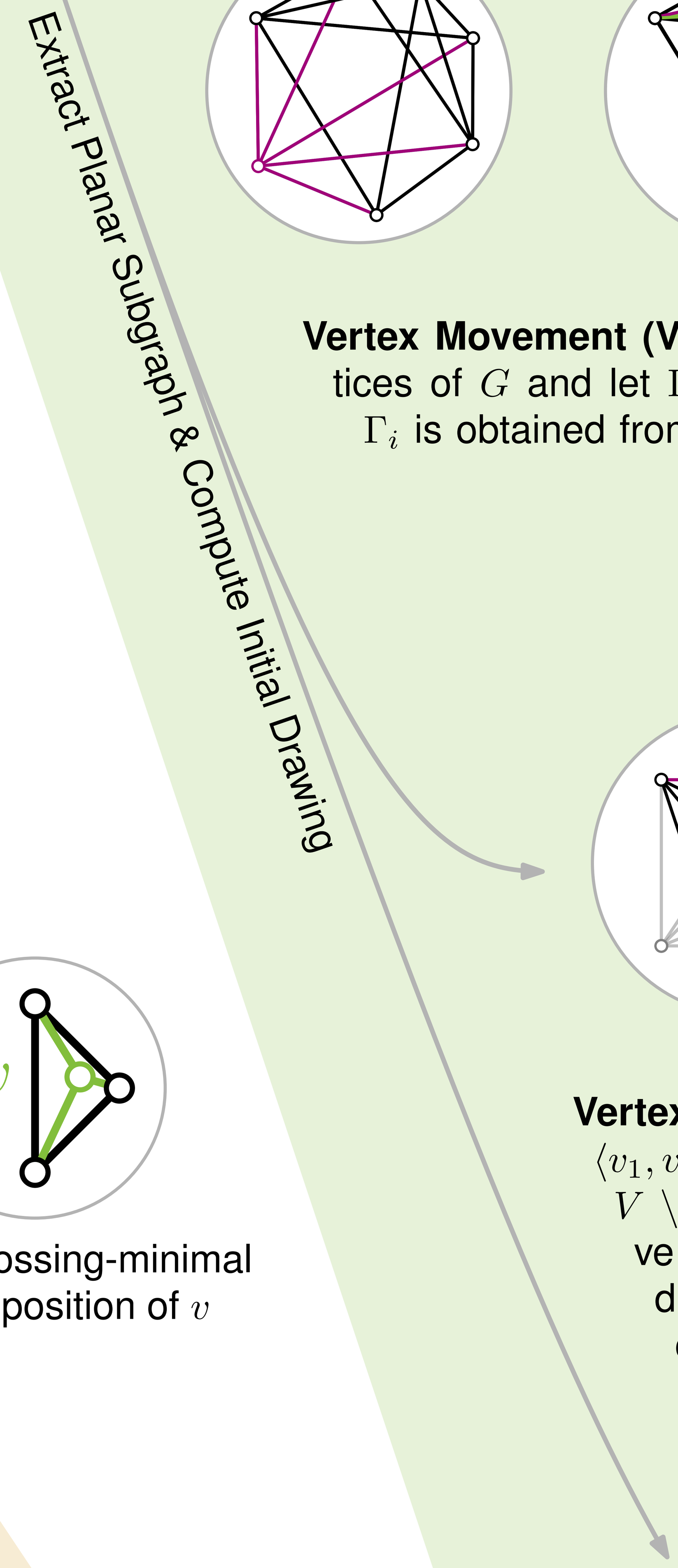
Conclusion: EI has the lowest number of crossings but a high running time. If time is crucial, EP is the best choice.

Heuristics

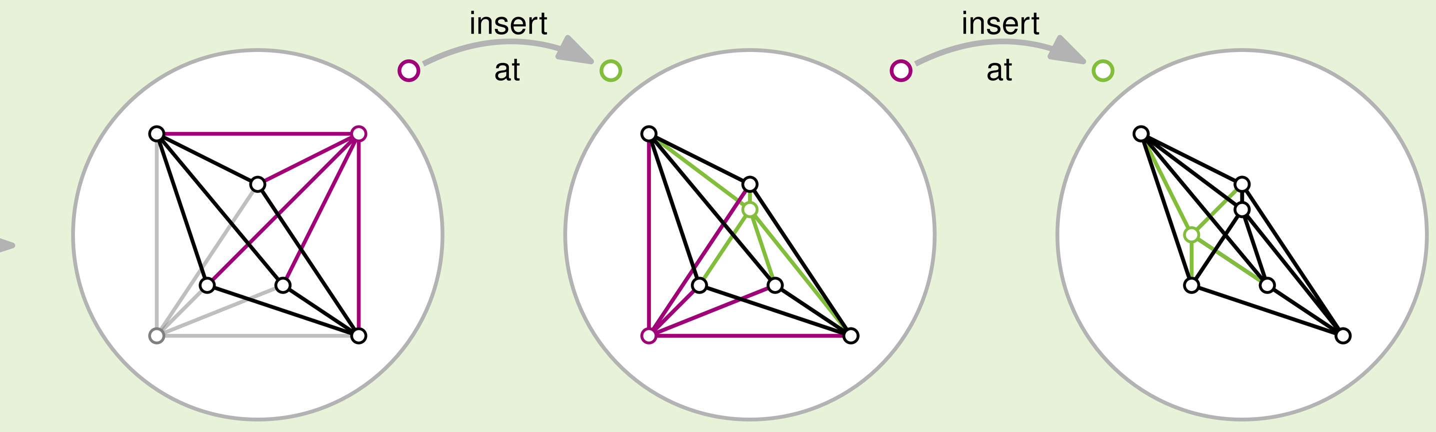
Let $\Gamma[v \mapsto p]$ be the drawing obtained from a drawing Γ where v is moved to p. A *crossing-minimal position* $p^\star$ of v corresponds to a drawing $\Gamma[v \mapsto p^\star]$ with a minimum number of crossings.



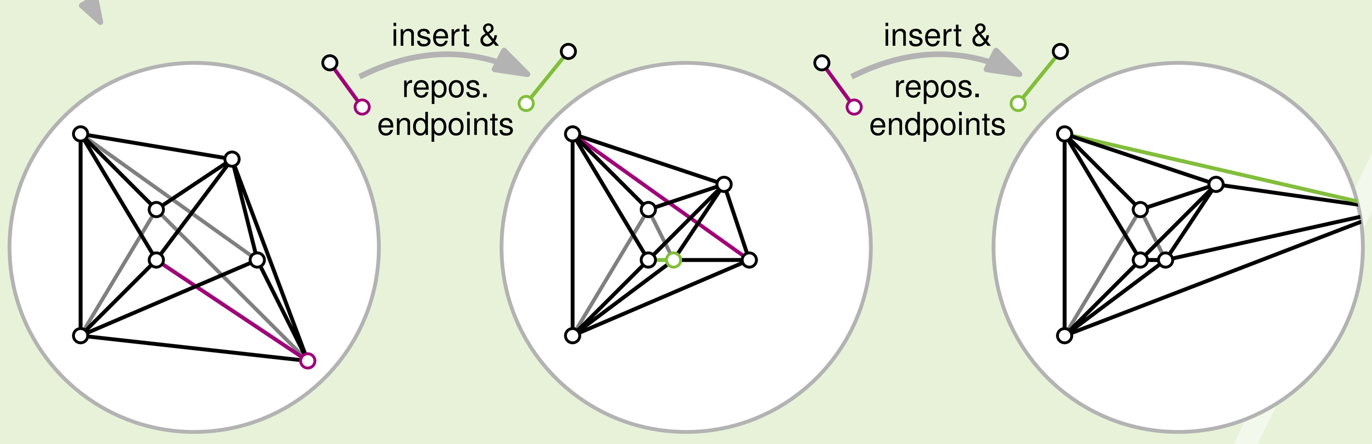




Vertex Movement (VM) Let $S = \langle v_1, v_2, \ldots, v_k \rangle$, $k \in \mathbb{N}$, be a sequence of vertices of G and let Γ_0 be an arbitrary straight-line drawing of G. The drawing Γ_i is obtained from Γ_{i-1} by moving vertex v_i to its crossing-minimal position.



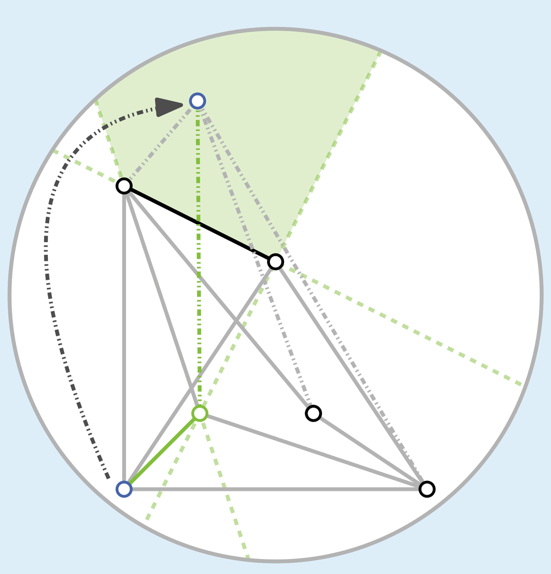
Vertex Insertion (VI) We identify a sequence $T = \langle v_1, v_2, \ldots, v_k \rangle$, $k \leq n$, so that the induced subgraph G_P of $V \setminus V(T)$ is a planar subgraph of G. We iteratively remove the vertex with the highest crossing number from a drawing Γ of G until Γ is planar. We reinsert the vertices in reverse order at their crossing-minimal positions.



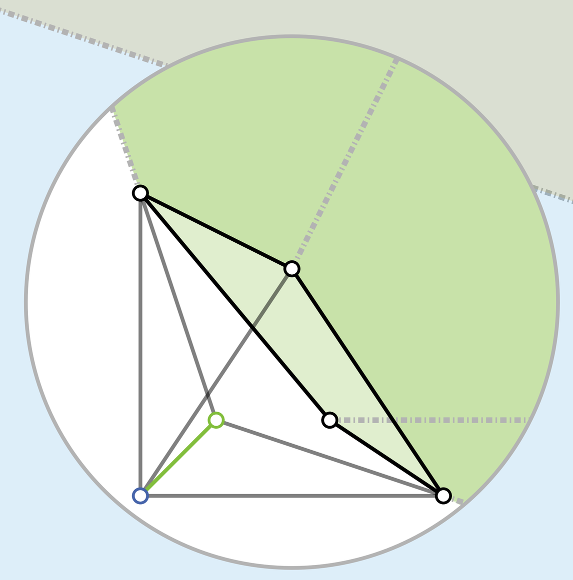
Edge Insertion We start with a maximal planar subgraph of G and iteratively reinsert edges e into the previous drawing. We modify each drawing so that we can add the edge e with a small number of crossings. We evaluated two strategies, (i) either only moving the endpoints of e (**EP**), or (ii) moving vertices incident to edges crossing e (**EI**).

Crossing-Minimal Position

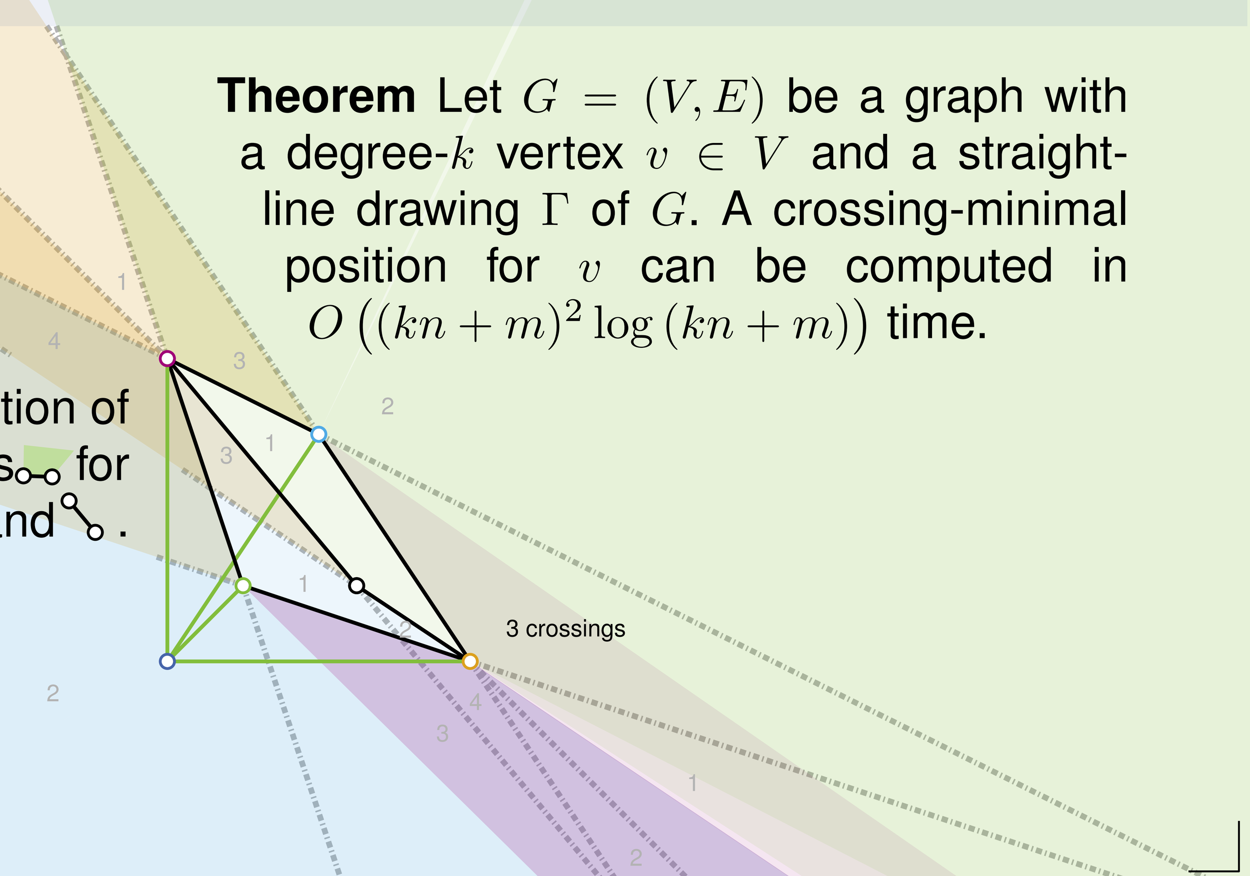
and cross if and only if is moved into .



A cell corresponds to a set of edges crossed by .



Intersection of visibility regions for all pairs of and .



Theorem Let $G = (V, E)$ be a graph with a degree-k vertex $v \in V$ and a straight-line drawing Γ of G. A crossing-minimal position for v can be computed in $O\left((kn+m)^2 \log(kn+m)\right)$ time.