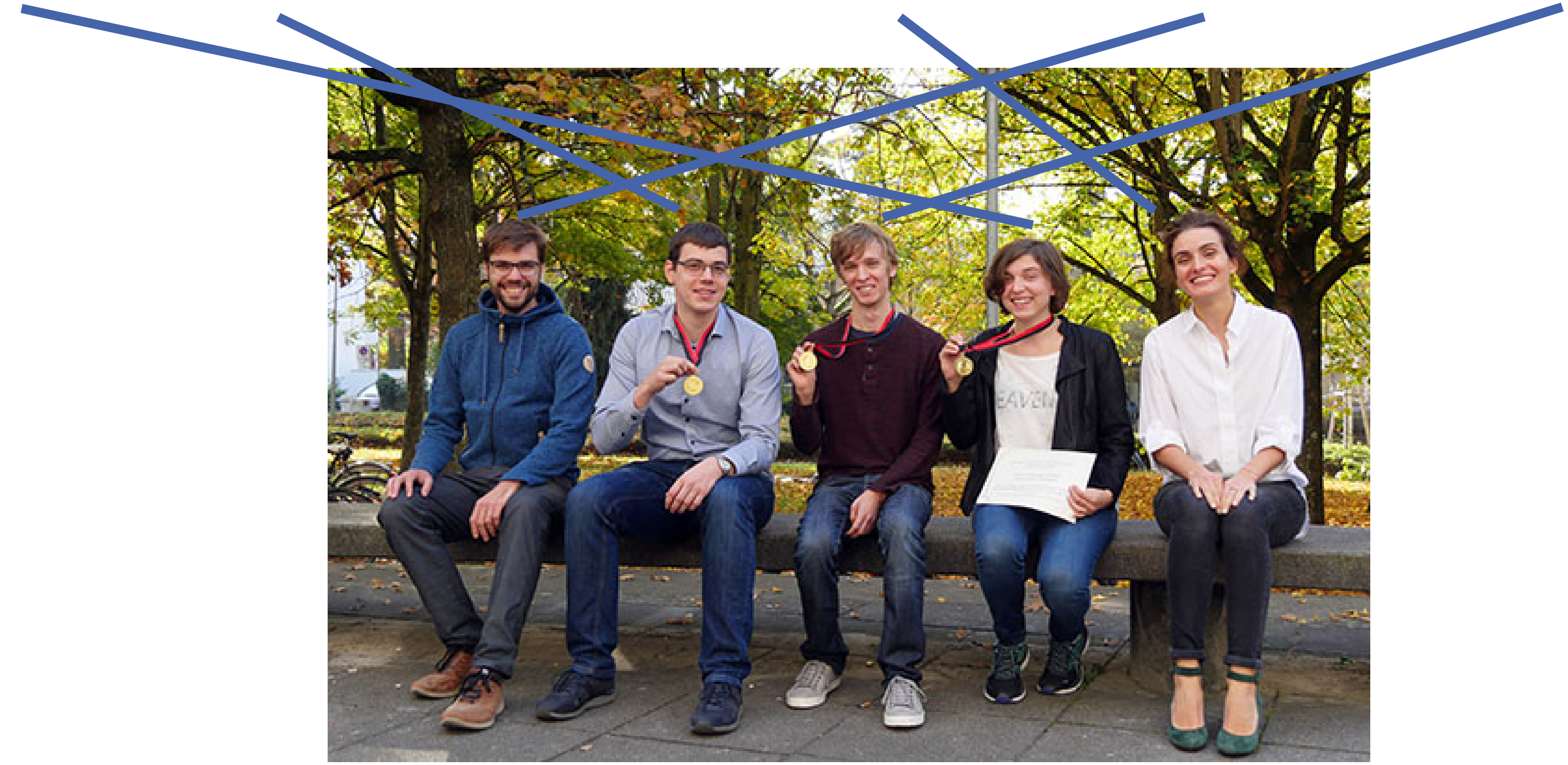
A Greedy Heuristic for Crossing-Angle Maximization

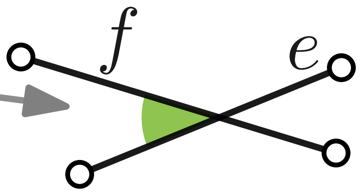
Graph Drawing 2018

Almut Demel
Dominik Dürrschnabel
Tamara Mchedlidze
Marcel Radermacher
Lasse Wulff



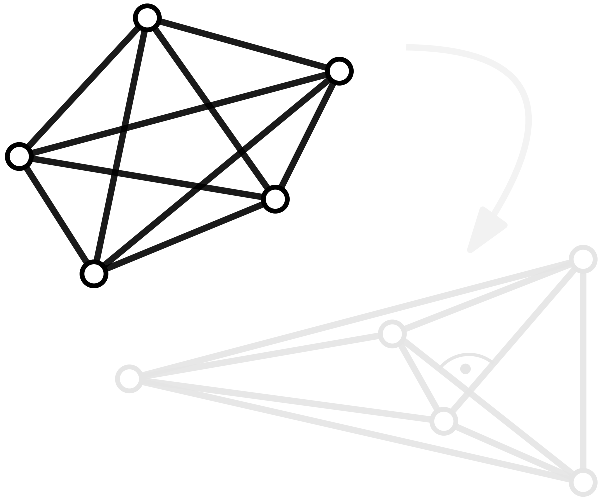
Crossing-Angle Maximization

crossing angle $\mathrm{cr}\text{-}\angle(e, f)$



crossing angle $\mathrm{cr}\text{-}\angle(\Gamma)$ of a drawing:

smallest crossing angle of two crossing edges



Crossing-Angle Maximization

Compute a straight-line drawing Γ of G that maximizes $\mathrm{cr}\text{-}\angle(\Gamma)$

A Lot of Theory

NP-Hardness

1, 2, 3 - bends per edge

Counting edges of RAC graphs

…

[Argyriou et al., Arikushi et al, Didimo et al, Djumović et al., …]

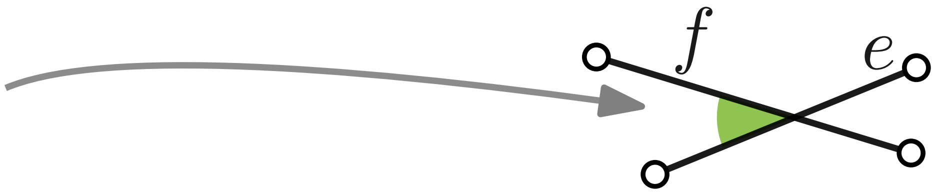
Practice before GD'17 contest

2 force-directed algorithms

[Argyriou et al. '13, Huang et al.'10]

Crossing-Angle Maximization

crossing angle $\text{cr-}\angle(e,f)$

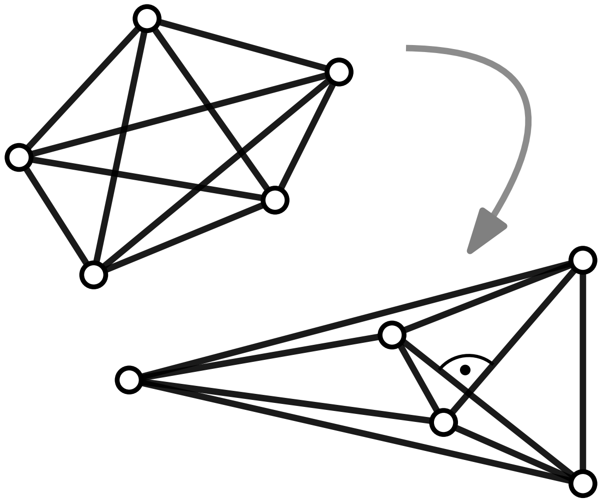


crossing angle $\text{cr-}\angle(\Gamma)$ of a drawing:

smallest crossing angle of two crossing edges

Crossing-Angle Maximization

Compute a straight-line drawing Γ of G that maximizes $\text{cr-}\angle(\Gamma)$



A Lot of Theory

- NP-Hardness
- 1, 2, 3 - bends per edge
- Counting edges of RAC graphs
- …

[Argyriou et al., Arikushi et al, Didimo et al, Djumović et al., …]

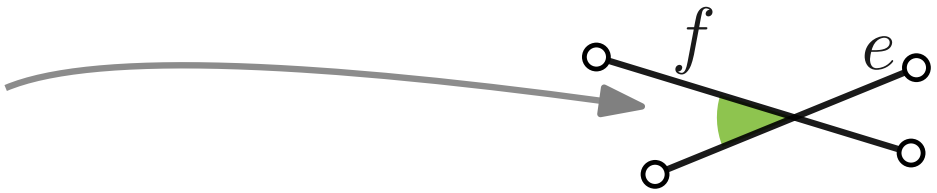
Practice before GD'17 contest

2 force-directed algorithms

[Argyriou et al. '13, Huang et al.'10]

Crossing-Angle Maximization

crossing angle $\mathrm{cr}\text{-}\angle(e, f)$

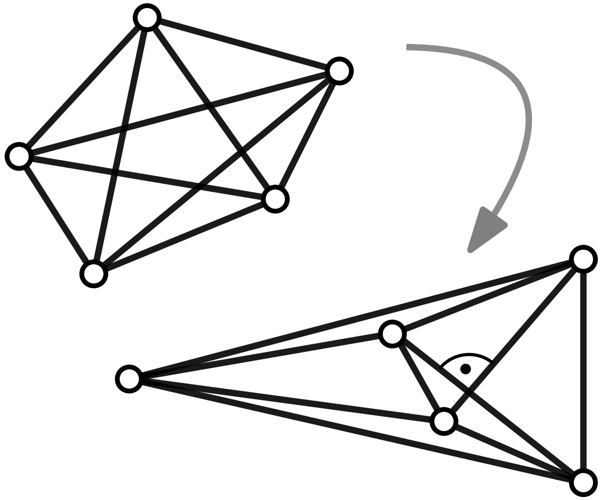


crossing angle $\mathrm{cr}\text{-}\angle(\Gamma)$ of a drawing:

smallest crossing angle of two crossing edges

Crossing-Angle Maximization

Compute a straight-line drawing Γ of G that maximizes $\mathrm{cr}\text{-}\angle(\Gamma)$



A Lot of Theory

NP-Hardness
1, 2, 3 - bends per edge
Counting edges of RAC graphs
…

[Argyriou et al., Arikushi et al, Didimo et al, Djumović et al., …]

Practice before GD’17 contest

2 force-directed algorithms

[Argyriou et al. ’13, Huang et al.’10]

Our Heuristic

Motivation: Win Graph Drawing Contest

Our Heuristic

Motivation: Win Graph Drawing Contest

Challenge: No Restriction on Input

Our Heuristic

Motivation: Win Graph Drawing Contest

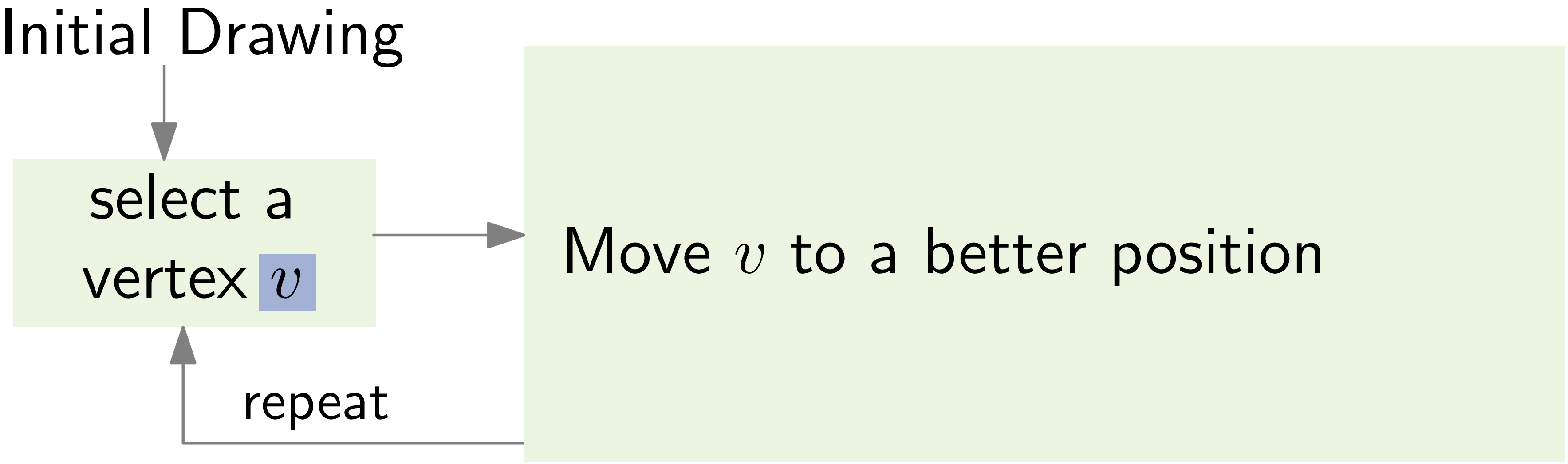
Challenge: No Restriction on Input

Our Heuristic

Design Goals

- Fast
- Generic
- Easy to implement

Our Heuristic

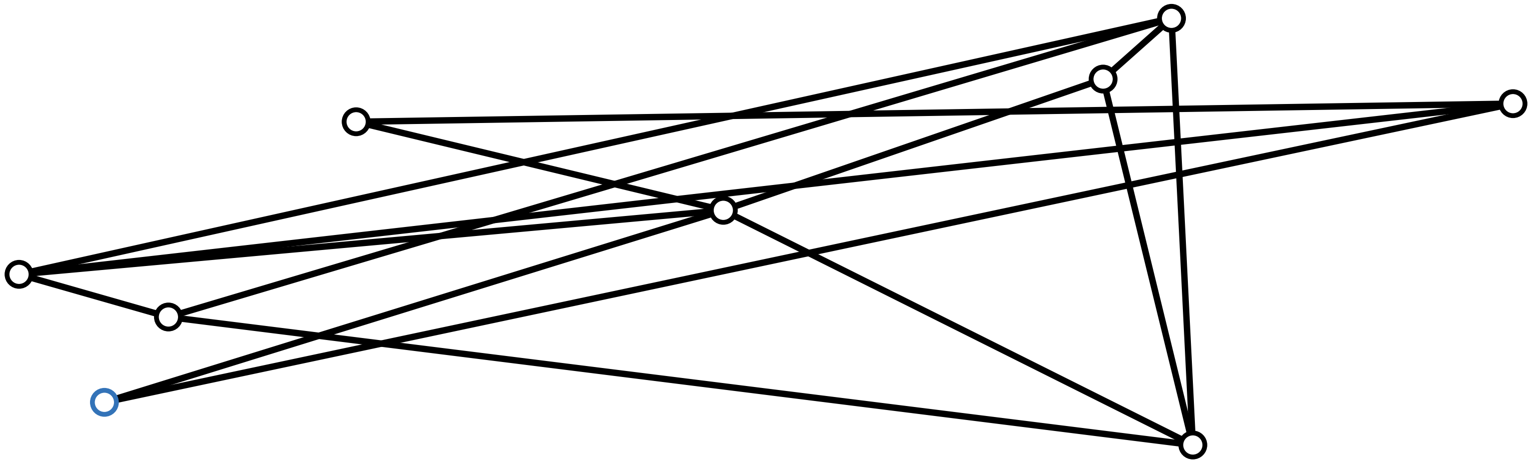


Initial Drawing

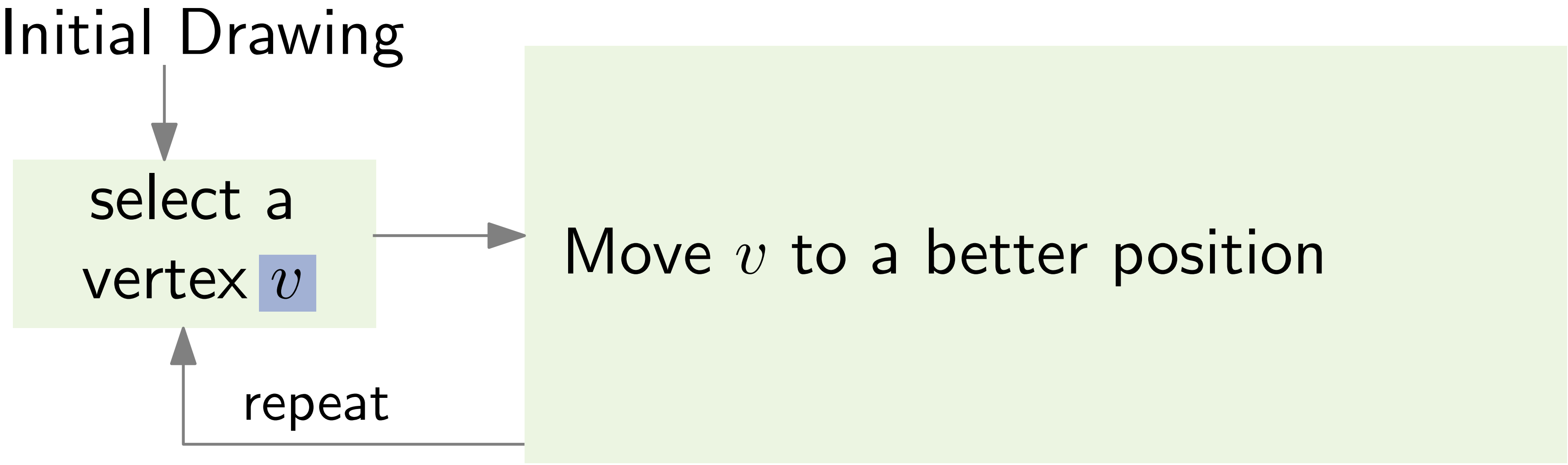
select a vertex v

Move v to a better position

repeat



Our Heuristic

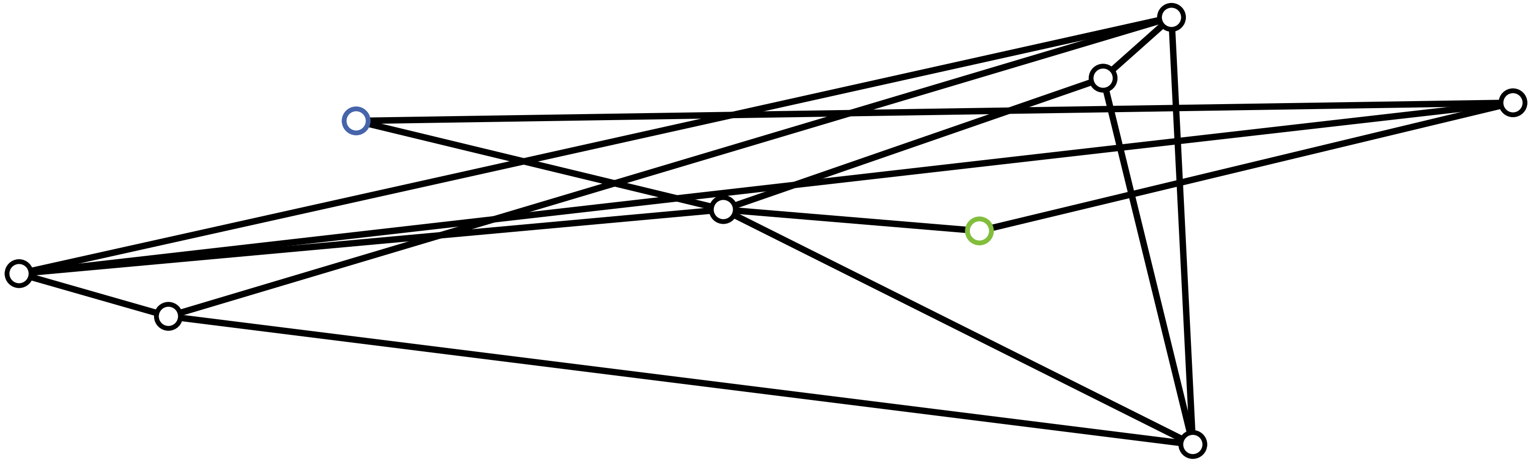


Initial Drawing

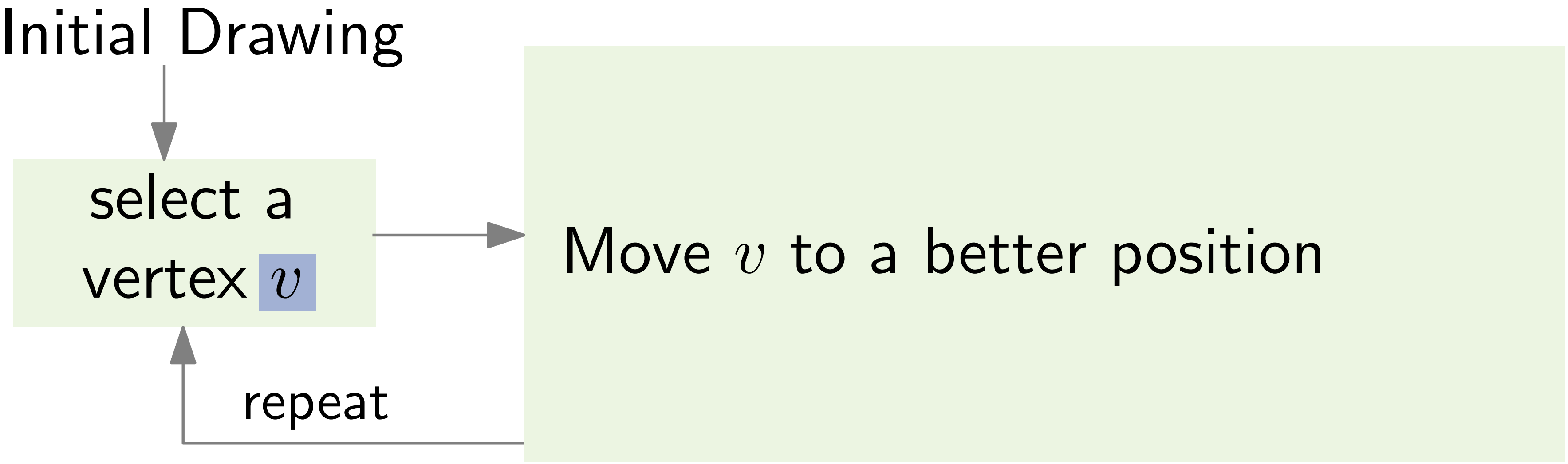
select a vertex v

Move v to a better position

repeat



Our Heuristic

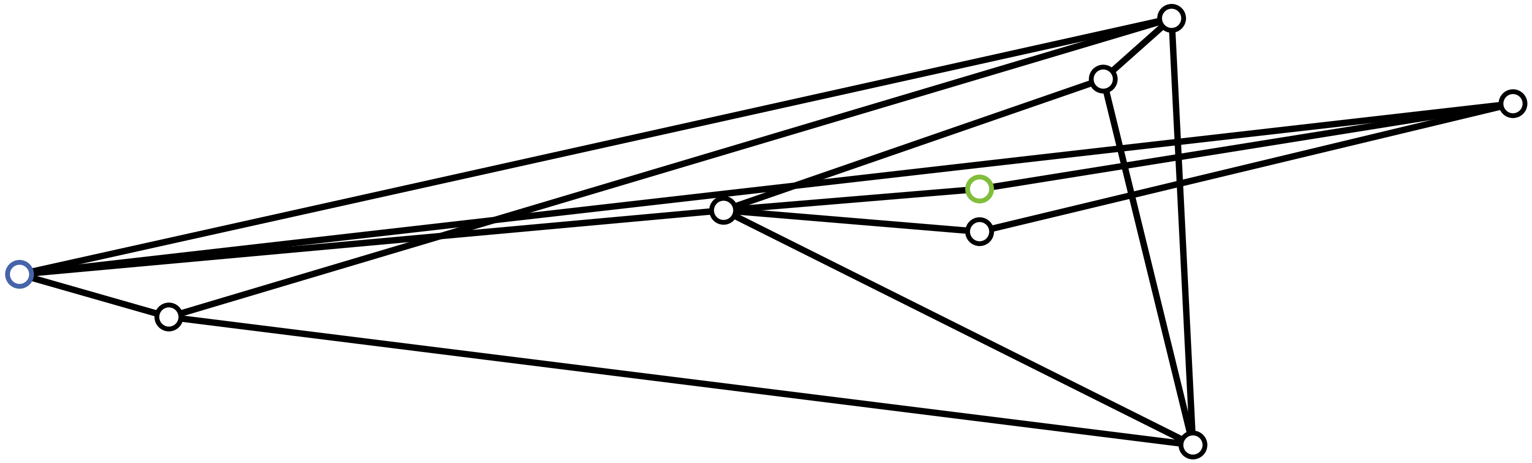


Initial Drawing

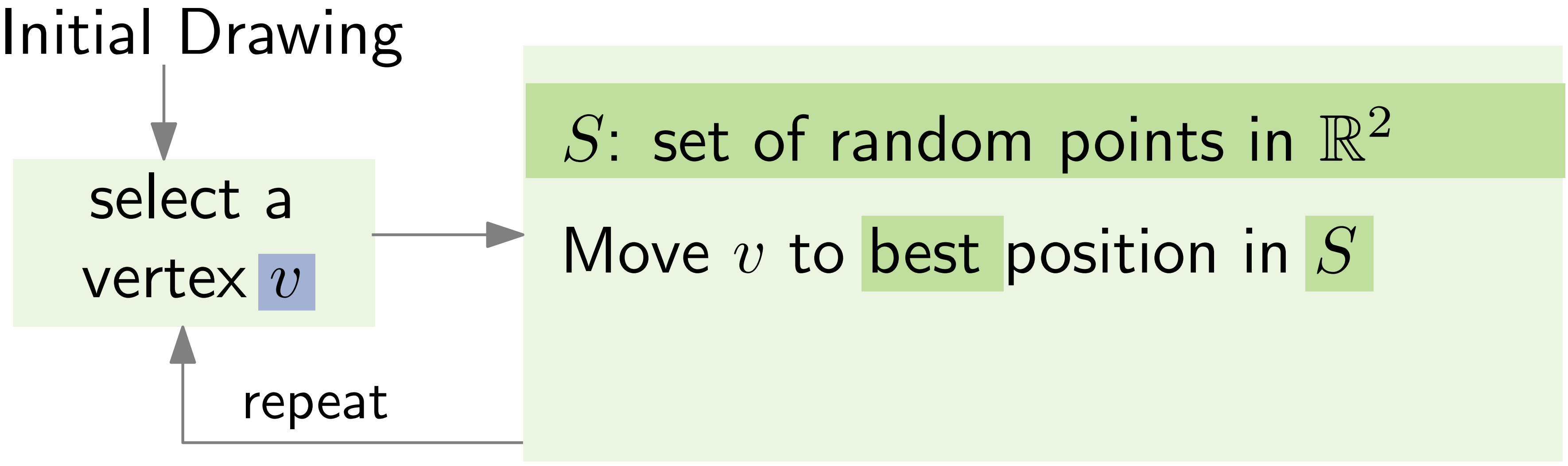
select a vertex v

Move v to a better position

repeat



Our Heuristic



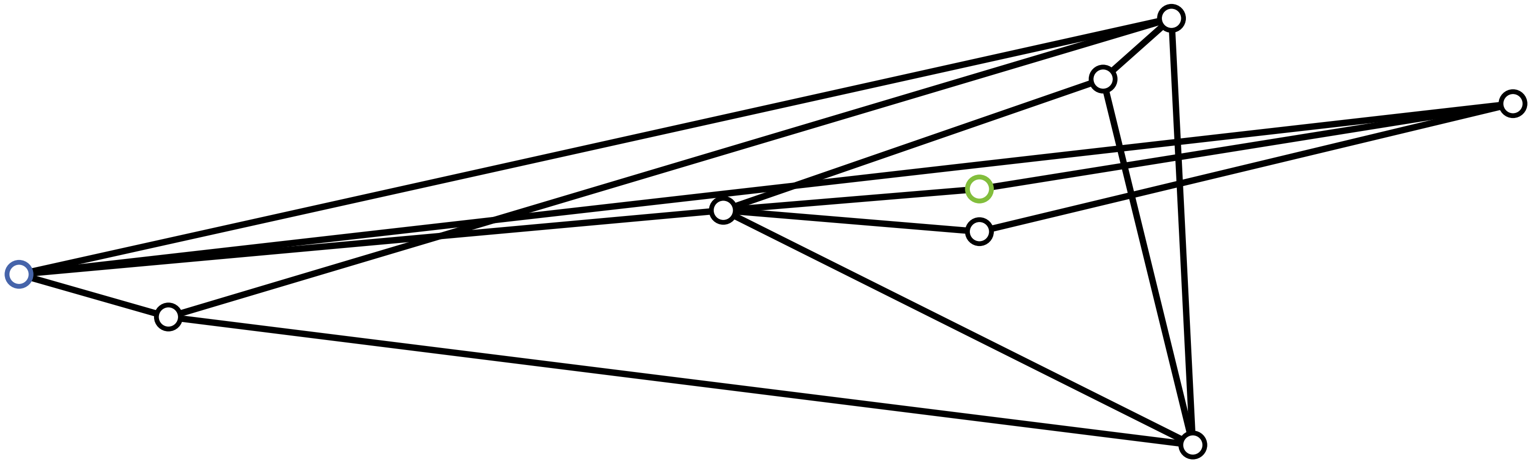
Initial Drawing

select a vertex v

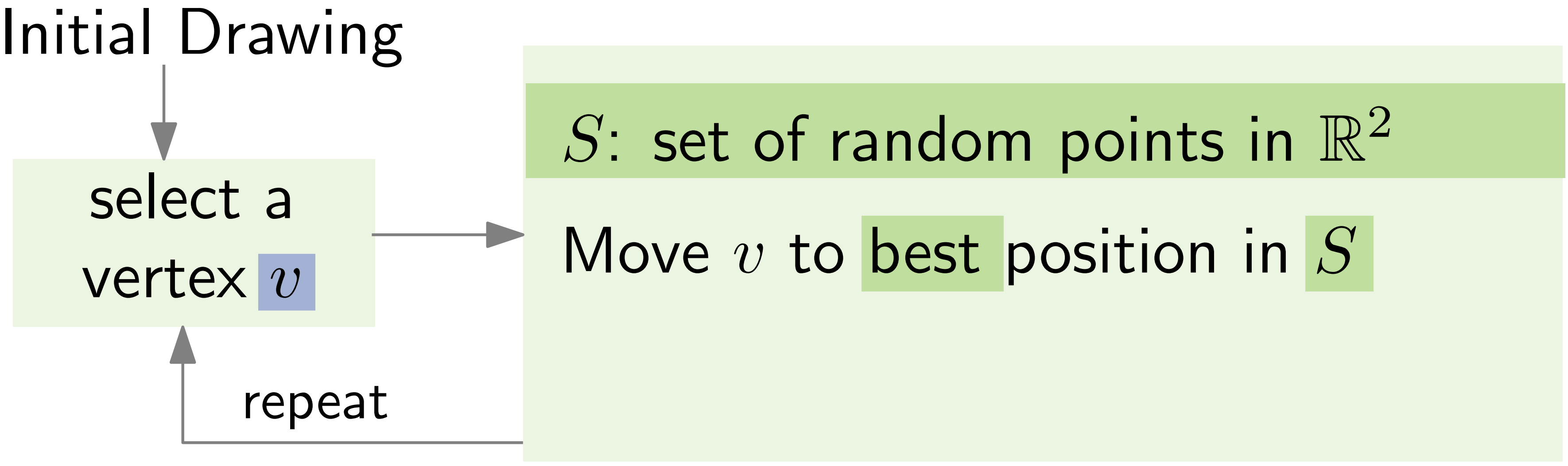
S: set of random points in $\mathbb{R}^2$

Move v to best position in S

repeat



Our Heuristic



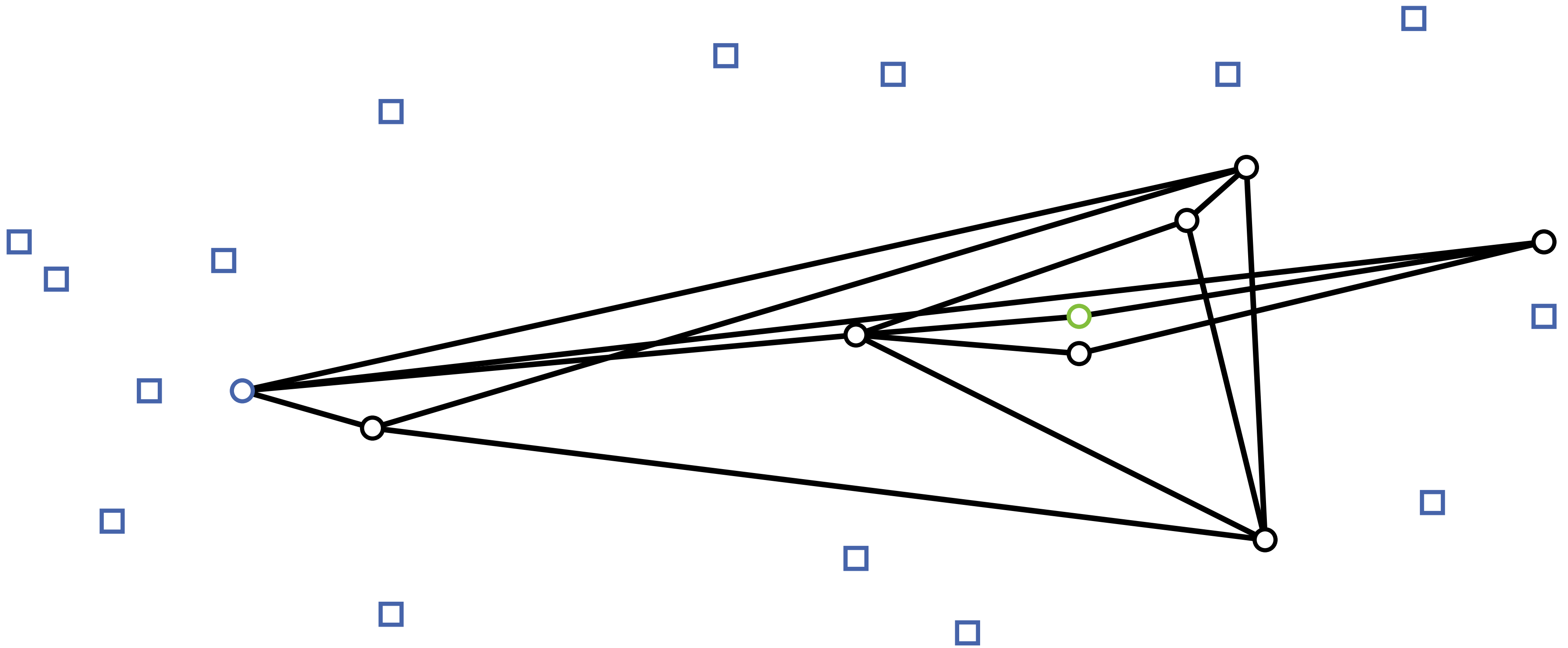
Initial Drawing

select a vertex v

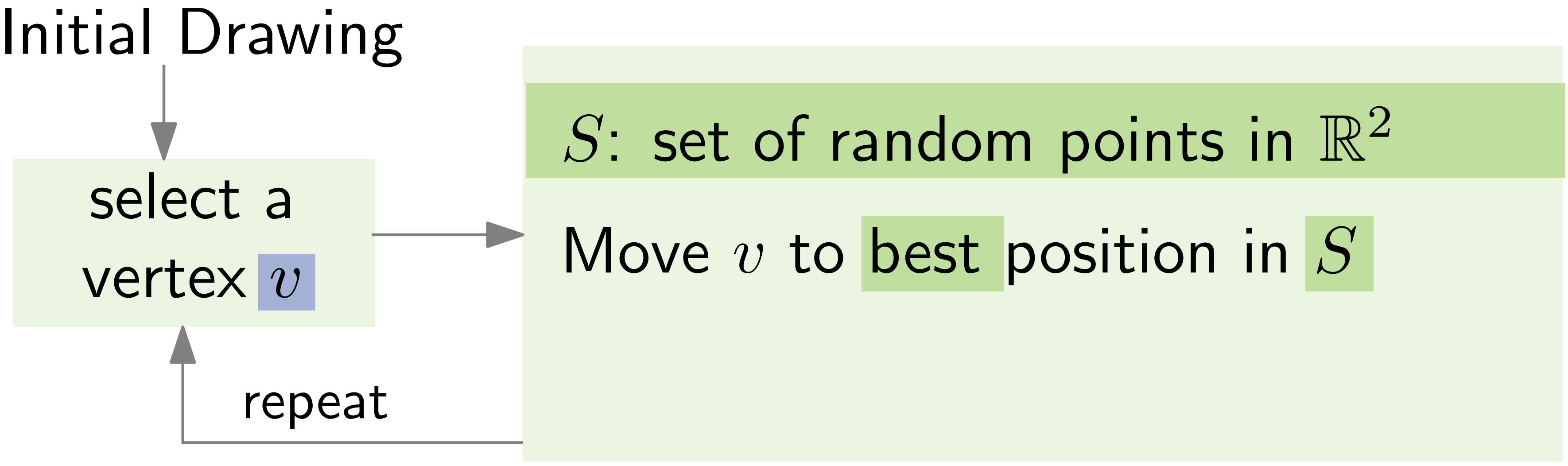
S: set of random points in $\mathbb{R}^2$

Move v to best position in S

repeat



Our Heuristic



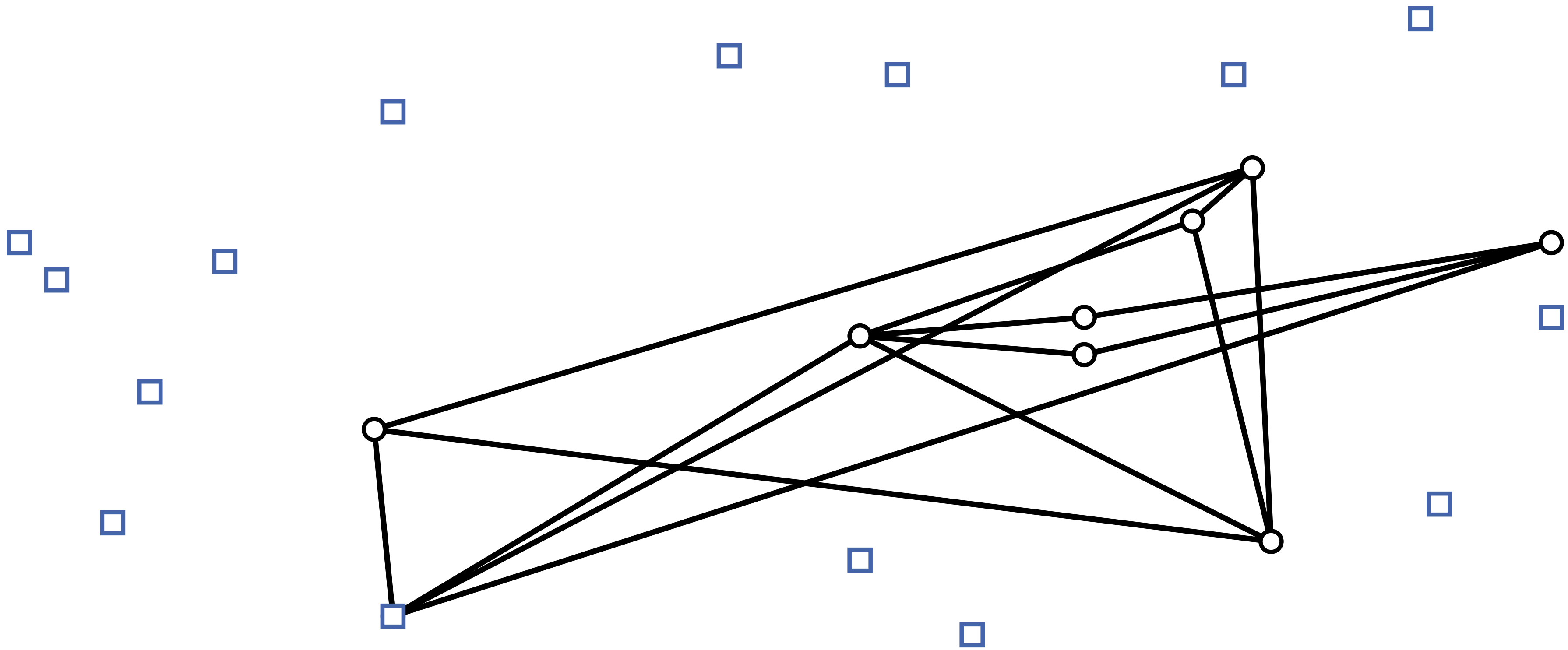
Initial Drawing

select a vertex v

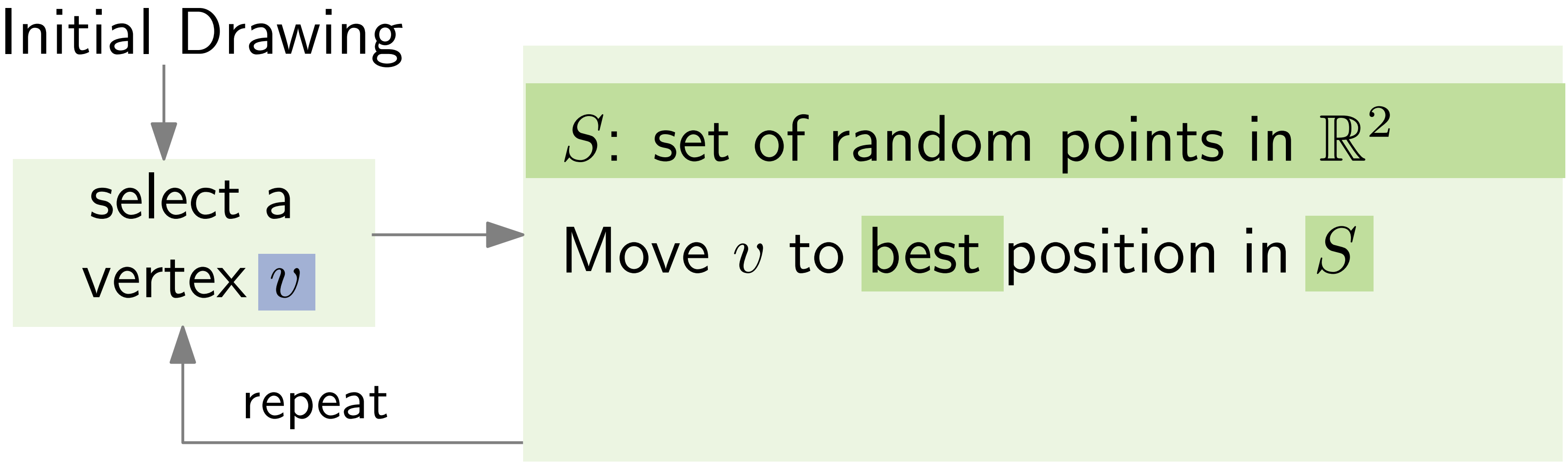
S: set of random points in $\mathbb{R}^2$

Move v to best position in S

repeat



Our Heuristic



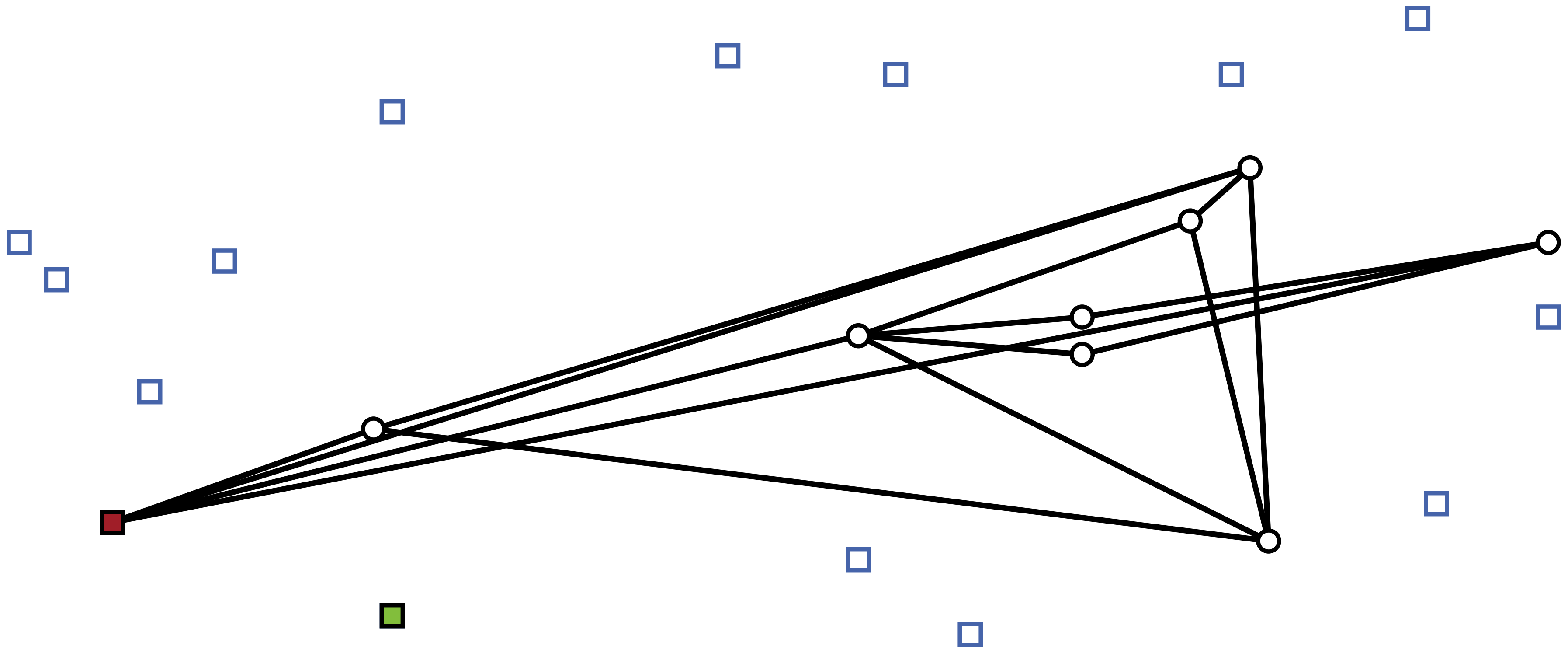
Initial Drawing

select a vertex v

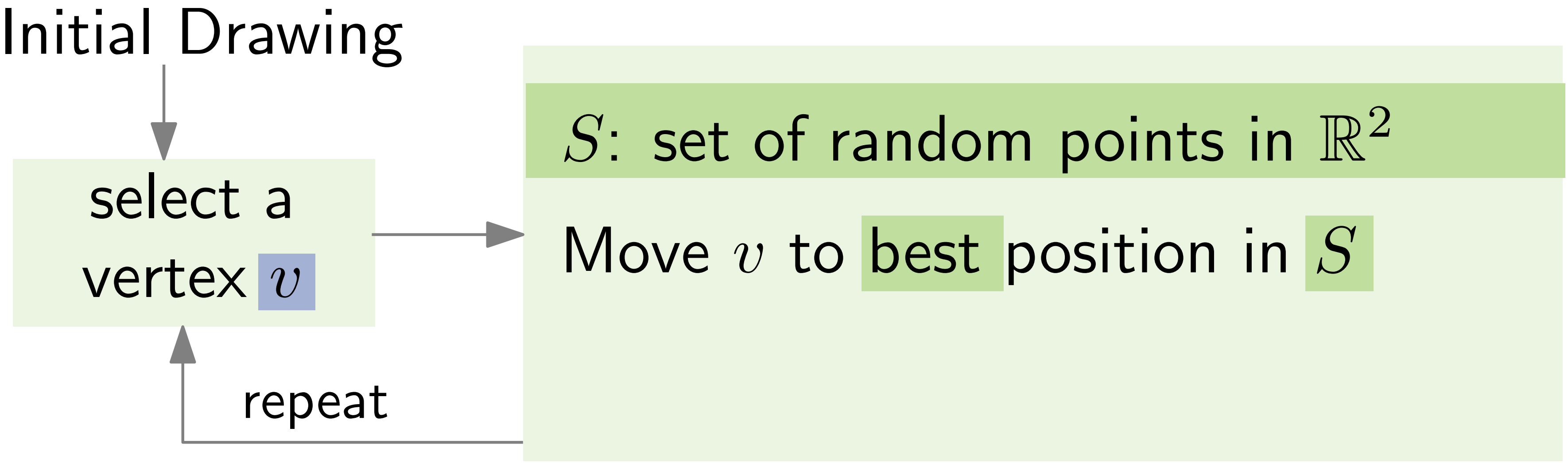
repeat

S: set of random points in $\mathbb{R}^2$

Move v to best position in S



Our Heuristic



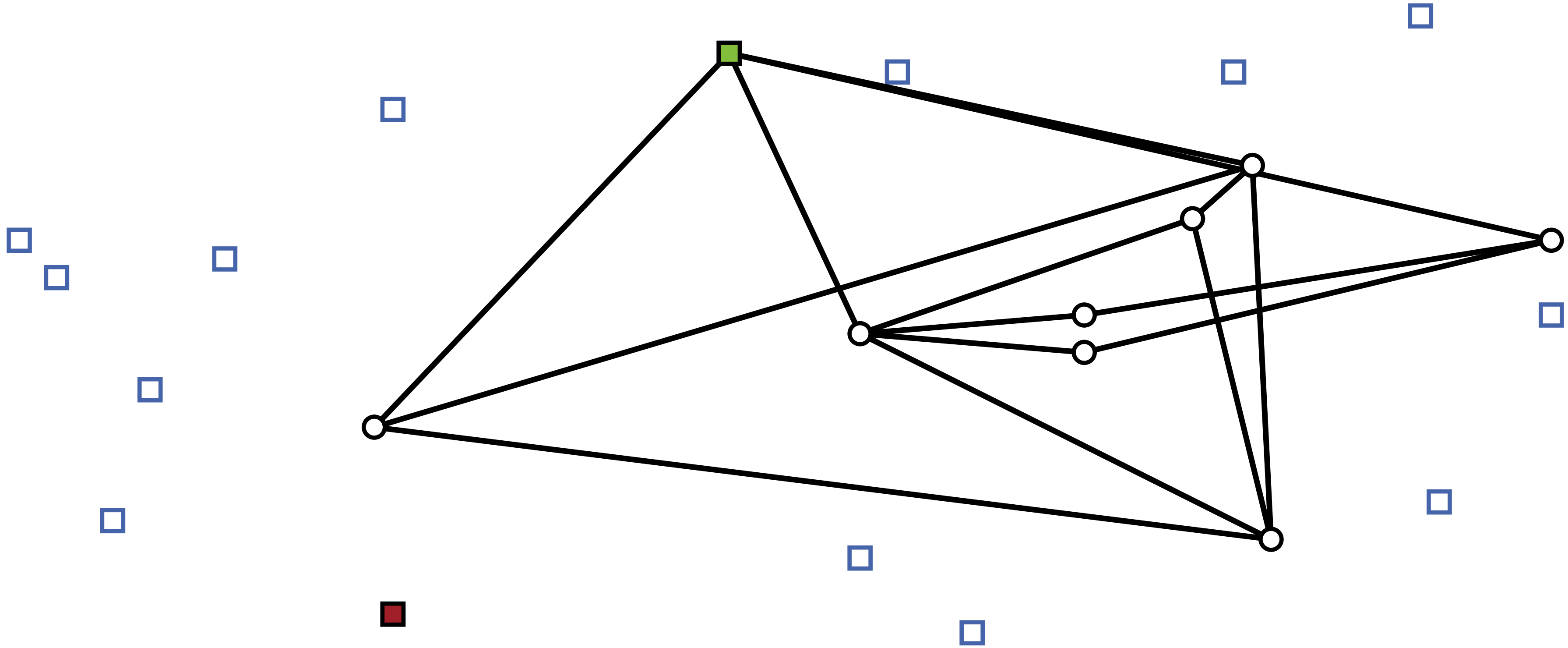
Initial Drawing

select a vertex v

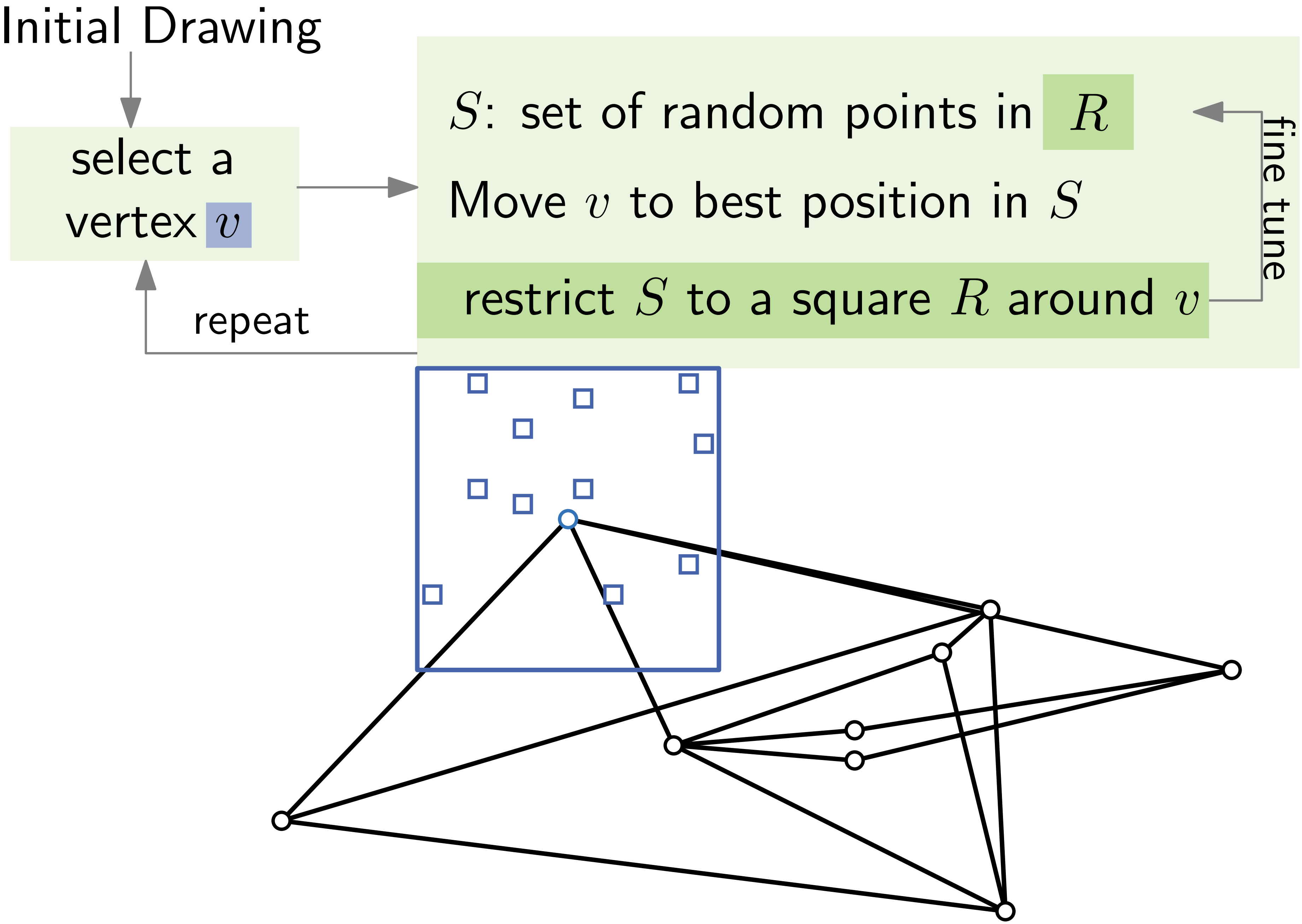
S: set of random points in $\mathbb{R}^2$

Move v to best position in S

repeat



Our Heuristic



Initial Drawing

select a vertex v

S: set of random points in R

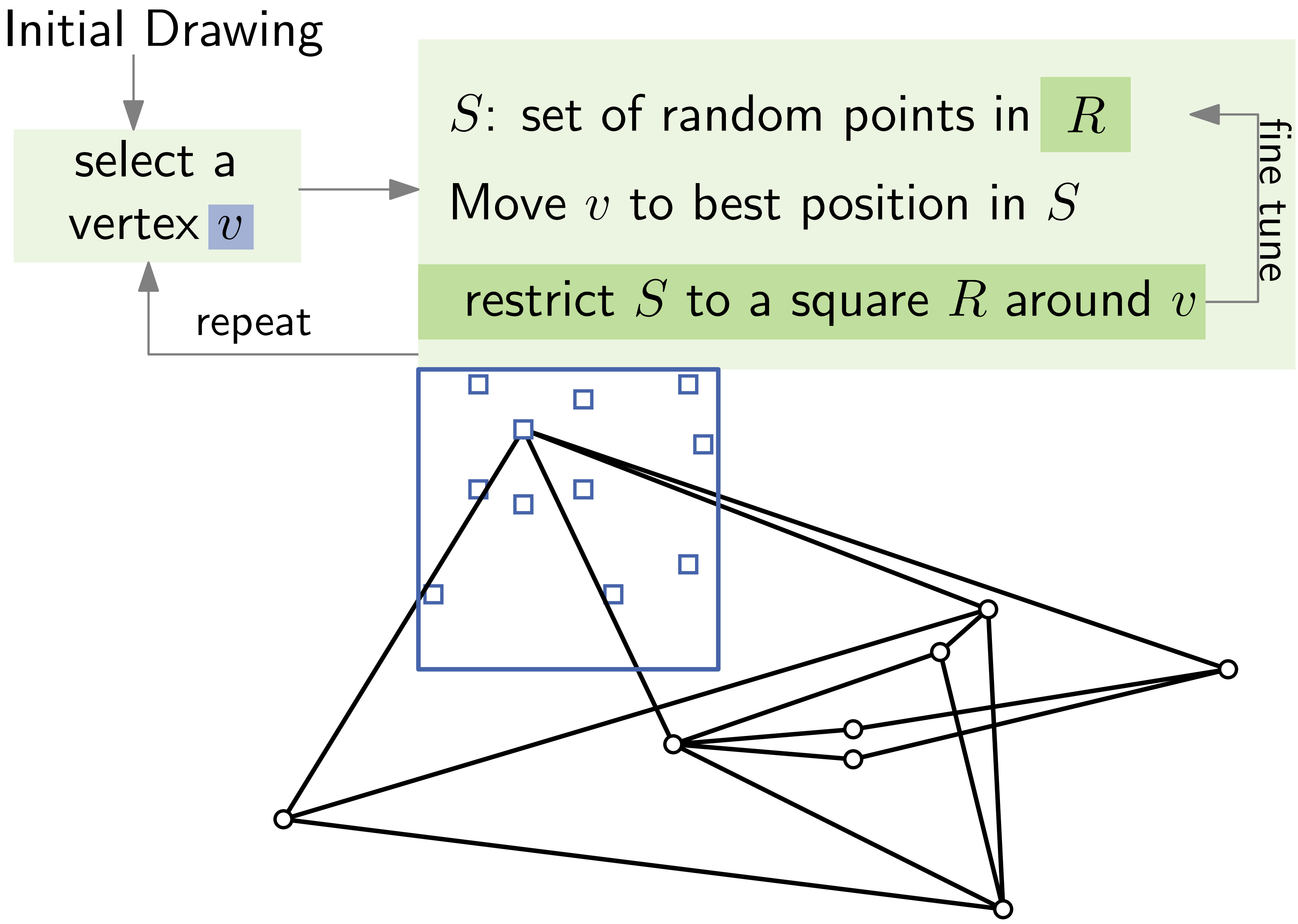
Move v to best position in S

restrict S to a square R around v

fine tune

repeat

Our Heuristic



Initial Drawing

select a vertex v

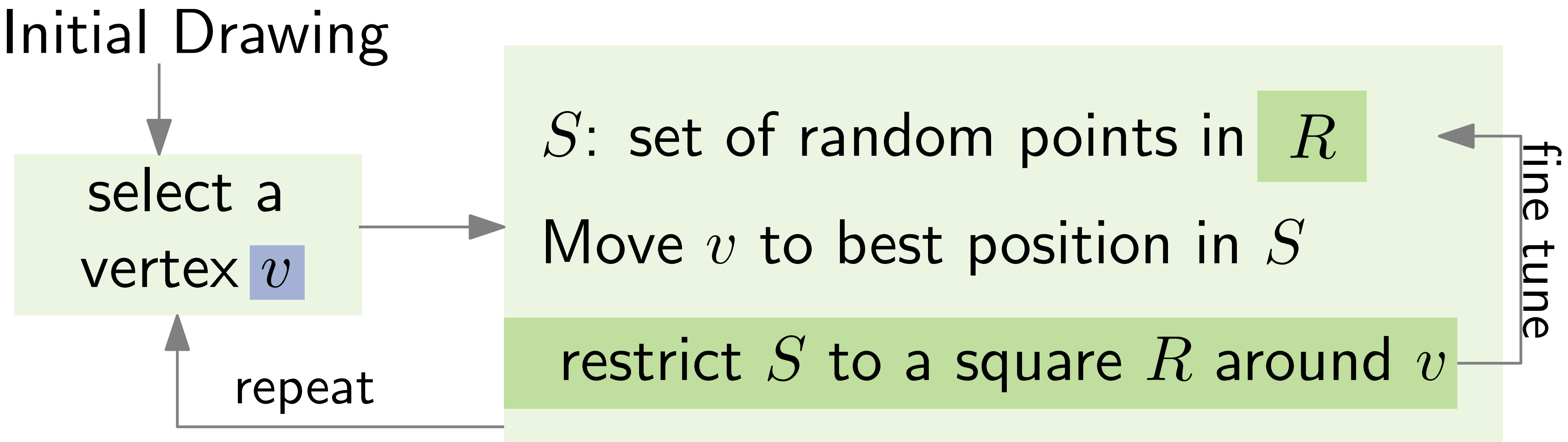
S: set of random points in R

Move v to best position in S

restrict S to a square R around v

fine tune
repeat

Our Heuristic



Initial Drawing

select a vertex v

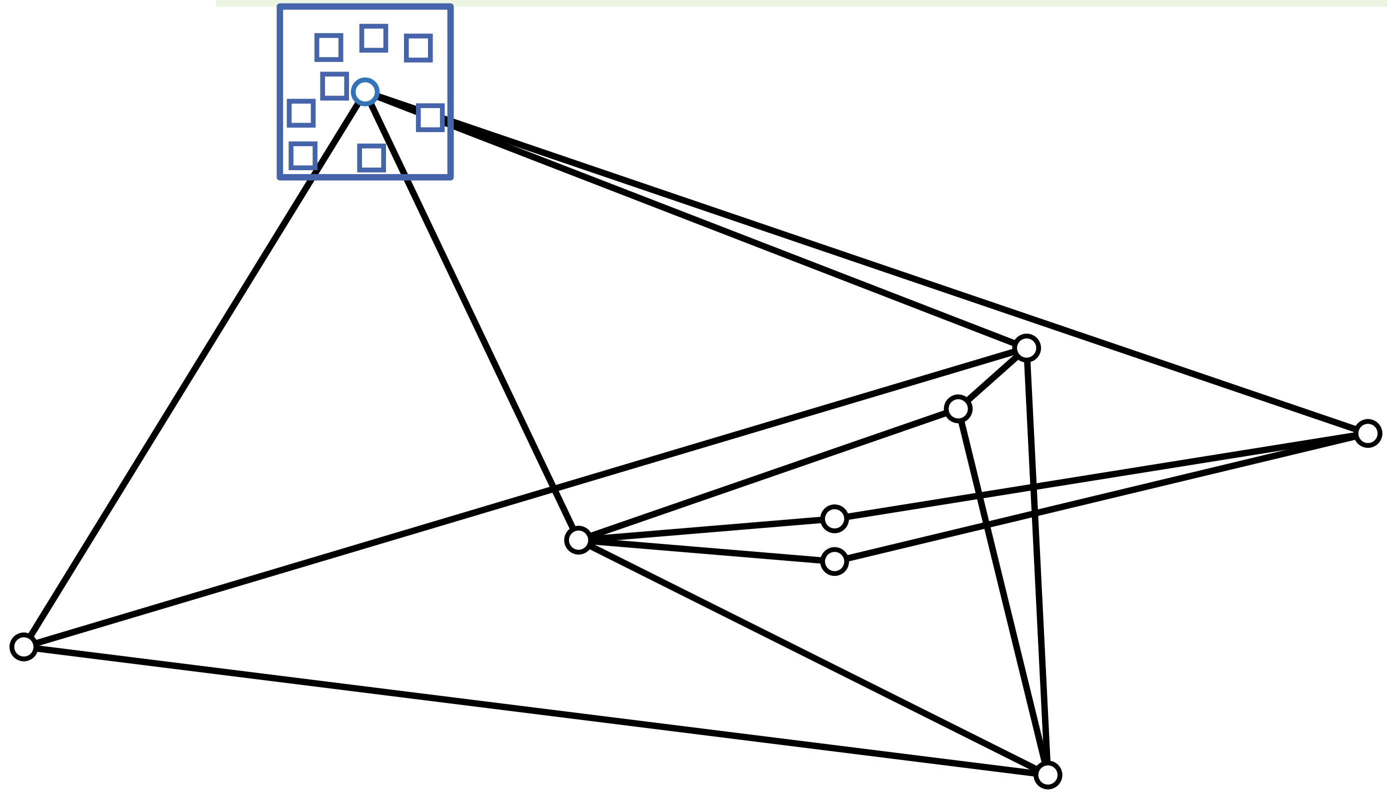
S: set of random points in R

Move v to best position in S

restrict S to a square R around v

fine tune

repeat



Our Heuristic

Initial Drawing

select a vertex v

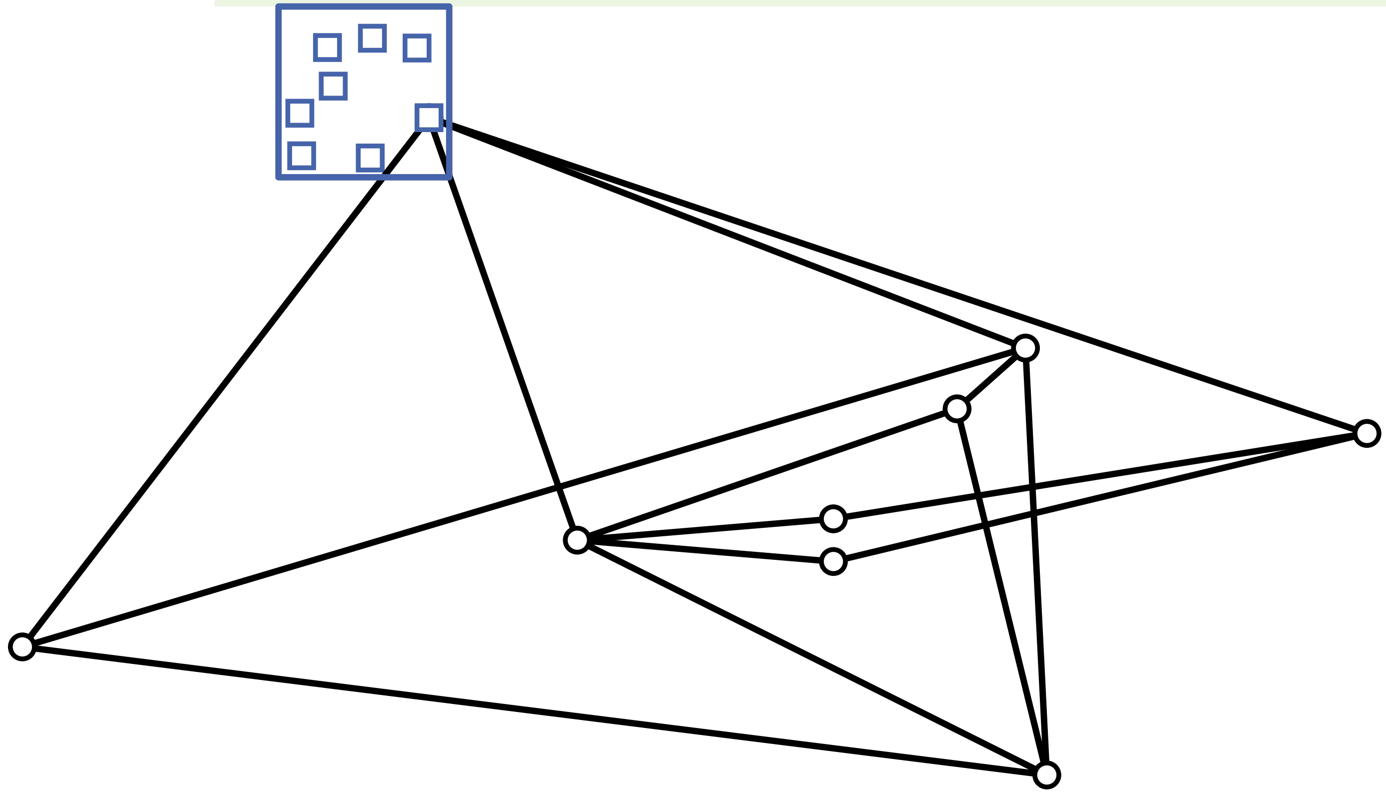
S: set of random points in R

Move v to best position in S

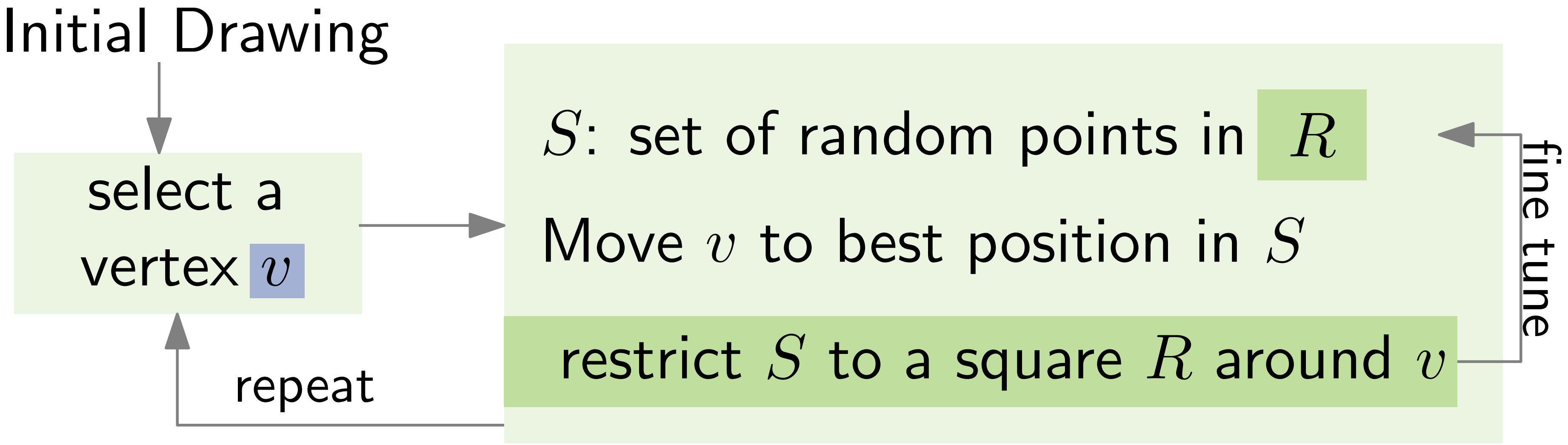
restrict S to a square R around v

fine tune

repeat



Our Heuristic



Initial Drawing

select a vertex v

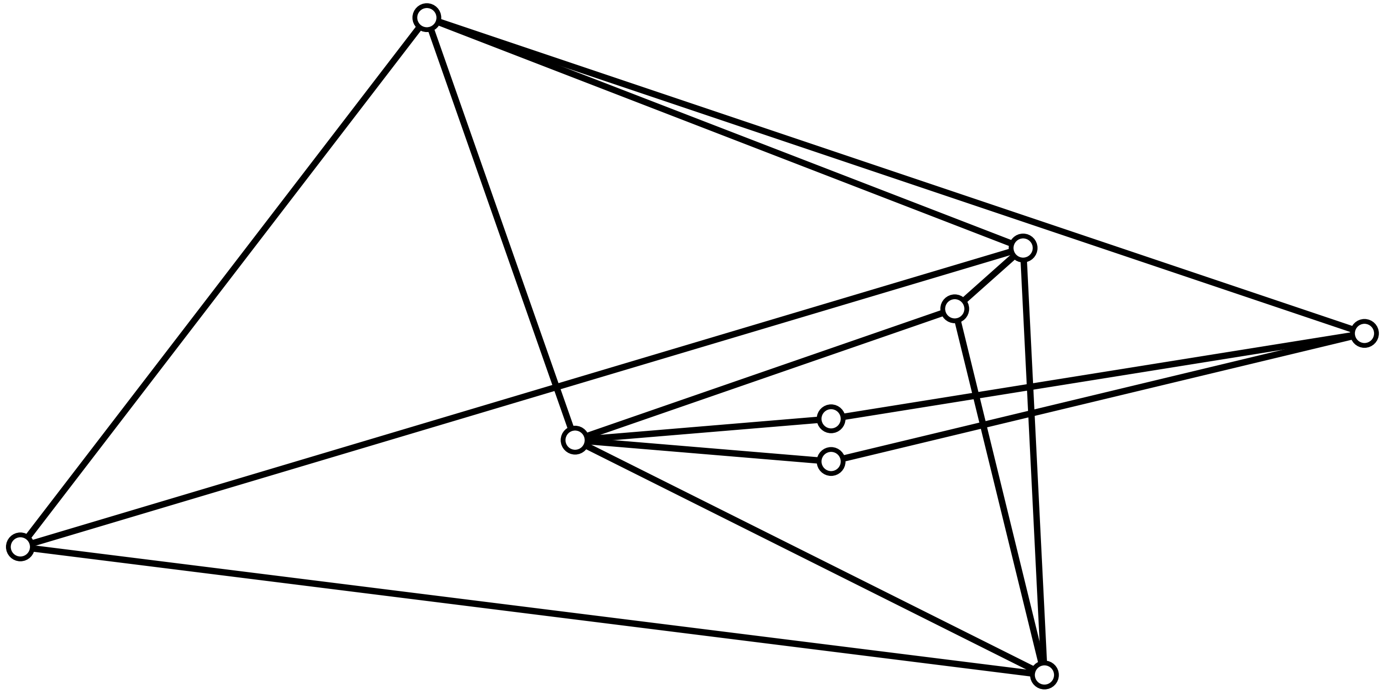
S: set of random points in R

Move v to best position in S

restrict S to a square R around v

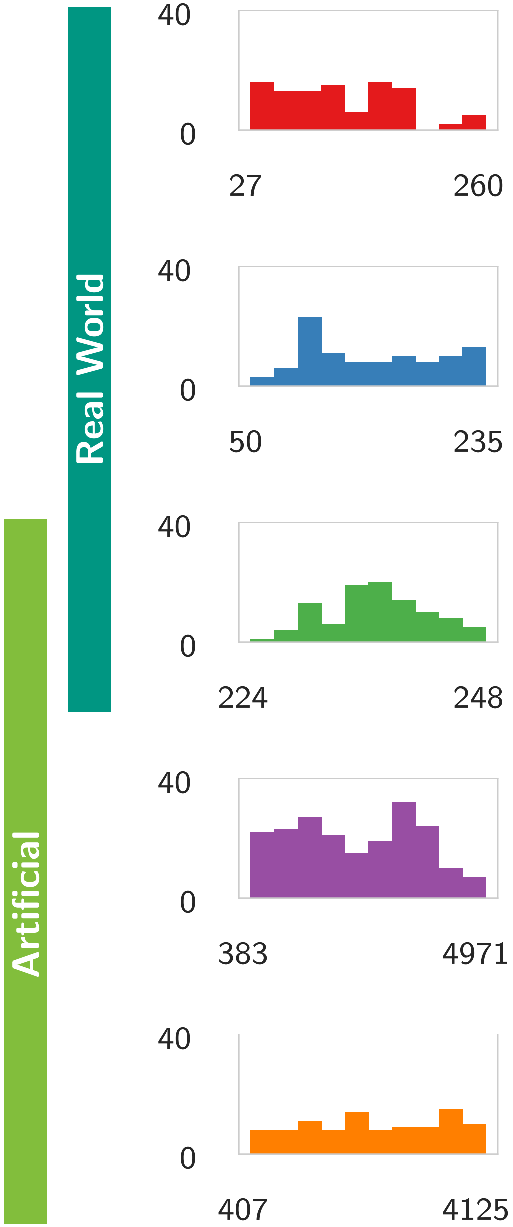
repeat

fine tune



Evaluation

Test Instances



North

- collection of small real world graphs

Rome

- collection of small real world graphs

Community

- Resembles community structure

1-Planar

Triangulation + X

- Triangulation + set of random edges

100 randomly selected graphs per class

Research Questions

Q: What is good parametrization of our algorithm?

Q: What is a good choice for an initial drawing?

Q: Does our heuristic improve the crossing angle?

Research Questions

Q: What is good parametrization of our algorithm?

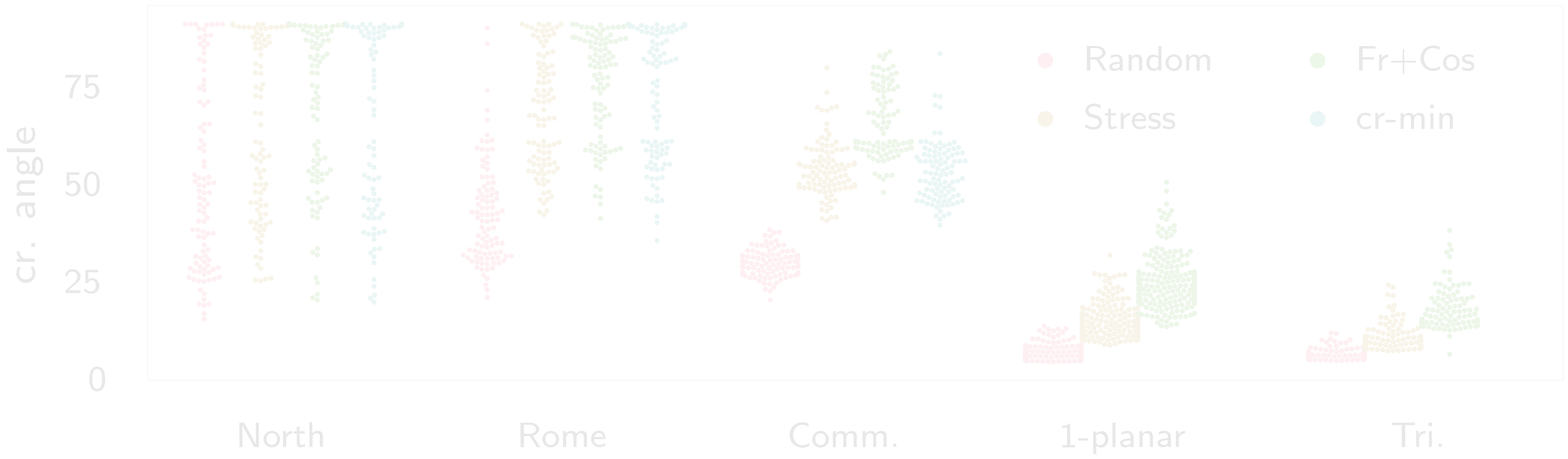
Q: What is a good choice for an initial drawing?

Q: Does our heuristic improve the crossing angle?

Good Initial Drawing

Initial Drawing Styles:

Random: random position per vertex
Fr+Cos: Force-Directed + Angle Max. Force [Huang et al. '14]
Stress: Stress Majorization [Gansner et al. '05, OGDF]
cr-small: Drawing with small number of crossings [R. et al.'18]



Observations

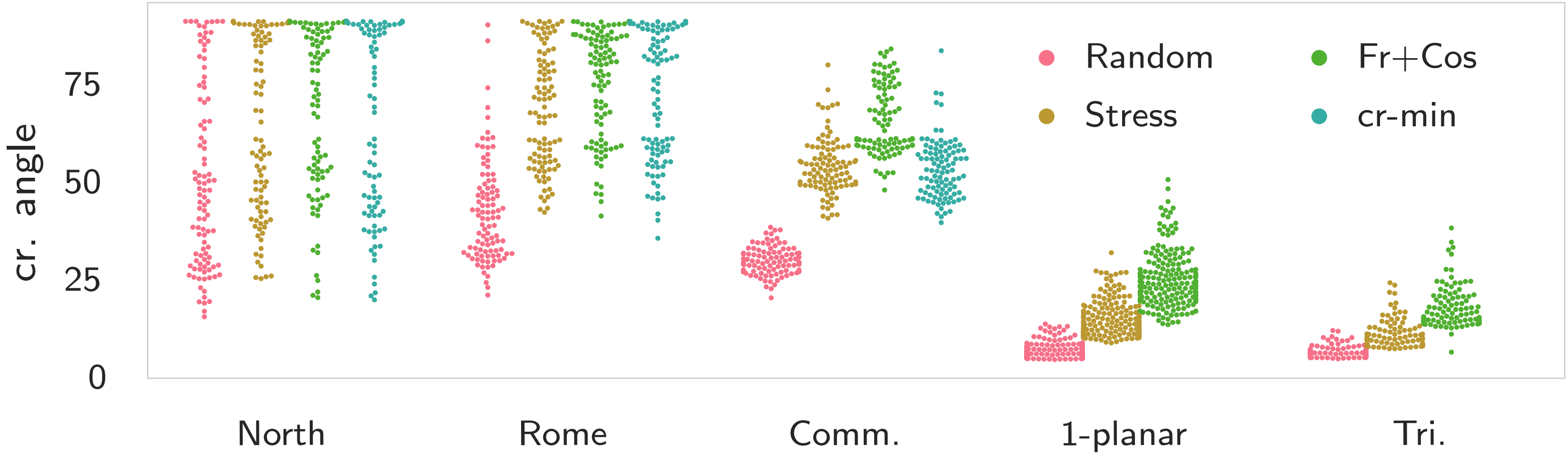
Random seems to be a bad choice

Tendence towards Fr+Cos

Good Initial Drawing

Initial Drawing Styles:

Random: random position per vertex
Fr+Cos: Force-Directed + Angle Max. Force [Huang et al. '14]
Stress: Stress Majorization [Gansner et al. '05, OGDF]
cr-small: Drawing with small number of crossings [R. et al.'18]



Observations

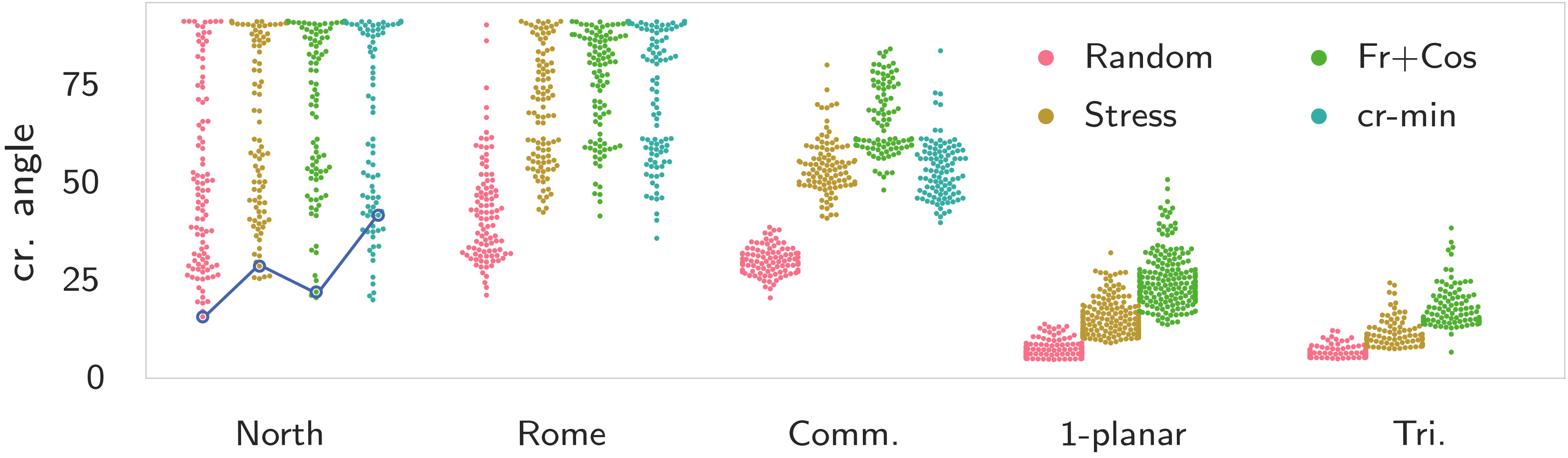
Random seems to be a bad choice

Tendence towards Fr+Cos

Good Initial Drawing

Initial Drawing Styles:

Random: random position per vertex
Fr+Cos: Force-Directed + Angle Max. Force [Huang et al. '14]
Stress: Stress Majorization [Gansner et al. '05, OGDF]
cr-small: Drawing with small number of crossings [R. et al.'18]



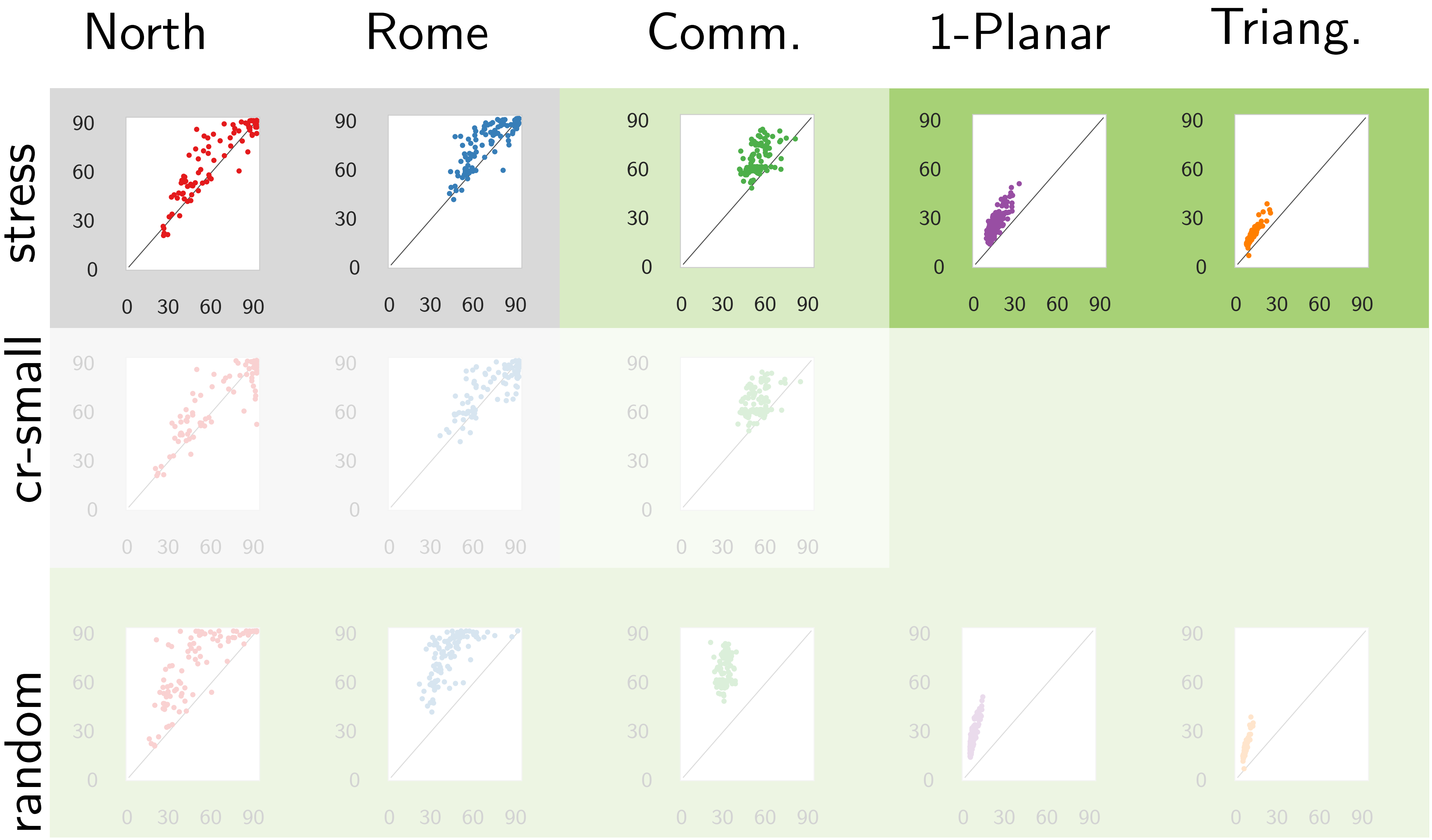
Observations

Random seems to be a bad choice

Tendence towards Fr+Cos

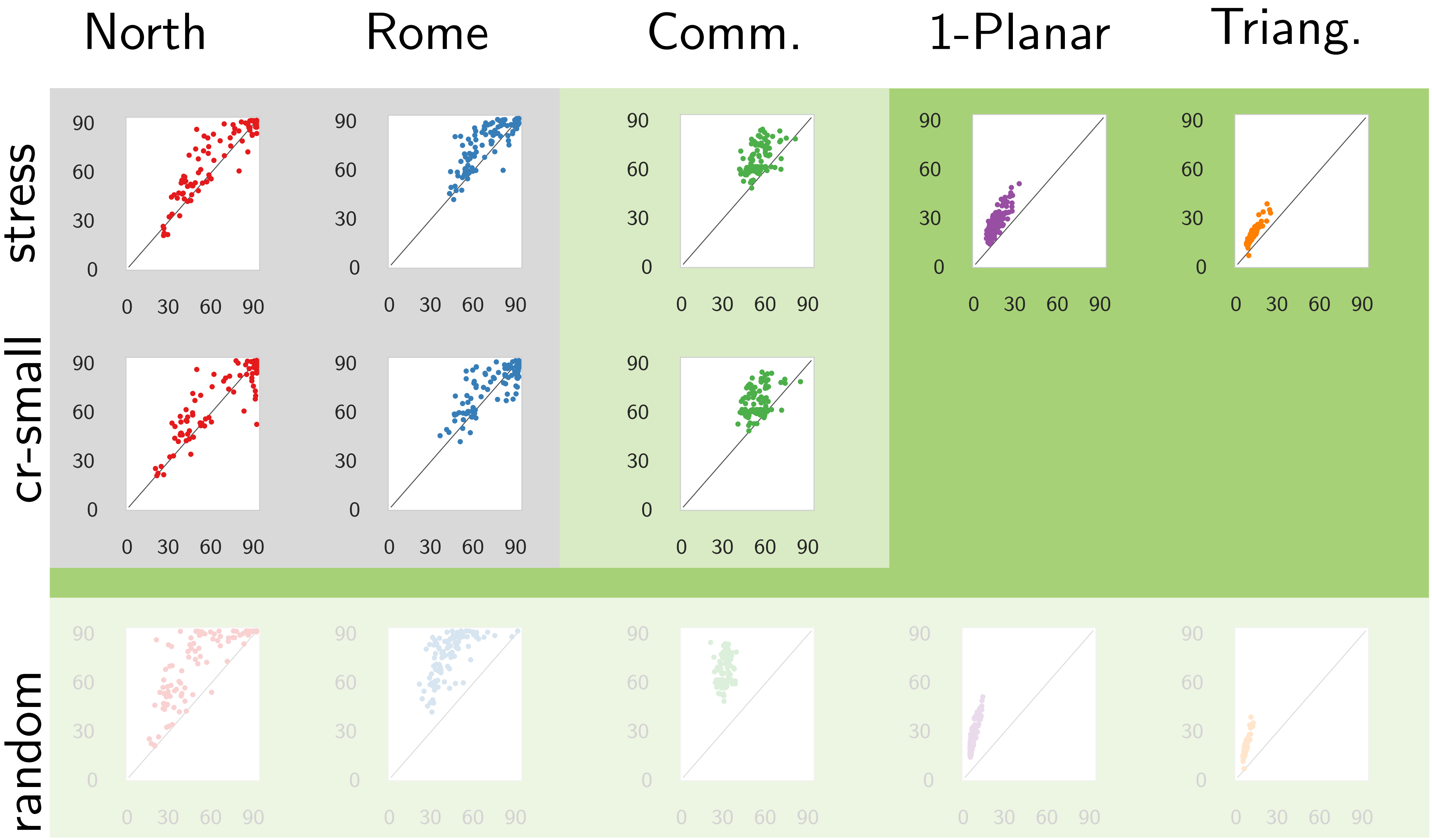
Good Initial Drawing

Q: Is Fr+Cos a good initial drawing?



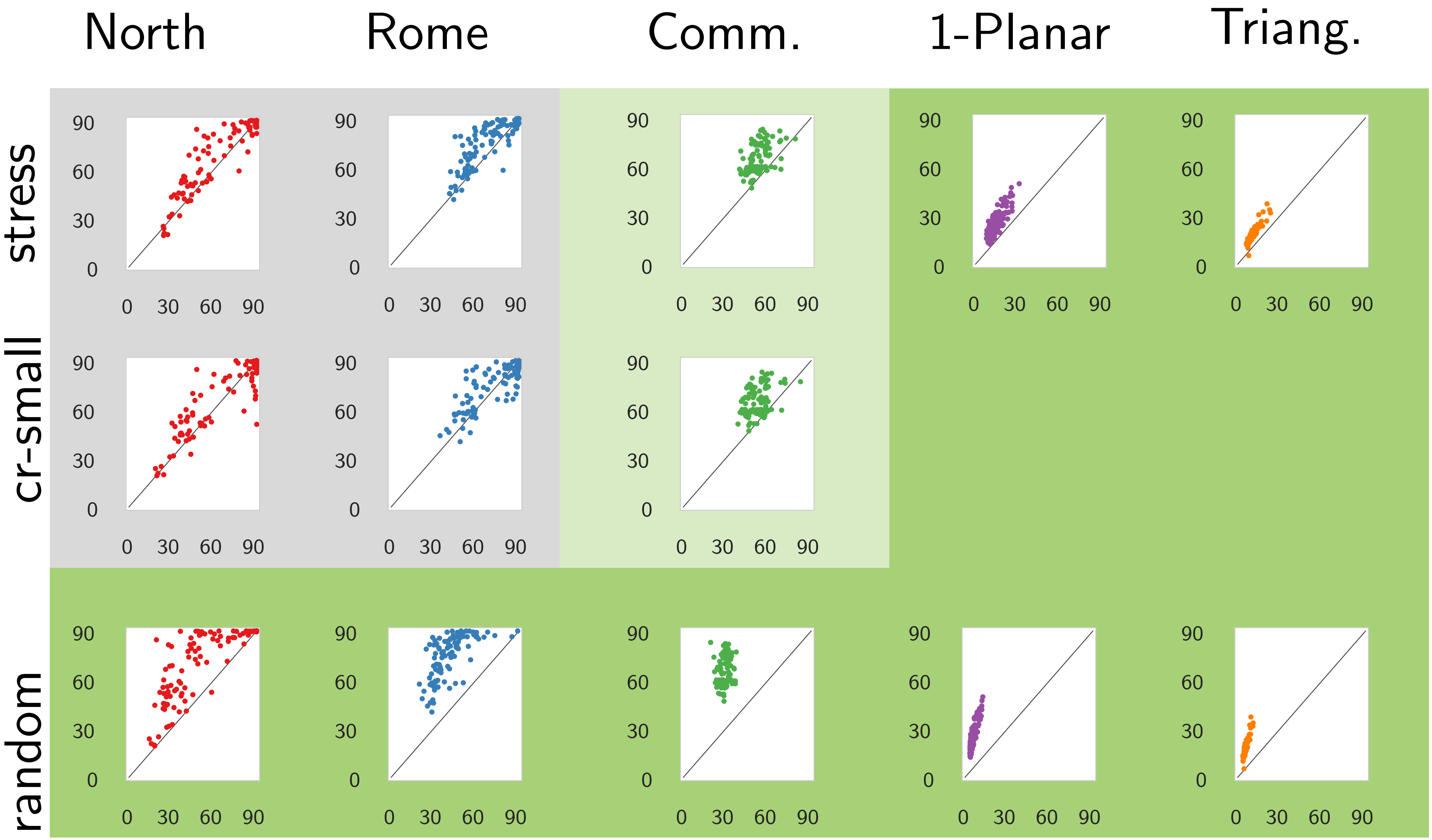
Good Initial Drawing

Q: Is Fr+Cos a good initial drawing?



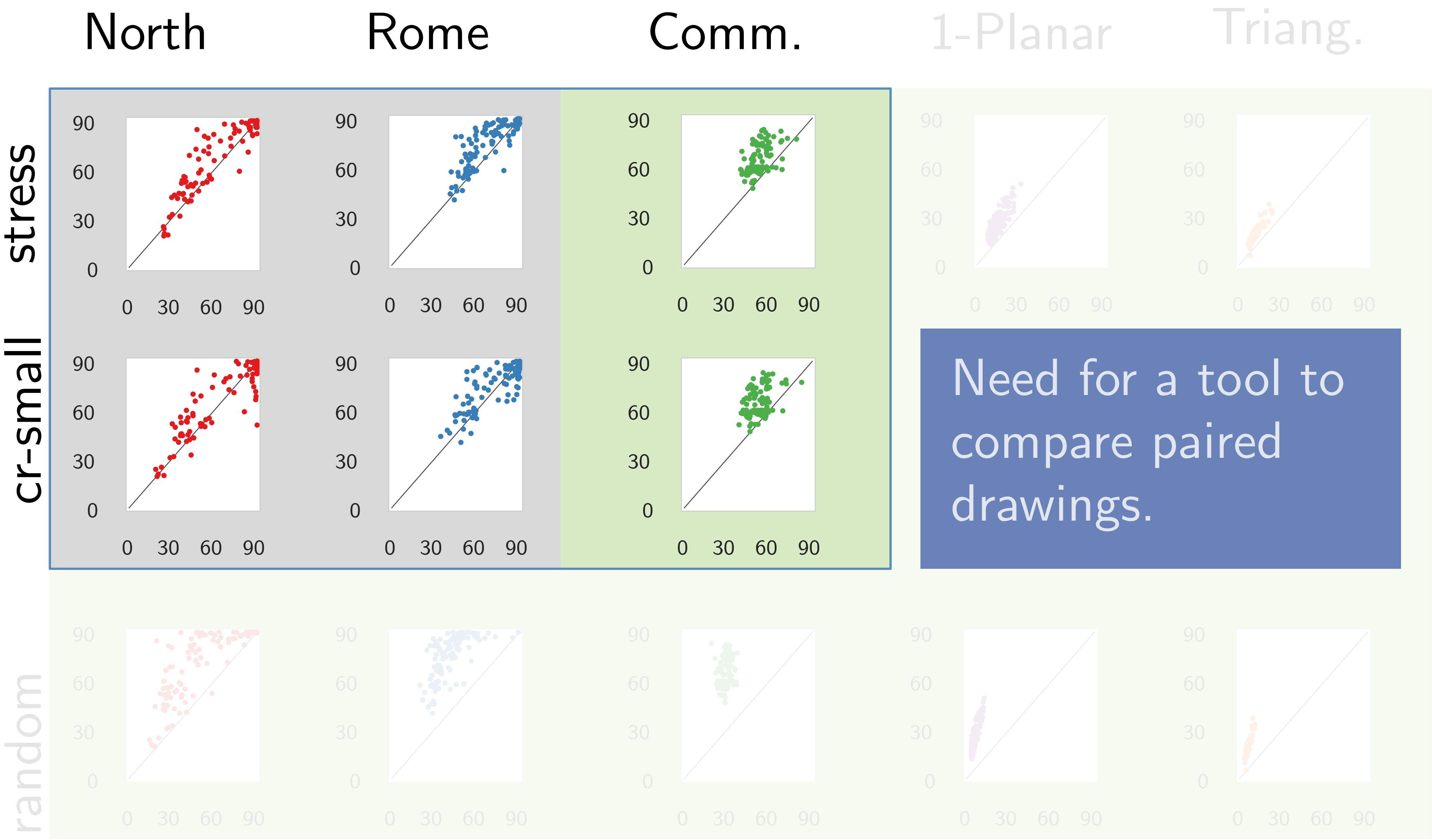
Good Initial Drawing

Q: Is Fr+Cos a good initial drawing?



Good Initial Drawing

Q: Is Fr+Cos a good initial drawing?

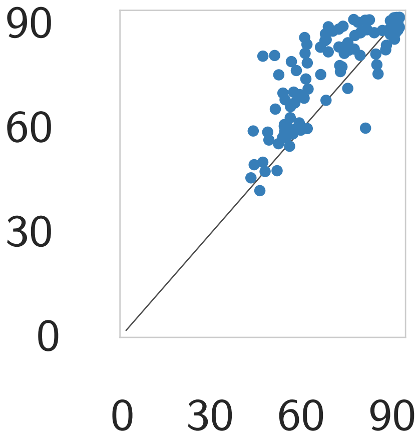


Need for a tool to compare paired drawings.

Tool to Compare Paired Drawings

Input: Ground set of Graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$

Two sets of drawings of $\mathcal{G}$ $\{\Gamma[G_i] \mid G_i \in \mathcal{G}\}$ $\{\Pi[G_i] \mid G_i \in \mathcal{G}\}$



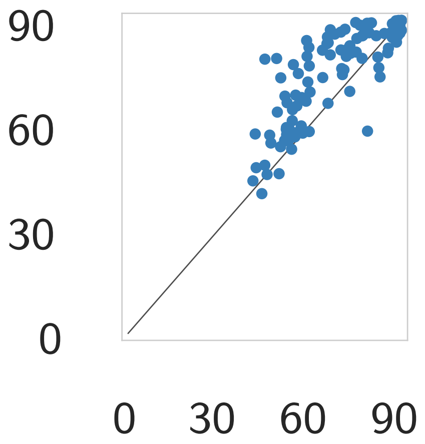
Tool to Compare Paired Drawings

Input: Ground set of Graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$

Two sets of drawings of $\mathcal{G}$ $\{\Gamma[G_i] \mid G_i \in \mathcal{G}\}$ $\{\Pi[G_i] \mid G_i \in \mathcal{G}\}$

Q: Do the drawings Γ have a larger crossing angle than Π?

for all G_i $\quad$ $\text{cr-}\angle(\Gamma[G_i]) > \text{cr-}\angle(\Pi[G_i])$



Tool to Compare Paired Drawings

Input: Ground set of Graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$

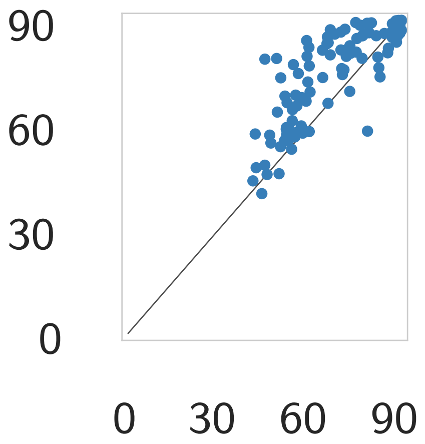
Two sets of drawings of $\mathcal{G}$ $\{\Gamma[G_i] \mid G_i \in \mathcal{G}\}$ $\{\Pi[G_i] \mid G_i \in \mathcal{G}\}$

A number $p \in [0, 1]$

Q: Do the drawings Γ have a larger crossing angle than Π?

Is there: a subset $\mathcal{G}' \subseteq \mathcal{G}, |\mathcal{G}'| > p \cdot |\mathcal{G}|$ such that

for all $G_i \in \mathcal{G}'$: $\text{cr-}\angle(\Gamma[G_i]) > \text{cr-}\angle(\Pi[G_i])$



Tool to Compare Paired Drawings

Input: Ground set of Graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$

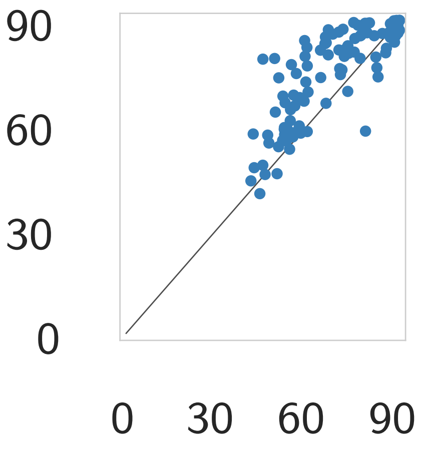
Two sets of drawings of $\mathcal{G}$ $\{\Gamma[G_i] \mid G_i \in \mathcal{G}\}$ $\{\Pi[G_i] \mid G_i \in \mathcal{G}\}$

A number $p \in [0, 1]$, $\Delta > 0$

Q: Do the drawings Γ have a larger crossing angle than Π?

Is there: a subset $\mathcal{G}' \subseteq \mathcal{G}, |\mathcal{G}'| > p \cdot |\mathcal{G}|$ such that

for all $G_i \in \mathcal{G}'$: $\text{cr-}\angle(\Gamma[G_i]) > \text{cr-}\angle(\Pi[G_i]) + \Delta$



Tool to Compare Paired Drawings

Input: Ground set of Graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$

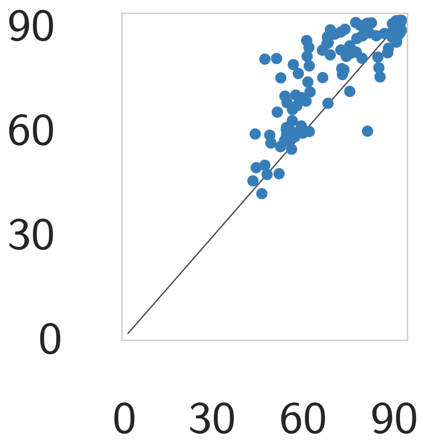
Two sets of drawings of $\mathcal{G}$ $\{\Gamma[G_i] \mid G_i \in \mathcal{G}\}$ $\{\Pi[G_i] \mid G_i \in \mathcal{G}\}$

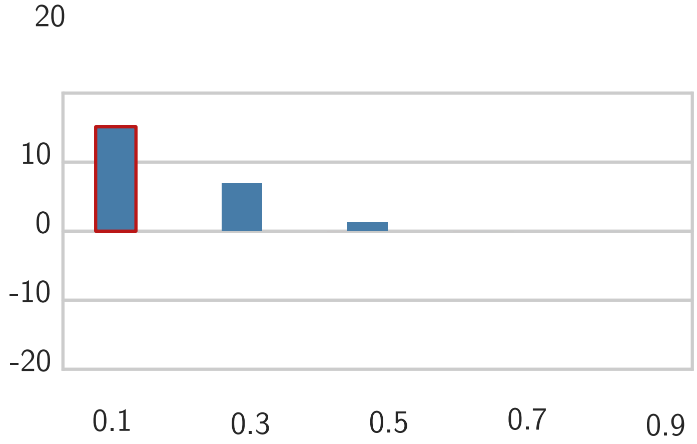
A number $p \in [0,1]$, $\Delta > 0$

Q: Do the drawings Γ have a larger crossing angle than Π?

Is there: a subset $\mathcal{G}' \subseteq \mathcal{G}, |\mathcal{G}'| > p \cdot |\mathcal{G}|$ such that

for all $G_i \in \mathcal{G}'$: $\text{cr-}\angle(\Gamma[G_i]) > \text{cr-}\angle(\Pi[G_i]) + \Delta$





fr-cos cr-small

Tool to Compare Paired Drawings

Input: Ground set of Graphs $\mathcal{G} = \{G_1, G_2, \dots, G_n\}$

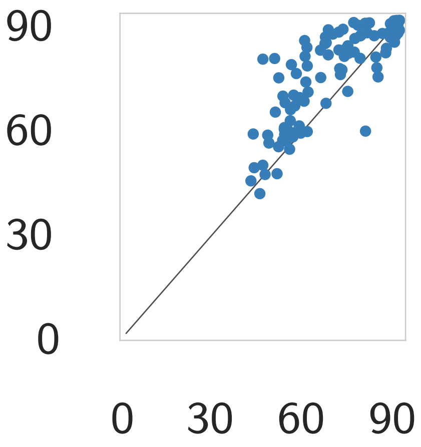
Two sets of drawings of $\mathcal{G}$ $\{\Gamma[G_i] \mid G_i \in \mathcal{G}\}$ $\{\Pi[G_i] \mid G_i \in \mathcal{G}\}$

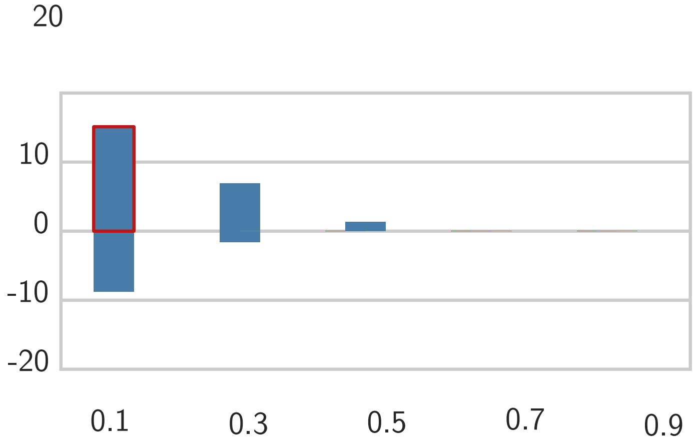
A number $p \in [0, 1]$, $\Delta > 0$

Q: Do the drawings Γ have a larger crossing angle than Π?

Is there: a subset $\mathcal{G}' \subseteq \mathcal{G}, |\mathcal{G}'| > p \cdot |\mathcal{G}|$ such that

for all $G_i \in \mathcal{G}'$: $\text{cr-}\angle(\Gamma[G_i]) > \text{cr-}\angle(\Pi[G_i]) + \Delta$

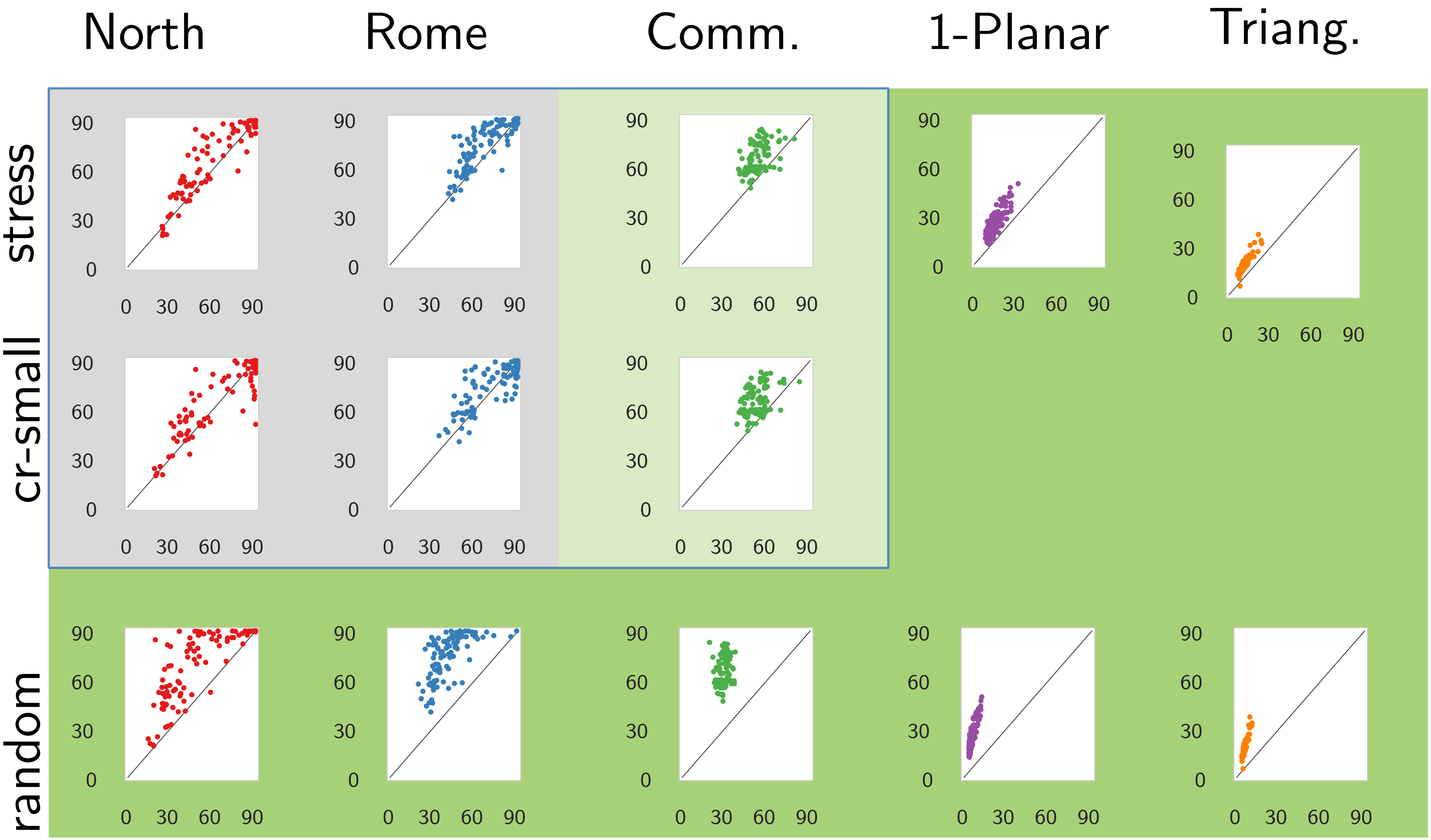




fr-cos cr-small

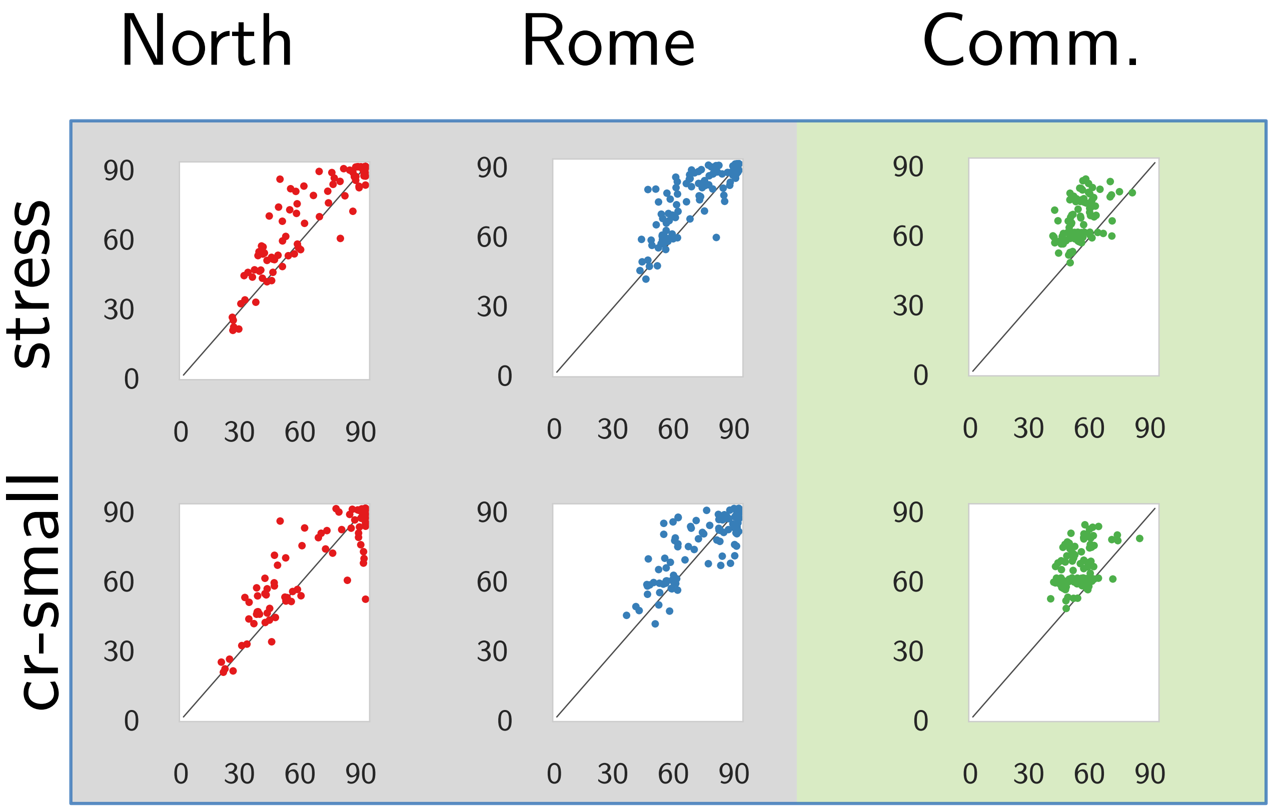
Good Initial Drawing

Q: Is Fr+Cos a good initial drawing?

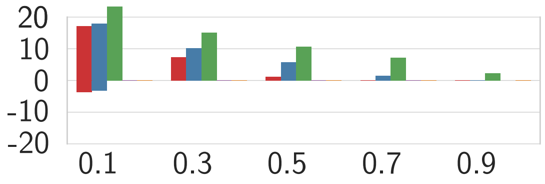


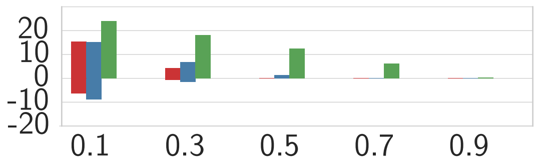
Good Initial Drawing

Q: Is Fr+Cos a good initial drawing?



fr-cos vs stress





fr-cos vs cr-small

Research Questions

Q: What is good parametrization of our algorithm?

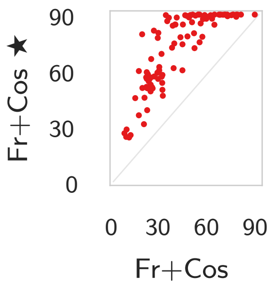
Q: What is a good choice for an initial drawing?

Q: Does our heuristic improve the crossing angle?

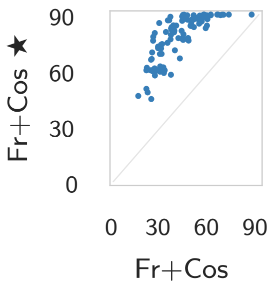
Improvement of the Crossing Angle

Q: Does our Heuristic improve the Crossing Angle?

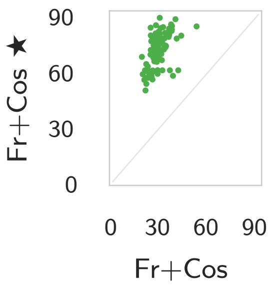
North



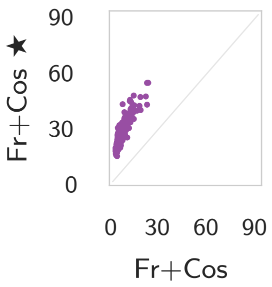
Rome



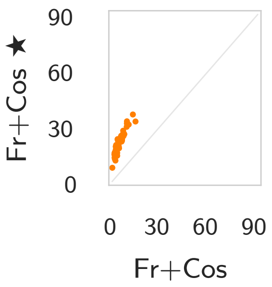
Comm.



1-Planar

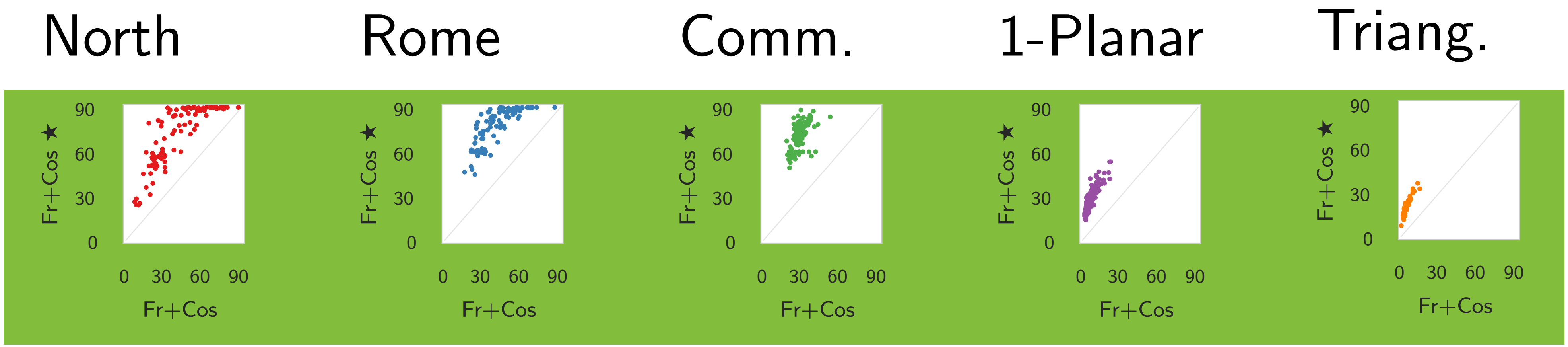


Triang.



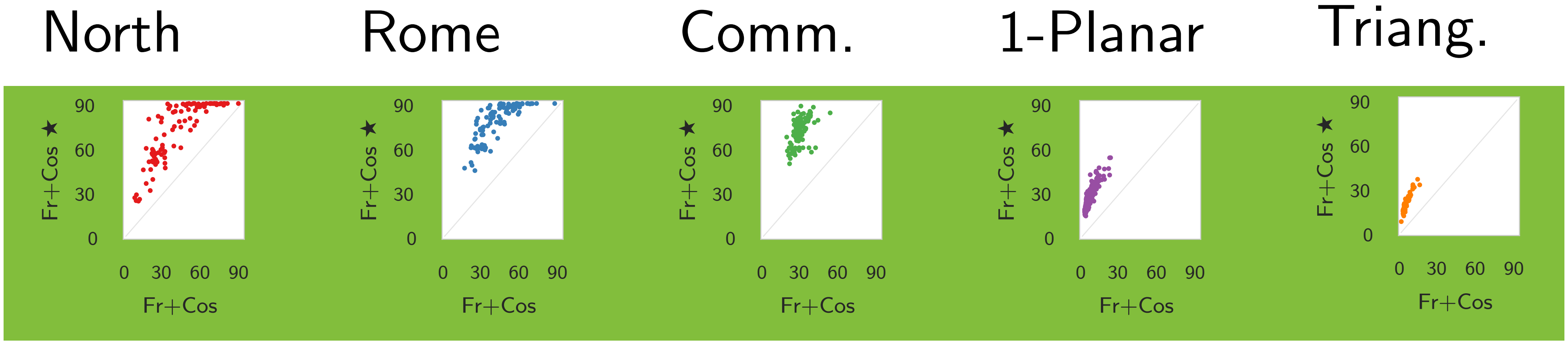
Improvement of the Crossing Angle

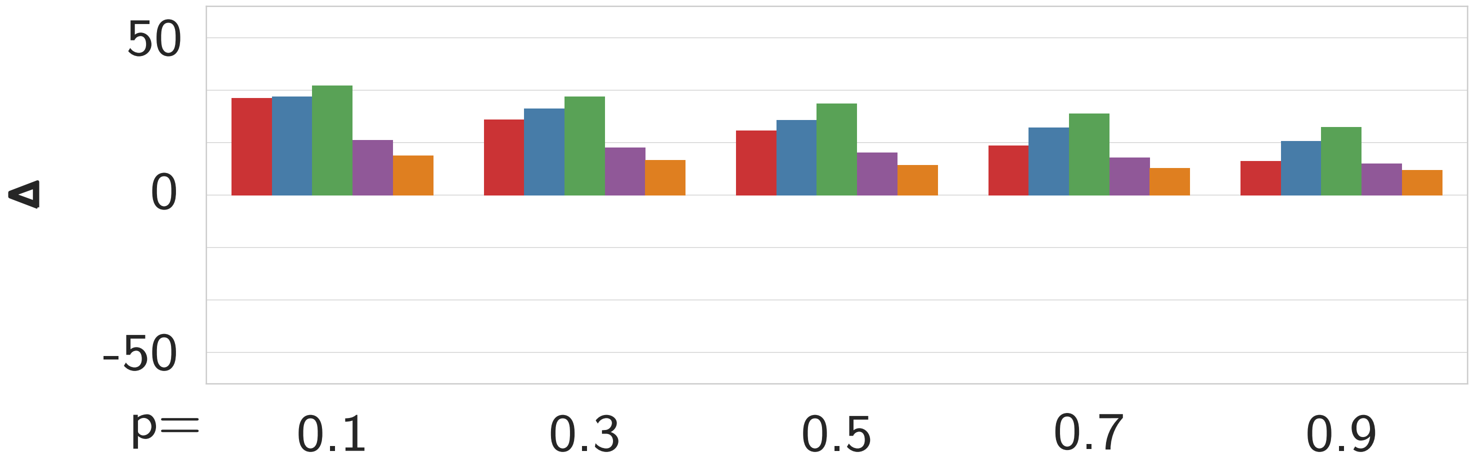
Q: Does our Heuristic improve the Crossing Angle?



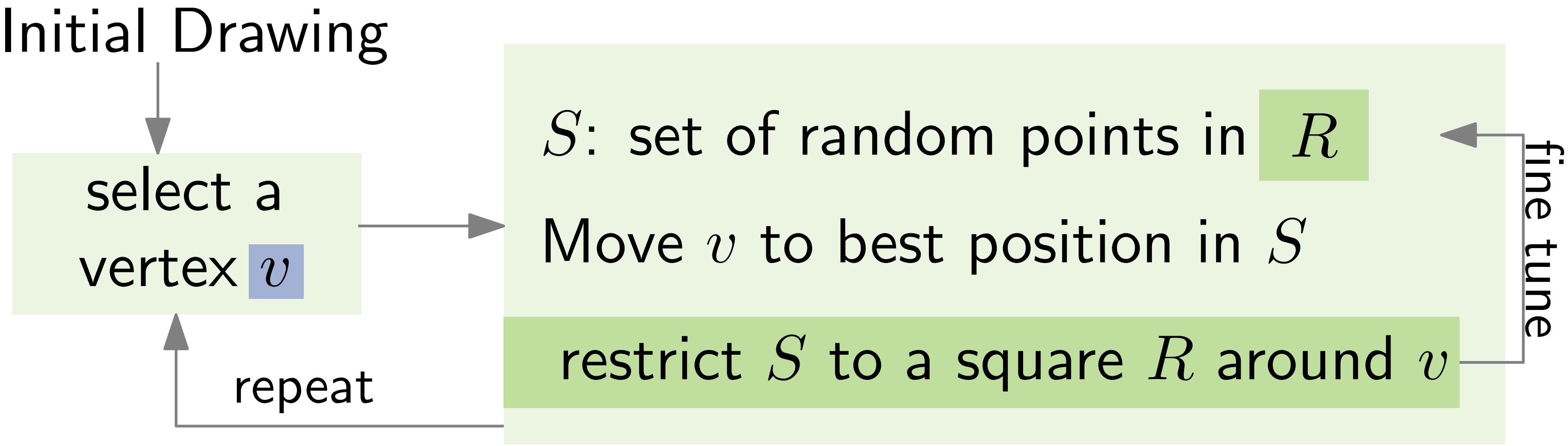
Improvement of the Crossing Angle

Q: Does our Heuristic improve the Crossing Angle?





Our Heuristic



Initial Drawing

select a vertex v

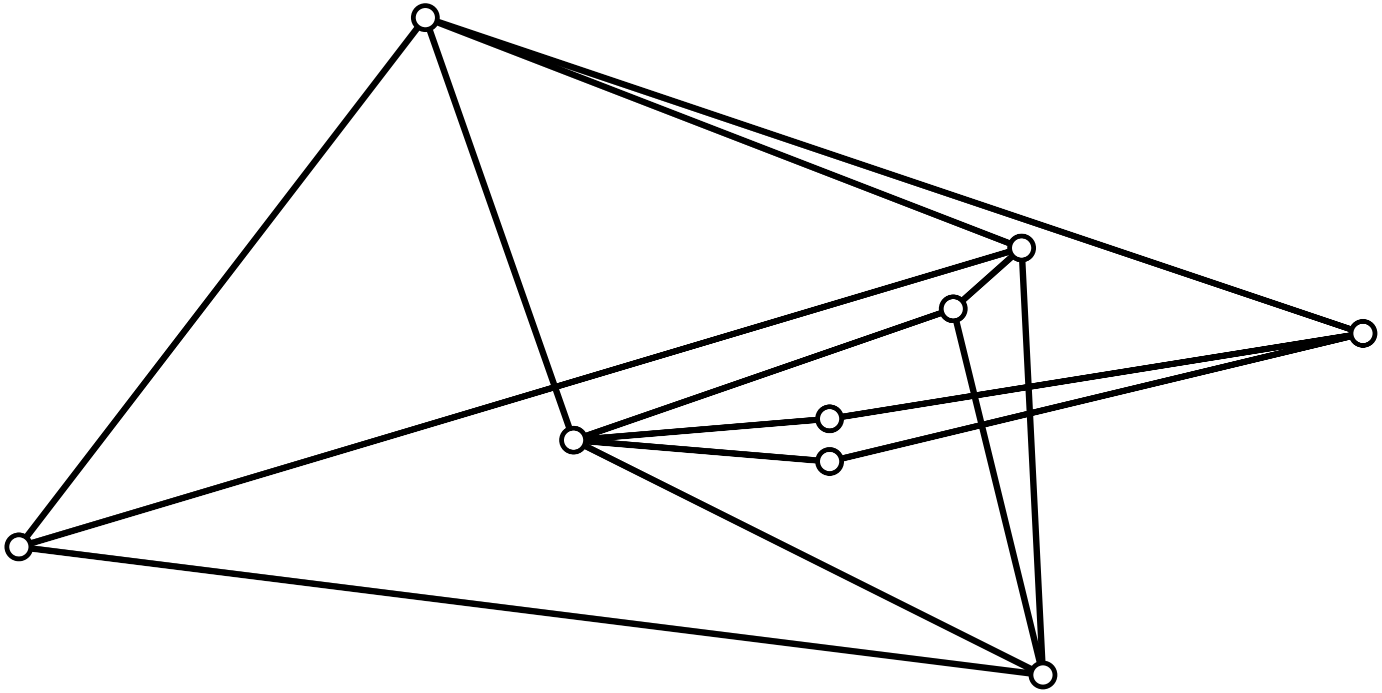
S: set of random points in R

Move v to best position in S

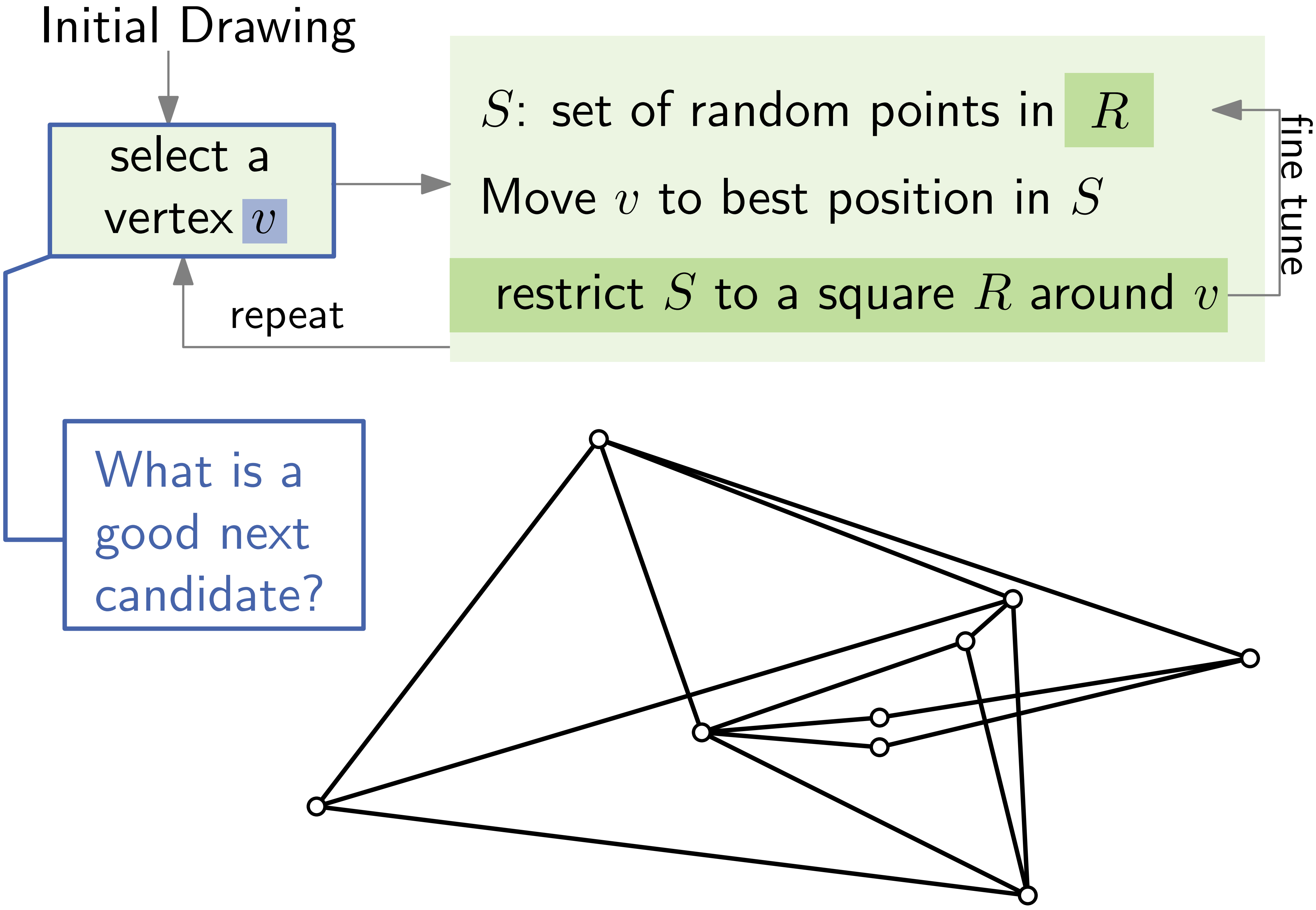
restrict S to a square R around v

fine tune

repeat



Our Heuristic



Initial Drawing

select a vertex v

S: set of random points in R

Move v to best position in S

restrict S to a square R around v

fine tune

repeat

What is a good next candidate?

Running Time

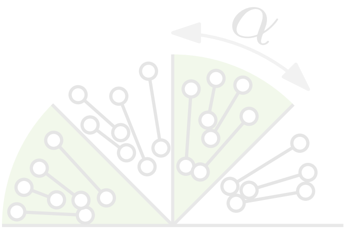
Task Find edges e,f s.t. $\text{cr-}\angle(\Gamma, e, f) = \text{cr-}\angle(\Gamma)|$

Possibility *Sweep*: Sweep-Line Algorithm

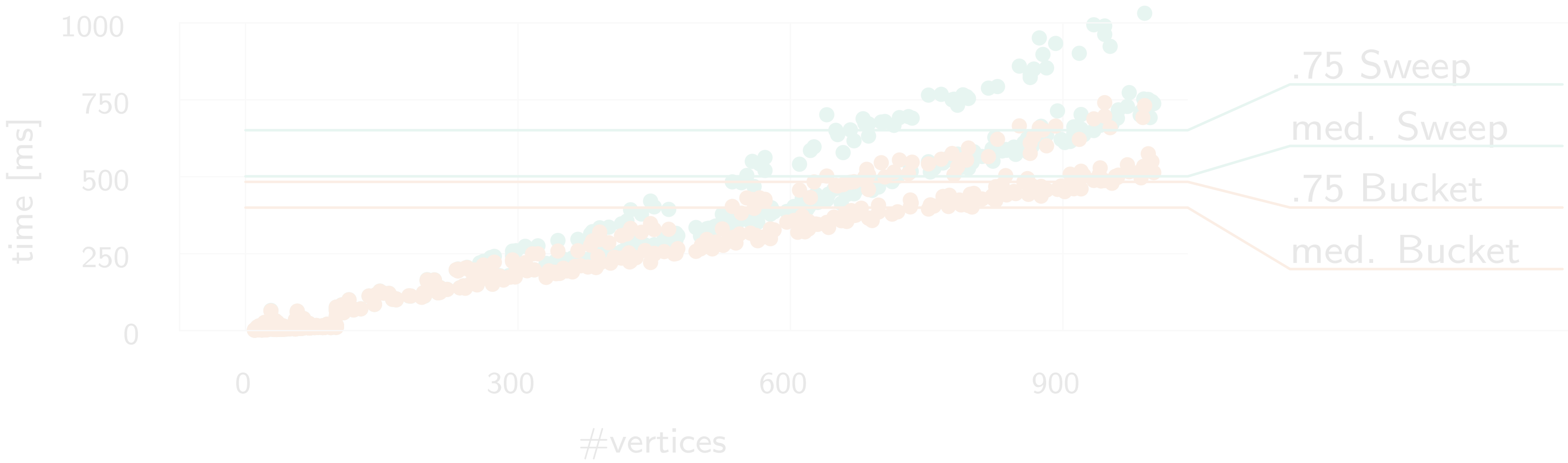
Possibility *Bucket*:

sort edges into buckets according to slopes

edges of adjacent buckets form $\text{cr-}\angle(\Gamma)$



Time to move a single vertex



Running Time

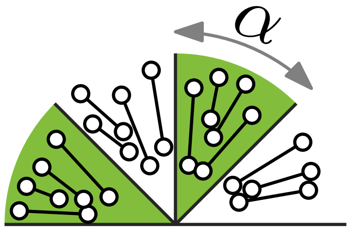
Task Find edges e,f s.t. $\text{cr-}\angle(\Gamma, e, f) = \text{cr-}\angle(\Gamma)|$

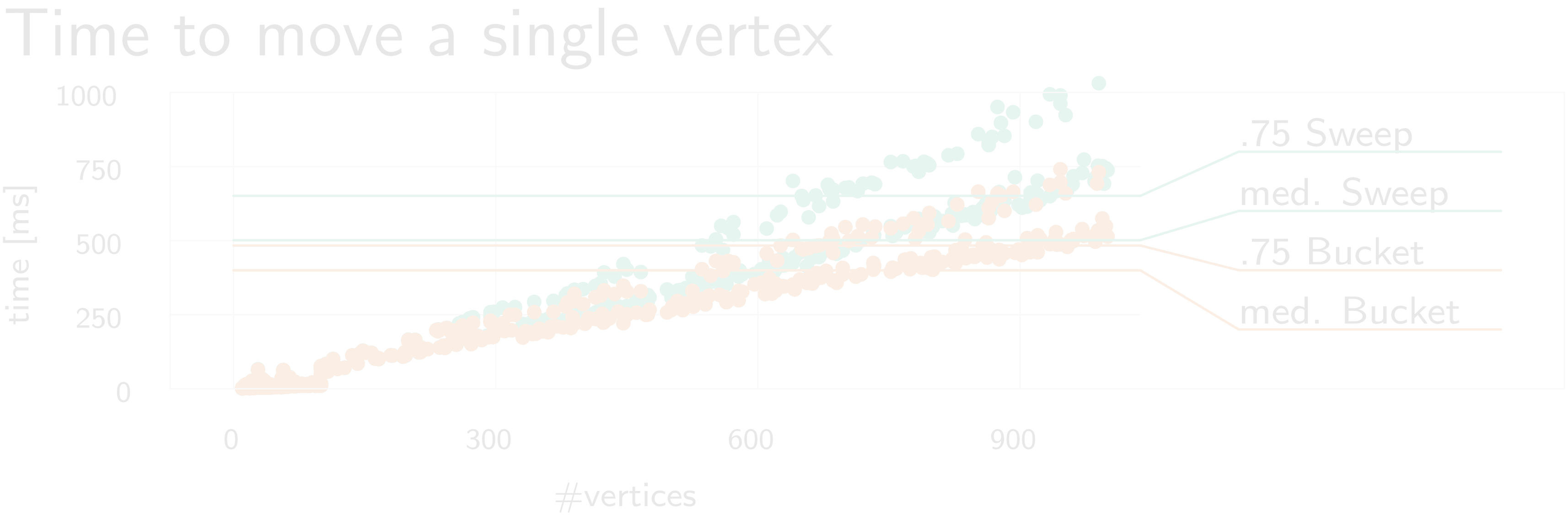
Possibility *Sweep*: Sweep-Line Algorithm

Possibility *Bucket*:

sort edges into buckets according to slopes

edges of adjacent buckets form $\text{cr-}\angle(\Gamma)$





Running Time

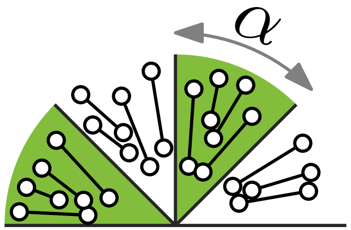
Task Find edges e,f s.t. $\text{cr-}\angle(\Gamma, e, f) = \text{cr-}\angle(\Gamma)|$

Possibility *Sweep*: Sweep-Line Algorithm

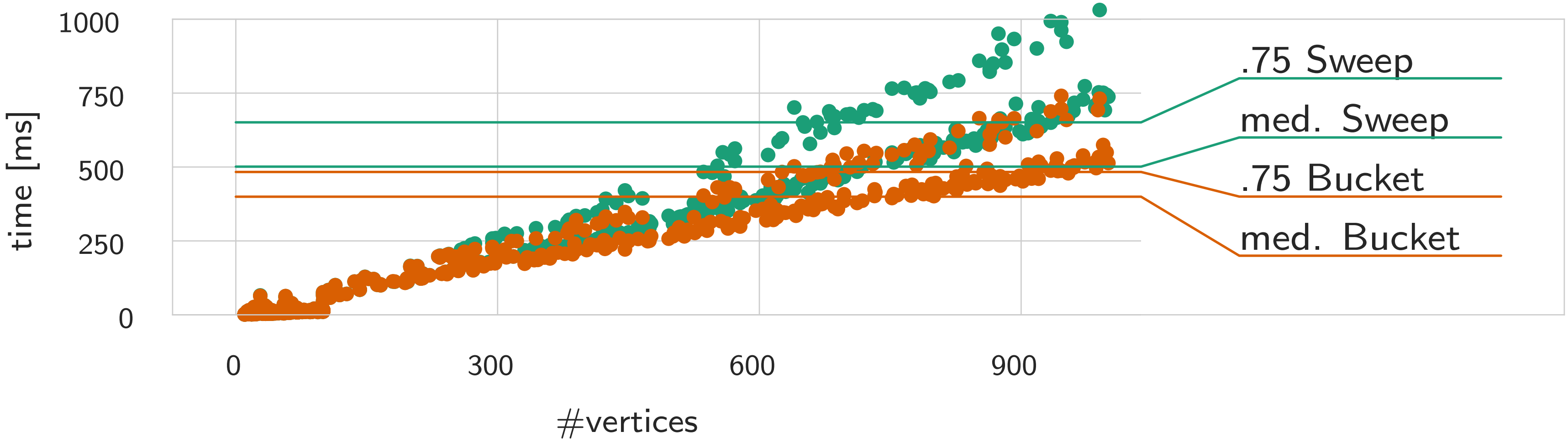
Possibility *Bucket*:

sort edges into buckets according to slopes

edges of adjacent buckets form $\text{cr-}\angle(\Gamma)$

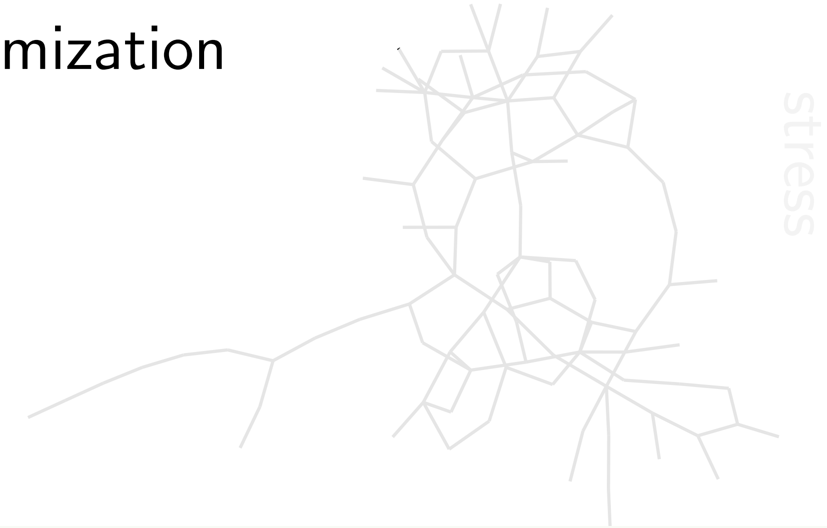


Time to move a single vertex



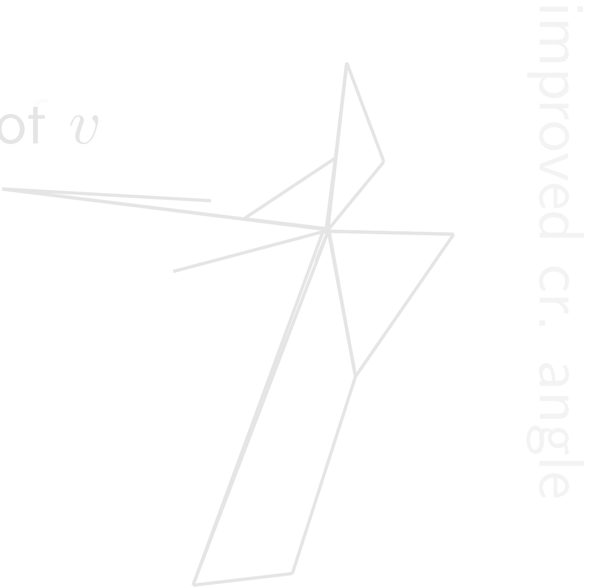
Conclusion

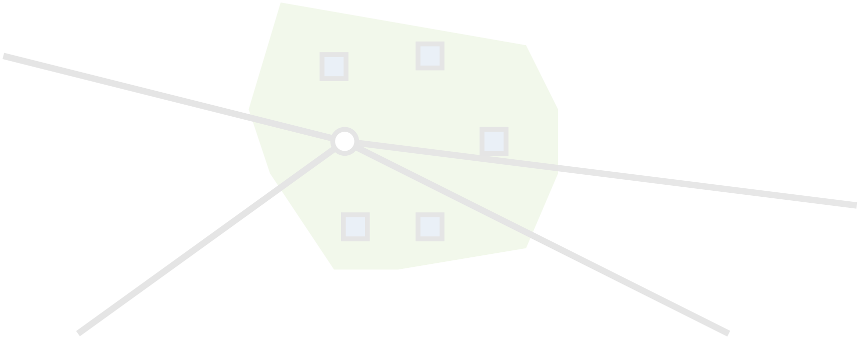
- a simple heuristic for Crossing-Angle Maximization
    - easy to implement
    - generic



Future Work

- Drawings are not necessarily *readable*
    - Let R be a region that ensure some properties of v
    - Optimize position of v within R

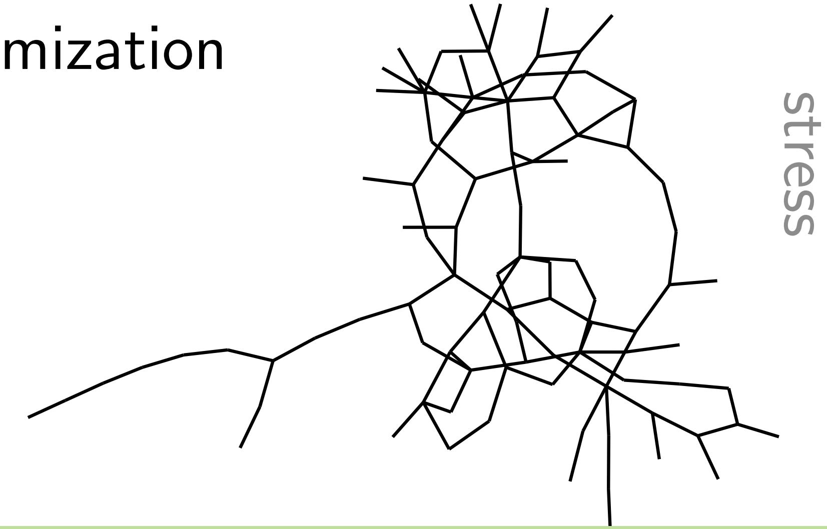

improved cr. angle



Conclusion

a simple heuristic for Crossing-Angle Maximization

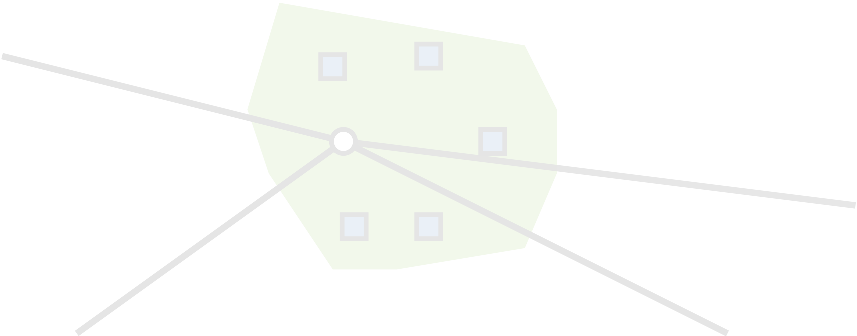
- easy to implement
- generic

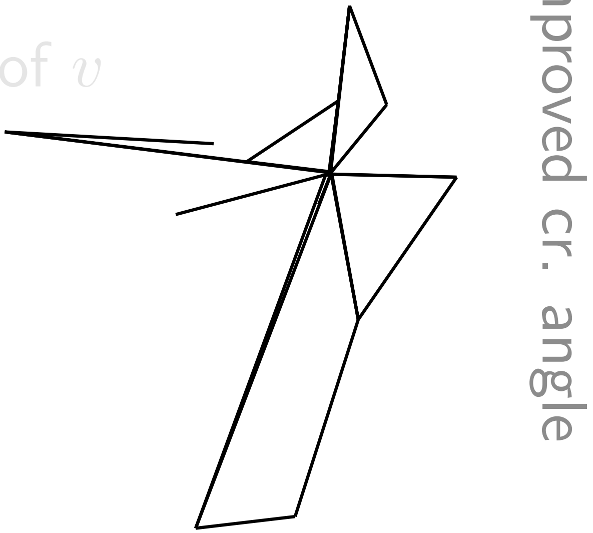


Future Work

Drawings are not necessarily *readable*

- Let R be a region that ensure some properties of v
- Optimize position of v within R

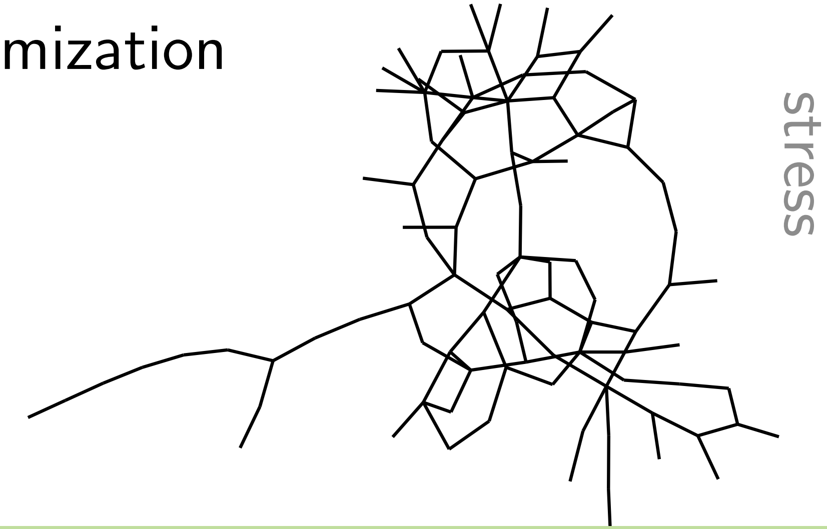



improved cr. angle

Conclusion

a simple heuristic for Crossing-Angle Maximization

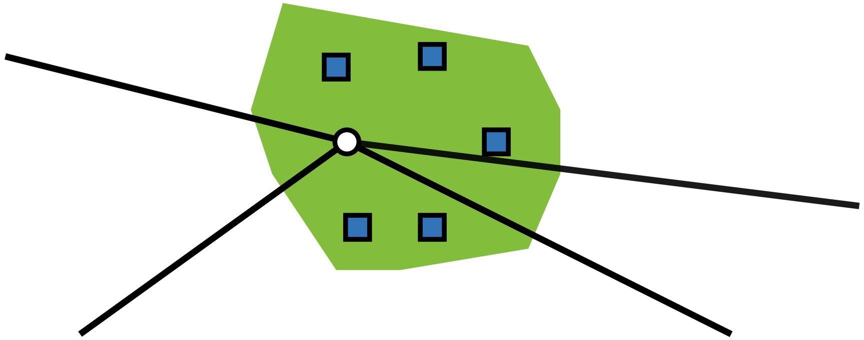
- easy to implement
- generic

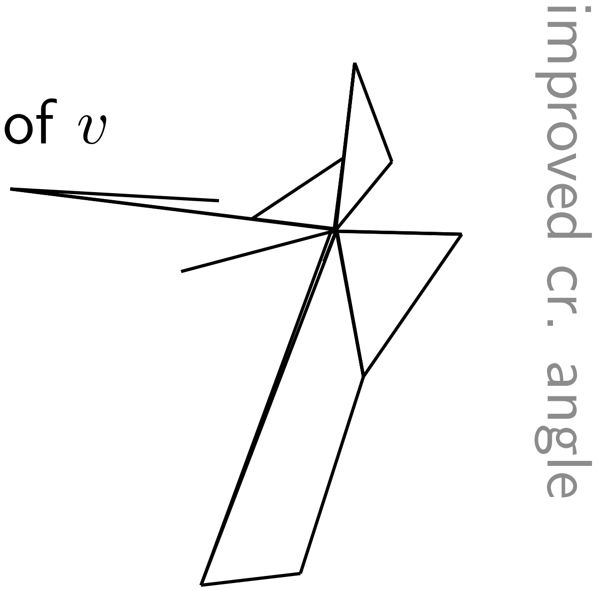


Future Work

Drawings are not necessarily *readable*

- Let R be a region that ensure some properties of v
- Optimize position of v within R

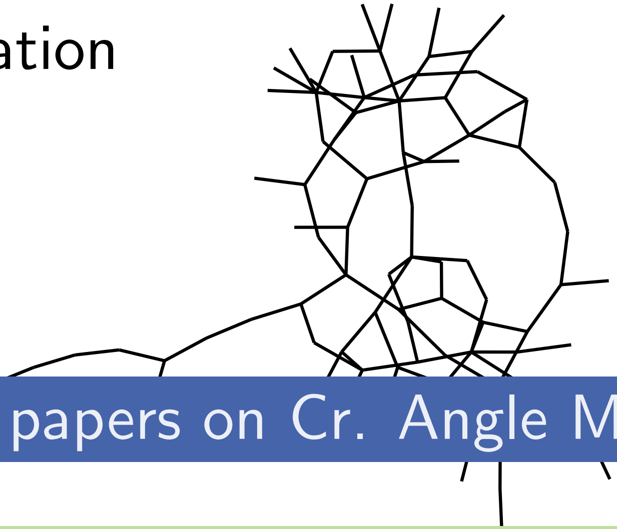



improved cr. angle

Conclusion

a simple heuristic for Crossing-Angle Maximization

- easy to implement
- generic

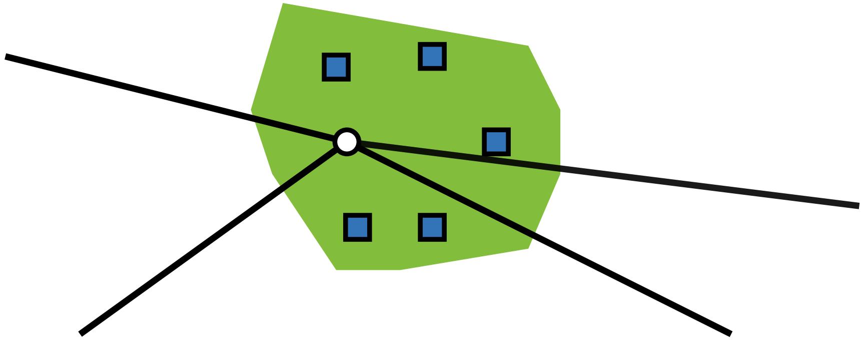


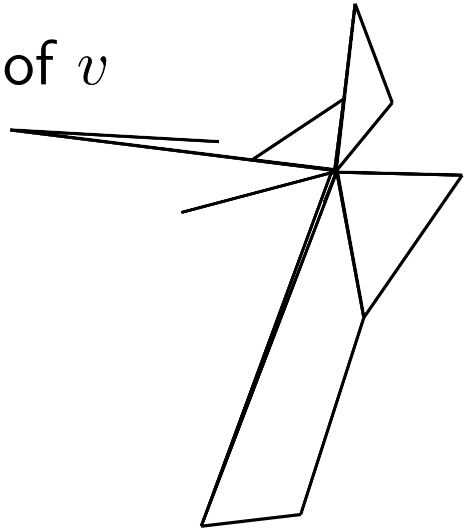
GD Contest duplicated the number of applied papers on Cr. Angle Max ;-)

Future Work

Drawings are not necessarily *readable*

- Let R be a region that ensure some properties of v
- Optimize position of v within R





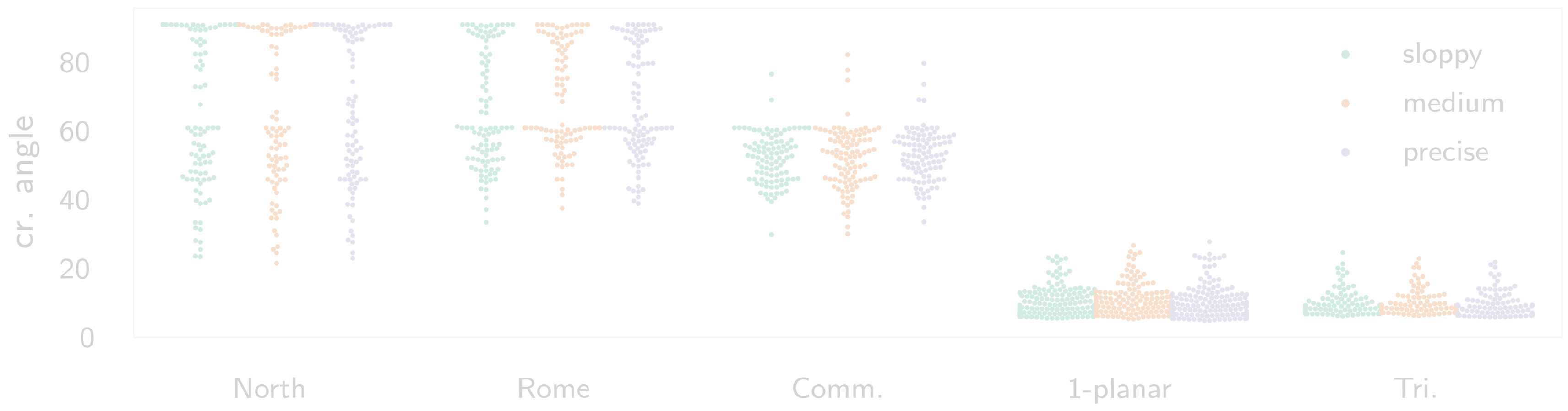
Thank you.

Good Parameter

Configurations:

Sloppy	Fast and *inaccurate*
Medium	Trade of between speed and accuracy
Precise	Slow and *accurate*

Time Limit: n seconds for an n-vertex graph

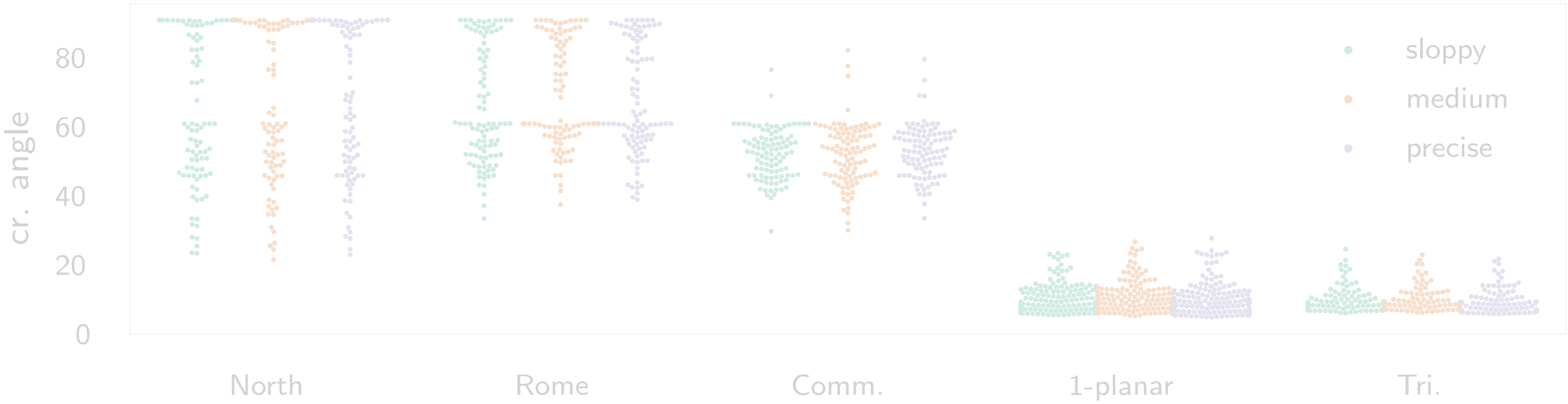


Good Parameter

Configurations:

Sloppy	Fast and *inaccurate*
Medium	Trade of between speed and accuracy
Precise	Slow and *accurate*

Time Limit: n seconds for an n-vertex graph ← allows fair comparison

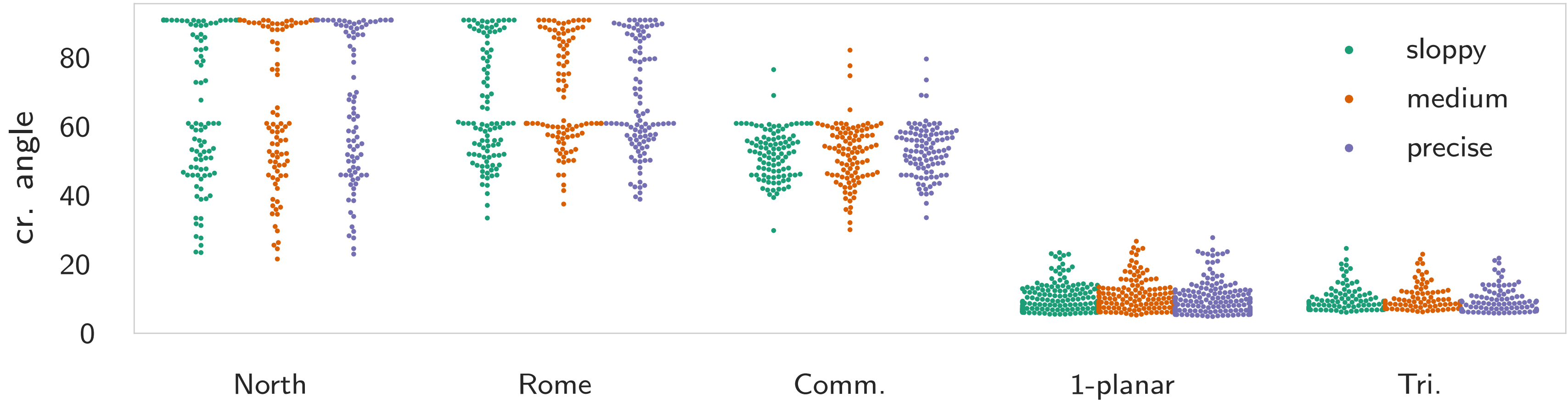


Good Parameter

Configurations:

Sloppy	Fast and *inaccurate*
Medium	Trade of between speed and accuracy
Precise	Slow and *accurate*

Time Limit: n seconds for an n-vertex graph ← allows fair comparison



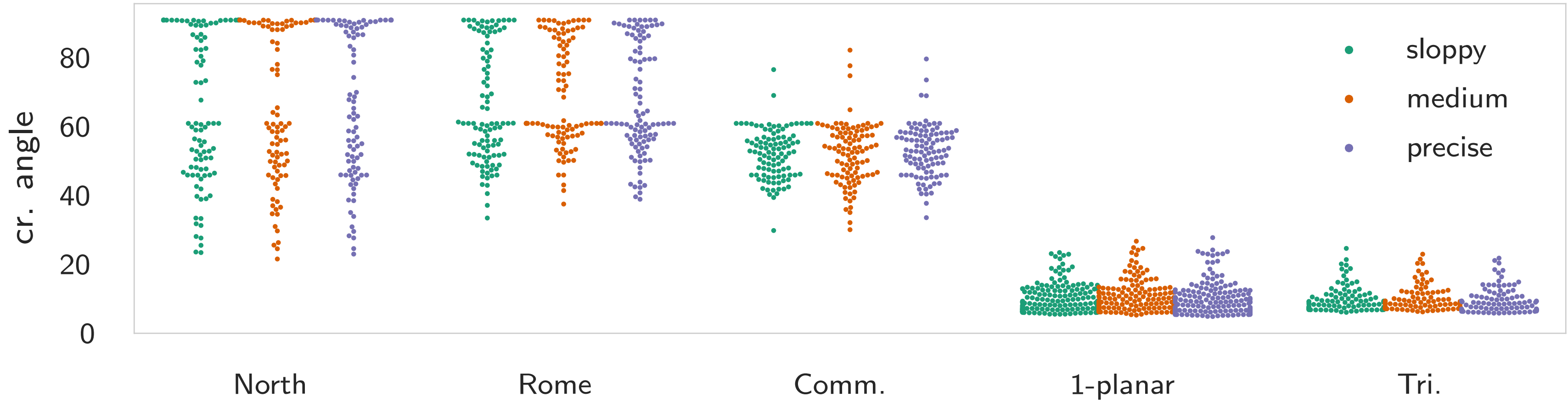
There is no obvious difference between the configurations

Good Parameter

Configurations:

Sloppy	Fast and *inaccurate*
Medium	Trade of between speed and accuracy
Precise	Slow and *accurate*

Time Limit: n seconds for an n-vertex graph ← allows fair comparison



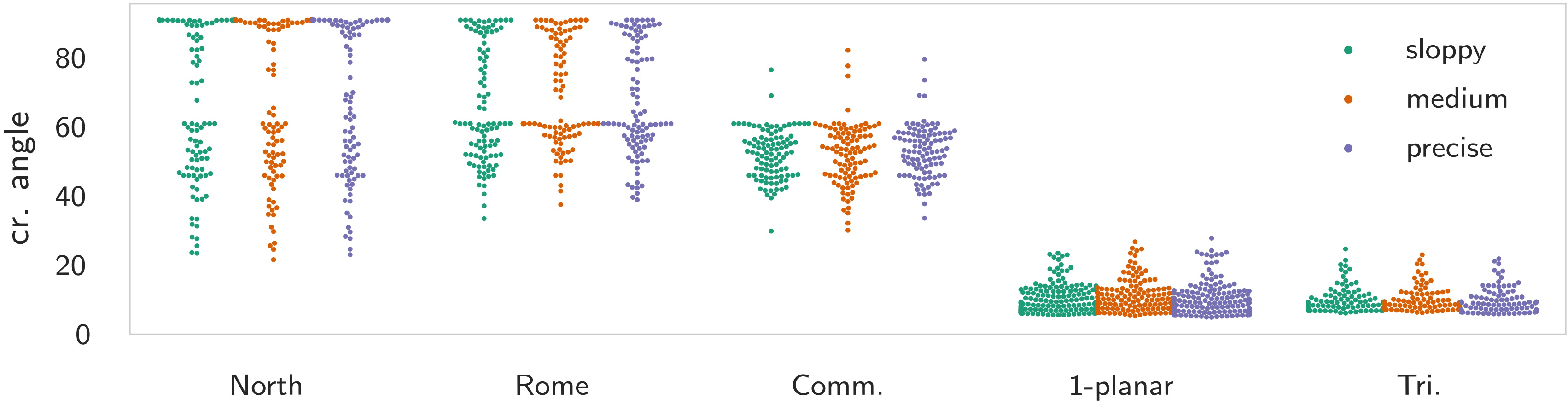
There is no obvious difference between the configurations

Good Parameter

Configurations:

Sloppy	Fast and *inaccurate*
Medium	Trade of between speed and accuracy
Precise	Slow and *accurate*

Time Limit: n seconds for an n-vertex graph ← allows fair comparison



There is no obvious difference between the configurations?