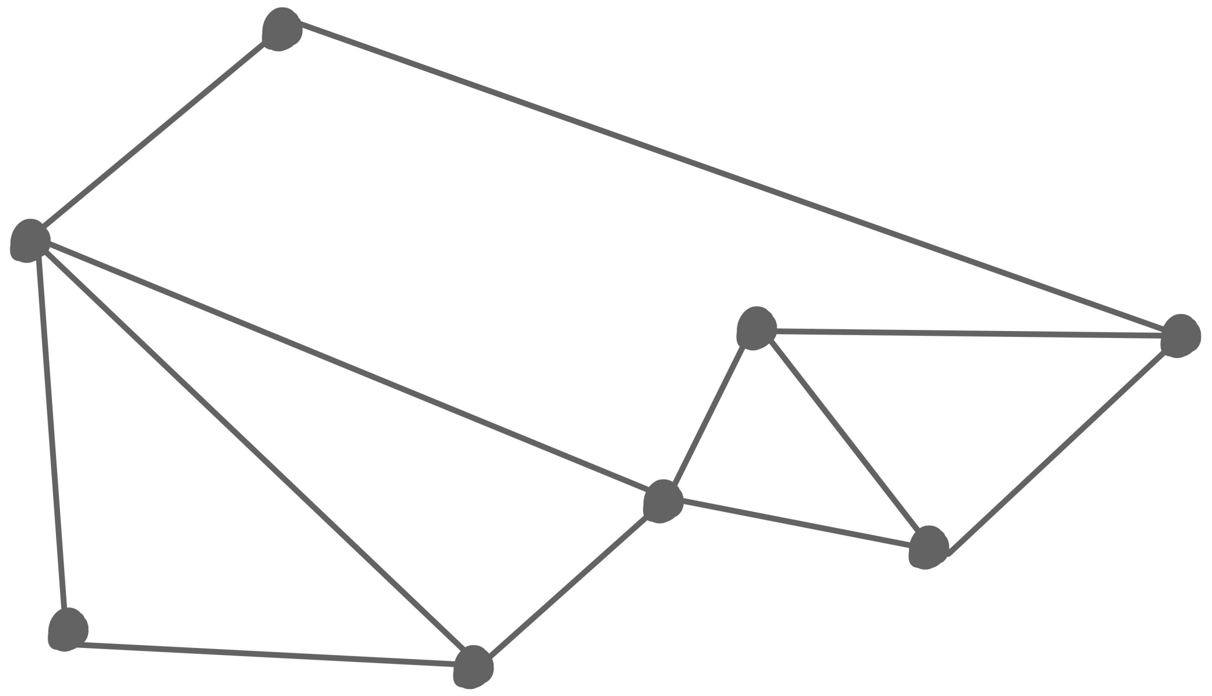
Embedding Planar Graphs into Graphs of Treewidth $O(\log^3 n)$

Hsien-Chih Chang Vincent Cohen-Addad **Jonathan Conroy**

Hung Le Marcin Pilipczuk Michał Pilipczuk

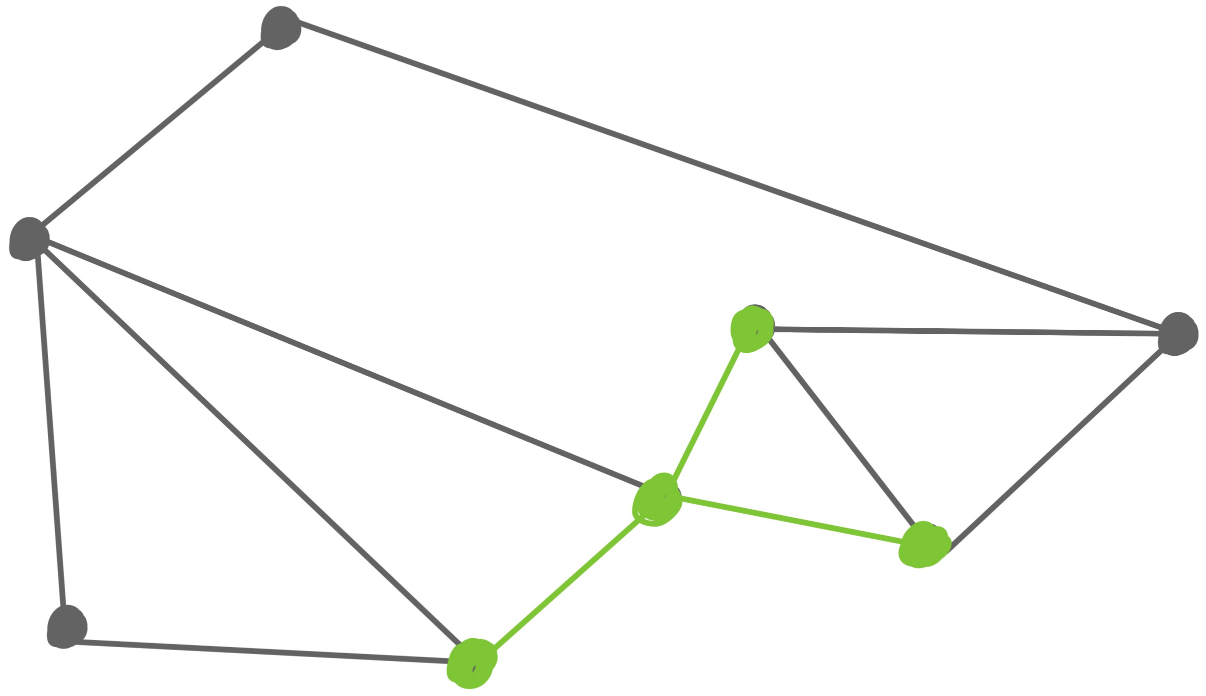
Network design problems



k-MST

Capacitated Vehicle Routing

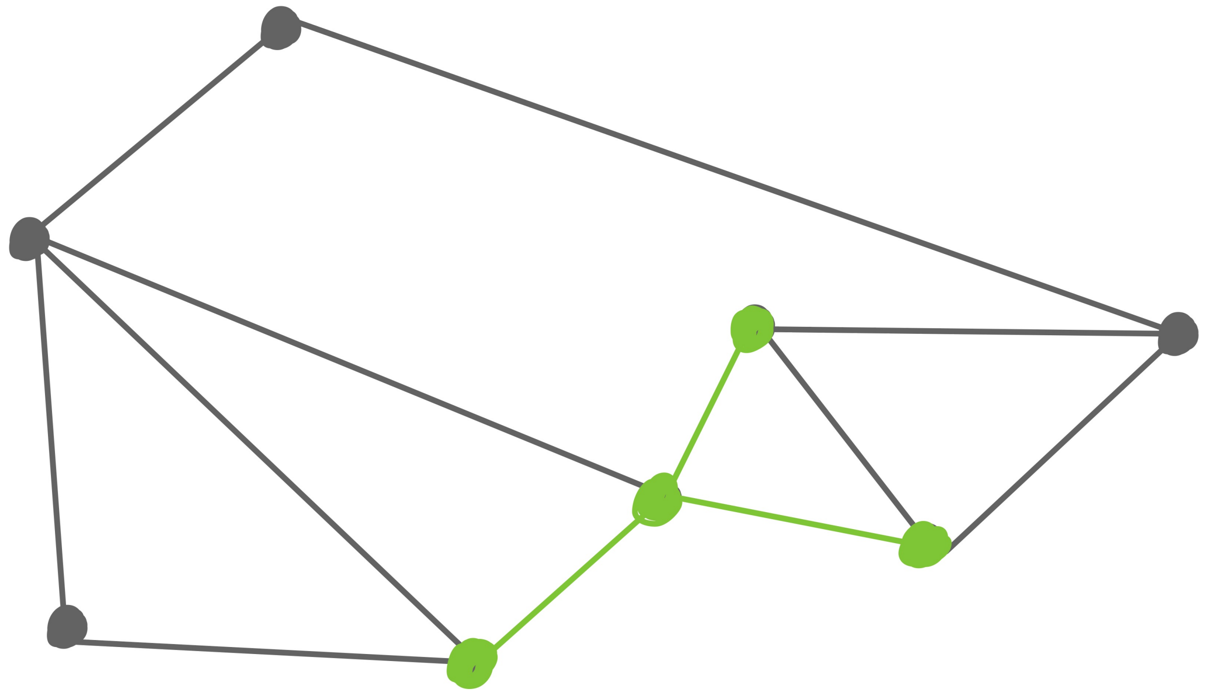
Network design problems



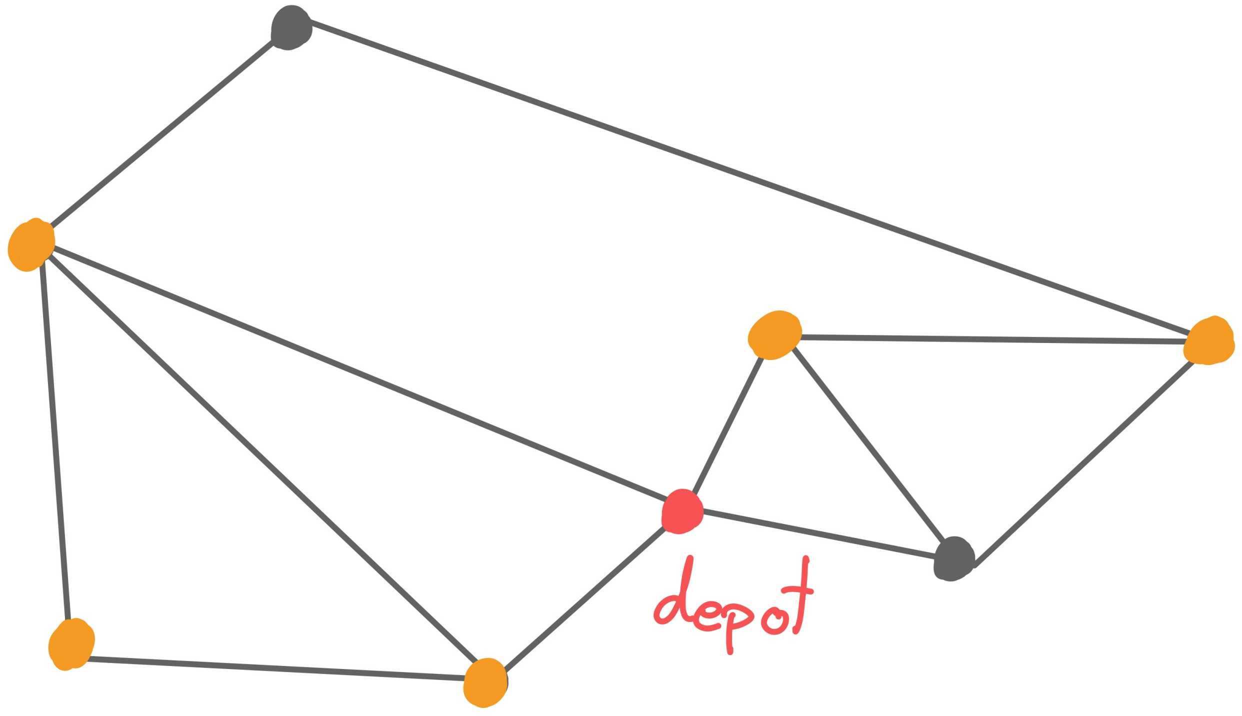
k-MST

Capacitated Vehicle Routing

Network design problems

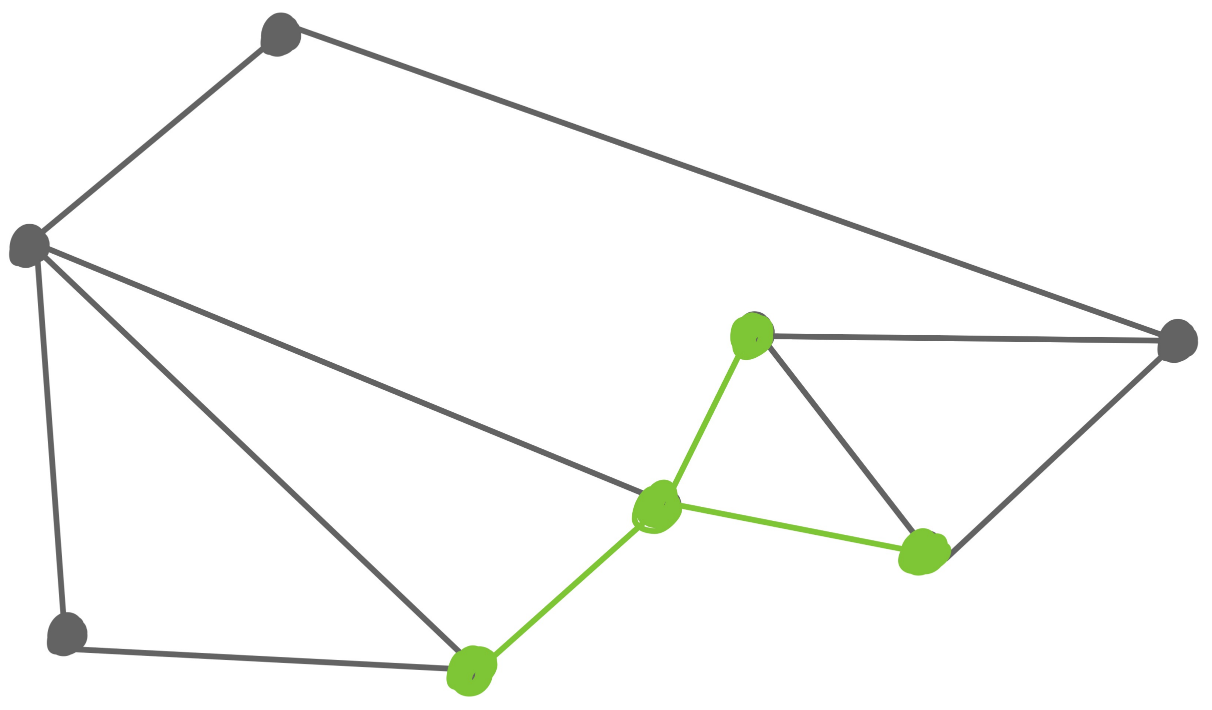


k-MST

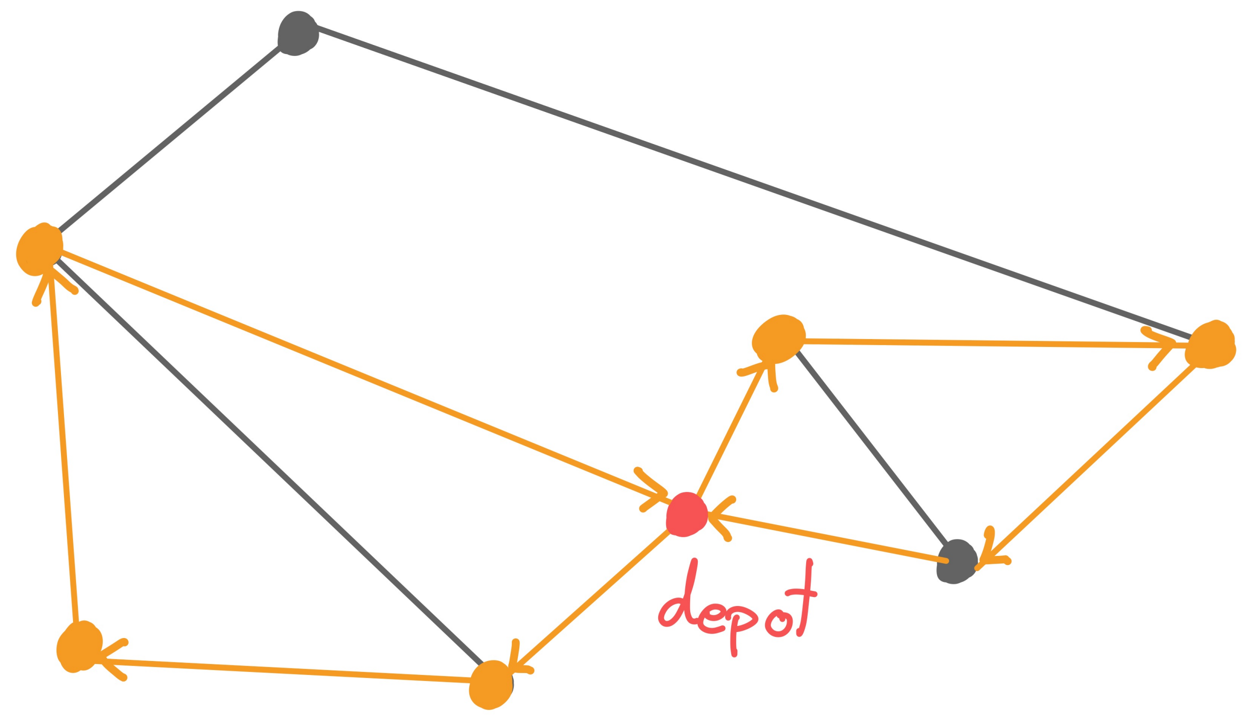


Capacitated Vehicle Routing

Network design problems



k-MST



Capacitated Vehicle Routing

Tree embedding

Thm [Fakcharoenphol, Rao, Talwar '03]. For any n-vertex graph G, one can randomly construct tree H such that, for all vertices u and v,

$$\mathrm{dist}_G(u,v) \leq \mathbb{E}\mathrm{dist}_H(u,v) \leq O(\log n) \cdot \mathrm{dist}_G(u,v)$$

.

$\Rightarrow O(\log n)$-approximation for network design problems

$\Omega(\log n)$ distortion necessary for tree embedding, even for planar graphs

Bounded-treewidth embedding

Thm [CLPP '23]. For any n-vertex planar graph G, one can randomly construct graph H with treewidth $O(\varepsilon^{-1}\log^{13} n)$ such that, for all vertices u and v,

$$\mathrm{dist}_G(u,v) \le \mathbb{E}\mathrm{dist}_H(u,v) \le (1+\varepsilon)\cdot\mathrm{dist}_G(u,v)$$

.

$\Rightarrow$ a *uniform* reason for QPTAS for network design problems in planar graphs

Example: a $1+\varepsilon$ approximation for planar k-MST in time $2^{O(\varepsilon^{-1}\log^{13} n)}$

Bounded-treewidth embedding

Planar graphs embed with expected $1+\varepsilon$ distortion into treewidth:

$\Omega(\log n)$
[CG'04]

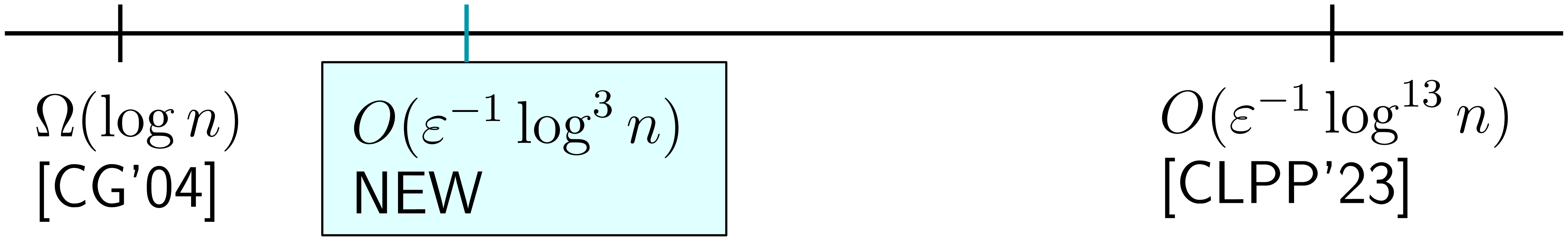
??

$O(\varepsilon^{-1}\log^{13} n)$
[CLPP'23]

Conjecture. Planar graphs embed into graphs of $O(\varepsilon^{-1}\log n)$ treewidth with expected distortion $1+\varepsilon$

Bounded-treewidth embedding

Planar graphs embed with expected $1+\varepsilon$ distortion into treewidth:

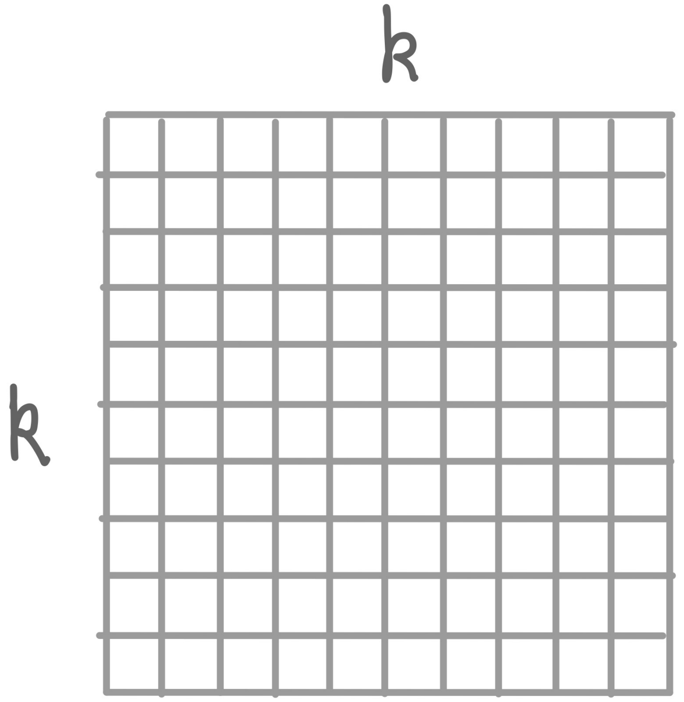


Conjecture. Planar graphs embed into graphs of $O(\varepsilon^{-1}\log n)$ treewidth with expected distortion $1+\varepsilon$

Outline

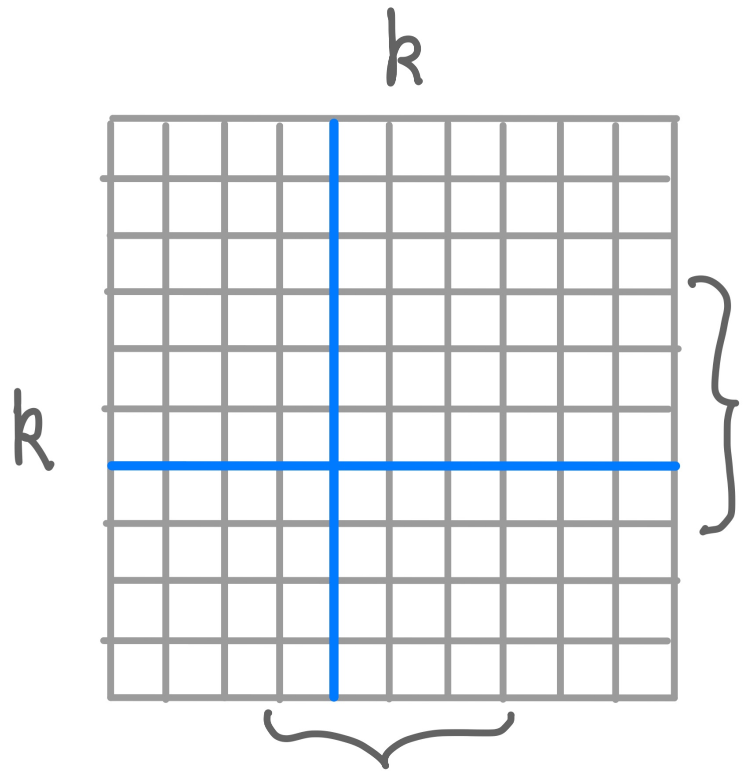
1. Intro/Motivation
2. Review of [CLPP'23] idea
3. Our improvements

[CLPP'23] Grid example



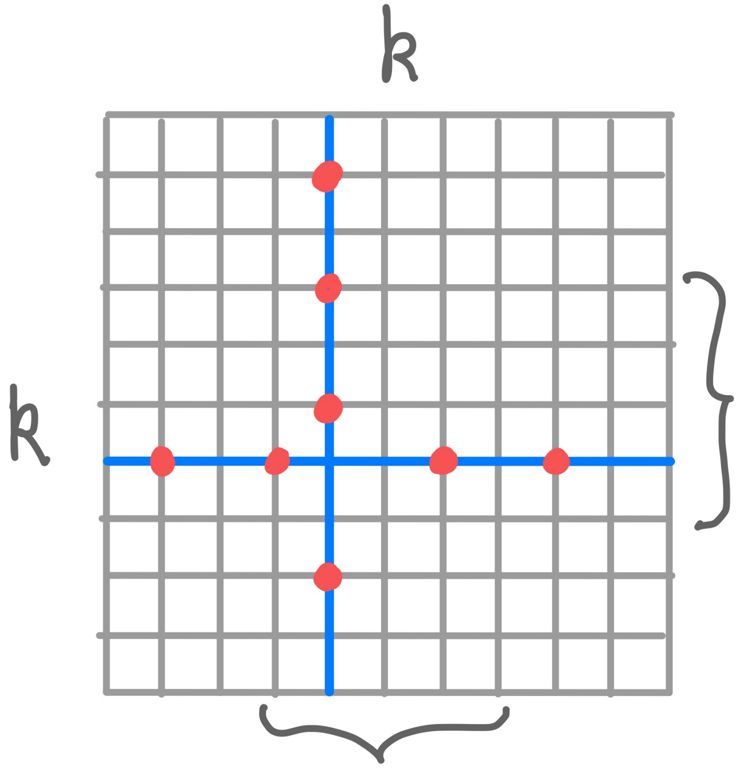
1. Pick a random row and column from the middle ones

[CLPP'23] Grid example



1. Pick a random row and column from the middle ones

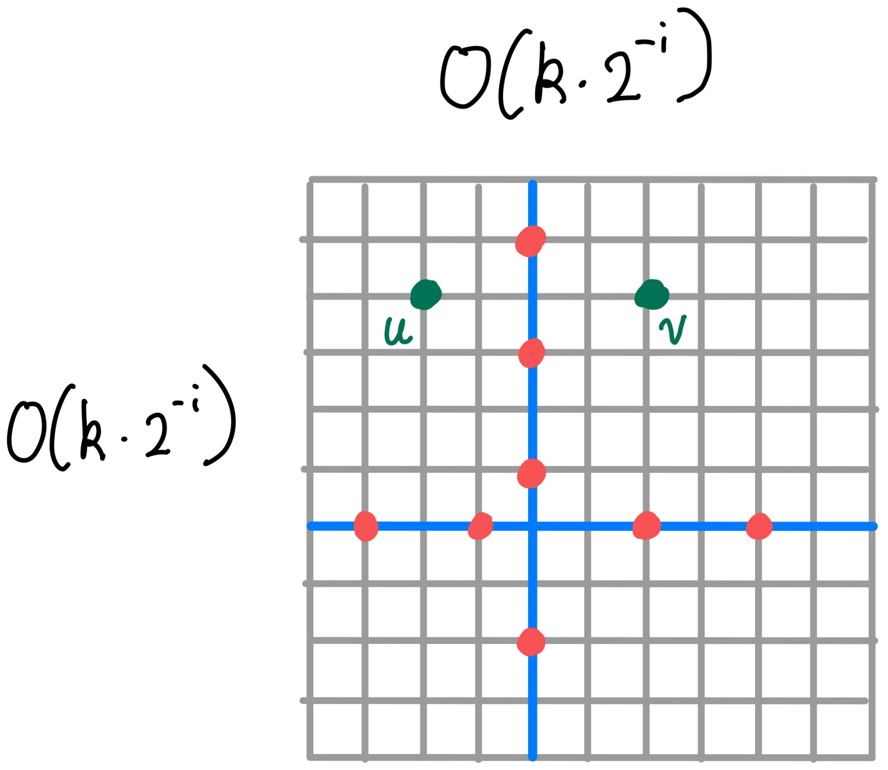
[CLPP'23] Grid example



1. Pick a random row and column from the middle ones

2. Select $O(\varepsilon^{-1} \log n)$ portals. For each portal p, add (to H) an edge from p to every vertex. Recurse on the four quadrants.

[CLPP'23] Grid example, analysis

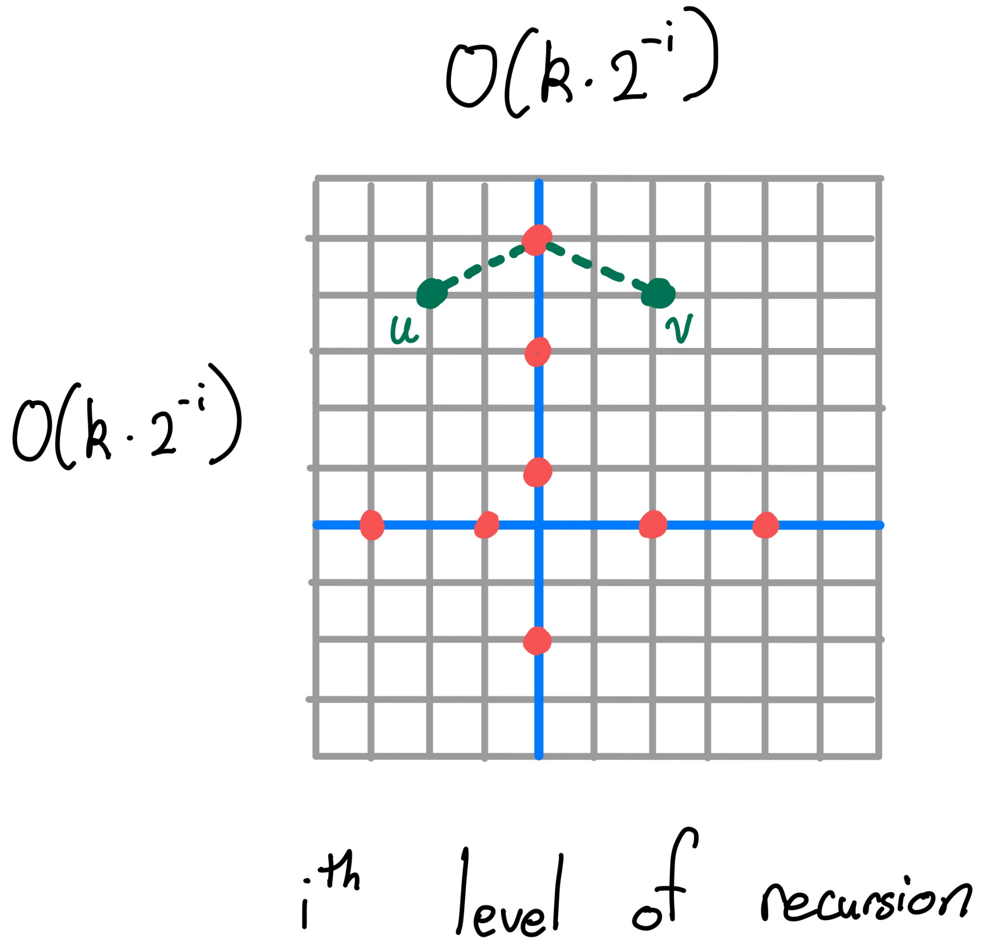


ith level of recursion

Prob that u and v are separated in ith level of recursion:

$$\frac{\mathrm{dist}_G(u,v)}{O(k \cdot 2^{-i})}$$

[CLPP'23] Grid example, analysis



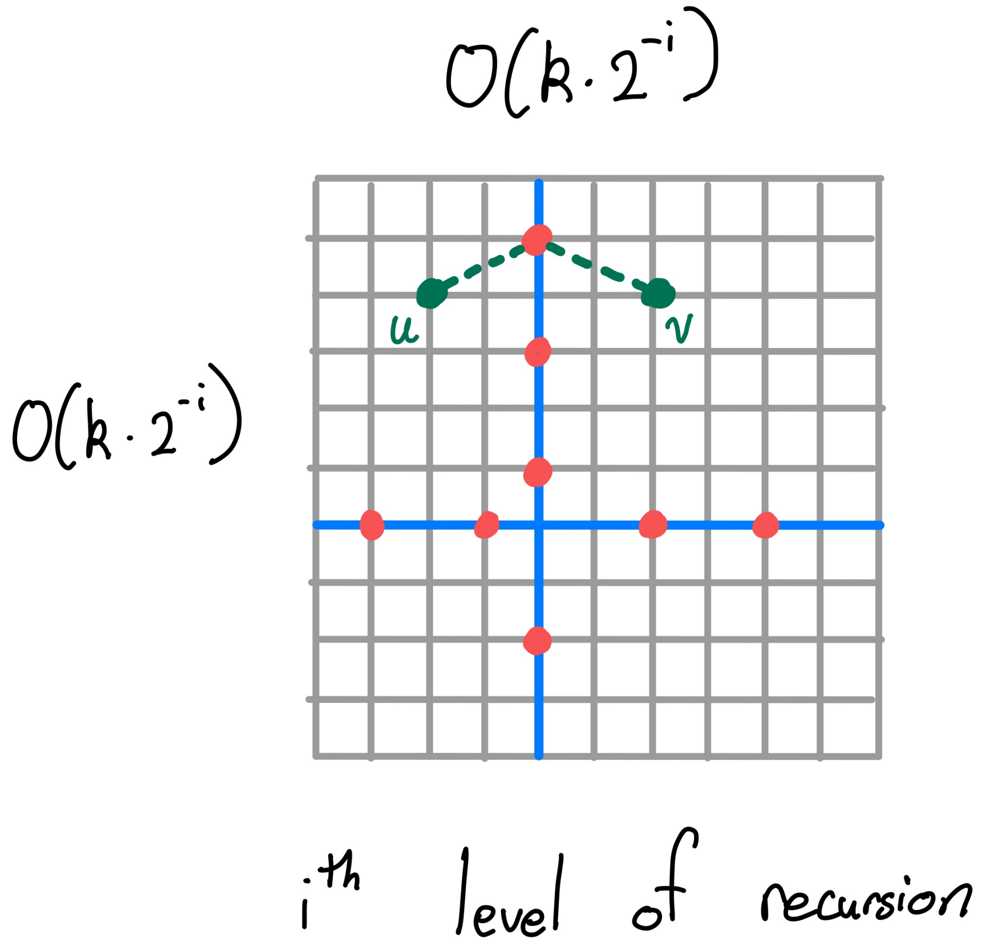
Prob that u and v are separated in ith level of recursion:

$$\frac{\mathrm{dist}_G(u,v)}{O(k\cdot 2^{-i})}$$

If u, v separated at ith level:

$$\mathrm{dist}_H(u,v) \leq \mathrm{dist}_G(u,v) + \frac{\varepsilon}{\log n} \cdot O(k \cdot 2^{-i})$$

[CLPP'23] Grid example, analysis



Prob that u and v are separated in ith level of recursion: $\frac{\mathrm{dist}_G(u,v)}{O(k\cdot 2^{-i})}$

If u, v separated at ith level:

$$\mathrm{dist}_H(u,v) \leq \mathrm{dist}_G(u,v) + \frac{\varepsilon}{\log n} \cdot O(k \cdot 2^{-i})$$

$$\mathbb{E}\mathrm{dist}_H(u,v) \leq \mathrm{dist}_G(u,v) + \sum_i^{\log n} \left(\frac{\mathrm{dist}_G(u,v)}{O(k \cdot 2^{-i})} \cdot \frac{\varepsilon}{\log n} \cdot O(k \cdot 2^{-i}) \right)$$
$$= (1+\varepsilon)\mathrm{dist}_G(u,v)$$

[CLPP'23] Beyond grid graphs

1. Pick a random "column"

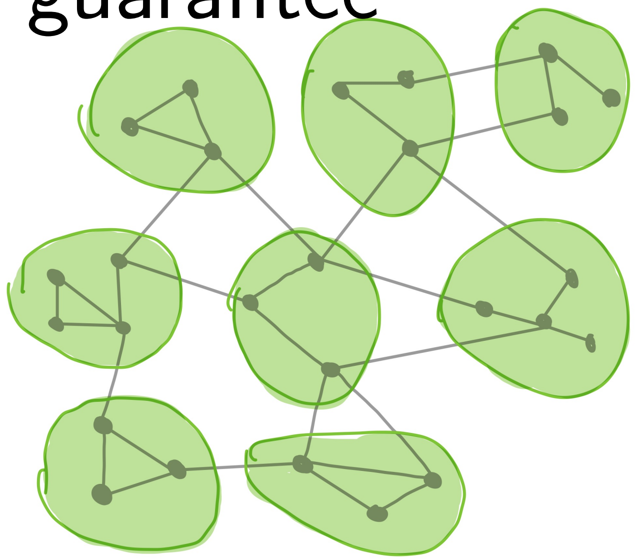
2. Add edges from portals, and recurse

Q: How can we find lots of disjoint "columns" to choose from?

[CLPP'23] Beyond grid graphs

Q: How can we find lots of (almost-)disjoint columns to choose from?

1: Low-diameter decomposition with hop guarantee

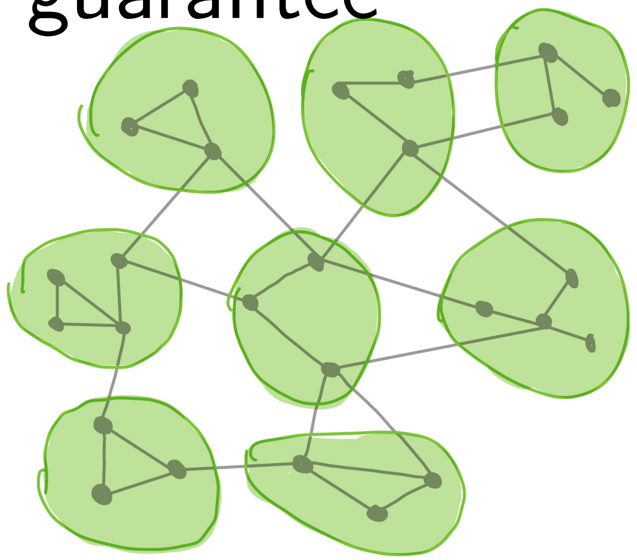


- Chop G into clusters of diameter Δ
- u, v in different clusters w.p. $\leq \beta \cdot \frac{\mathrm{dist}_G(u,v)}{\Delta}$
- After contracting every cluster, graph has hop-diameter h

[CLPP'23] Beyond grid graphs

Q: How can we find lots of (almost-)disjoint columns to choose from?

1: Low-diameter decomposition with hop guarantee

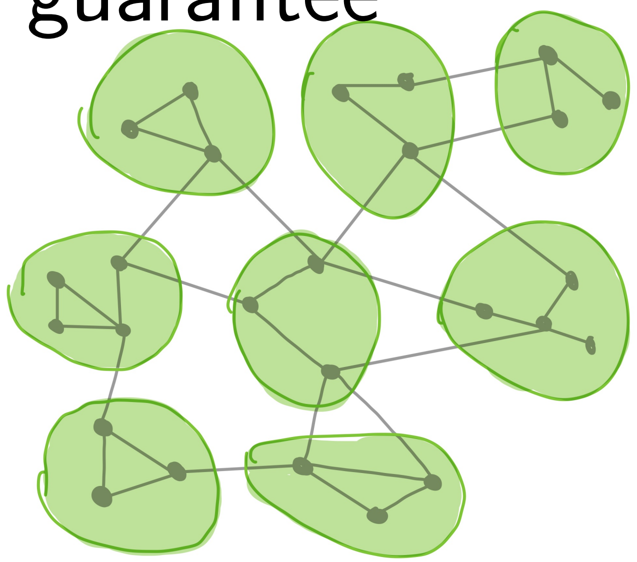


- Chop G into clusters of diameter Δ
- u, v in different clusters w.p. $\leq \beta \cdot \frac{\mathrm{dist}_G(u,v)}{\Delta}$
- After contracting every cluster, graph has hop-diameter h $\beta = O(\log n)$, $h = O(\log n)$

[CLPP'23] Beyond grid graphs

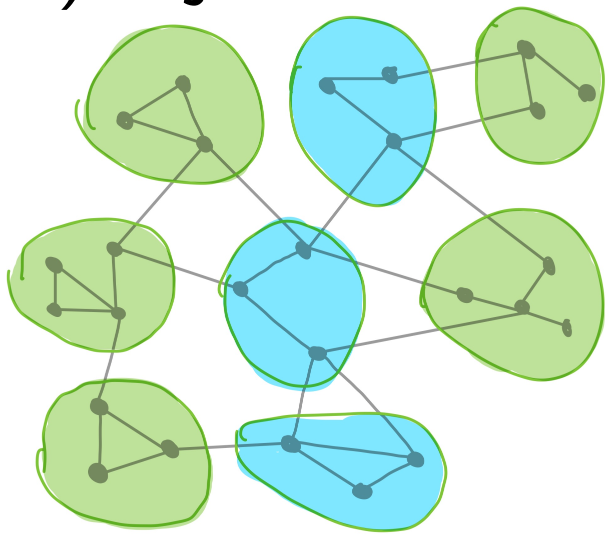
Q: How can we find lots of (almost-)disjoint columns to choose from?

1: Low-diameter decomposition with hop guarantee



- Chop G into clusters of diameter Δ
- u, v in different clusters w.p. $\leq \beta \cdot \frac{\mathrm{dist}_G(u,v)}{\Delta}$
- After contracting every cluster, graph has hop-diameter h $\beta = O(\log n)$, $h = O(\log n)$

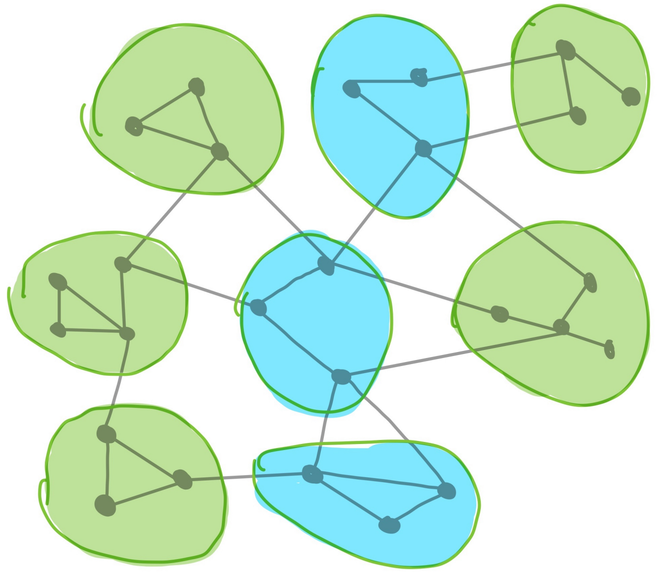
2: Use decomposition to find (almost-)disjoint columns



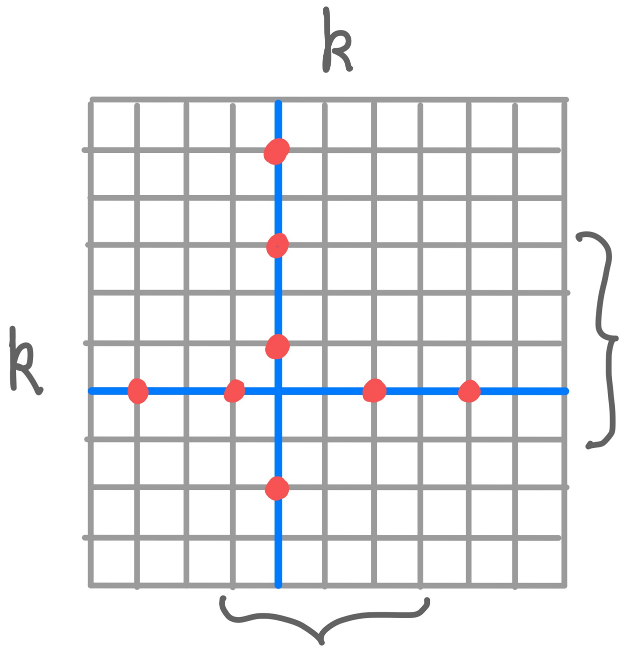
Proof exploits *diameter-treewidth* property of planar graphs

[CLPP'23] Summary

1: Low-diameter decomposition with hop guarantee



2: Use decomposition to find almost-disjoint "columns"



3. Add edges from each portals on column, and then recurse

Outline

1. Intro/Motivation

2. Review of [CLPP'23] idea

3. Our improvements

Our improvements

1: Low-diameter decomposition
with hop guarantee

2: Use decomposition to find
almost-disjoint “columns”

3. Add edges from each portals
on column, and then recurse

Our improvements

1: Low-diameter decomposition with hop guarantee

$\beta = O(1)$ and $h = O(1)$
Idea: Adapt *buffered cop decomposition* of [CCLMST'24]

2: Use decomposition to find almost-disjoint "columns"

3. Add edges from each portals on column, and then recurse

Our improvements

1: Low-diameter decomposition with hop guarantee

$\beta = O(1)$ and $h = O(1)$
Idea: Adapt *buffered cop decomposition* of [CCLMST'24]

2: Use decomposition to find almost-disjoint "columns"

Improved analysis of *contraction sequence* from [CLPP'23]

3. Add edges from each portals on column, and then recurse

Our improvements

1: Low-diameter decomposition with hop guarantee

$\beta = O(1)$ and $h = O(1)$
Idea: Adapt *buffered cop decomposition* of [CCLMST'24]

2: Use decomposition to find almost-disjoint "columns"

Improved analysis of *contraction sequence* from [CLPP'23]

3. Add edges from each portals on column, and then recurse

Use single hierarchy of low-diameter decomposition, and exploit treewidth vs. treedepth

Our improvements

1: Low-diameter decomposition with hop guarantee

Contributes $O(1)$ treewidth

$\beta = O(1)$ and $h = O(1)$
Idea: Adapt *buffered cop decomposition* of [CCLMST'24]

2: Use decomposition to find almost-disjoint "columns"

Contributes $O(\log^2 n)$ treewidth

Improved analysis of *contraction sequence* from [CLPP'23]

3. Add edges from each portals on column, and then recurse

Contributes $O(\log n)$ treewidth

Use single hierarchy of low-diameter decomposition, and exploit treewidth vs. treedepth

Conclusion

We show planar graphs embed into $O(\varepsilon^{-1} \log^3 n)$ treewidth

Still open... can we embed into $O(\varepsilon^{-1} \log n)$ treewidth?

Conclusion

We show planar graphs embed into $O(\varepsilon^{-1} \log^3 n)$ treewidth

Still open... can we embed into $O(\varepsilon^{-1} \log n)$ treewidth?

Thank you!