Hop Spanners for Geometric Intersection Graphs

Jonathan Conroy, Csaba Tóth

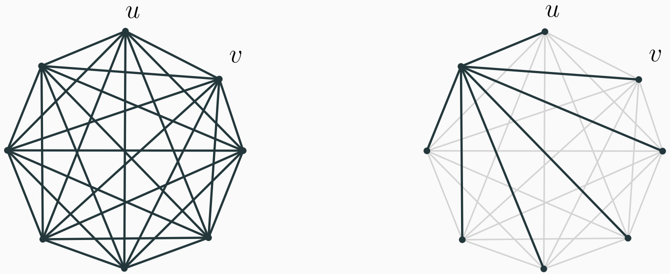
Tufts University, California State University Northridge

Introduction

Spanners

k-spanner: a subgraph such that $dist(u, v)$ in subgraph is at most k times the distance bewteen u and v in the original graph

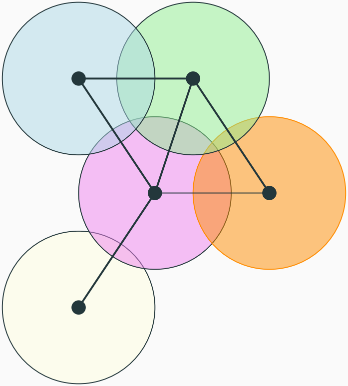
k-hop spanner: a k-spanner for a graph with edges of weight 1

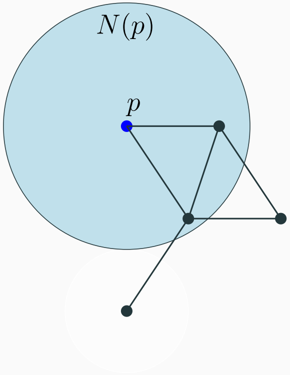


A sparse 2-hop spanner for the complete graph

Geometric Intersection Graphs

Unit disk graphs, motivated by wireless communication



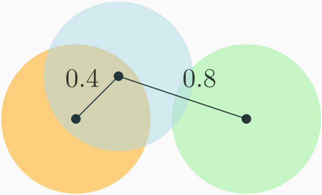


Edge iff corresponding disks of radius **0.5** intersect

Equivalently: edge iff vertices are within distance 1

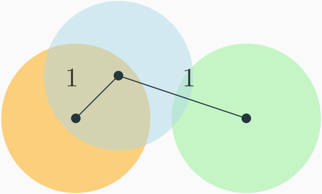
Prior Work

Weighted Spanner for UDG



Many results in the literature (see paper)

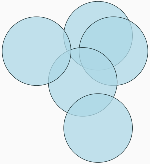
Hop Spanner for UDG



- 2-hop, with $O(n \log n)$ edges [Dumitrescu et. al., 2021]
- 3-hop, with $O(n)$ edges [Dumitrescu et. al., 2021]

Our Results

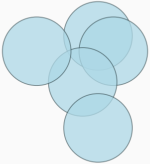
Stretch	Number of Edges	Intersection Graph of
2	$O(n)$	unit disks



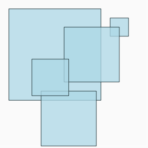
Unit Disks

Our Results

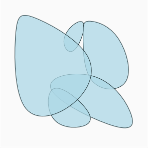
Stretch	Number of Edges	Intersection Graph of
2	$O(n)$	unit disks
2	$O(n \log n)$	axis-aligned squares
3	$O(n \log n)$	fat convex bodies
3	$O(n \log^2 n)$	axis-aligned rectangles



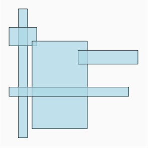
Unit Disks



Squares



Fat Convex Bodies

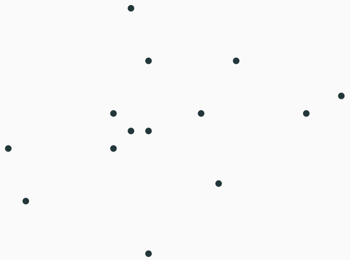


Rectangles

2-Hop Spanners of Size $O(n)$ for Unit Disk Graphs

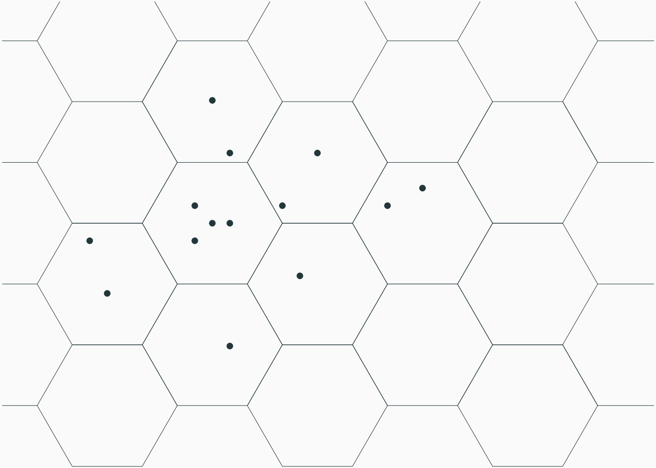
Reduction to Bipartite Setting

Reduction: Find spanner for general UDGs $\rightarrow$ Find spanner for UDGs in bipartite setting [Biniaz, 2020]



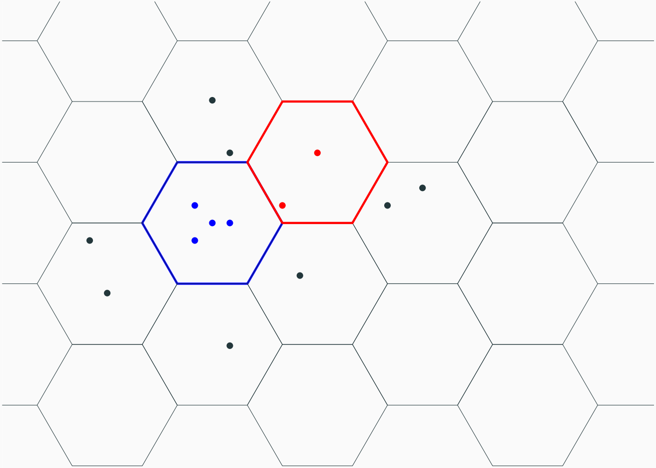
Reduction to Bipartite Setting

Reduction: Find spanner for general UDGs $\rightarrow$ Find spanner for UDGs in bipartite setting [Biniaz, 2020]

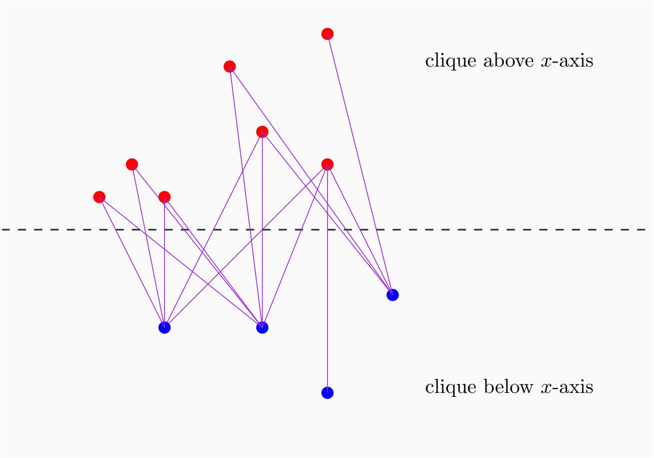


Reduction to Bipartite Setting

Reduction: Find spanner for general UDGs $\rightarrow$ Find spanner for UDGs in bipartite setting [Biniaz, 2020]

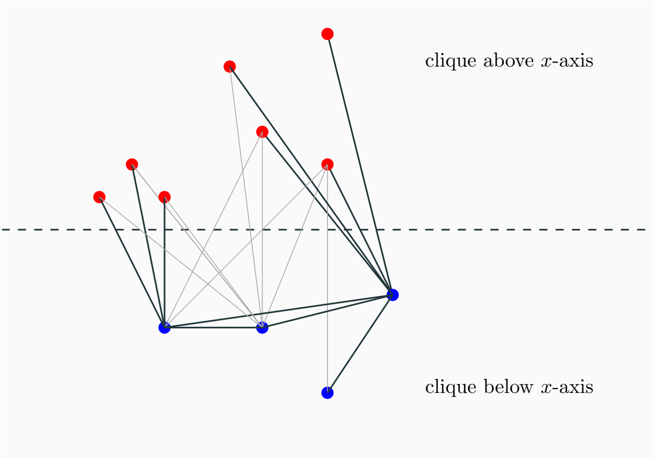


Reduction to Bipartite Setting



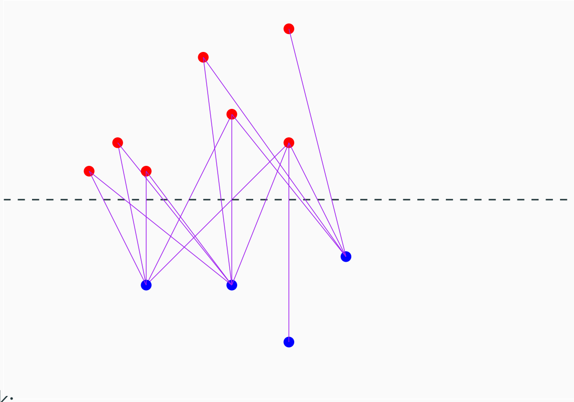
Goal: Construct a 2-hop spanner for bipartite edges

Reduction to Bipartite Setting



Goal: Construct a 2-hop spanner for bipartite edges

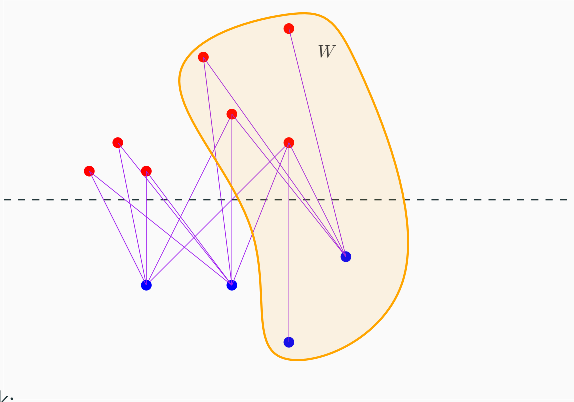
Incremental Construction



Framework:

- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

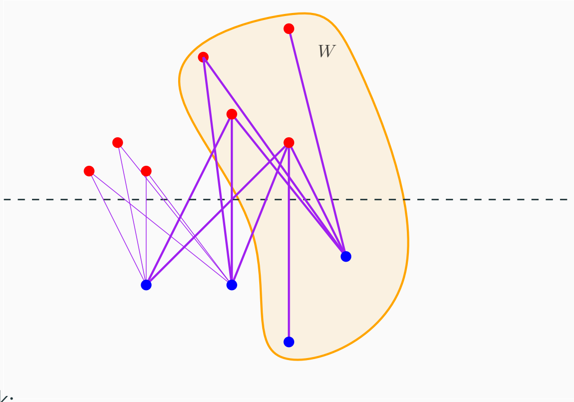
Incremental Construction



Framework:

- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

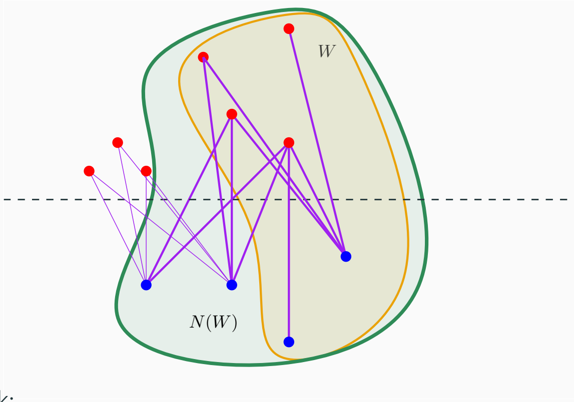
Incremental Construction



Framework:

- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

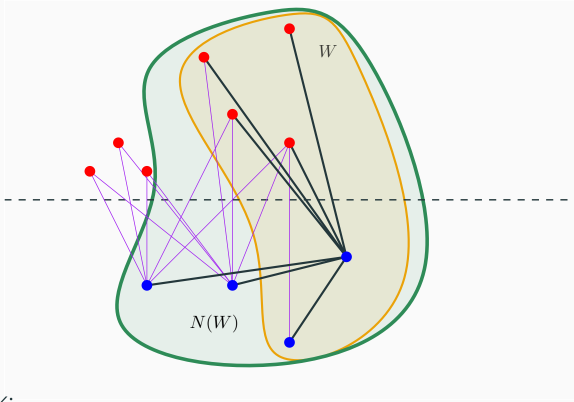
Incremental Construction



Framework:

- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

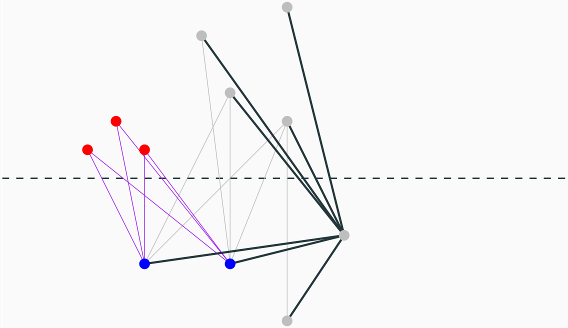
Incremental Construction



Framework:

- Select subset *W* of vertices
- Construct 2-hop spanner for edges involving *W*
- Remove *W*, and recurse on remainder

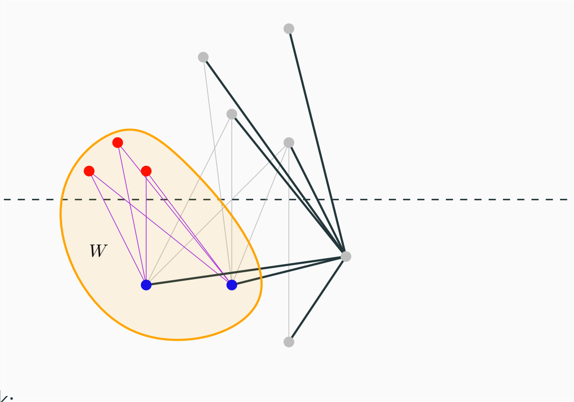
Incremental Construction



Framework:

- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

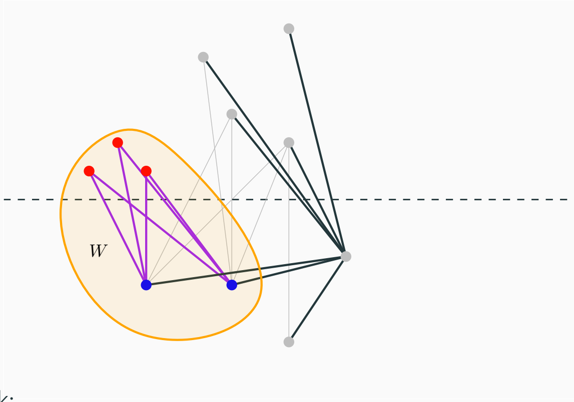
Incremental Construction



Framework:

- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

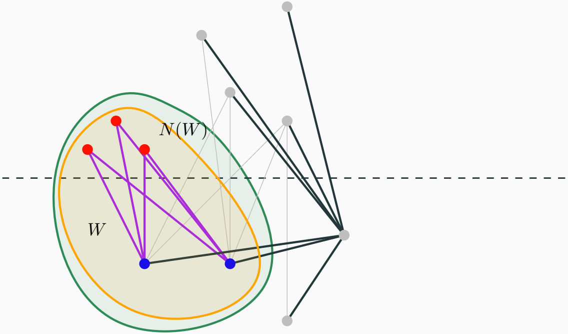
Incremental Construction



Framework:

- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

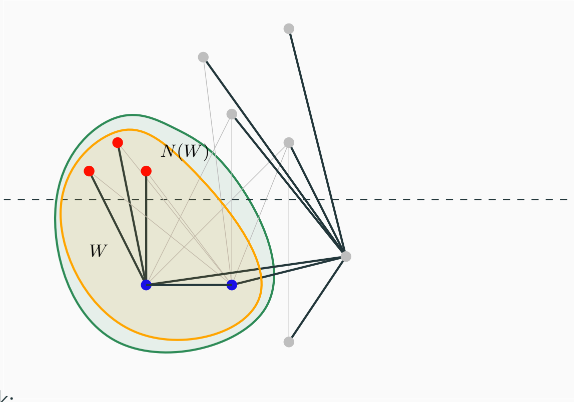
Incremental Construction



Framework:

- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

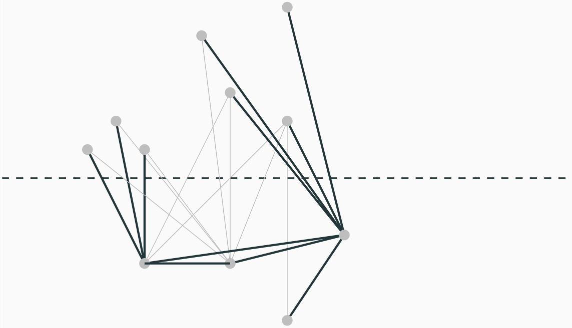
Incremental Construction



Framework:

- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

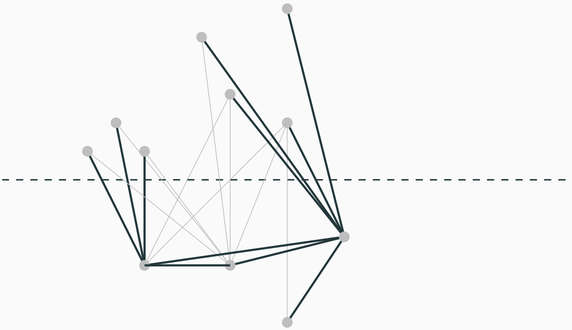
Incremental Construction



Framework:

- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

Incremental Construction



Framework:

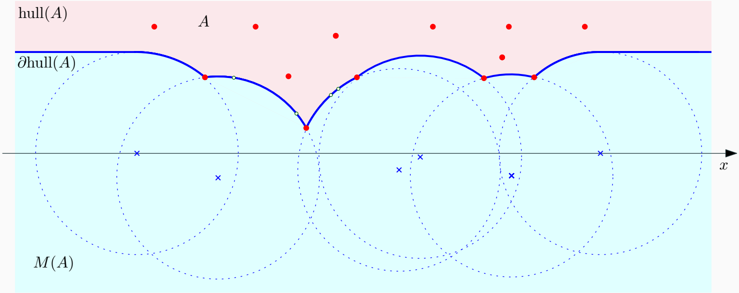
- Select subset W of vertices
- Construct 2-hop spanner for edges involving W
- Remove W, and recurse on remainder

Need: W such that # edges added = $O(|W|)$

Technical Tool: α-Hull

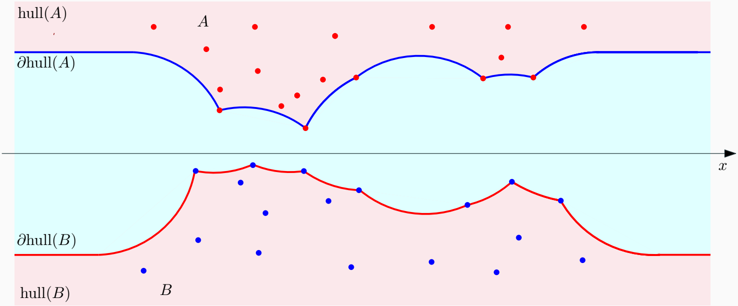
α-shapes [Edelsbrunner et. al., 1983]

∂hull(A): Boundary of union of all unit disks centered below x-axis that do not intersect A [Dumitrescu et. al., 2021]

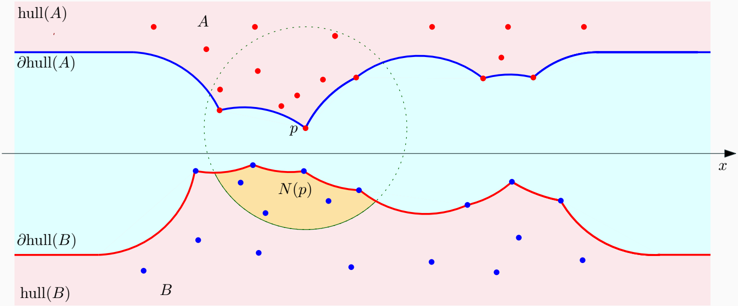


- Similar properties to convex hull
- x-monotone

Incremental Construction: Finding W

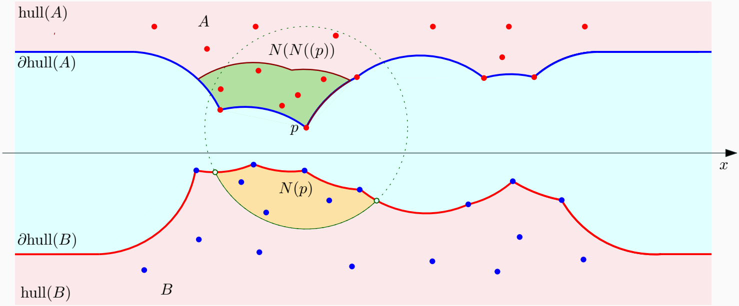


Incremental Construction: Finding W



Remove $W = N(p) \cup p$?

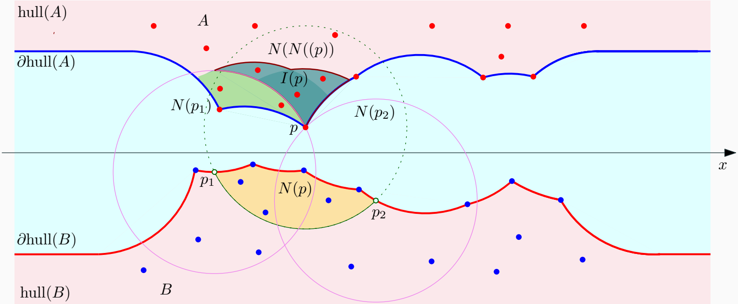
Incremental Construction: Finding W



Remove $W = N(p) \cup p$? ✗

Spanning star connecting $N(p) \cup N(N(p))$ is too large

Incremental Construction: Finding W

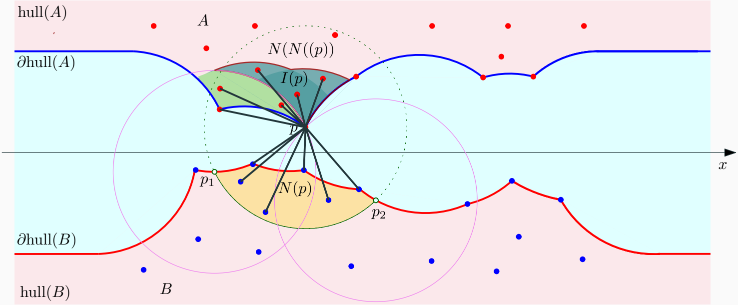


Remove $W = N(p) \cup p$? ✗

Spanning star connecting $N(p) \cup N(N(p))$ is too large

Remove $W = N(p) \cup I(p) \cup p$. ✓

Incremental Construction: Finding W



Remove $W = N(p) \cup p$? ✗

Spanning star connecting $N(p) \cup N(N(p))$ is too large

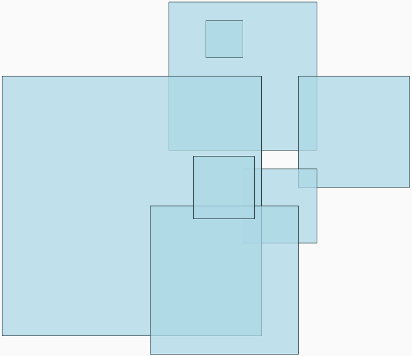
Remove $W = N(p) \cup I(p) \cup p$. ✓

Found: $W = N(p) \cup p \cup I(p)$ such that # edges added = $O(|W|)$.

2-Hop Spanners of Size $O(n \log n)$ for Square Intersection Graphs

Square Intersection Graphs

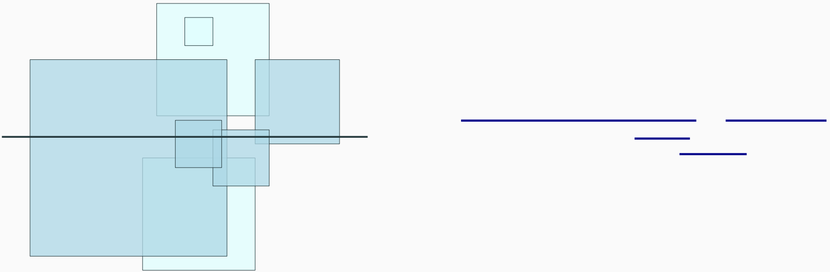
Reduction to bipartite case no longer works



New idea: Divide and conquer, using 1D case as subroutine

Motivation for Divide and Conquer

Squares intersecting common line $\rightarrow$ interval graph



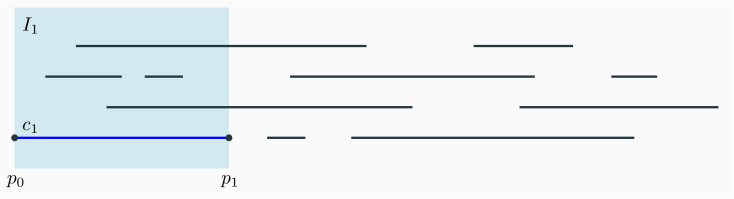
We can construct 2-hop spanners of size $O(n)$ for interval graphs.

1D Case

1D Construction:

- Greedily construct a cover.

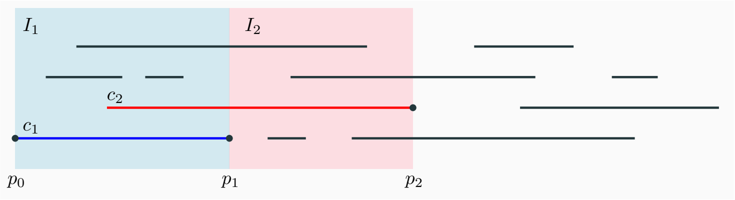
1D Case



1D Construction:

- Greedily construct a cover.

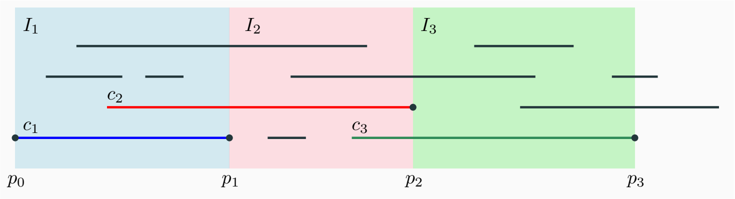
1D Case



1D Construction:

- Greedily construct a cover.

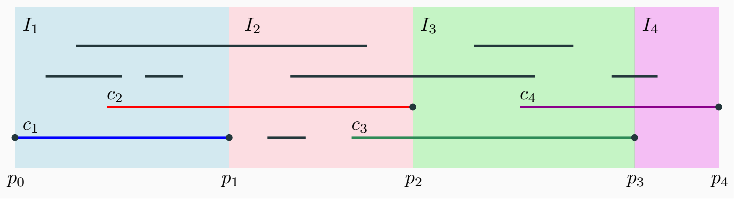
1D Case



1D Construction:

- Greedily construct a cover.

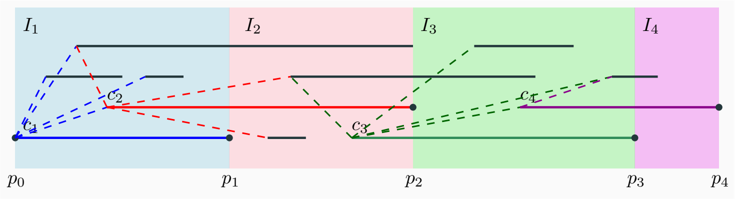
1D Case



1D Construction:

- Greedily construct a cover.

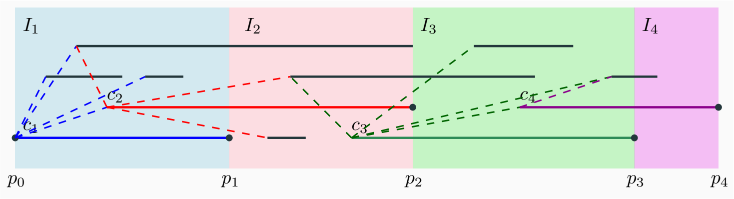
1D Case



1D Construction:

- Greedily construct a cover.
- Add spanning star for each covering segment

1D Case



1D Construction:

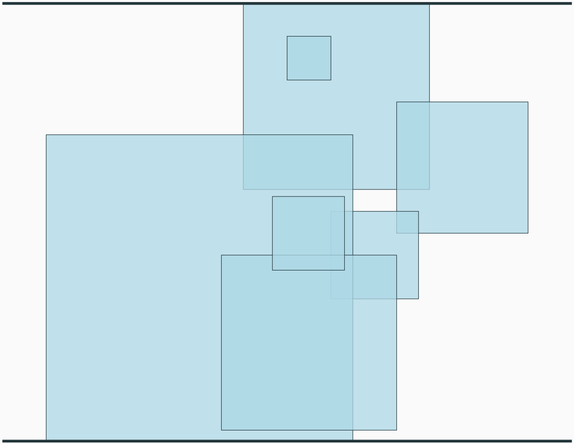
- Greedily construct a cover.
- Add spanning star for each covering segment

Result: 2-hop spanner with $O(n)$ edges for interval graph

Divide and Conquer Framework

2D Construction:

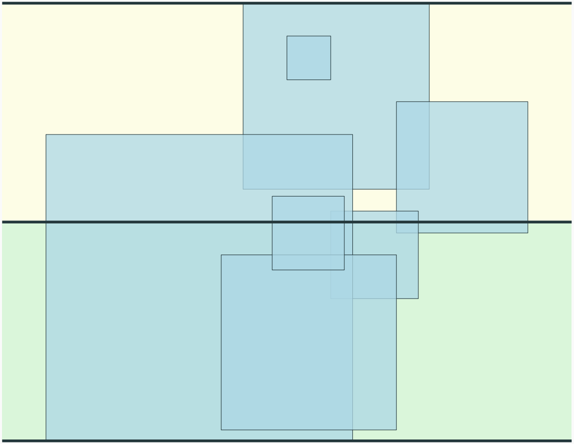
- Split space in half
- Remove squares that go across slab
- Recursively split slab



Divide and Conquer Framework

2D Construction:

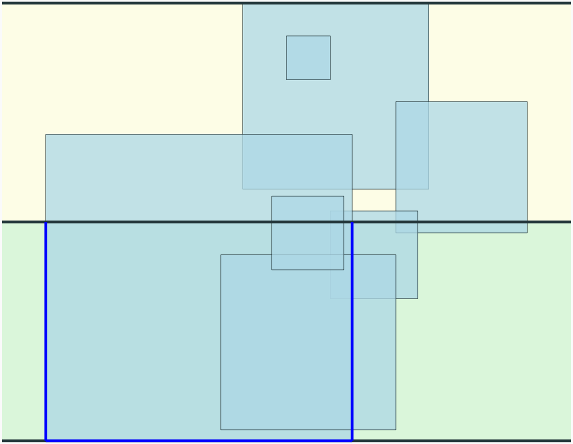
- Split space in half
- Remove squares that go across slab
- Recursively split slab



Divide and Conquer Framework

2D Construction:

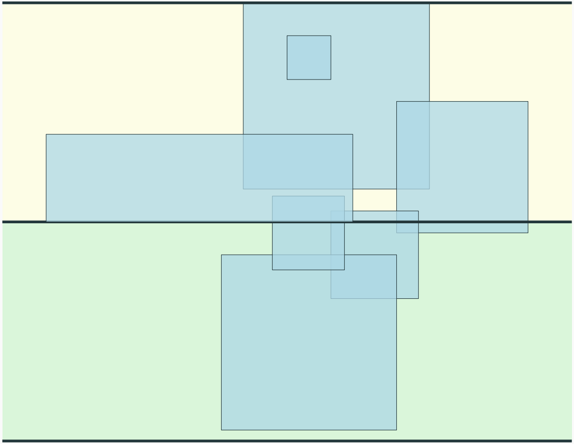
- Split space in half
- Remove squares that go across slab
- Recursively split slab



Divide and Conquer Framework

2D Construction:

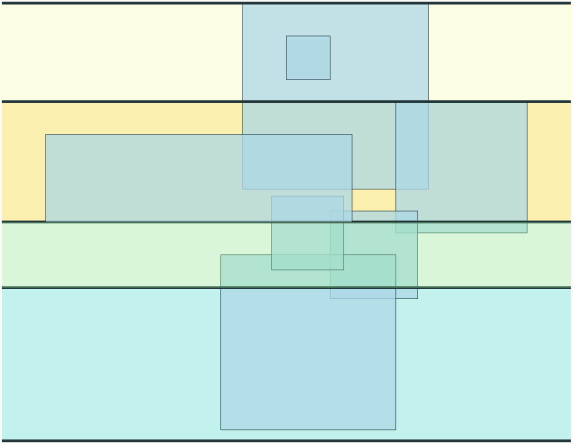
- Split space in half
- Remove squares that go across slab
- Recursively split slab



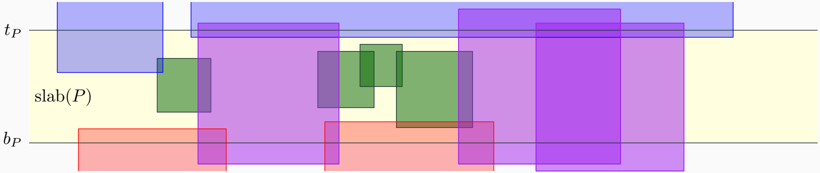
Divide and Conquer Framework

2D Construction:

- Split space in half
- Remove squares that go across slab
- Recursively split slab



Goal: Eliminate Across squares from recursive calls

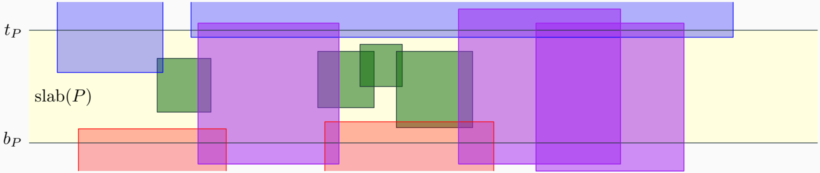


Need to deal with edges between:

- Across-Across
- Across-Bottom
- Across-Top
- Across-Inside

Removing Squares From a Slab

Goal: Eliminate Across squares from recursive calls

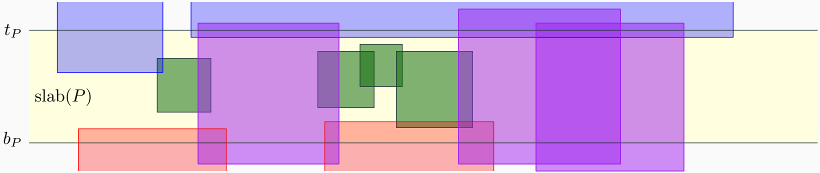


Need to deal with edges between:

- Across-Across: Interval graph
- Across-Bottom: Interval graph
- Across-Top: Interval graph
- Across-Inside: Similar to interval graph (see paper)

Removing Squares From a Slab

Goal: Eliminate Across squares from recursive calls



Need to deal with edges between:

- Across-Across: Interval graph
- Across-Bottom: Interval graph
- Across-Top: Interval graph
- Across-Inside: Similar to interval graph (see paper)

Result: 2-hop spanner of size O(# squares intersecting slab)

Each square is involved in $O(\log n)$ slabs $\implies$ total size $O(n \log n)$

Open Problems

Are there 2-hop spanners of size $O(n \log n)$ for disks of arbitrary radii?

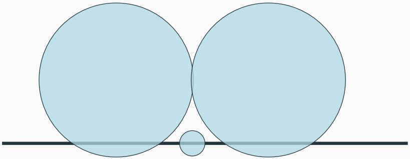


Figure: Arbitrary disks do not reduce easily to 1D

We construct 3-hop spanners of size $O(n \log n)$ for fat convex objects. Is $O(n)$ possible?