(Distance Approximating Minor)

Building a DAM in Planar and Minor-Free Graphs

Hsien-Chih Chang
Dartmouth College

Jonathan Conroy
Dartmouth College

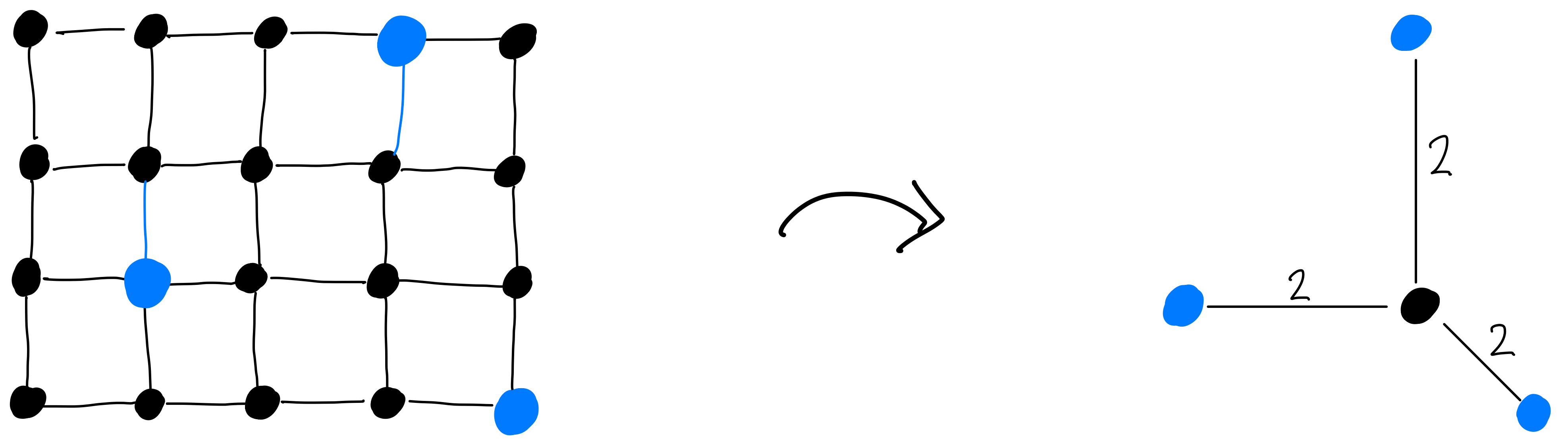
Part 1: What is a DAM?

Distance Approximating Minor (DAM)

Given a graph G and set of terminals $T \subseteq V(G)$.

a *c-DAM* is a minor H of G with $T \subseteq V(H)$, such that for every $u, v \in T$:

$$d_G(u,v) \leq d_H(u,v) \leq c \cdot d_G(u,v)$$



Goal: minimize the distortion c and the size of H

↳ #vertices in H

Prior Work

DAM is particularly useful for planar and minor-free graphs

Steiner Point Removal [Gup01]

Distortion	Size	Graph class	
$O(1)$	$\|T\|$	Minor-free	[C**C**LMST24]
1	$O(\|T\|^4)$	General	[KNZ14]
1	$\Omega(\|T\|^2)$	Grid graph	[KNZ14]
$1+\varepsilon$	$\tilde{O}(\|T\|^2)$	Minor-free	[CGH16]
$1+\varepsilon$	$\tilde{O}(\|T\|)$-size *emulator*	Planar	[CKT22]

Our Result

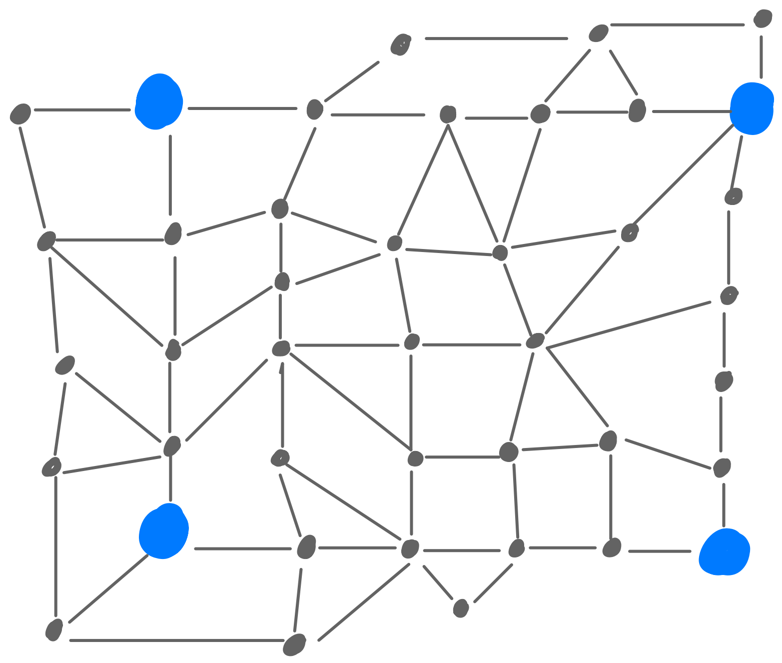
Thm. Every planar or minor-free graph G admits a $(1+\varepsilon)$-DAM of size $|T| \cdot \mathrm{poly}(\log |T|, \log \Phi) \approx \tilde{O}(|T|)$, where Φ is the aspect ratio of G

Our proof applies to any graph class with shortest-path separators

Part 2: How to build a DAM

“Path Overlaying” Algorithm

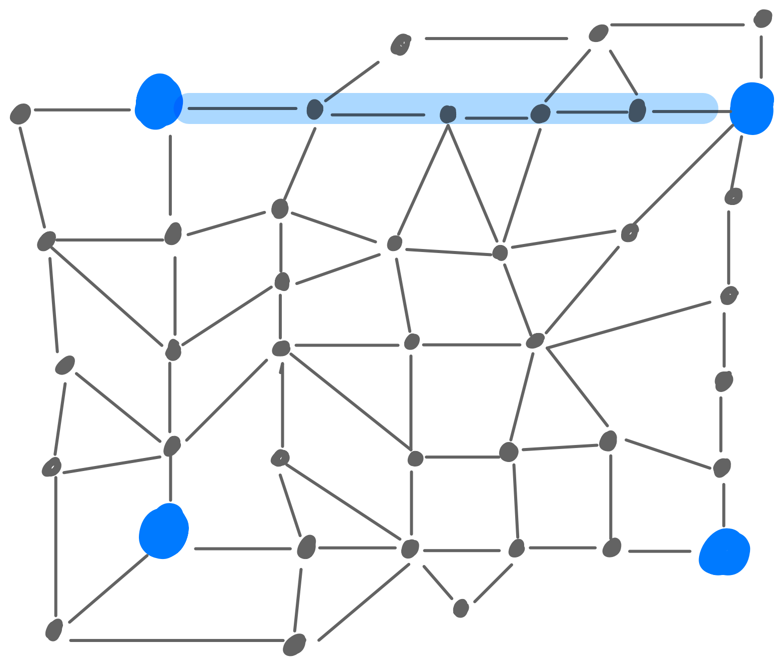
Thm. [KNZ14] Every graph G admits a 1-DAM with size $|T|^4$



1. Draw a shortest path in G between every pair of terminals
2. Contract away degree-2 vertices

"Path Overlaying" Algorithm

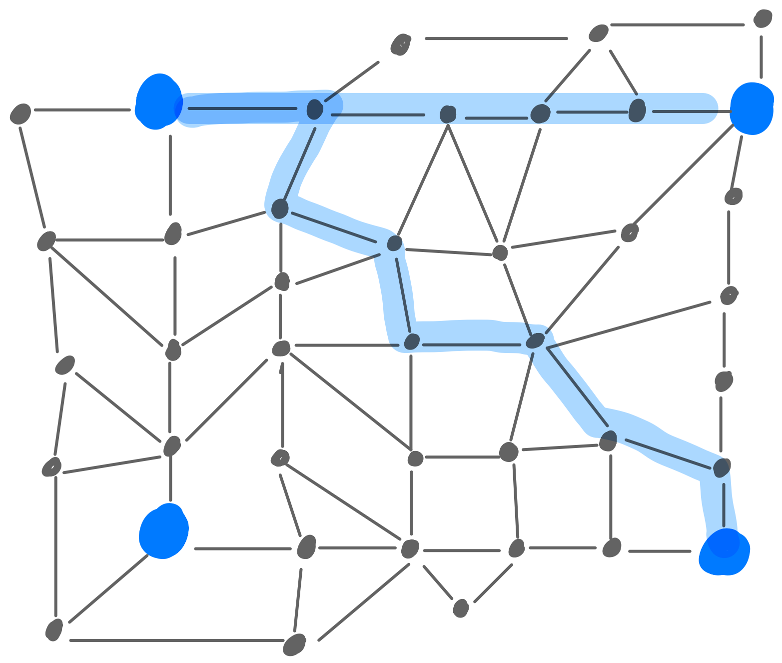
Thm. [KNZ14] Every graph G admits a 1-DAM with size $|T|^4$



1. Draw a shortest path in G between every pair of terminals
2. Contract away degree-2 vertices

“Path Overlaying” Algorithm

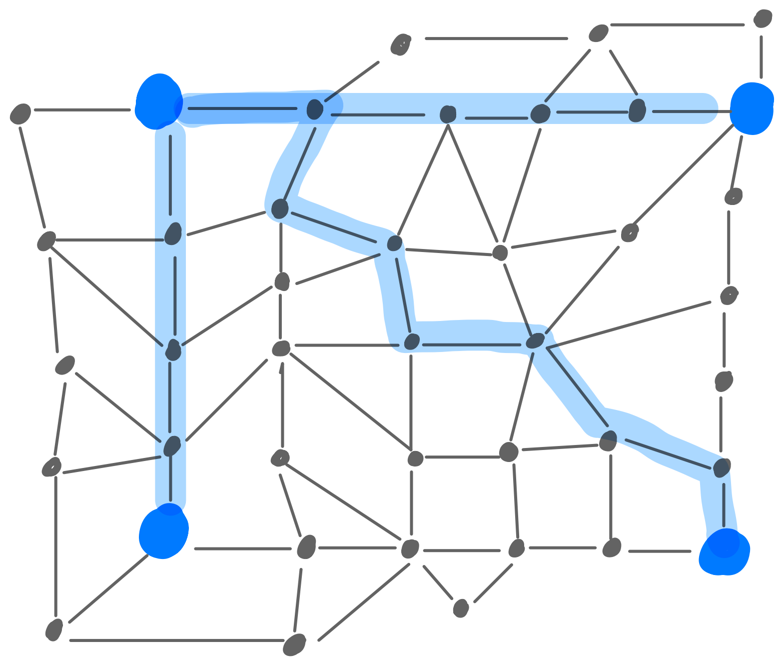
Thm. [KNZ14] Every graph G admits a 1-DAM with size $|T|^4$



1. Draw a shortest path in G between every pair of terminals
2. Contract away degree-2 vertices

“Path Overlaying” Algorithm

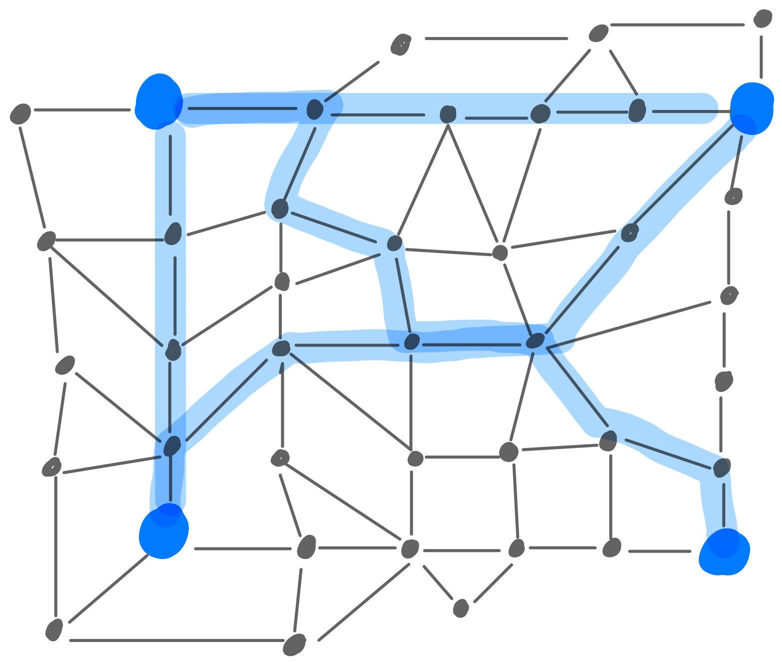
Thm. [KNZ14] Every graph G admits a 1-DAM with size $|T|^4$



1. Draw a shortest path in G between every pair of terminals
2. Contract away degree-2 vertices

“Path Overlaying” Algorithm

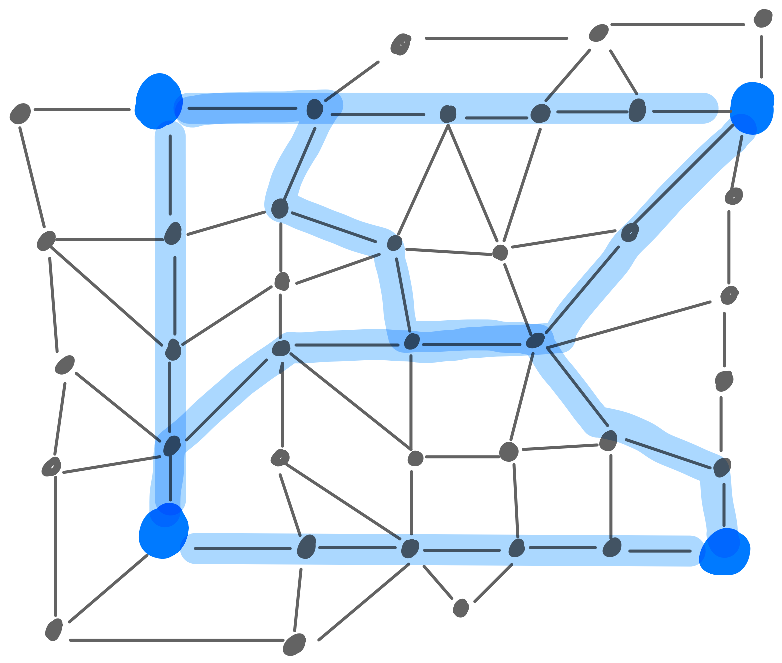
Thm. [KNZ14] Every graph G admits a 1-DAM with size $|T|^4$



1. Draw a shortest path in G between every pair of terminals
2. Contract away degree-2 vertices

“Path Overlaying” Algorithm

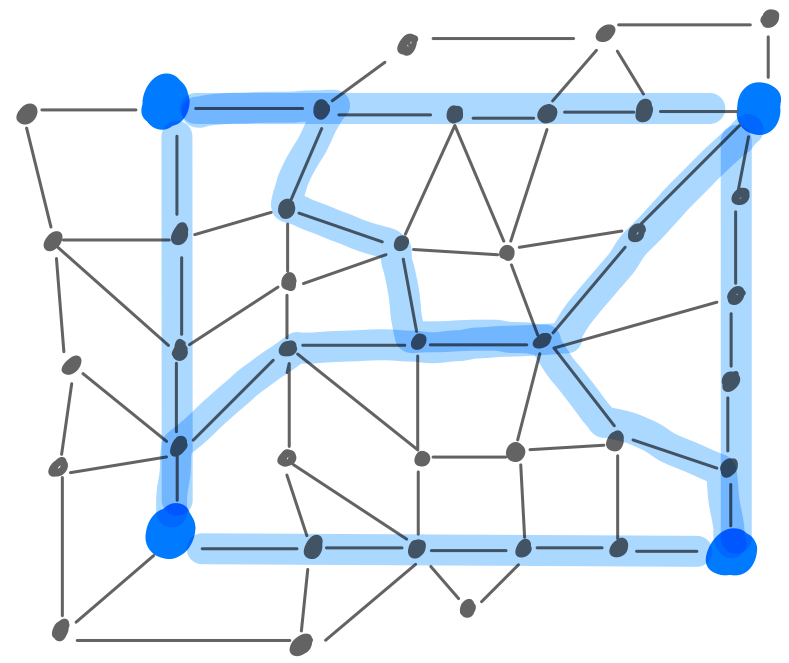
Thm. [KNZ14] Every graph G admits a 1-DAM with size $|T|^4$



1. Draw a shortest path in G between every pair of terminals
2. Contract away degree-2 vertices

“Path Overlaying” Algorithm

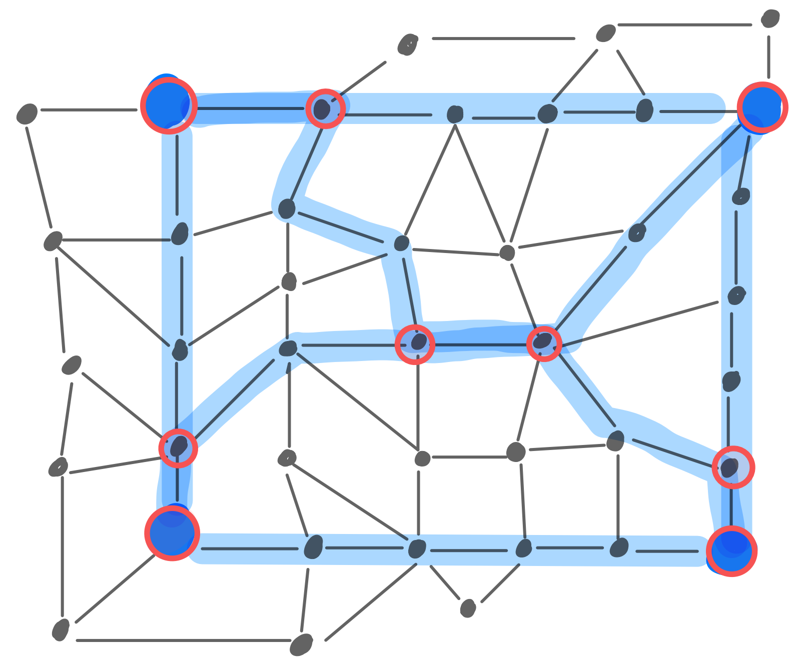
Thm. [KNZ14] Every graph G admits a 1-DAM with size $|T|^4$



1. Draw a shortest path in G between every pair of terminals
2. Contract away degree-2 vertices

"Path Overlaying" Algorithm

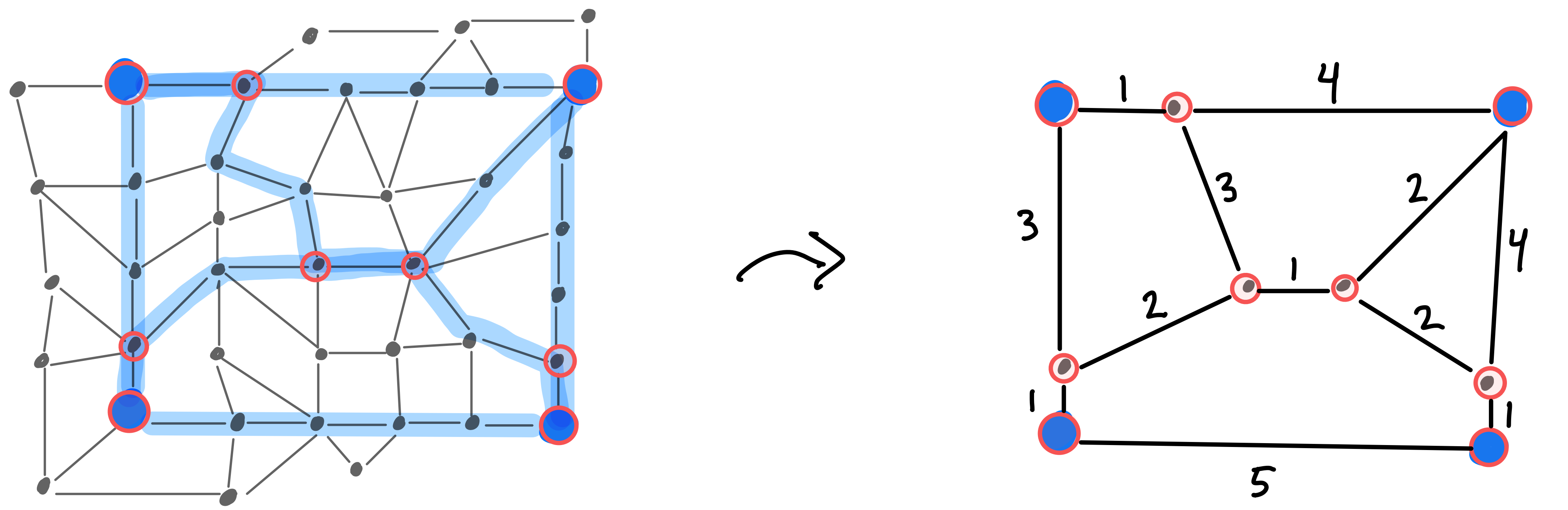
Thm. [KNZ14] Every graph G admits a 1-DAM with size $|T|^4$



1. Draw a shortest path in G between every pair of terminals
2. Contract away degree-2 vertices

“Path Overlaying” Algorithm

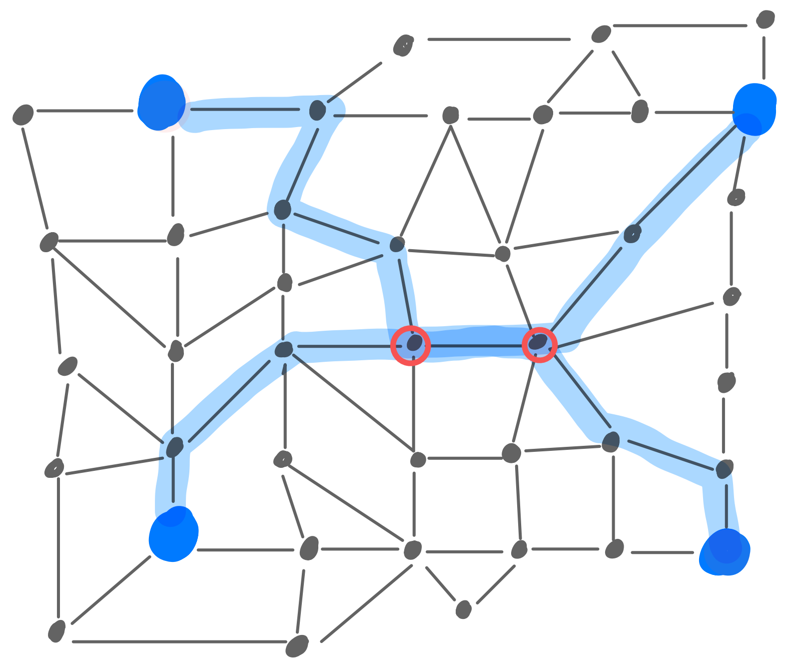
Thm. [KNZ14] Every graph G admits a 1-DAM with size $|T|^4$

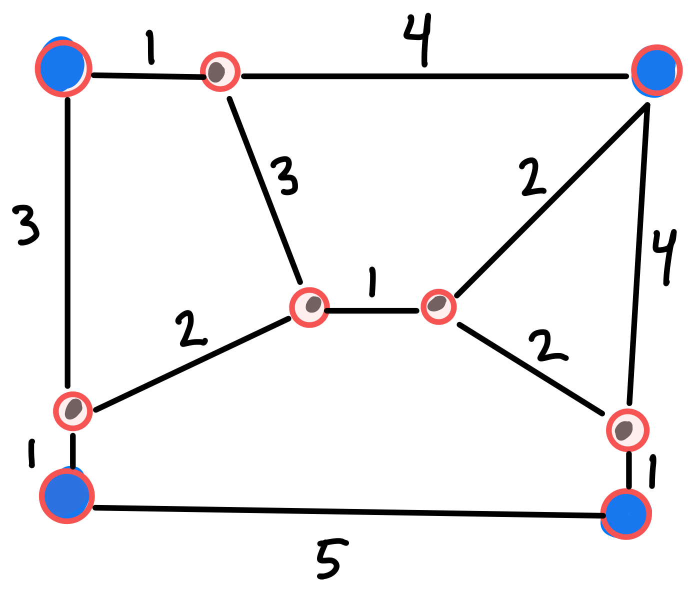


1. Draw a shortest path in G between every pair of terminals
2. Contract away degree-2 vertices

“Path Overlaying” Algorithm

Thm. [KNZ14] Every graph G admits a 1-DAM with size $|T|^4$



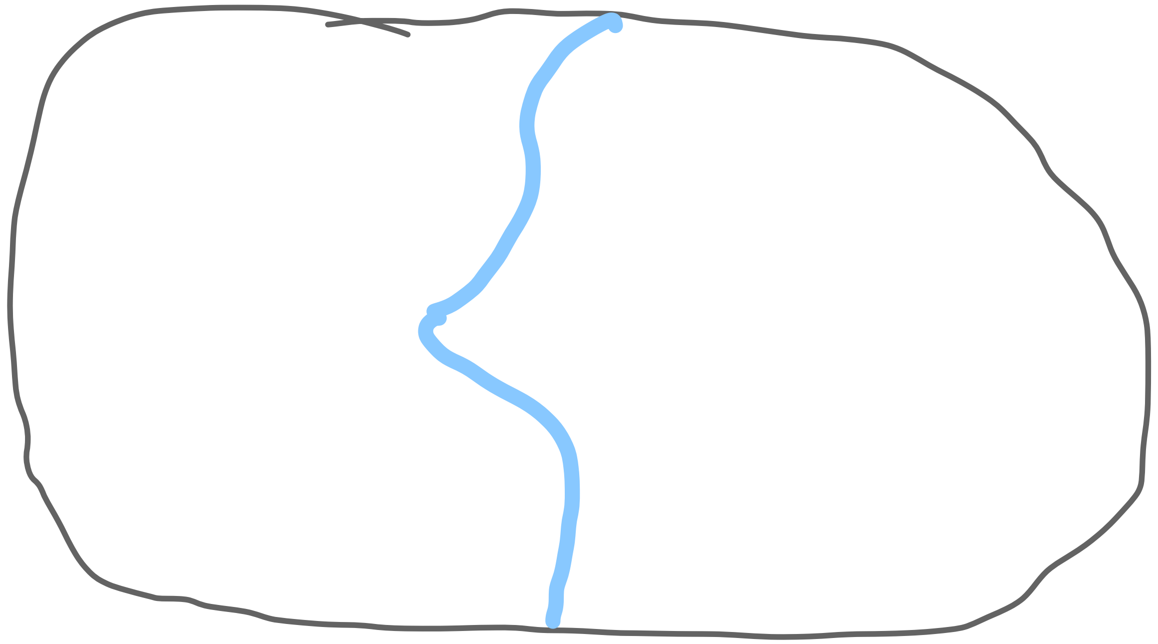


1. Draw a shortest path in G between every pair of terminals
2. Contract away degree-2 vertices

Analysis: $O(|T^2|)$ paths. Each *pair* of paths crosses at most 2 times.
$\implies O(|T|^4)$ vertices with degree ≥ 2.

Approximate DAMs for Planar Graphs

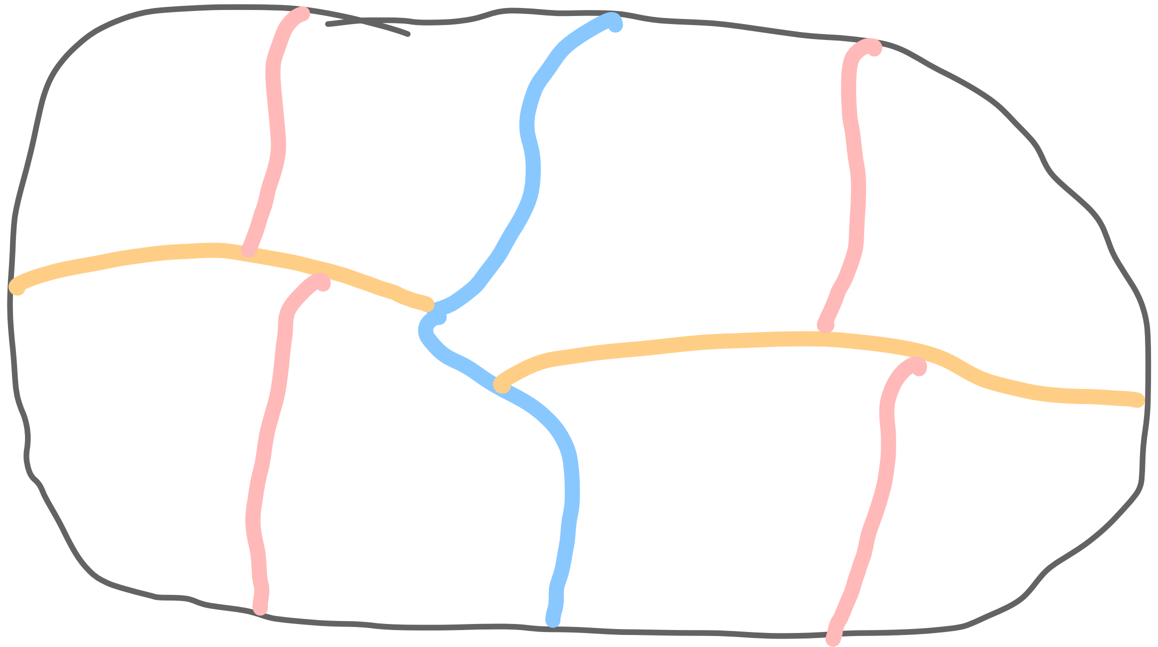
Thm. [CGH16] Every planar graph G admits a $(1+\varepsilon)$-DAM with size $|T|^2$



Shortest path separator [LT79]: there exist 2 shortest paths whose removal shatters G into components of size $\leq 2n/3$

Approximate DAMs for Planar Graphs

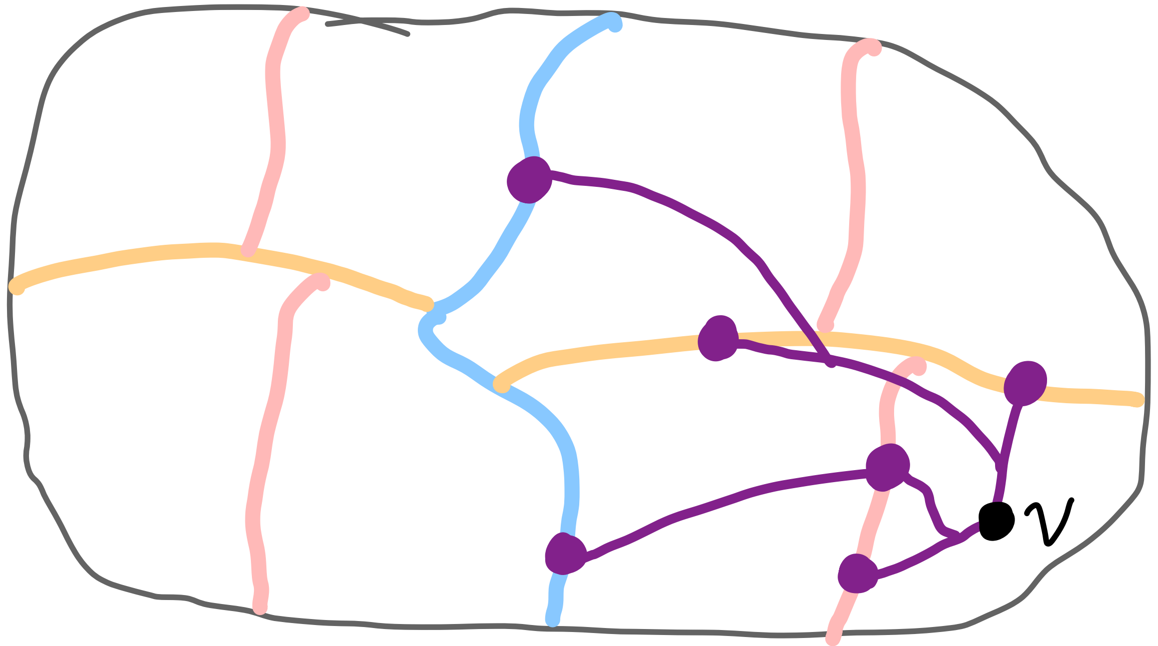
Thm. [CGH16] Every planar graph G admits a $(1+\varepsilon)$-DAM with size $|T|^2$



Shortest path separator [LT79]: there exist 2 shortest paths whose removal shatters G into components of size $\leq 2n/3$

Approximate DAMs for Planar Graphs

Thm. [CGH16] Every planar graph G admits a $(1+\varepsilon)$-DAM with size $|T|^2$



Shortest path separator [LT79]: there exist 2 shortest paths whose removal shatters G into components of size $\leq 2n/3$

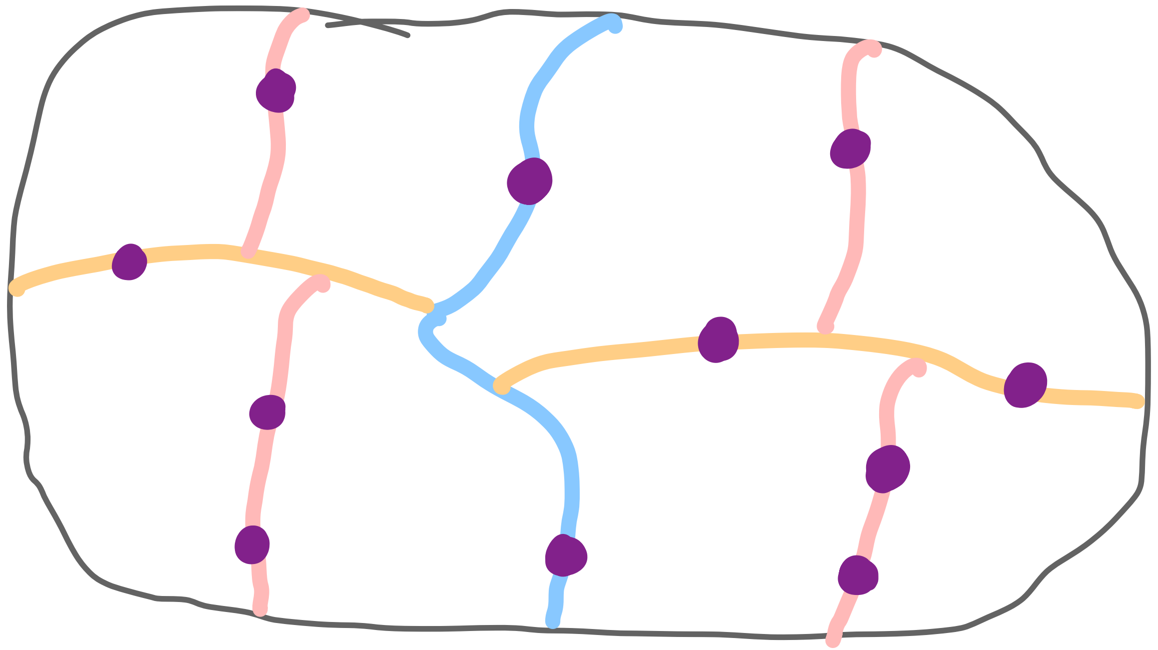
ε-cover [Tho04]: To preserve distances involving vertex v, it suffices to preserve distances between v and $O(\varepsilon^{-1}\log n) = \tilde{O}(1)$ *portals* on separators

$\implies$ DAM of size $\tilde{O}(|T|^2)$ by overlaying $\tilde{O}(|T|)$ paths

Our Approach to $\tilde{O}(|T|)$-Size DAMs

We use the same path overlaying approach!

Thm. There exist a set of $\tilde{O}(|T|)$ paths $\mathcal{P}$ that $(1+\varepsilon)$-approximately preserve pairwise distances between T. There is an order $\preceq$ on $\mathcal{P}$ such that each $P \in \mathcal{P}$ intersects $\tilde{O}(1)$ paths $P' \in \mathcal{P}$ with $P' \preceq P$.

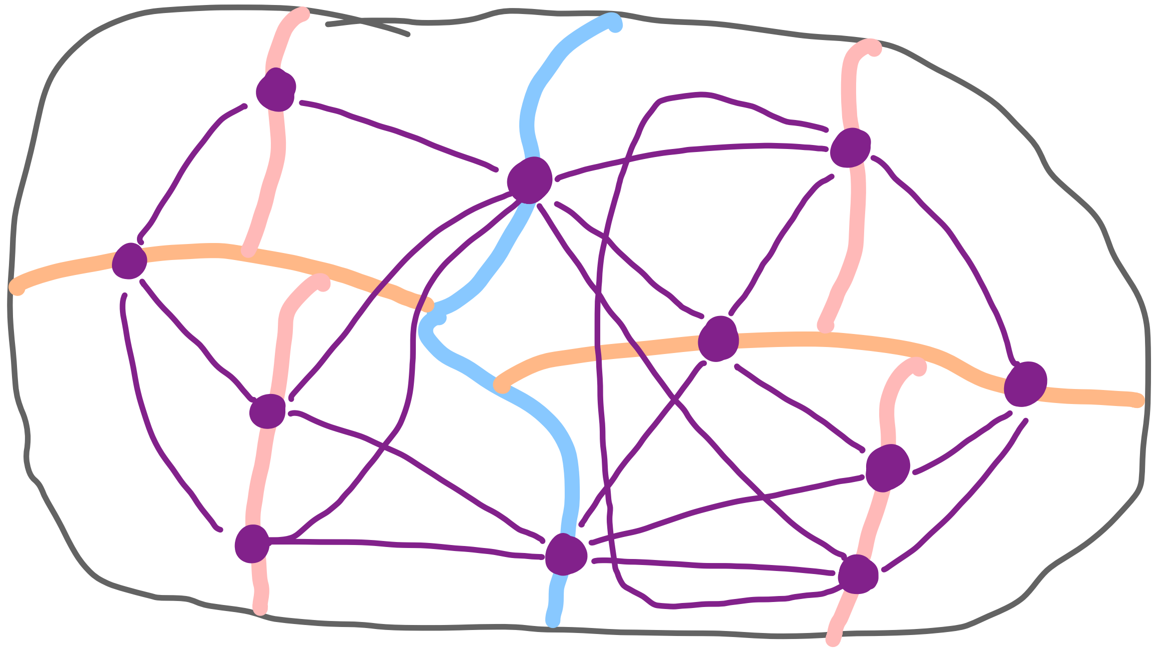


Start by drawing paths between portals on shortest-path separators...

Our Approach to $\tilde{O}(|T|)$-Size DAMs

We use the same path overlaying approach!

Thm. There exist a set of $\tilde{O}(|T|)$ paths $\mathcal{P}$ that $(1+\varepsilon)$-approximately preserve pairwise distances between T. There is an order $\preceq$ on $\mathcal{P}$ such that each $P \in \mathcal{P}$ intersects $\tilde{O}(1)$ paths $P' \in \mathcal{P}$ with $P' \preceq P$.

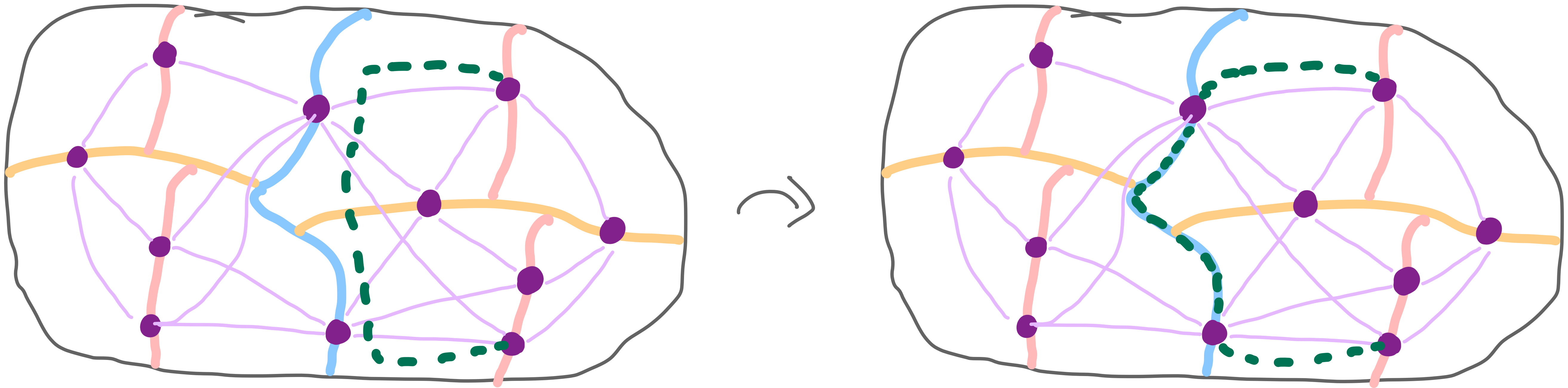


Start by drawing paths between portals on shortest-path separators...

Our Approach to $\tilde{O}(|T|)$-Size DAMs

We use the same path overlaying approach!

Thm. There exist a set of $\tilde{O}(|T|)$ paths $\mathcal{P}$ that $(1+\varepsilon)$-approximately preserve pairwise distances between T. There is an order $\preceq$ on $\mathcal{P}$ such that each $P \in \mathcal{P}$ intersects $\tilde{O}(1)$ paths $P' \in \mathcal{P}$ with $P' \preceq P$.

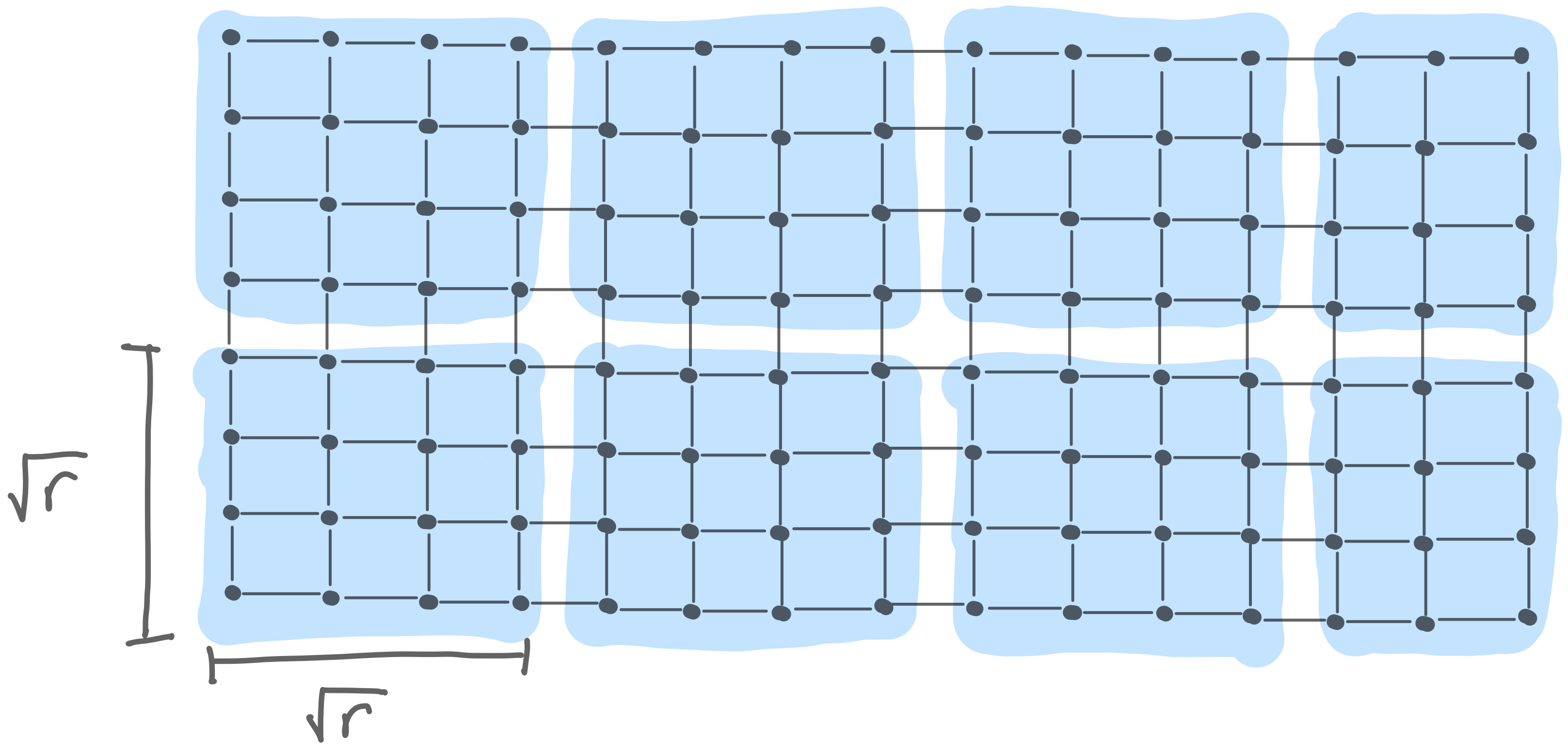


...if some path intersects many others, detour that path along a separator

Part 3: Why are DAMs Useful?

DAMs Help Speed Up Distance Computations

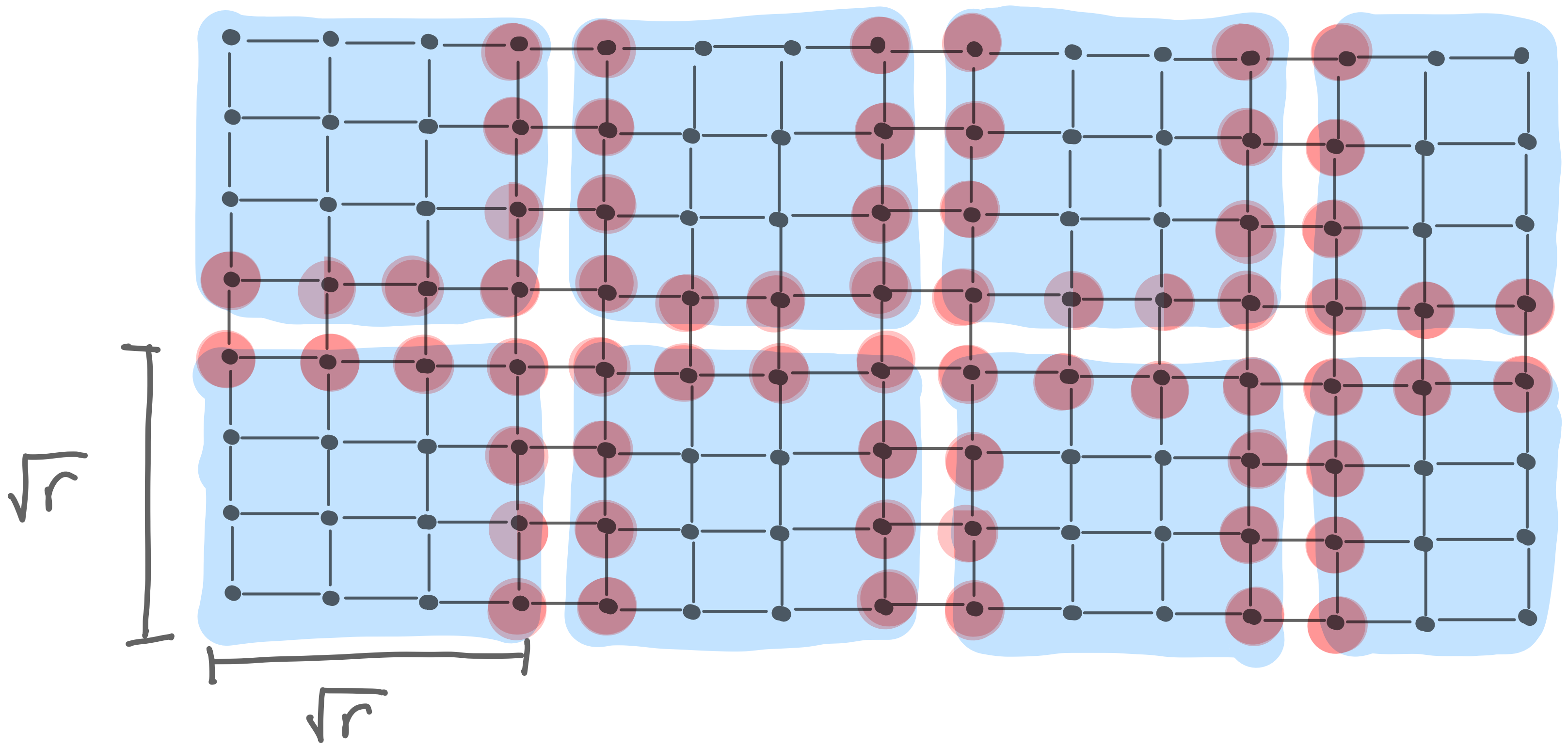
Example: $\mathrm{poly}(n)$-time construction of $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size
$\implies$ $\tilde{O}(n)$-time $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size



Tool: *r-division*. A planar G can be partitioned into n/r regions of $\leq r$ vertices, each with $O(\sqrt{r})$ boundary vertices.

DAMs Help Speed Up Distance Computations

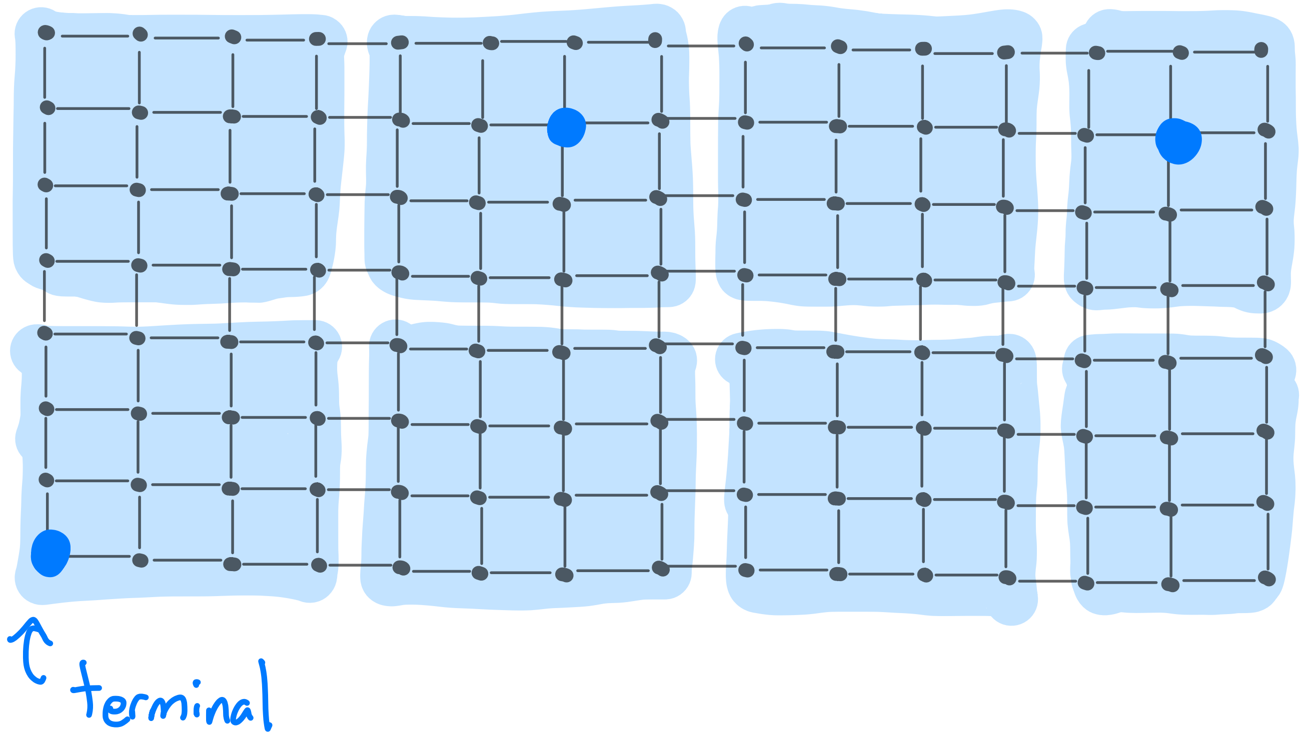
Example: $\mathrm{poly}(n)$-time construction of $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size
$\implies$ $\tilde{O}(n)$-time $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size



Tool: *r-division*. A planar G can be partitioned into n/r regions of $\leq r$ vertices, each with $O(\sqrt{r})$ boundary vertices.

DAMs Help Speed Up Distance Computations

Example: $\mathrm{poly}(n)$-time construction of $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size
$\implies$ $\tilde{O}(n)$-time $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size

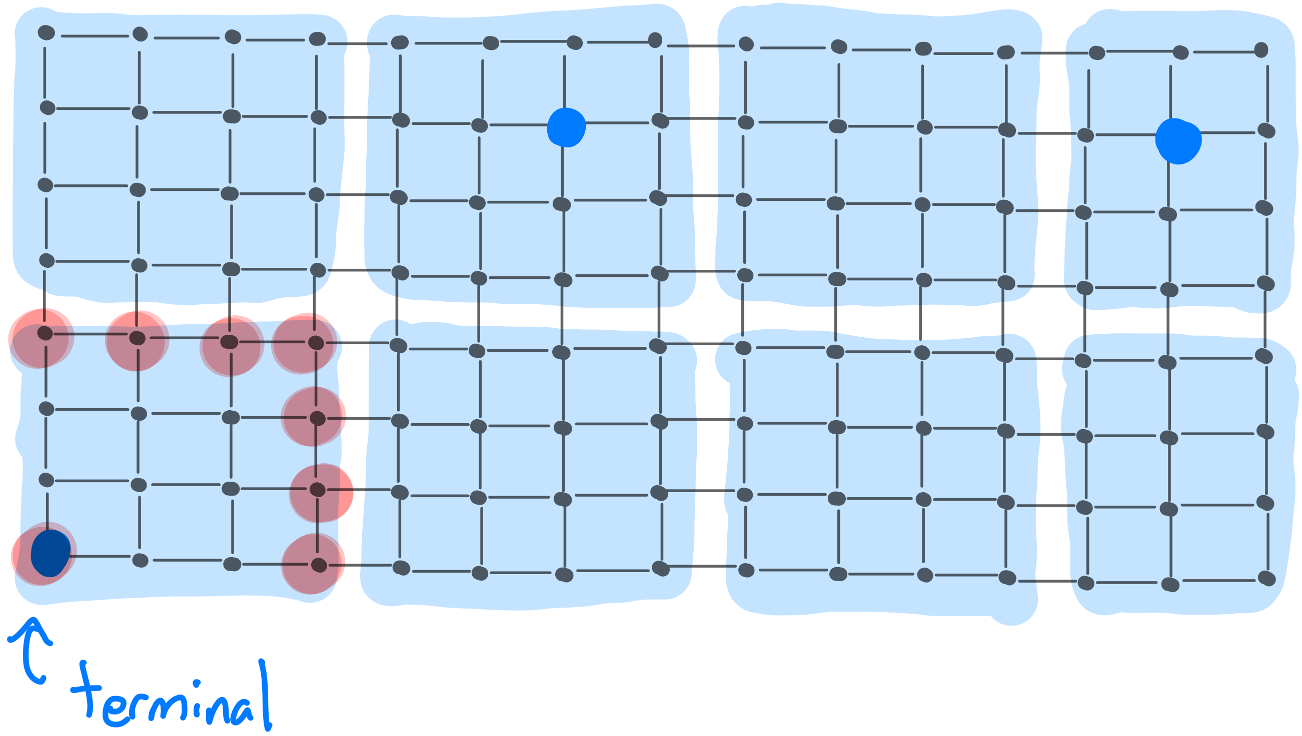


1. Compute r-division with $r = O(\log n)$
2. Replace each region of r vertices with a DAM on the $\sqrt{r}$ boundary vertices (plus T).

Tool: *r-division*. A planar G can be partitioned into n/r regions of $\leq r$ vertices, each with $O(\sqrt{r})$ boundary vertices.

DAMs Help Speed Up Distance Computations

Example: $\mathrm{poly}(n)$-time construction of $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size
$\implies$ $\tilde{O}(n)$-time $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size

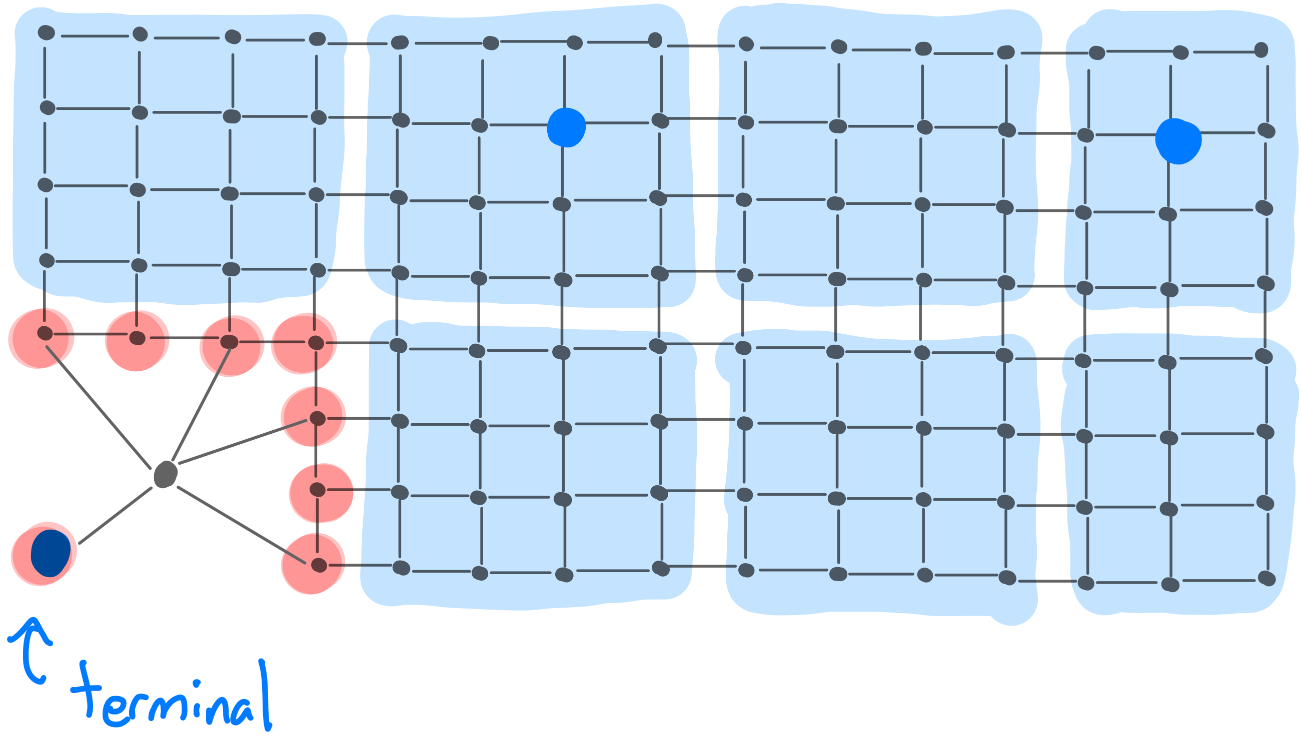


1. Compute r-division with $r = O(\log n)$
2. Replace each region of r vertices with a DAM on the $\sqrt{r}$ boundary vertices (plus T).

Tool: *r-division*. A planar G can be partitioned into n/r regions of $\leq r$ vertices, each with $O(\sqrt{r})$ boundary vertices.

DAMs Help Speed Up Distance Computations

Example: $\mathrm{poly}(n)$-time construction of $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size
$\implies$ $\tilde{O}(n)$-time $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size

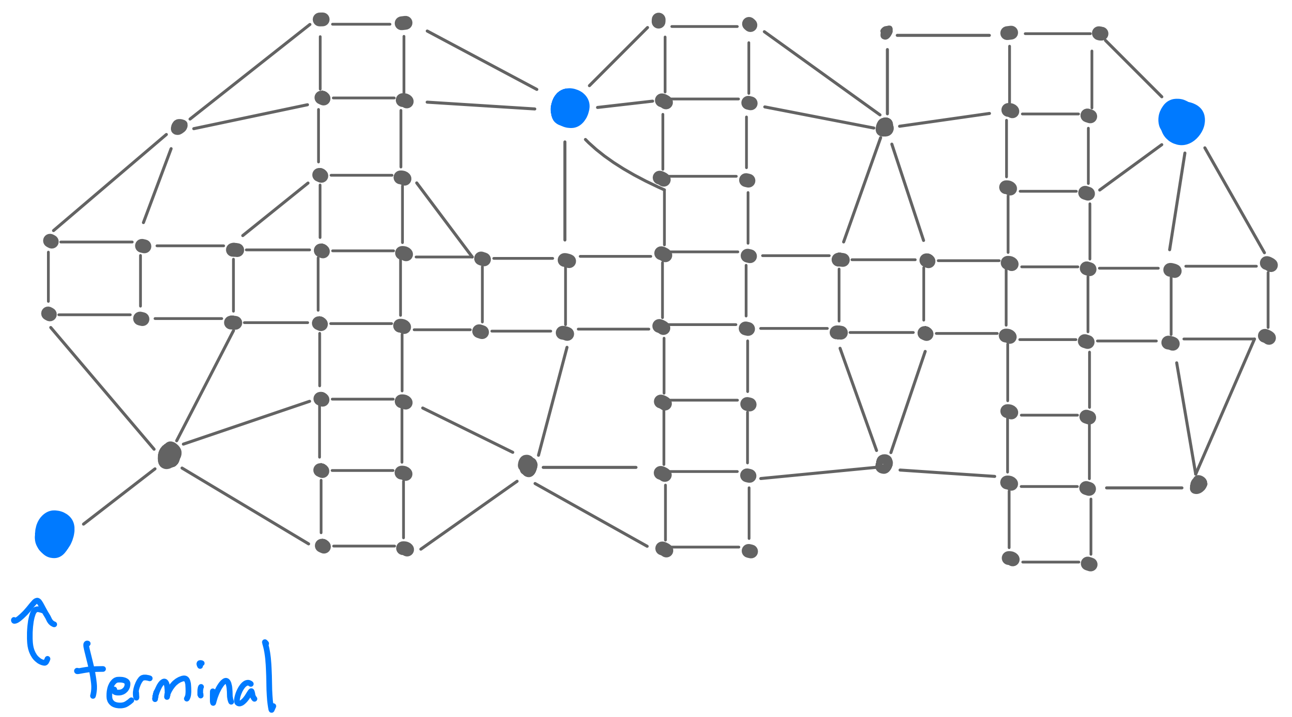


1. Compute r-division with $r = O(\log n)$
2. Replace each region of r vertices with a DAM on the $\sqrt{r}$ boundary vertices (plus T).

Tool: *r-division*. A planar G can be partitioned into n/r regions of $\leq r$ vertices, each with $O(\sqrt{r})$ boundary vertices.

DAMs Help Speed Up Distance Computations

Example: $\text{poly}(n)$-time construction of $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size
$\implies$ $\tilde{O}(n)$-time $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size

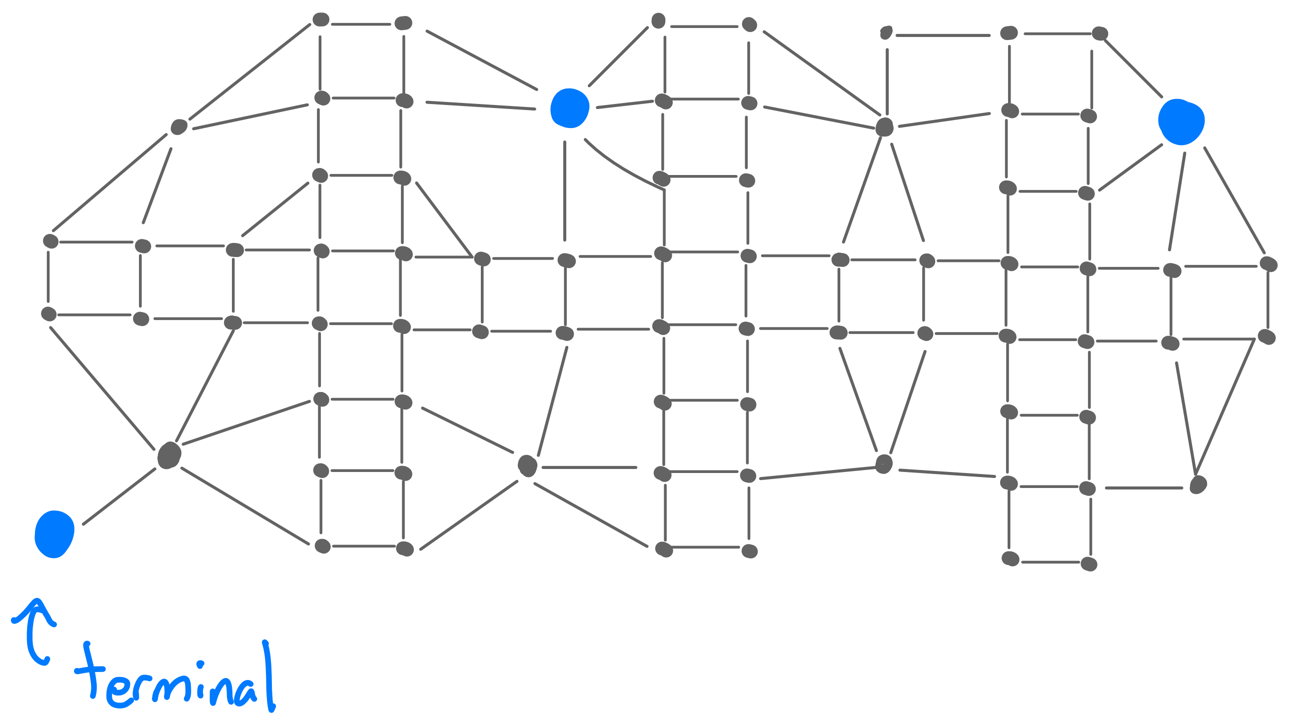


1. Compute r-division with $r = O(\log n)$
2. Replace each region of r vertices with a DAM on the $\sqrt{r}$ boundary vertices (plus T).

Tool: *r-division*. A planar G can be partitioned into n/r regions of $\leq r$ vertices, each with $O(\sqrt{r})$ boundary vertices.

DAMs Help Speed Up Distance Computations

Example: $\mathrm{poly}(n)$-time construction of $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size
$\implies$ $\tilde{O}(n)$-time $(1+\varepsilon)$-DAM with $\tilde{O}(|T|)$ size



1. Compute r-division with $r = O(\log n)$
2. Replace each region of r vertices with a DAM on the $\sqrt{r}$ boundary vertices (plus T).

Application [CGHPS20]: Offline $(1+\varepsilon)$-approximate dynamic distance oracle for minor-free graphs

Conclusion

We design $(1+\varepsilon)$-DAM for planar and minor-free graphs, with size $\tilde{O}(|T|) \cdot \mathrm{poly}\log \Phi$, using a path overlaying approach.

Open questions:

1. Remove the dependence on aspect ratio Φ?
2. Construct fully dynamic distance oracle for minor-free graphs?

Conclusion

We design $(1+\varepsilon)$-DAM for planar and minor-free graphs, with size $\tilde{O}(|T|) \cdot \operatorname{poly}\log \Phi$, using a path overlaying approach.

Open questions:

1. Remove the dependence on aspect ratio Φ?
2. Construct fully dynamic distance oracle for minor-free graphs?