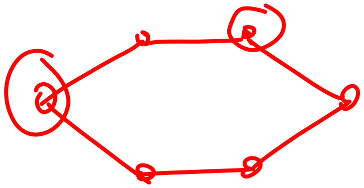
Algorithmic Applications of Metrics Embeddings 3
Tree covers and distance labelings

R. Ravi

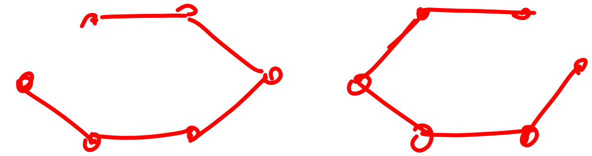
ravi@cmu.edu

Overview

- Day before Yesterday: Introduction, Spanners (Embedding into subgraph metrics)
- Yesterday : Embedding into tree metrics
- Today : Tree covers for routing and approximate distance oracles



Tree Covers



- Alternate to approximating distances by random trees
- Pick a few trees such that for every pair, some tree has a shortest path for the pair
- Defn: A tree cover for a graph (and its metric) is a family of trees such that
  - Each tree dilates distances and
  - For any pair of nodes (x,y), some tree in the family maintains the distance between x and y
- Natural measure for optimization: Minimize the number of trees in the cover; Do better for special classes of graphs

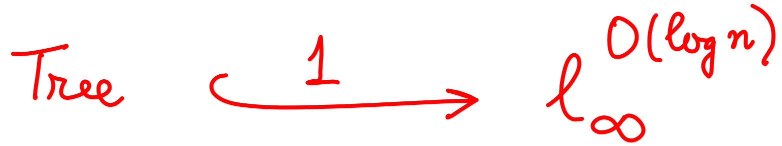
Applications of Tree Covers

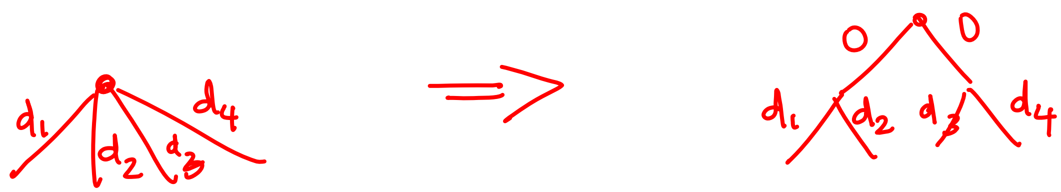
- Network Routing: Given a routing protocol for trees and a tree cover, we can assign the routing between pairs to the trees that preserve the distance for the pair and use the tree protocols
  - Overhead: If k trees in cover, O(log k) extra bits in each packet to specify which tree to use

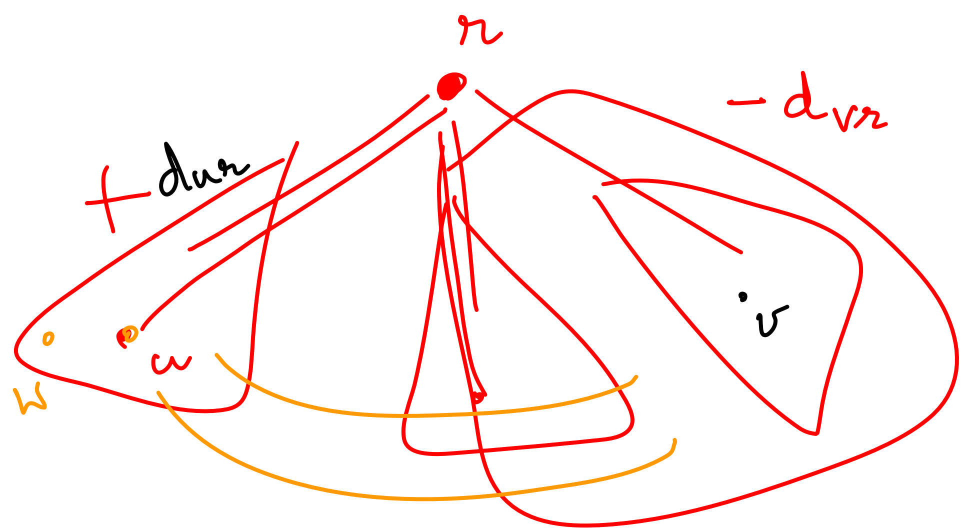
Applications of Tree Covers

- Distance labeling: Assign labels to vertices so that one can infer distances between any two vertices by just looking at the labels
  - Trivial with n labels at every vertex
  - Trees have distance labelings with O(log n) labels per node
  - With k trees in the cover, can get a distance labeling with O(k log n) labels

Trees have distance labeling with O(log n) labels?

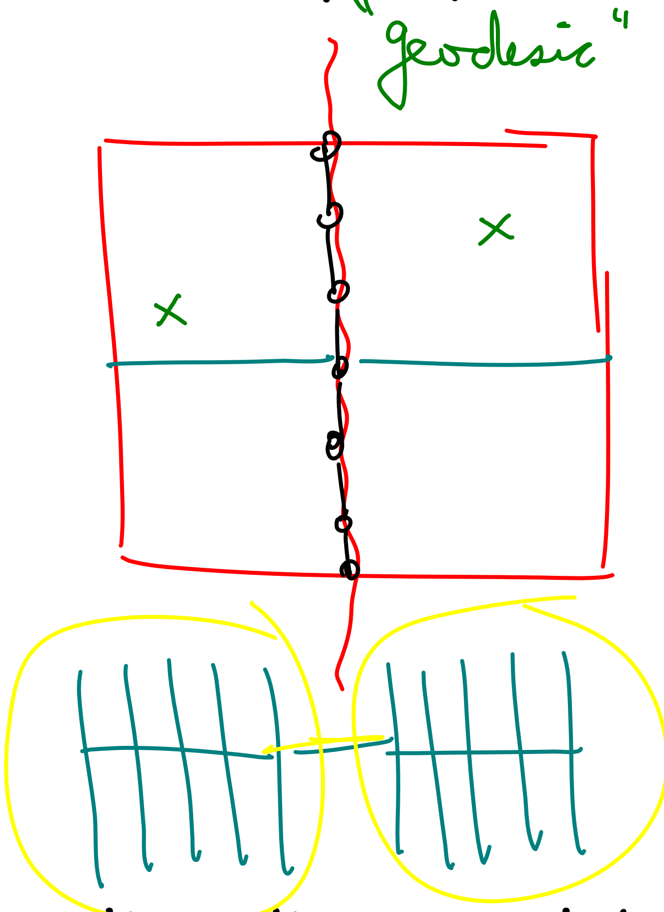


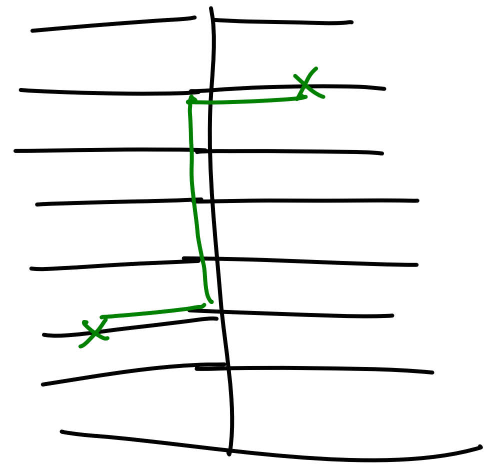




Grids have O(log n)-sized tree covers

- Idea: Use the natural separators and compute distances from it

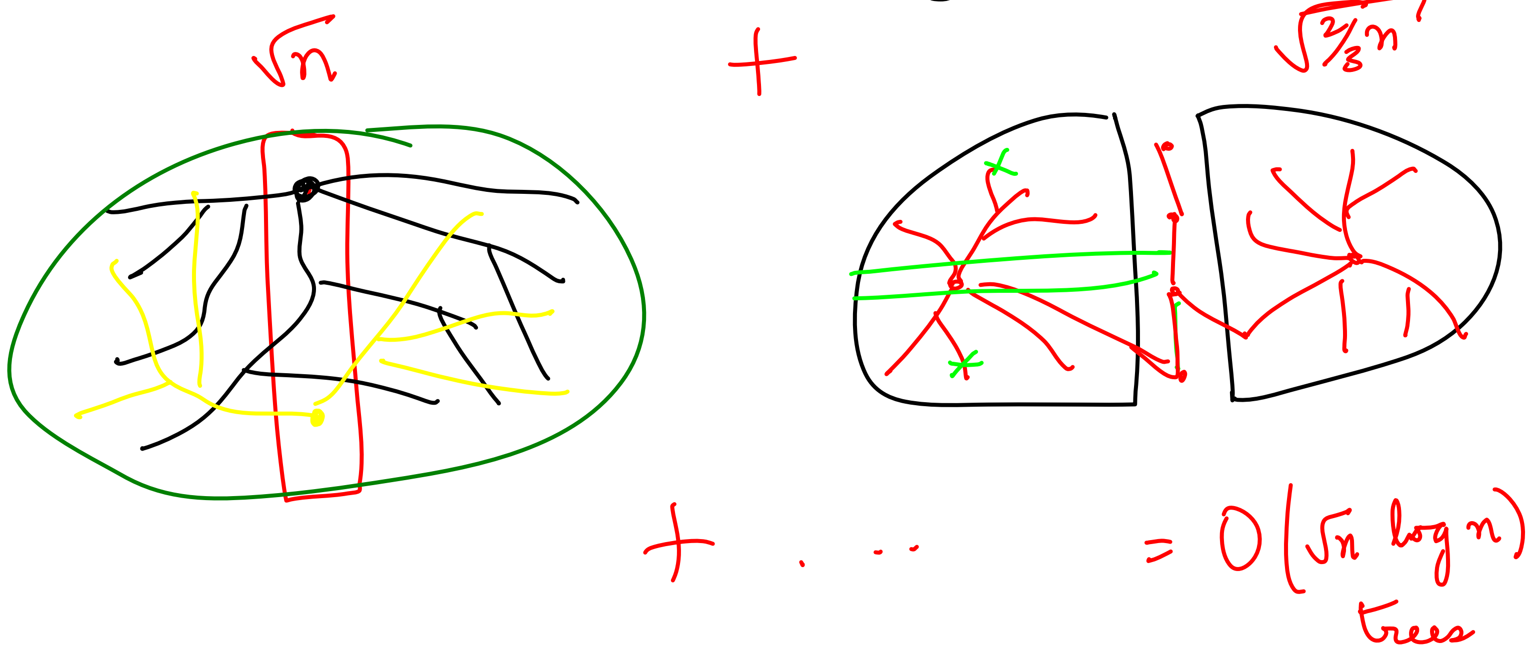




- Implies distance labeling with $O(\log^2 n)$ labels

Planar graphs have $O^*(\sqrt{n})$ size tree covers

- Use $O(\sqrt{n})$-size separators
- Include a shortest path tree from every separator node in the tree cover
- Recurse on the two pieces and pair up their trees in the final cover to get the bound



Extensions to recursive small separator graphs

- Defn: G has 1/3-balanced r(n) sized vertex separator if there is a r(n)-sized set of vertices, which if removed, breaks the graph in two components of size at least n/3.
- G has a recursive 1/3-balanced r(n)-sized vertex separator if the same holds recursively for the components formed after deleting the separator
  - E.g. outerplanar and series-parallel graphs have 2-sized recursive separators; treewidth-k graphs have k-sized recursive separators.

- Theorem: Every graph with recursive 1/3-balanced $r(n)$-sized vertex separators has an $r(n) \log_{3/2} n$-sized tree cover.

Simple lower bound for size of tree cover

- The unweighted complete graph on n nodes does not have a (n/2-1) cover

$$\frac{|K_n|}{|tree|}$$

Tree covers with stretch

- Stretch-D tree cover is a family of trees such that
  - Every tree dilates the graph distances and
  - For any two nodes, there is a tree in the family that maintains the distance between the pair to within a factor ~~t~~ D

- [GKR '01, T '01] Every planar graph has a stretch-3 O(log n) sized tree cover

- [TZ '01] For all graphs there is a stretch (2k - 1) tree cover such that every vertex lies in $O^*(n^{1/k})$ trees

Stretch-3 planar tree covers

- Defn: G has a t-path separator if there are t paths in G such that
  - Each path is a *geodesic* (i.e., a shortest path between its endpoints)
  - The union of the paths is a 1/3-balanced separator

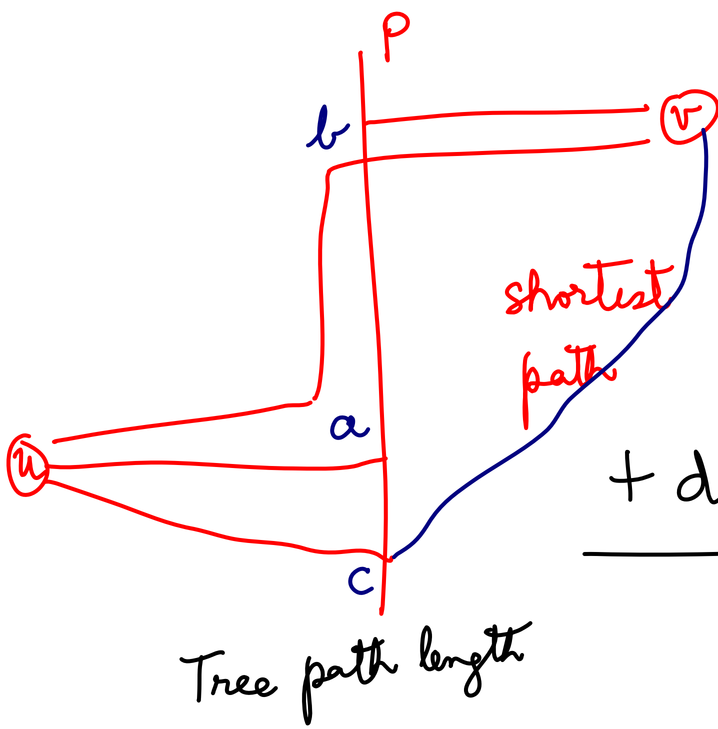
- Theorem: Every planar graph has a 2-path separator
- Proof idea:
  - Use Lipton-Tarjan separator algorithm starting with a shortest path tree T
  - Traingulate graph and argue existence of an edge e=(u,v) such that the fundamental cycle corresponding to this edge in T is a good separator
  - Paths from (u,lca) and (v,lca) in T give the 2-path separator

t-path separators to stretch-3 tree covers

- If G has t-path recursive separators, it has an O(t log n) sized stretch-3 tree cover
- Use the idea from grids building shortest path trees from each path in the separator and recurse to argue size

t-path separators give 3 stretch trees in cover

- Consider path P that intersects the shortest path for a pait of vertices u, v and use geodesic property



$$d(u,a) \leq d(u,c)$$

$$+\ d(v,b) \leq d(v,c)$$

↑ shortest distances from P

$$+\ d(a,b) \leq d(a,u) + d(u,v) + d(v,b)$$

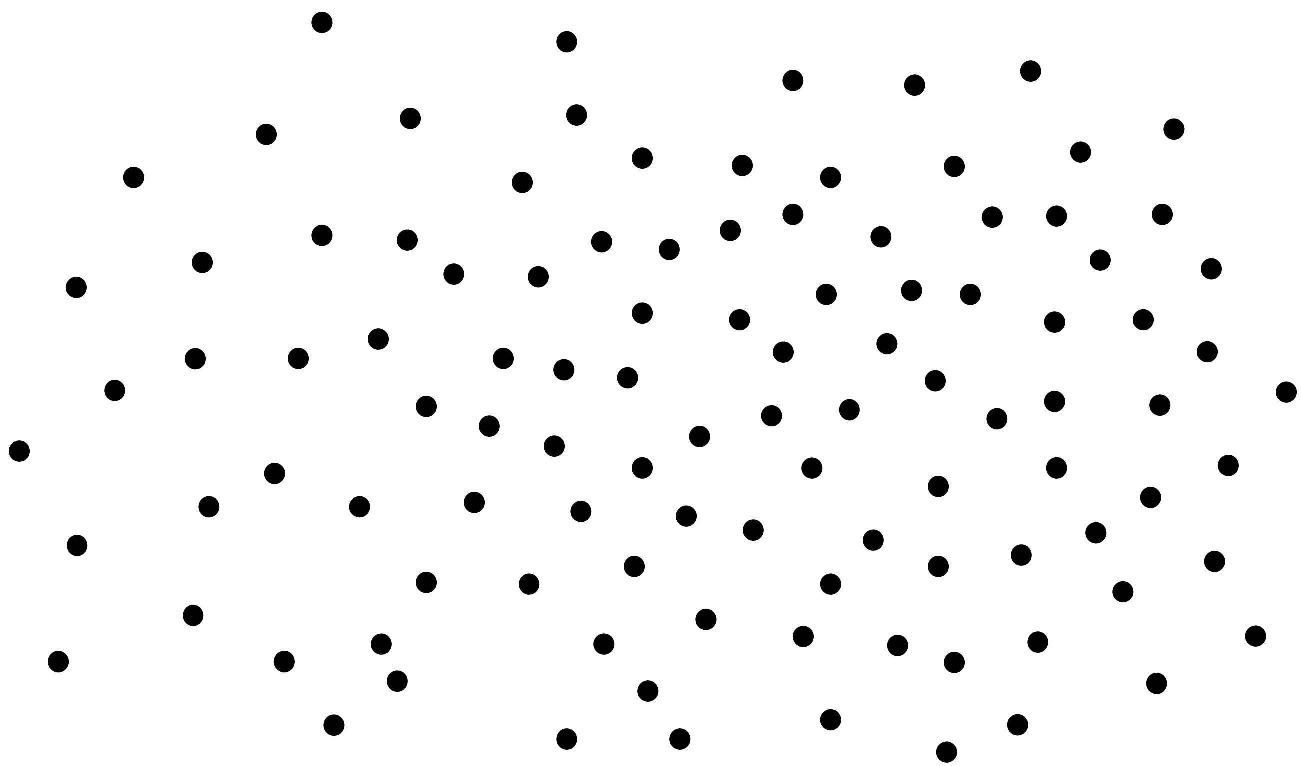
$$\leq 2\big(d(u,c) + d(v,c)\big) + d(u,v)$$

$$\leq 3\, d(u,v) \qquad \square$$

Stretch-t tree covers in general graphs

- We will see how to construct in a (2k -1) stretch distance labeling with space $O(kn^{1/k})$ labels per node in time $O(kmn^{1/k})$
- This can be used to construct a stretch (2k - 1) tree cover where every node is in at most $O(kn^{1/k})$ trees
- Another corollary is the construction of a (2k-1)-spanner with $O(kn^{1+1/k})$ edges.
- Following slides from Uri Zwick

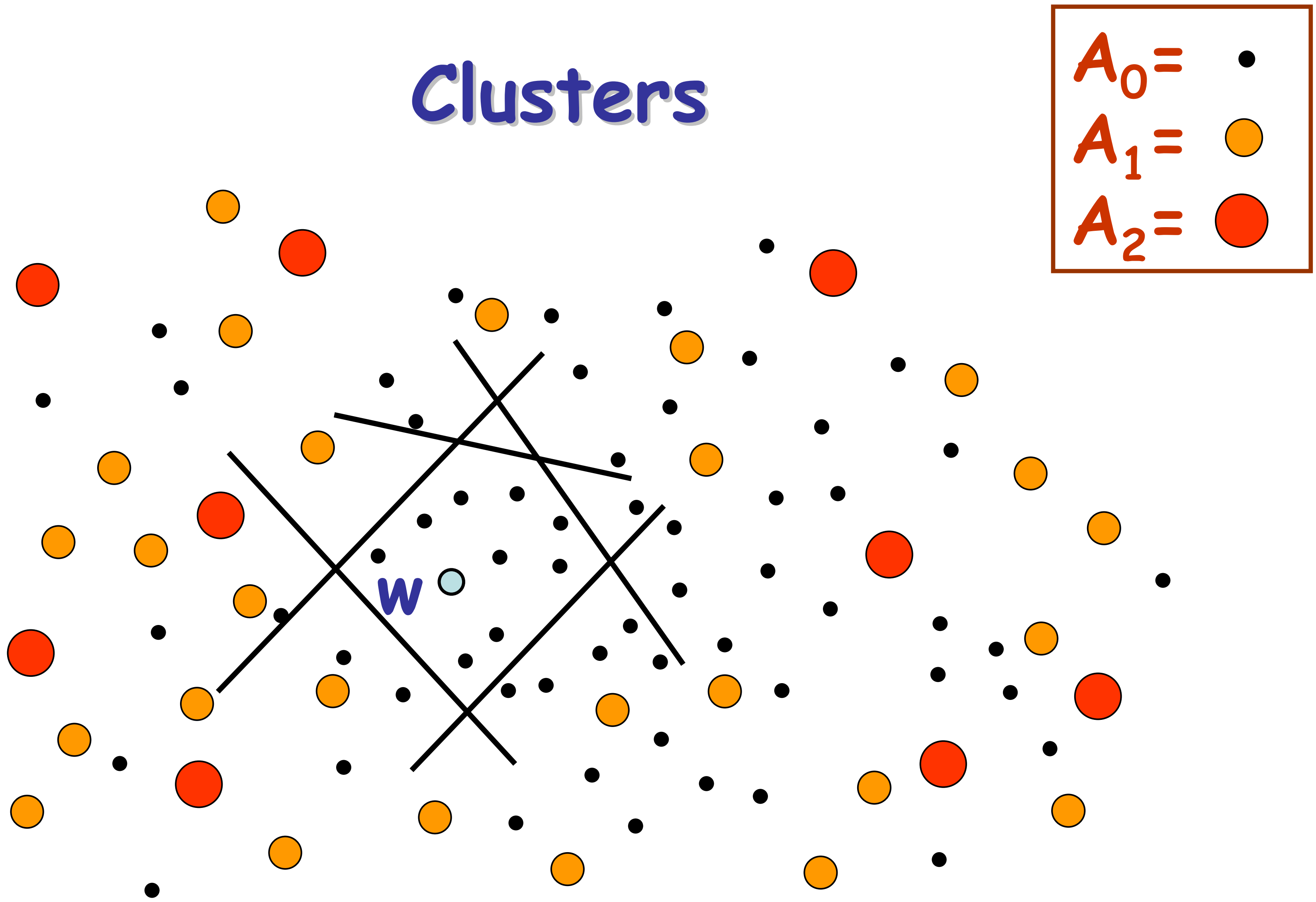
A hierarchy of centers



$A_0 \leftarrow V$; $A_k \leftarrow \emptyset$;

$A_i \leftarrow \text{sample}(A_{i-1}, n^{-1/k})$;

Clusters



$$C(w) \leftarrow \{v \in V \mid \delta(w,v) < \delta(A_{i+1},v)\} \quad , \quad w \in A_i - A_{i+1}$$

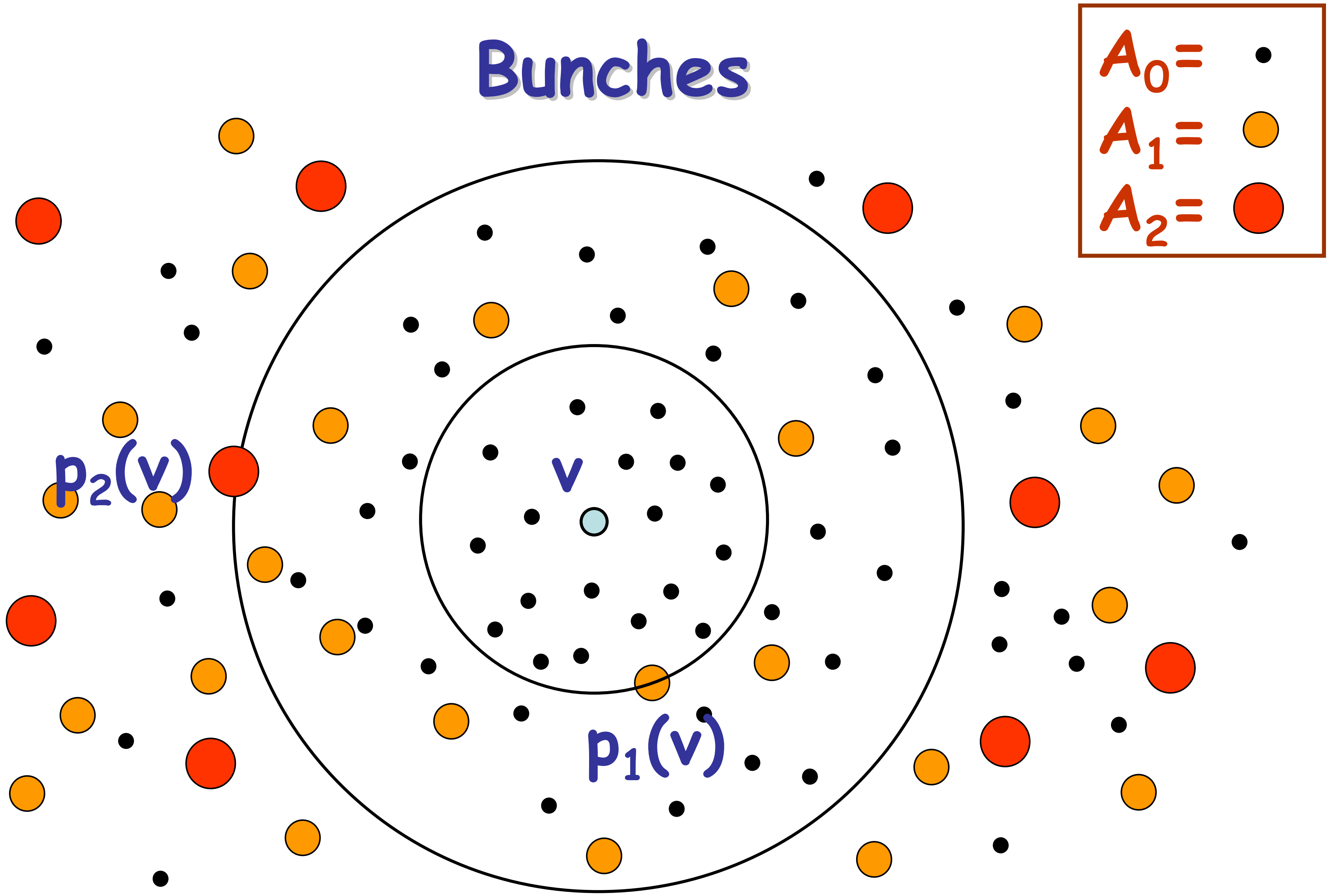
Bunches (inverse clusters)

$$w \in B(v) \quad \Leftrightarrow \quad v \in C(w)$$

$$C(w) \leftarrow \{v \in V \mid \delta(w,v) < \delta(A_{i+1},v)\} \quad ,$$
$$\text{if } w \in A_i - A_{i+1}$$

$$B(v) \leftarrow \bigcup_i \{w \in A_i - A_{i+1} \mid \delta(w,v) < \delta(A_{i+1},v)\}$$

Bunches



$$B(v) \leftarrow \bigcup_i \{ w \in A_i - A_{i+1} \mid \delta(w,v) < \delta(A_{i+1},v) \}$$

The data structure

For every vertex $v \in V$:

- The centers $p_1(v),\ p_2(v), \ldots,\ p_{k-1}(v)$
- A **hash table** holding $B(v)$

For every $w \in V$, we can check, in **constant time**, whether $w \in B(v)$, and if so, what is $\delta(v,w)$.

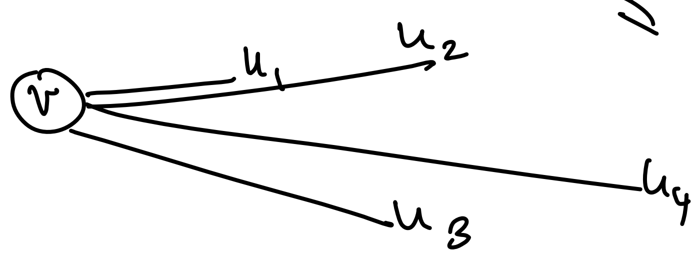
Lemma: $E[|B(v)|] \leq kn^{1/k}$

Proof: $|B(v) \cap A_i|$ is stochastically dominated by a geometric random variable with parameter $p = n^{-1/k}$.

$E(|B(v) \cap A_i|)$?

$u_j \in A_i$ in increasing distance from v

$$\leq \sum_i (1-p)^i$$



$$\leq \frac{1}{1-(1-p)} = \frac{1}{p}$$

$$= n^{1/k}$$

Query answering algorithm

Algorithm $dist_k(u,v)$

$w \leftarrow u$, $i \leftarrow 0$

while $w \notin B(v)$

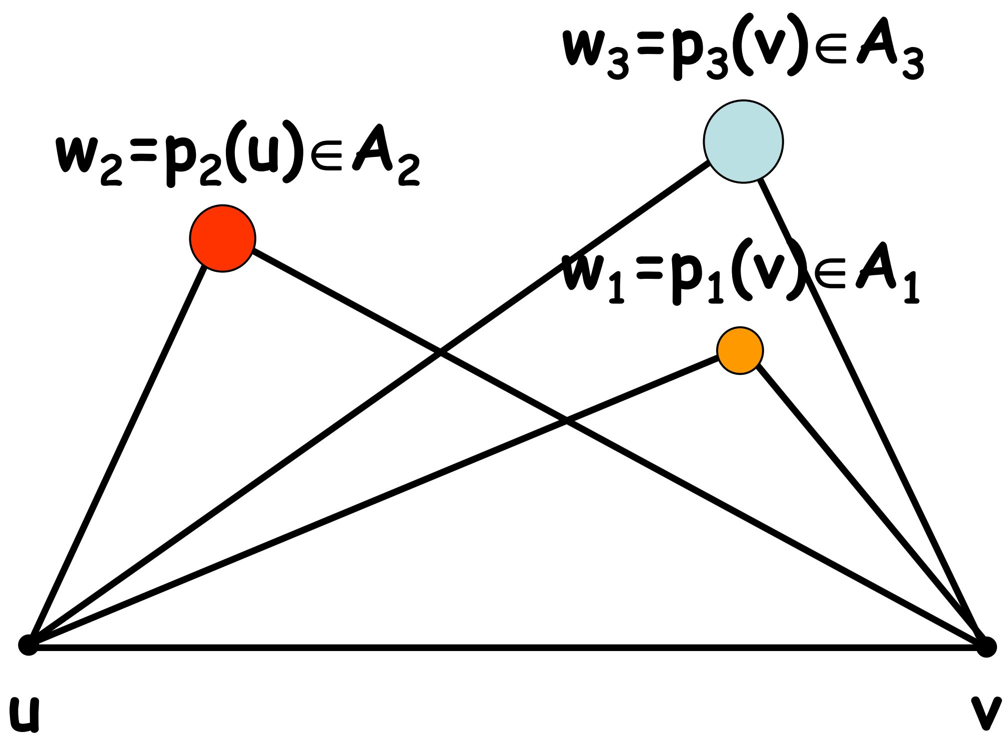
{ $i \leftarrow i+1$

$(u,v) \leftarrow (v,u)$

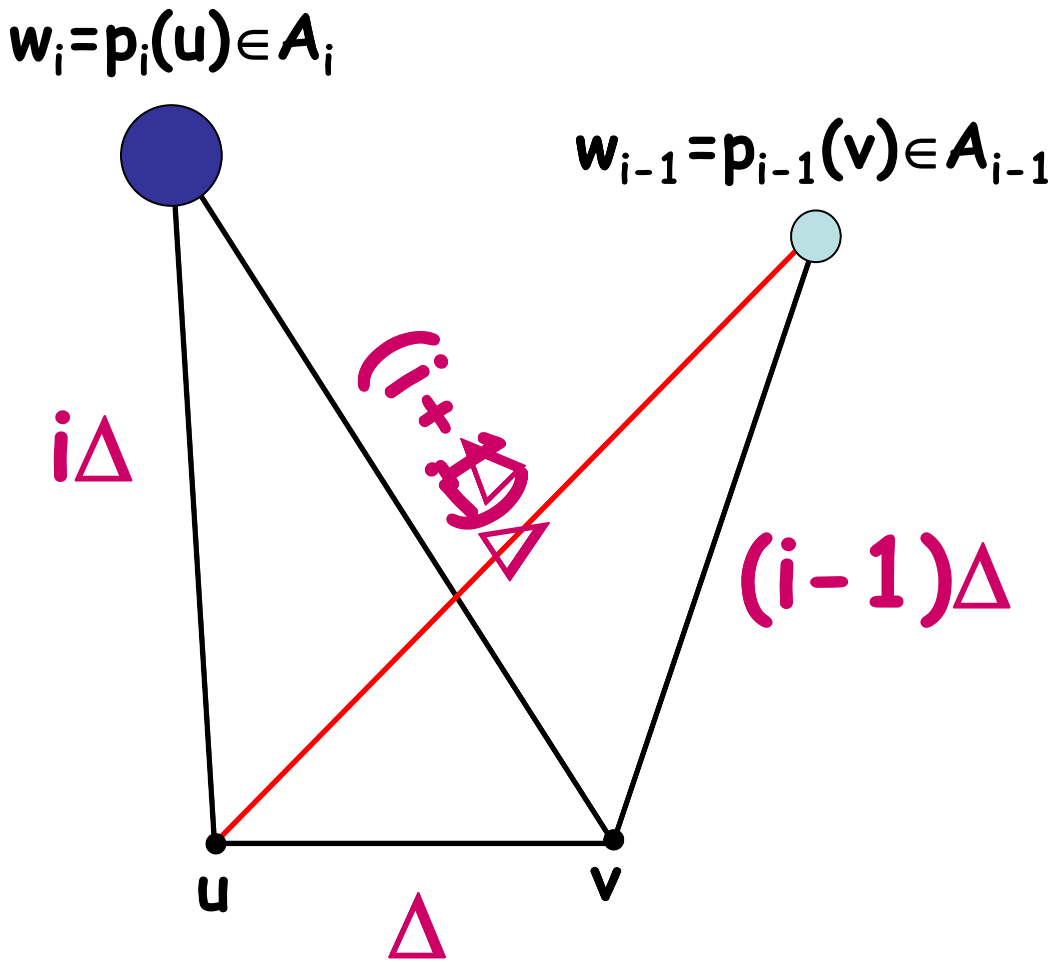
$w \leftarrow p_i(u)$ }

return $\delta(w,u) + \delta(w,v)$

Query answering algorithm



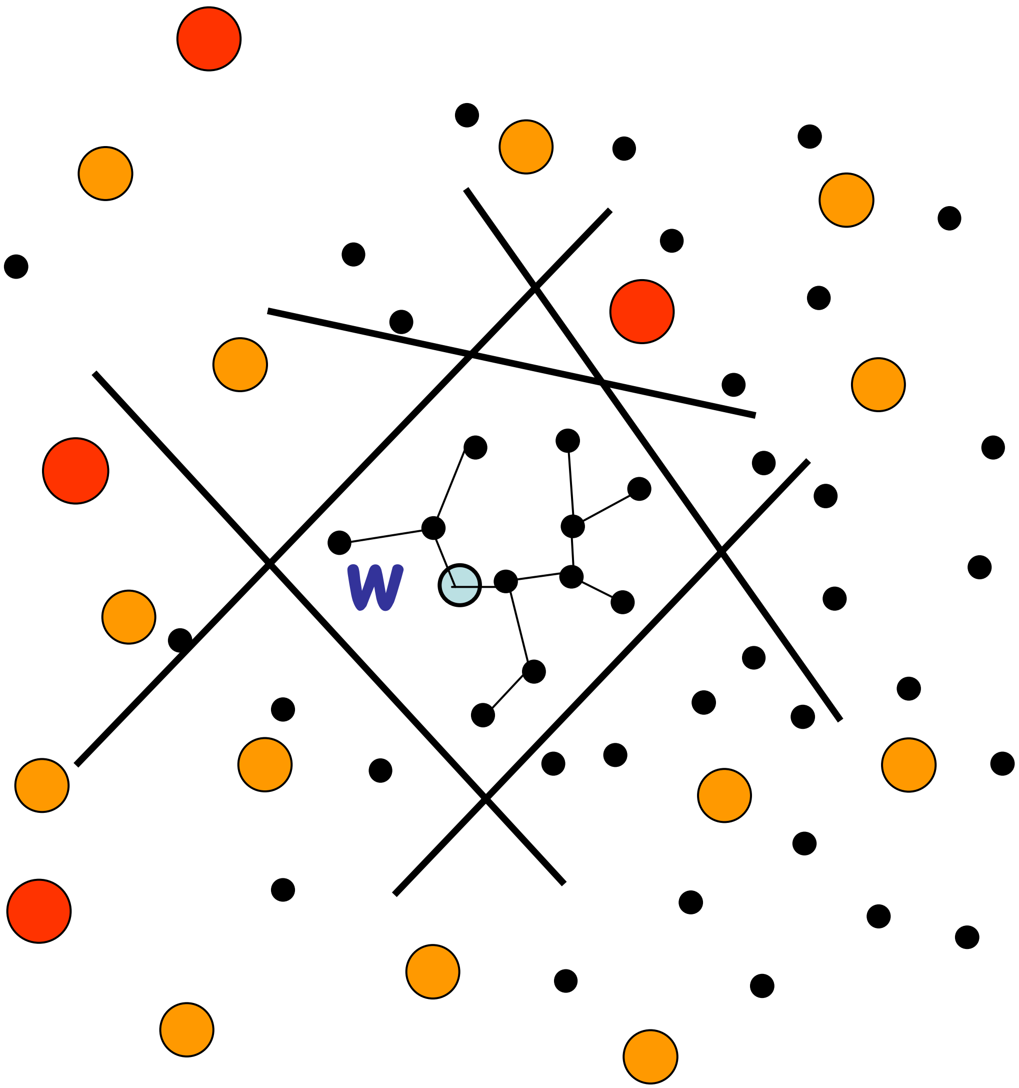
Analysis



Analysis of stretch - details

- Initially $\delta(w,u) = \delta(u,u) = 0$
- Show at every iteration, $\delta(w,u)$ does not increase by more than Δ
- Then final distance $\leq \delta(w,u) + \delta(w,v) \leq \delta(w,u) + \delta(w,u) + \Delta \leq [(k-1) + (k-1) + 1]\, \Delta$
- If i^{th} iteration passes while loop, w_{i-1} is not in $B(v_{i-1})$

Spanners / Tree covers

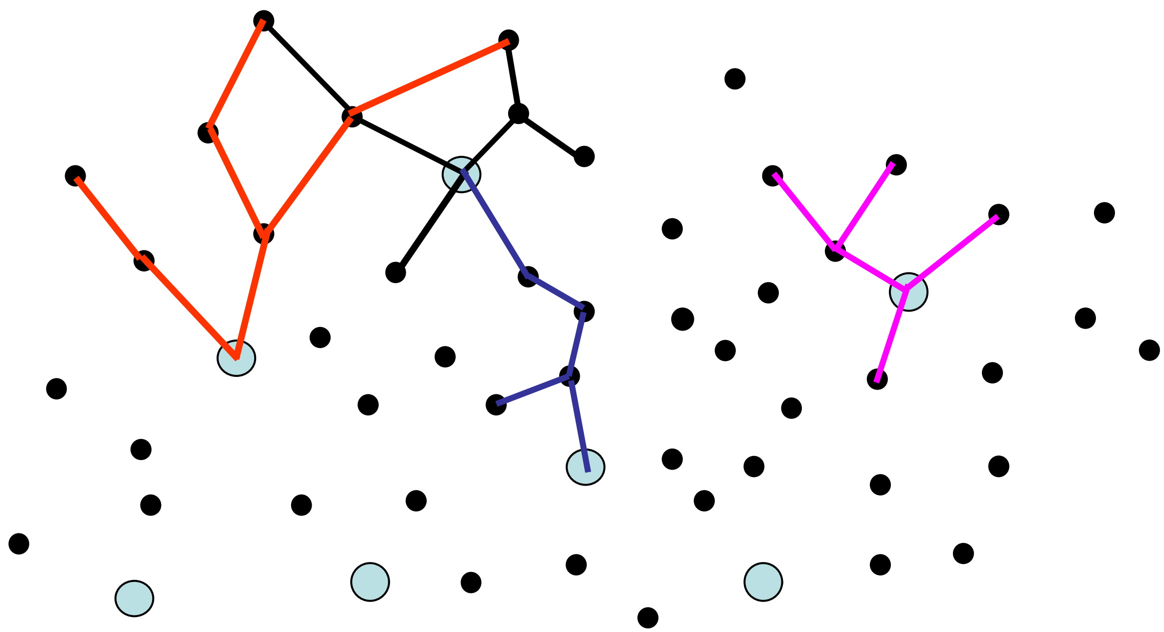


In each cluster, construct a tree of shortest paths

The union of all these trees in a (2k-1)-spanner with $kn^{1+1/k}$ edges.

Constructed in $O(kmn^{1/k})$ time!

Tree Cover



Each vertex contained in at most $n^{1/k}\log n$ trees.
For every u,v, there is a tree with a path of stretch at most $2k-1$ between them.

Summary

- Distances are vital for building and routing networks
- We discussed methods of handling generall distances by embedding into simpler ones and applications to various network problems
  - Day before Yesterday: Introduction, Spanners (Embedding into subgraph metrics)
  - Yesterday : Embedding into tree metrics
  - Today : Tree covers for routing and approximate distance oracles

- More on metric embeddings in our course website
- Good luck on framing new problems and applying these techniques in your own research!