Understanding and Analyzing the Algorithm for Approximating Arbitrary Metrics by Tree Metrics

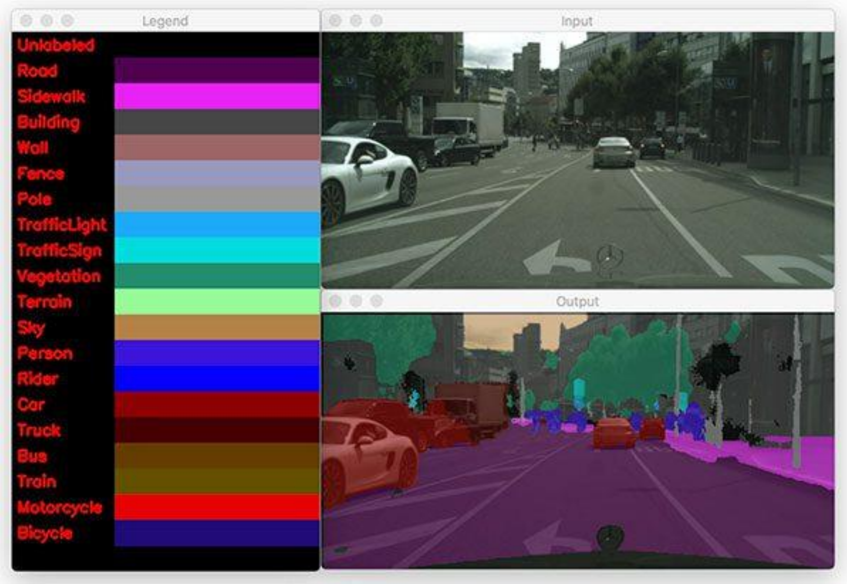
Adam Koziak

COMP 5112

Motivation

- Tree metrics are favorable from an algorithmic point of view.
- We'd like to approximate any metric with a shortest path tree metric, with minimal stretch.
- This method improves the prior bound from O(logn*loglogn) to a tight O(logn) distortion factor.
- Very important result by Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar from Kasetsart University and UC Berkeley.
- Significant impact on approximation algorithms in numerous applications.

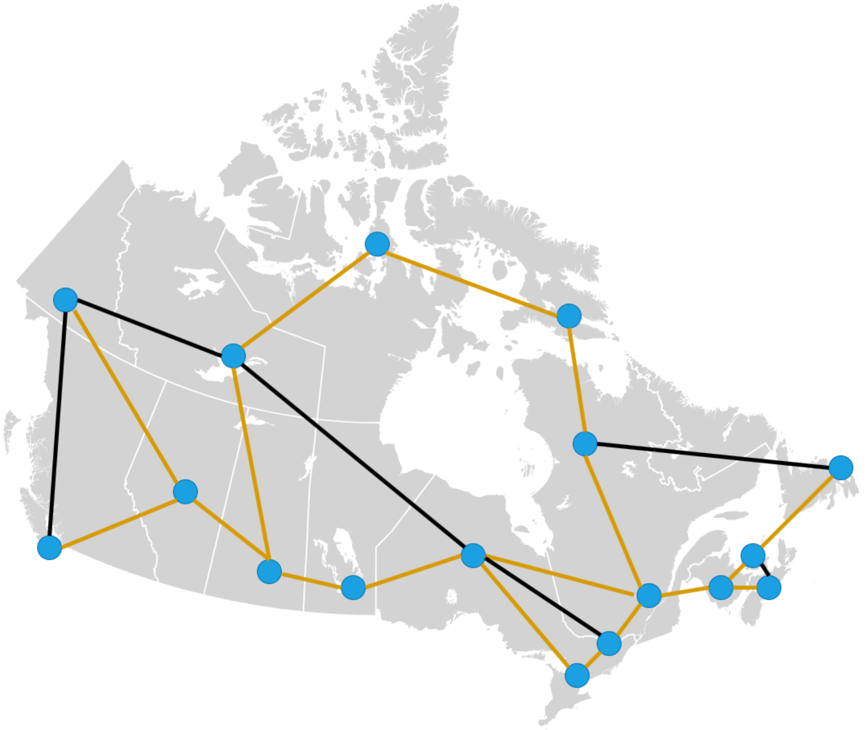
Application: Metric Labeling



- Used for image segmentation
- The image is modeled as a grid graph where each pixel is a node.
  - Edges connect neighboring pixels
  - Can optionally include other edges as well
- Edge weights represent dissimilarity between pixels
- Objective is to minimize the cost:

$$\sum_{\text{vertices } v} \text{cost of assigning label to } v + \sum_{\text{edges } (u,v)} w_{uv} \times \text{distance between labels of } u \text{ and } v.$$

Application: Buy-at-Bulk Network Design



Input: Undirected graph $G = (V, E)$

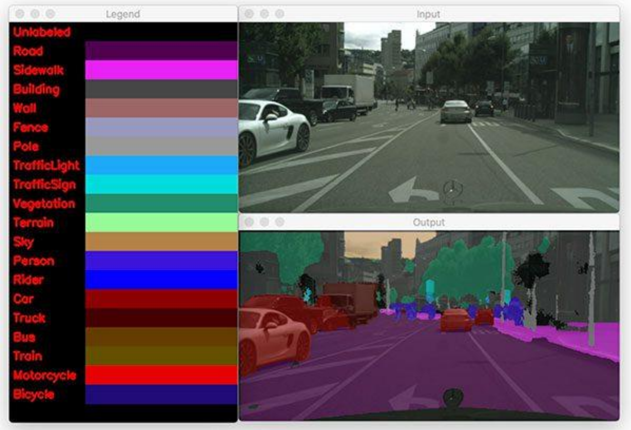
- Edge lengths $l : E \to \mathbb{R}$
- Demands: $b(s,t) \geq 0, \ \forall s,t \in V$
- For each edge $e \in E$: $f_e(x) > 0$
- $f_e(x)$ is subadditive: $f_e(x+y) \leq f_e(x) + f_e(y)$

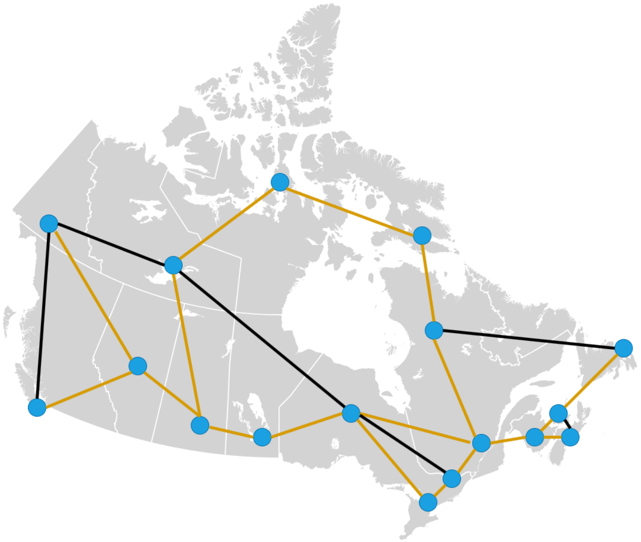
Output: (s,t)-path P_{st} $\forall s,t \in V$
Goal: minimize $\sum_{e \in E} l(e) f_e(u_e), \ u_w = \sum_{s,t: e \in p_{st}} b(s,t)$

Many metric-based problems

- Group Steiner Tree
- Metric Labeling
- Buy-at-Bulk Network Design
- Vehicle Routing
- Metrical Task System
- Min-Sum Clustering
- Distributed Computing
- K-Server Problem
- …

Such problems become easy with tree metrics.





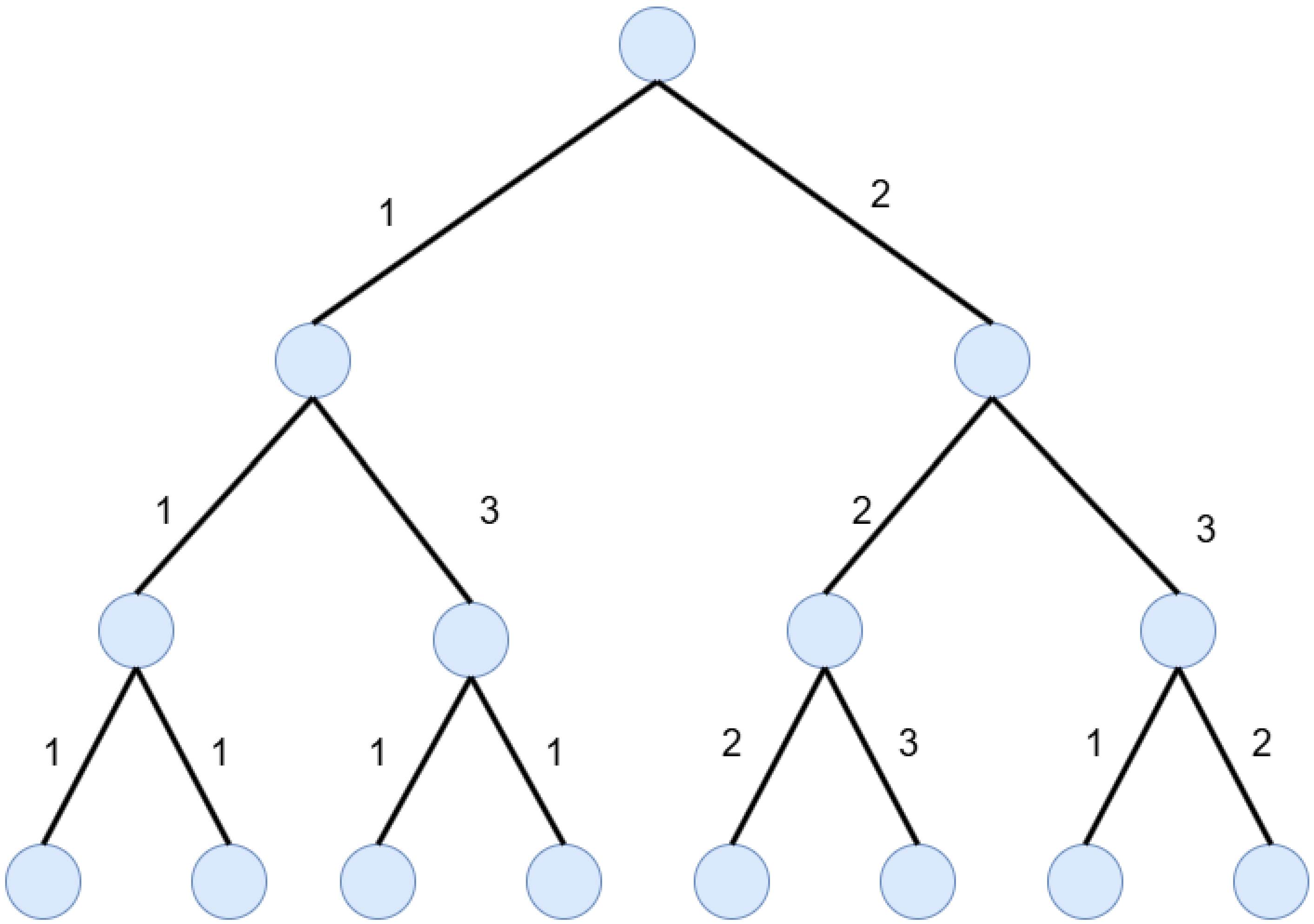
Tree Metrics

Shortest Paths Metric:

- O(mn) for general graphs

Trees have unique paths.

- Queries take O(logn) time
  - Least Common Ancestor
  - Path-to-root
  - Path Length
  - Path Sums



Approximation by Tree Metrics

Generally,
For an embedding $f : V \to V'$, the distortion is the minimal D such that:

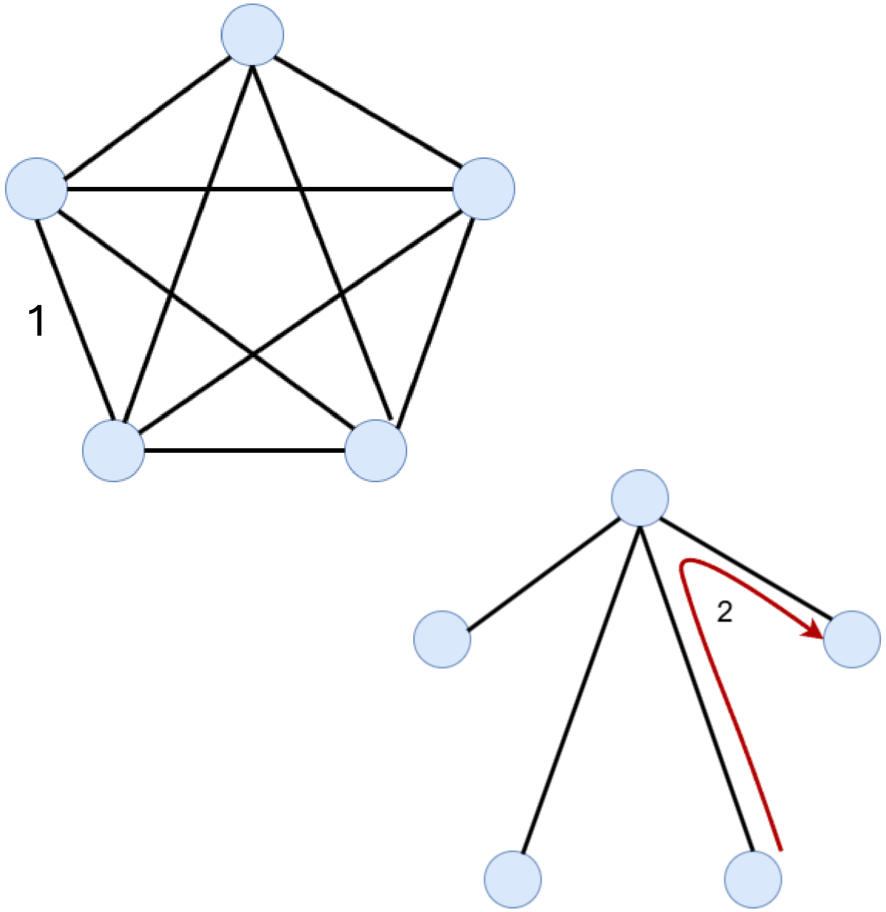
$$\forall u, v \in V, d(u, v) \leq d'(f(u), f(v)) \leq D \cdot d(u, v)$$

Input: Undirected graph $G = (V, E)$
Goal: Compute tree $T = (V, E')$ such that shortest paths on T are close to G

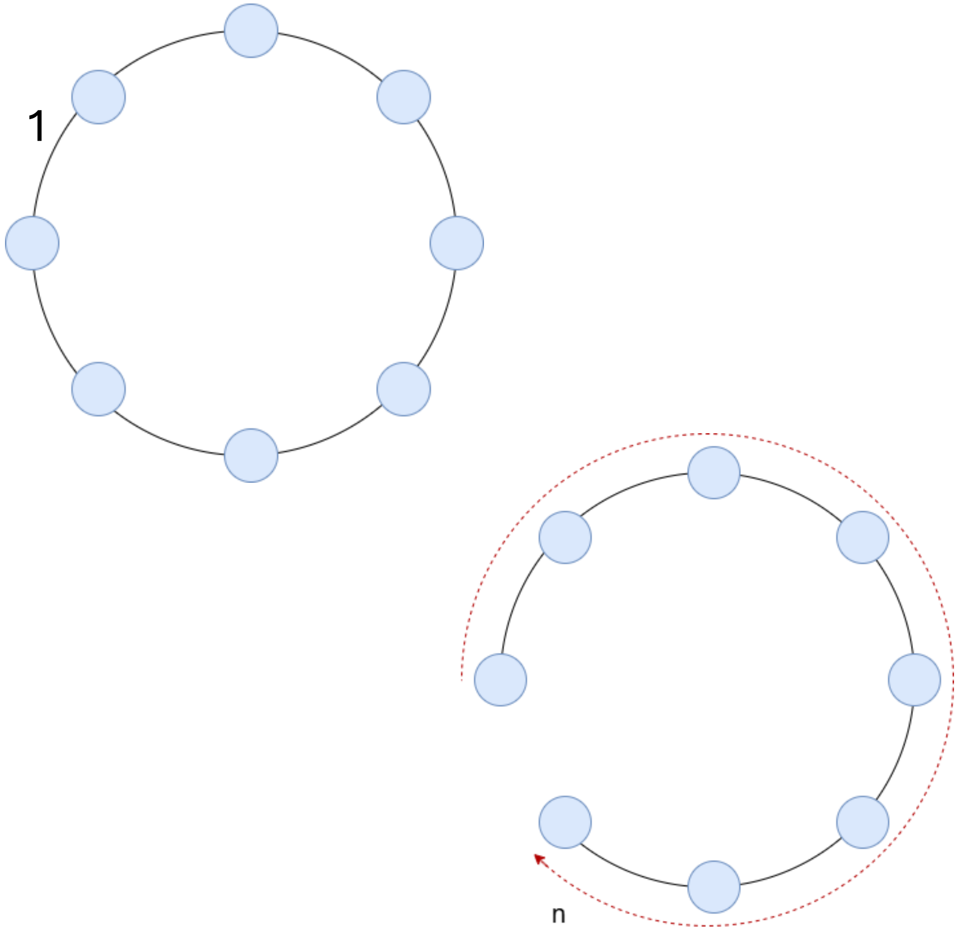
$$\text{stretch}(e) = \frac{d_T(u, v)}{d_G(u, v)} \text{ for edge } e \text{ between } u, v$$

Ideally, we want $\text{stretch}(e) = \text{polylog}(n)\ \forall e$

Naïve approach: Spanning Tree Metric



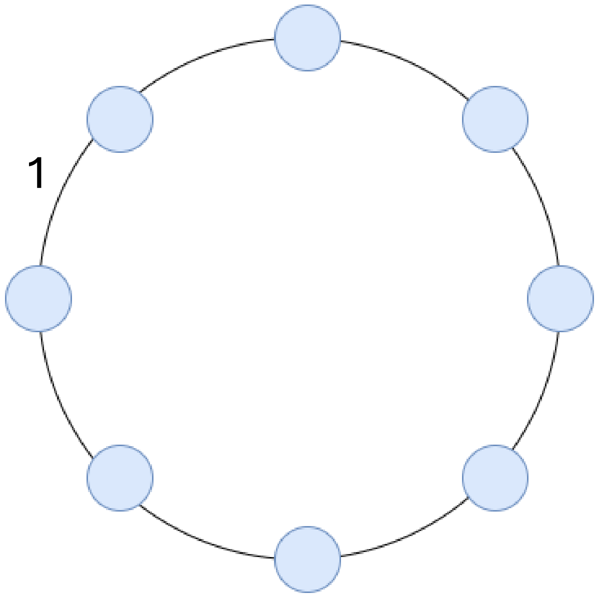
D=2, Not bad.



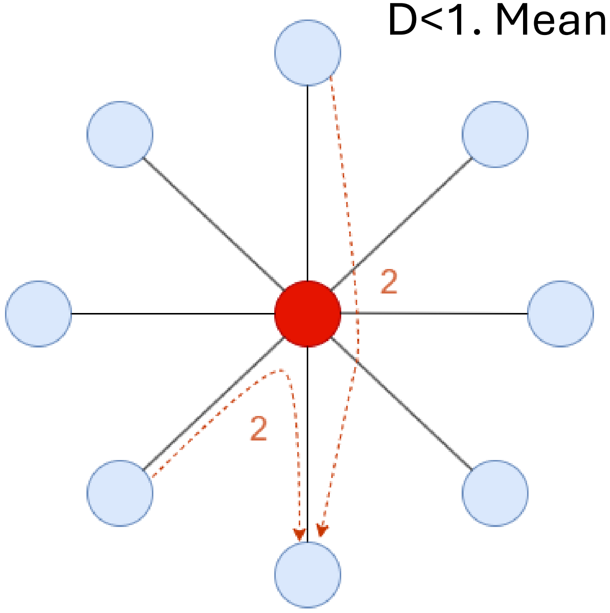
D=O(n). Terrible.

Auxiliary Tree Metric

- Auxiliary trees allow extra nodes.
  - “shortcuts”
- More flexible, but tends to compress distances.
  - Stretch calculations lose significance.

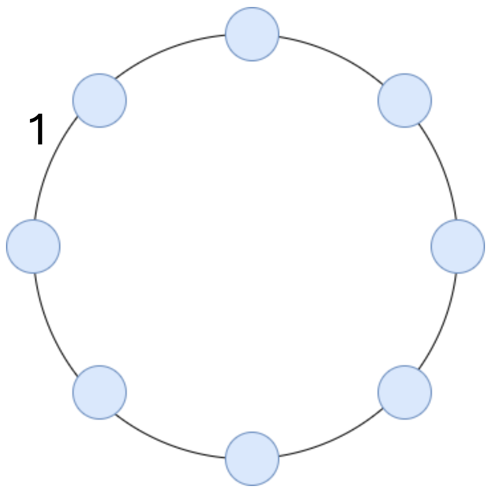


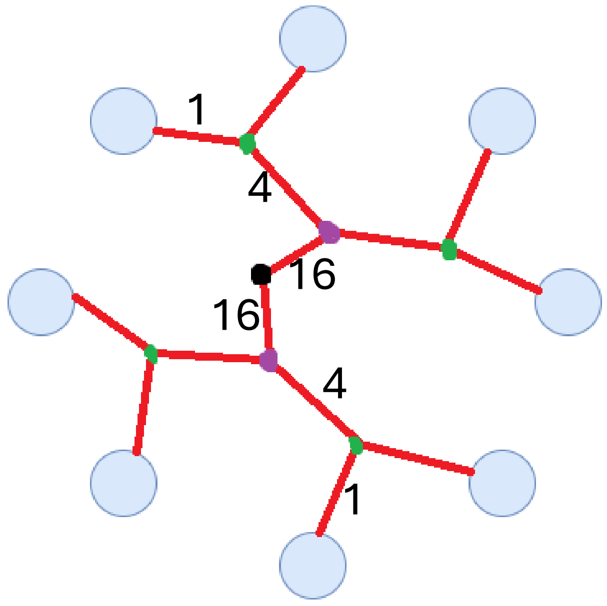
D<1. Meaningless.

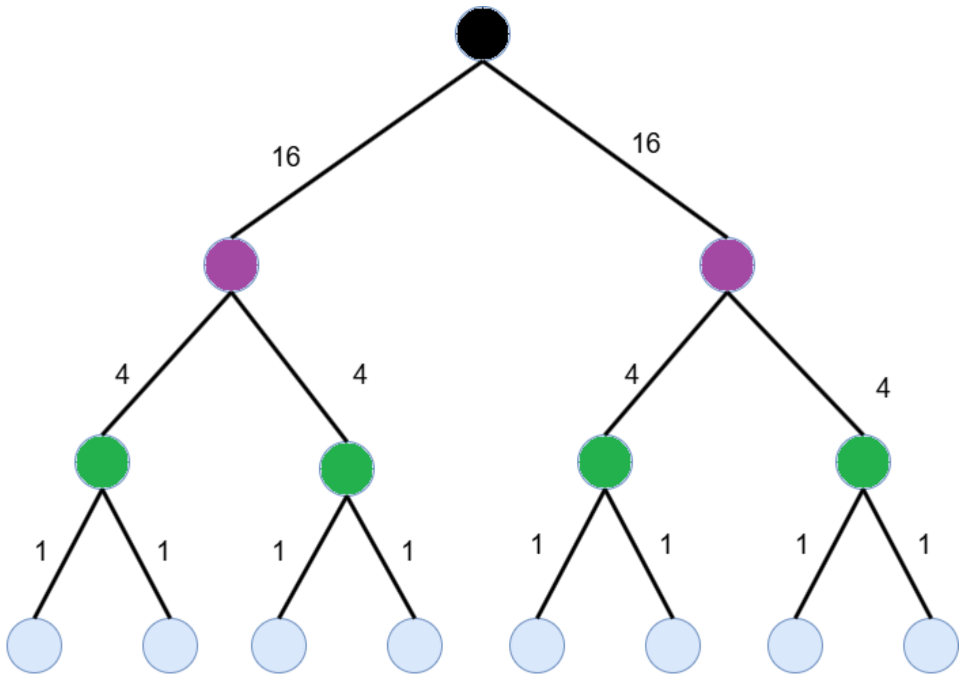


Hierarchal Tree Metric

Edge from height $i + 1$ to i has weight α^i for some value α.
Here, $\alpha = 4$





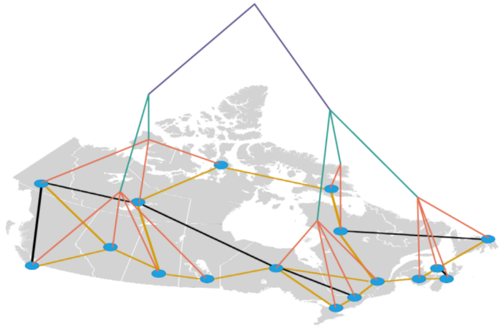


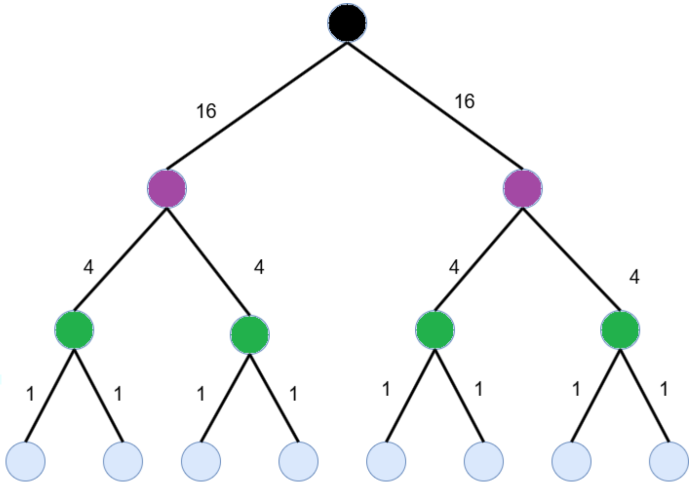
- Prevents compression, but edge cases inflate distortion bounds.
  - (e.g. cycles)
- Clever deterministic methods exist for low average stretch
- Key ingredient for further improvement: **Randomization**

Approximation by Tree Metrics

Input: Undirected graph $G = (V, E)$
Goal: Compute tree $T = (V, E')$ such that shortest paths on T are close to G

- Randomized dominating tree metric
- Introduced by Bartal in 1996, improved in 1998

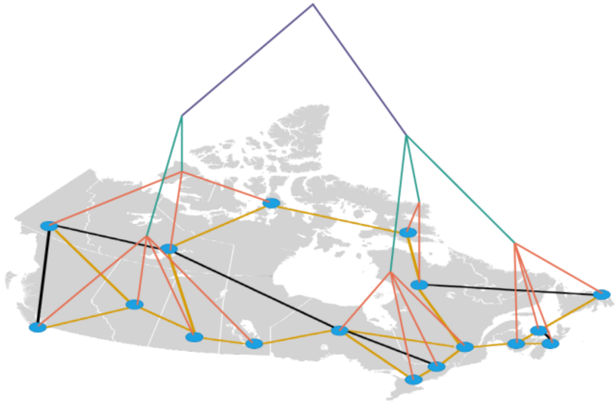




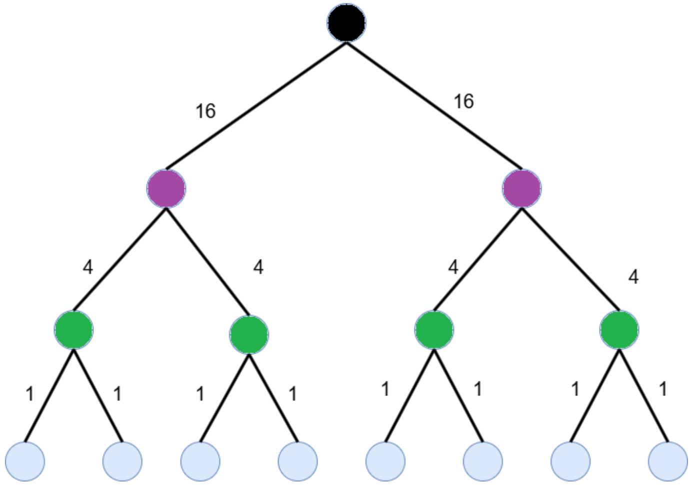
Construct auxiliary tree T with V as leaves such that:
T is **Dominating**: $d_T(u, v) \geq d_G(u, v) \ \ \forall u, v$ (no compression)
$E[\text{stretch}(e)] = O(\log n \log \log n) \forall e \in E$ (low stretch on average)

Tight O(logn) Bound

- In 2004, Fakcharoenphol, Rao, Talwar improved Bartal's stretch from O(lognloglogn) to O(logn)
- They demonstrated that O(logn) is the **optimal bound**
  - Better is impossible **unless P=NP**



Theorem: In randomized polynomial time:
We can construct a randomized hierarchal dominating tree metric such that:
$E[\text{stretch}(e)] = O(\log n) \forall e \in E$ (optimal stretch)



The Algorithm

Algorithm.*Partition*(*V*,*d*)

1. Choose a random permutation π of $v_1, v_2, \ldots, v_n$.
2. Choose β in $[1, 2]$ randomly from the distribution $p(x) = \frac{1}{x \ln 2}$.
3. $D_\delta \leftarrow V; i \leftarrow \delta - 1$.
4. while D_{i+1} has non-singleton clusters do

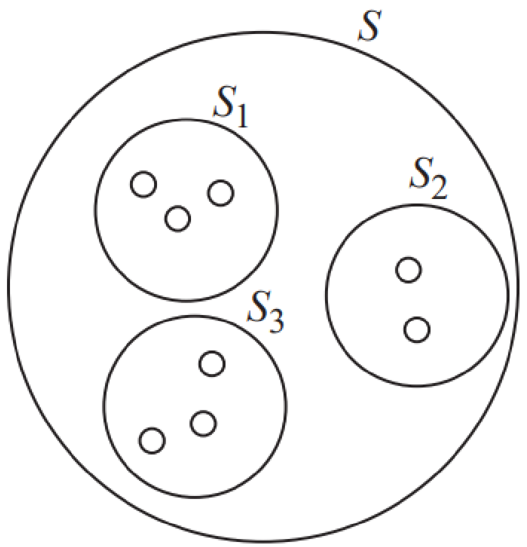
4.1 $\beta_i \leftarrow 2^{i-1}\beta$.

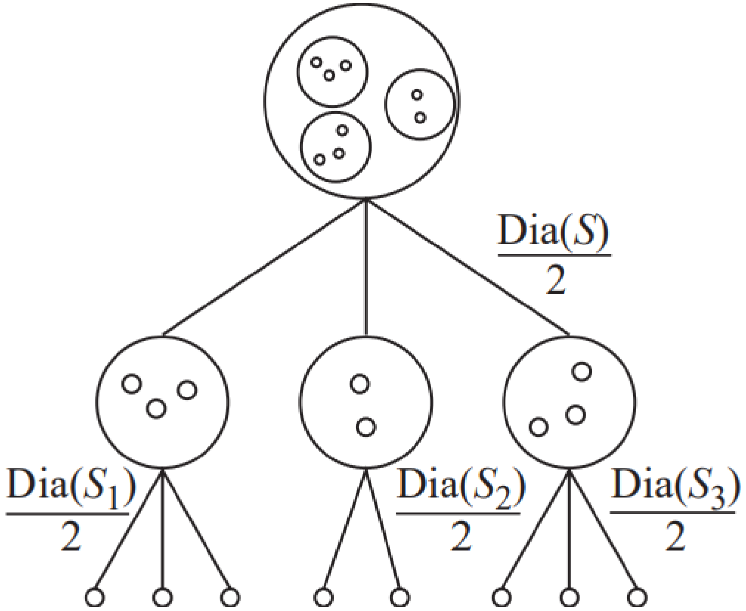
4.2 For $l = 1, 2, \ldots, n$ do

4.2.1 For every cluster S in D_{i+1}.

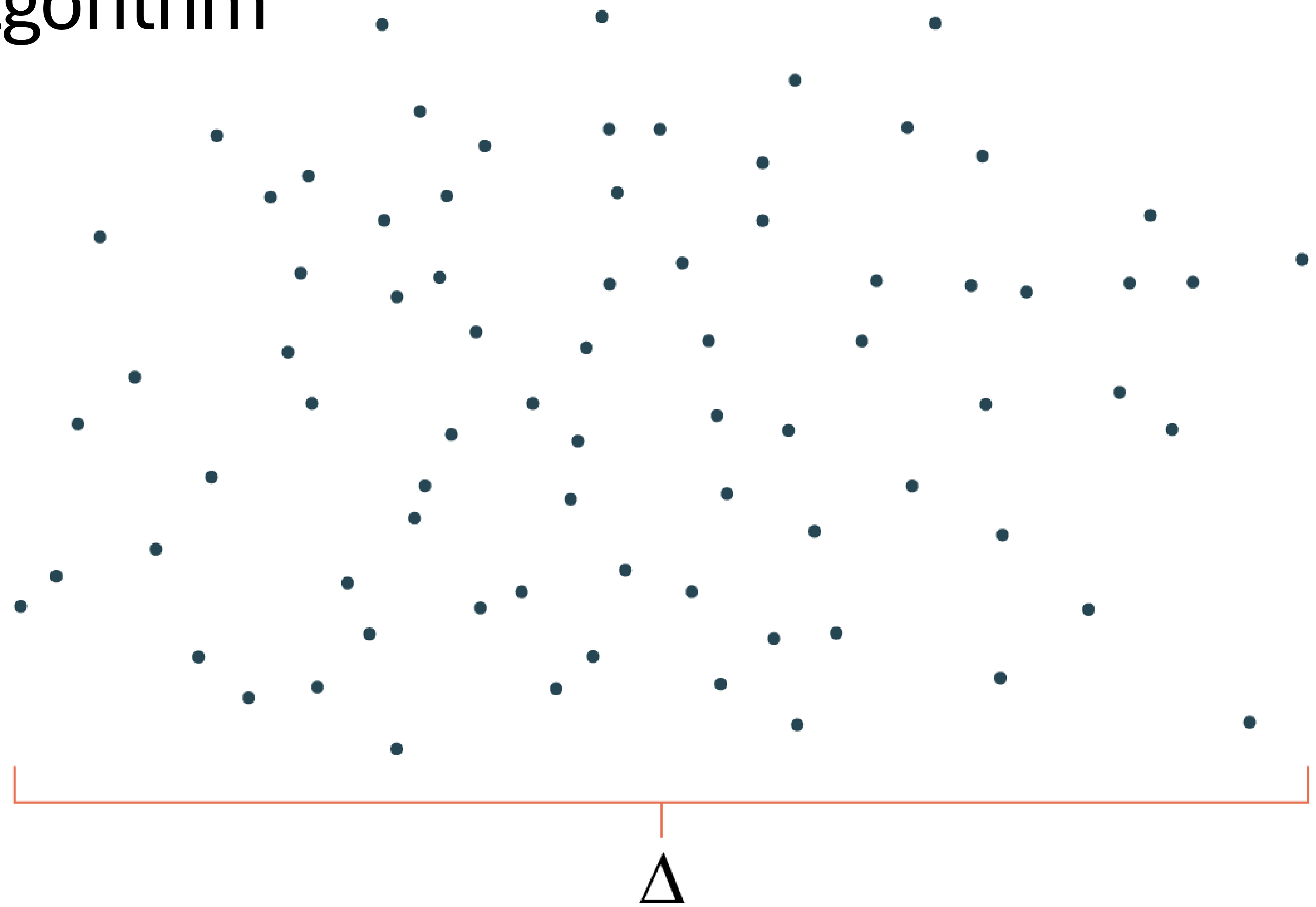
4.2.1.1 Create a new cluster consisting of all unassigned vertices in S closer than β_i to $\pi(l)$.

4.3 $i \leftarrow i - 1$.

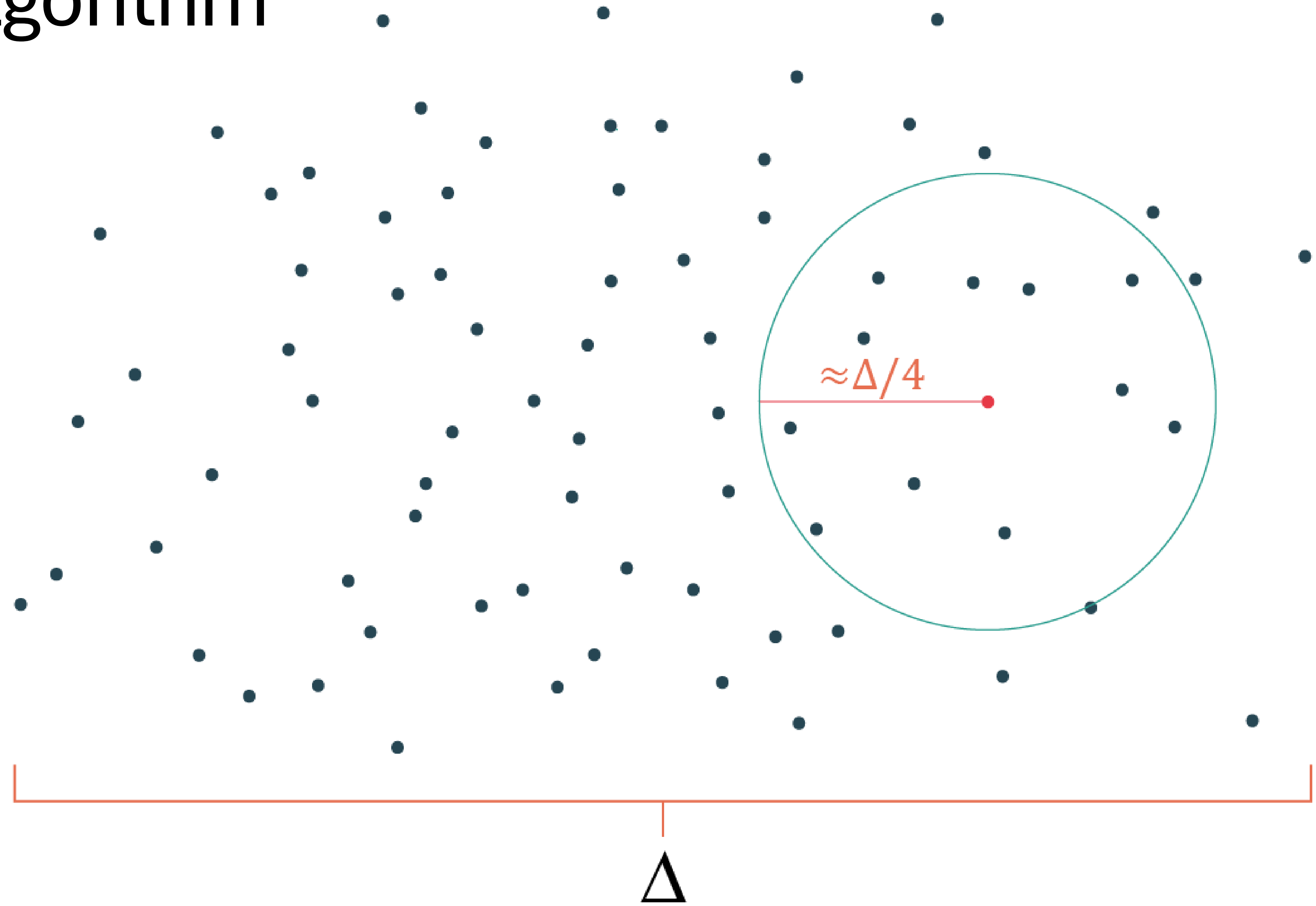




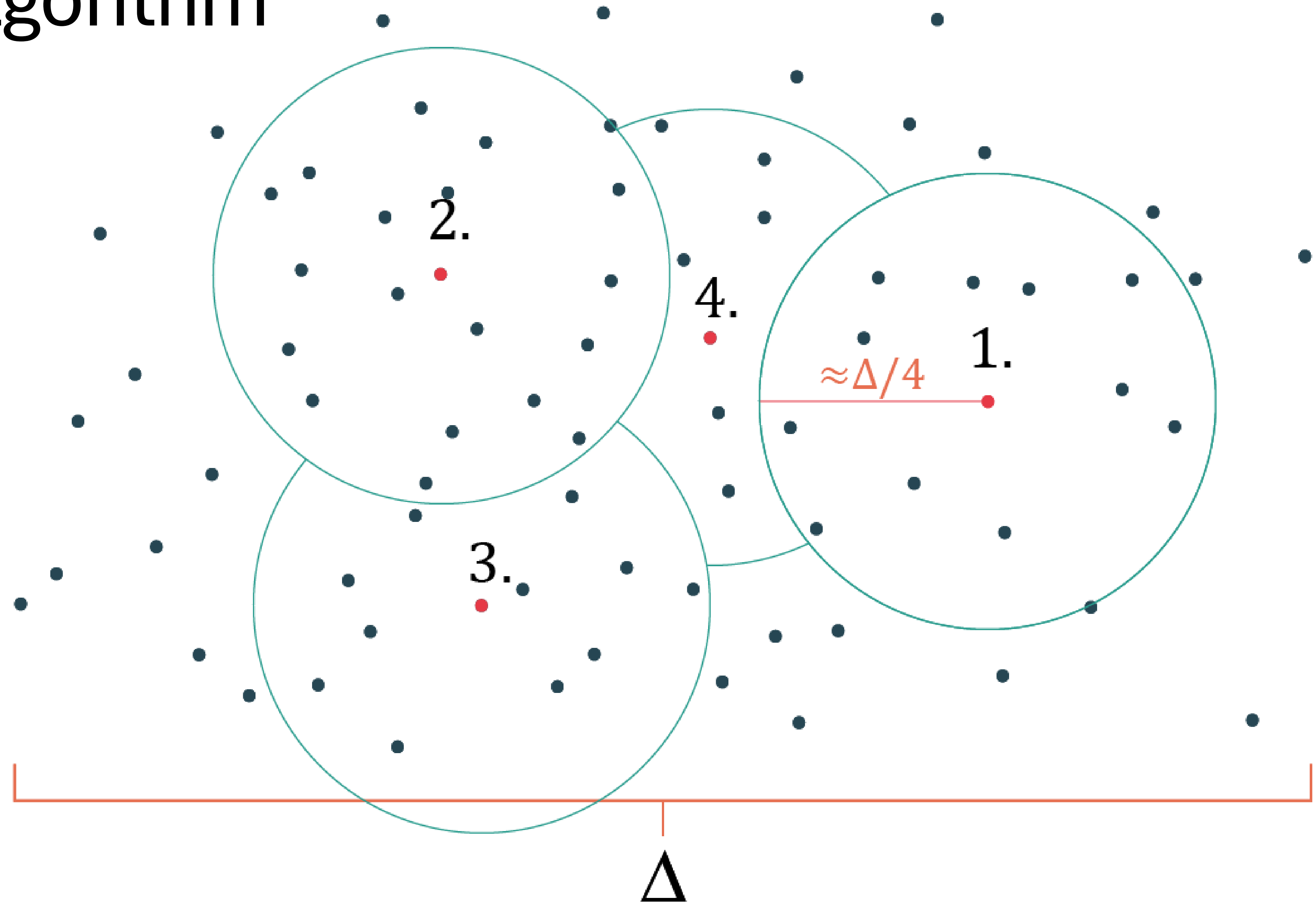
The Algorithm



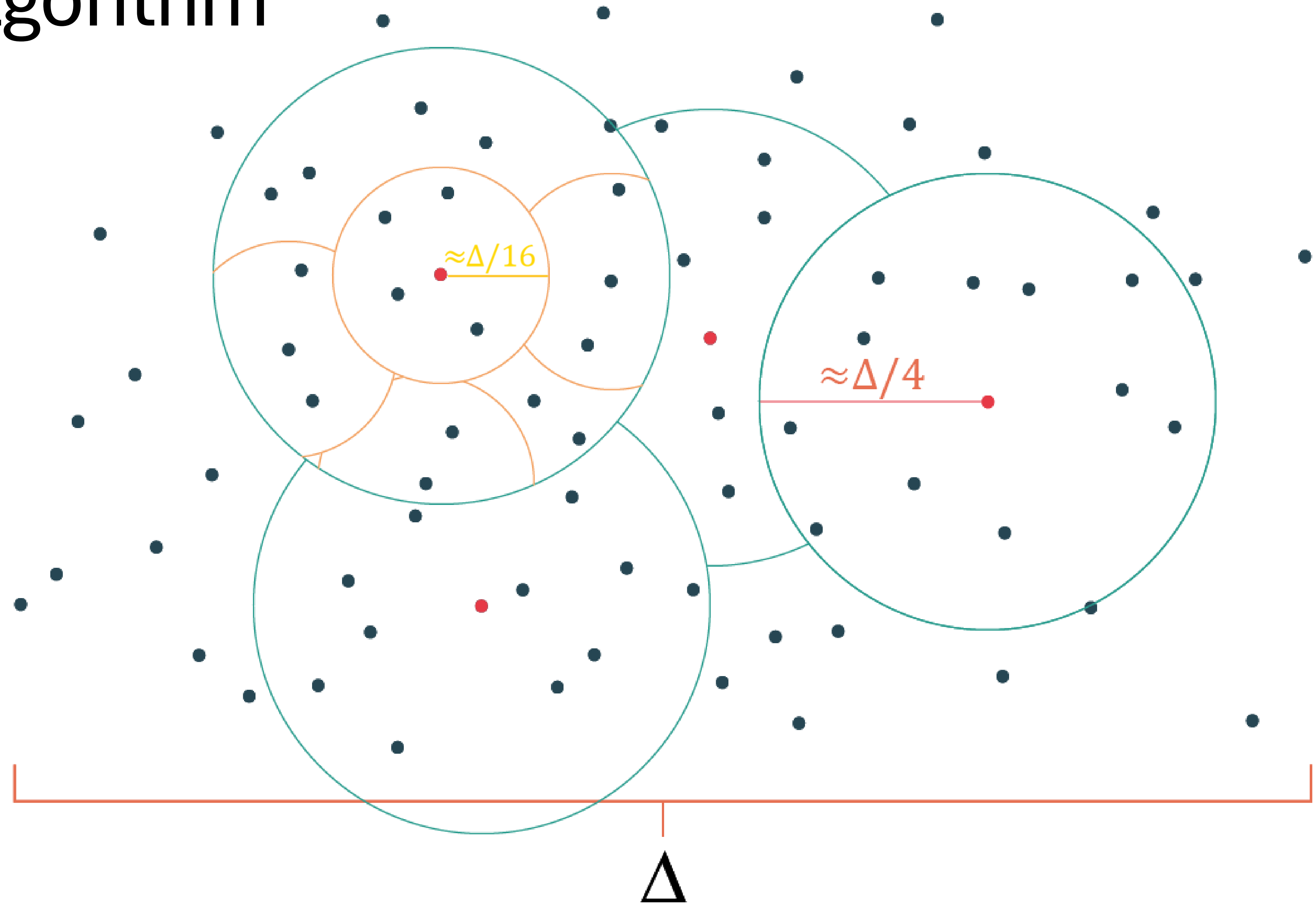
The Algorithm



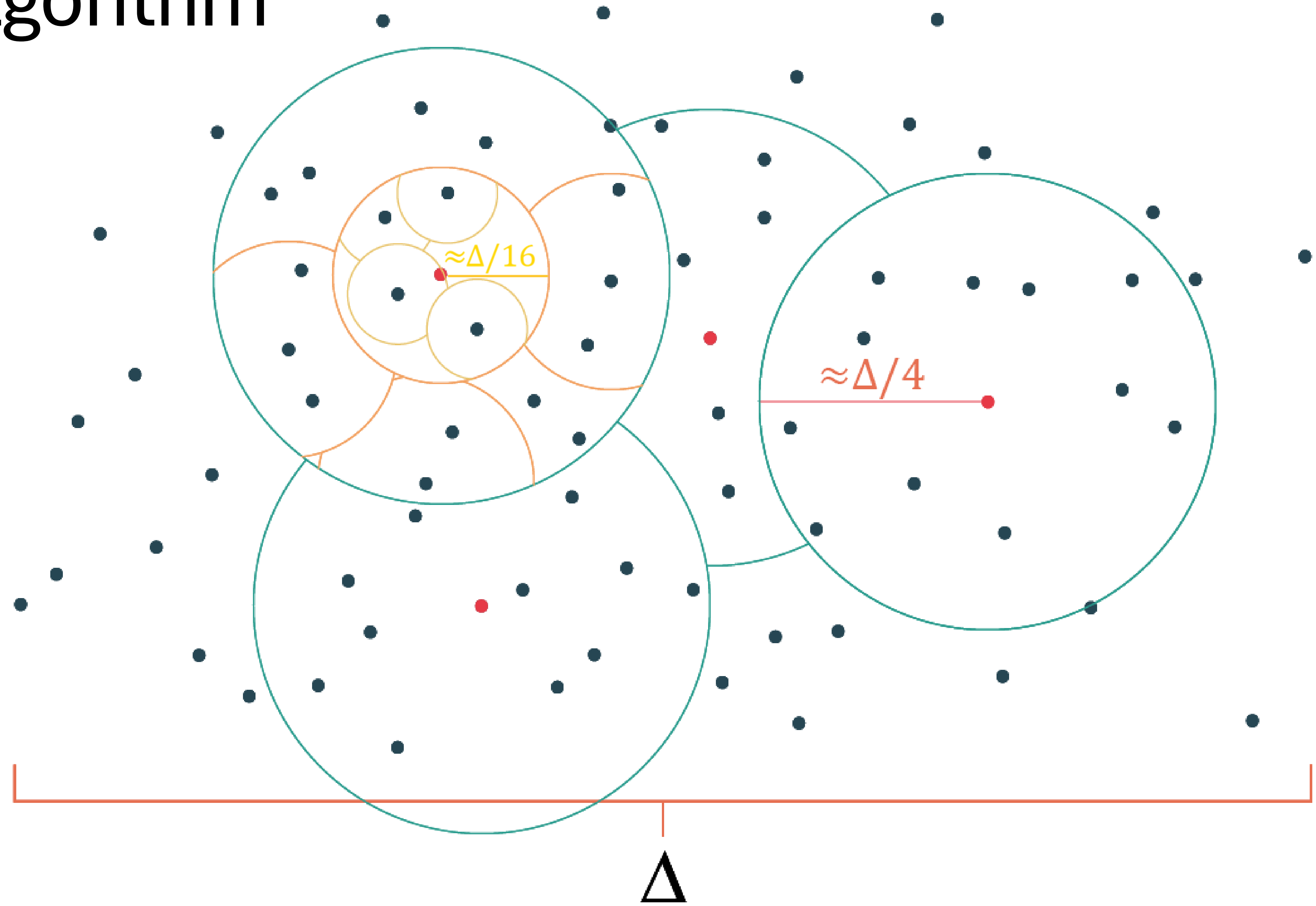
The Algorithm



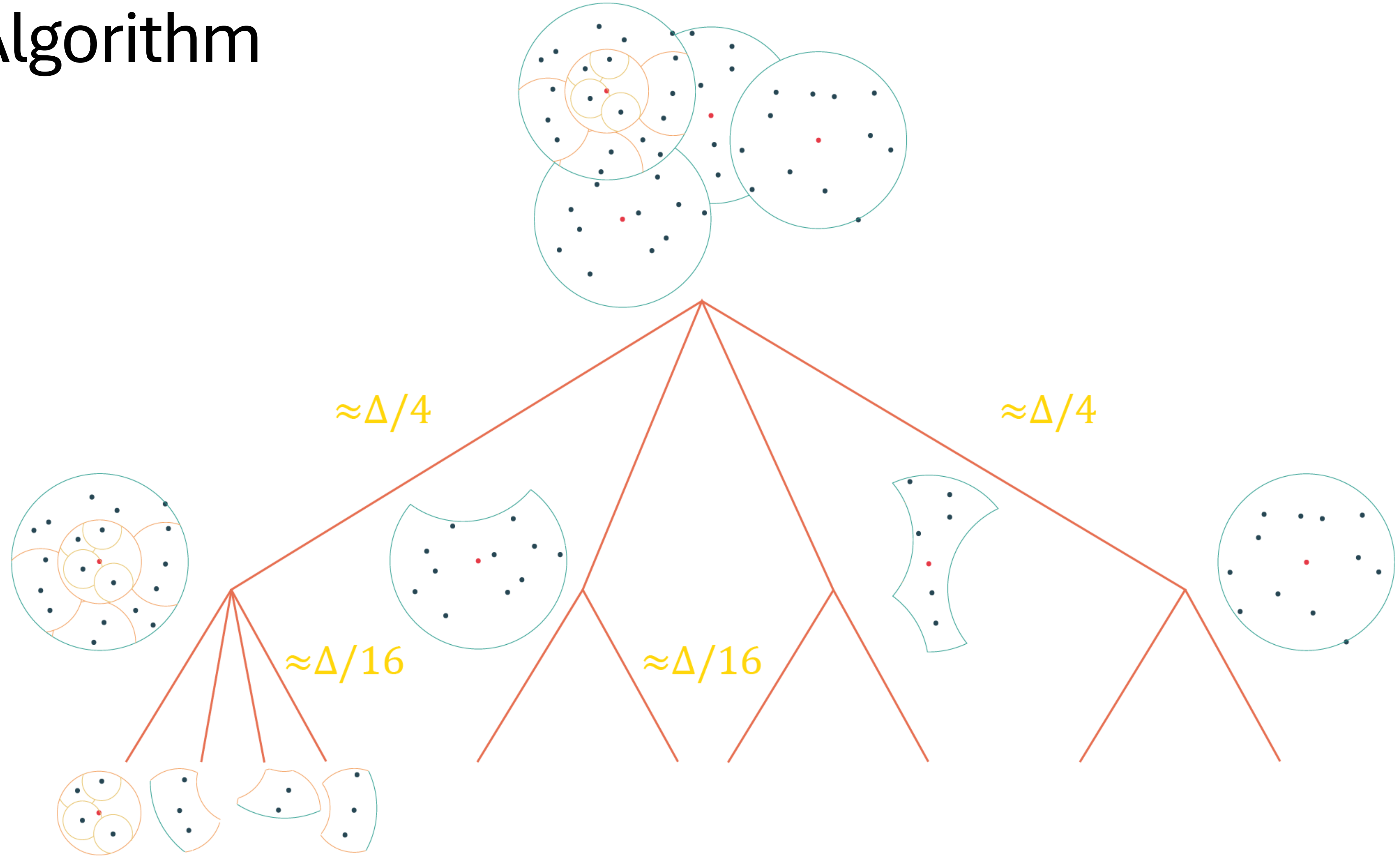
The Algorithm



The Algorithm



The Algorithm

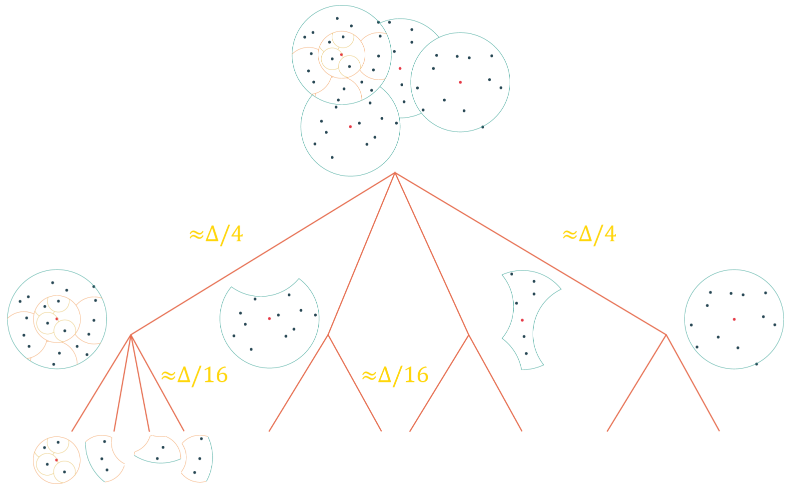


Analysis: Hierarchal Decomposition

We have constructed a hierarchical decomposition of the metric space to define the tree metric.

A *hierarchical decomposition* of (V, d) is a sequence of partitions $\{\mathcal{P}_i\}_{i \geq 0}$ such that:

1. $\mathcal{P}_0$ is the partition of V into singleton sets.
2. For each $i \geq 1$, $\mathcal{P}_i$ is a coarser partition than $\mathcal{P}_{i-1}$ (i.e., each part in $\mathcal{P}_{i-1}$ is contained in a part in $\mathcal{P}_i$).
3. There exists a sequence of scales $\{\Delta_i\}_{i \geq 0}$ with $\Delta_i = 2^i$, such that the diameter of each part in $\mathcal{P}_i$ is at most Δ_i.



Analysis: Properties of the Tree Metric

Lemma: The tree metric d_T defined by T dominates the original metric d; that is, for all $u, v \in V$, $d_T(u, v) \geq d(u, v)$.

Proof. Since clusters are formed by grouping points within a certain radius, the path in T from u to v must ascend to the lowest common ancestor (LCA) of u and v in T. The edge lengths are non-negative, and the cumulative length from u to the LCA and then to v is at least $d(u, v)$ because u and v are not in the same cluster at some level where the cluster diameter is less than $d(u, v)$. Therefore, $d_T(u, v) \geq d(u, v)$. □

Bounding the Expected Distance

Observation 1. For any $x \geq 1$,

$$\Pr[\text{some } b_i \text{ lies in } [x, x + dx)] = \frac{1}{x \ln 2}\, dx.$$

Fix an arbitrary edge (u, v) and show that the expected value of $d_T(u, v)$ is bounded by $O(\log n) \cdot d(u, v)$. Constants are not optimized in this analysis.

Clustering Step at Level i:

- In each iteration, all unassigned vertices v such that $d(v, p(l)) \leq b_i$ assign themselves to $p(l)$.
- For initial iterations, both u and v remain unassigned.
- At some step l, at least one of u or v gets assigned to center $p(l)$.

Edge Settlement and Cutting

Definitions:

- *Center w settles edge (u, v) at level i* if it is the first center to which at least one of u or v gets assigned.
- Exactly one center settles any edge (u, v) at any particular level.
- *Center w cuts edge $e = (u, v)$ at level i* if it settles e at this level, but exactly one of u or v is assigned to w at level i.

When w cuts edge (u, v) at level i, the tree length of the edge is about 2^{i+2}.

Defining Contribution

We attribute this length to vertex w and define:

$$d_T^w(u, v) = \sum_i \mathbf{1}[w \text{ cuts } (u, v) \text{ at level } i] \cdot 2^{i+2},$$

where $\mathbf{1}[\cdot]$ is the indicator function.

Clearly,

$$d_T(u, v) \leq \sum_w d_T^w(u, v).$$

Ordering Vertices

Arrange the vertices in V in order of increasing distance from edge (u, v) (breaking ties arbitrarily).

Consider the s-th vertex w_s in this sequence.
We will upper bound the expected value of $d_T^{w_s}(u, v)$ for an arbitrary w_s.

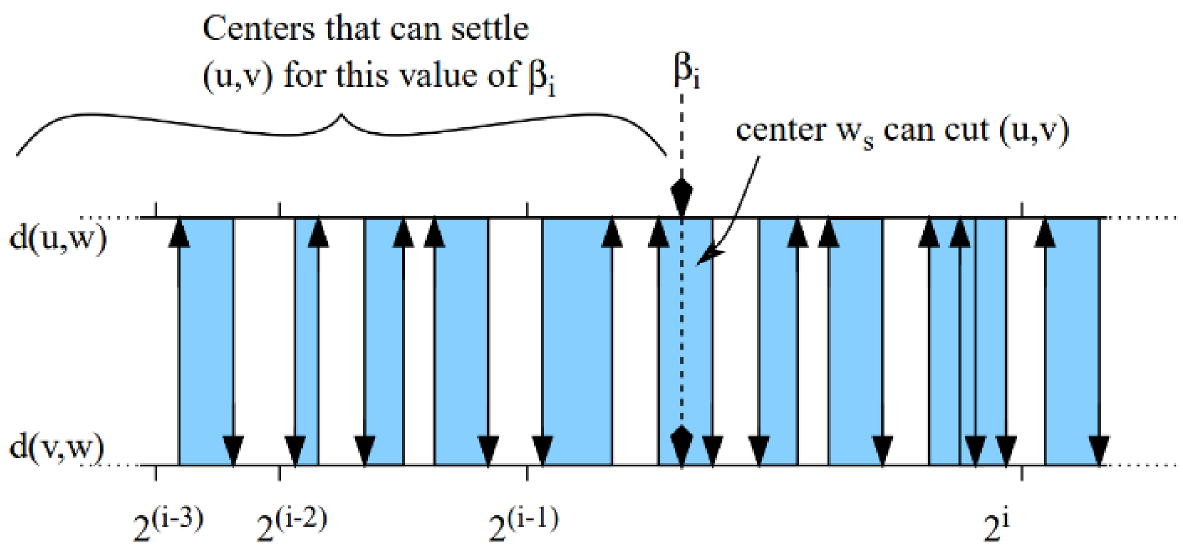
Conditions for Cutting Edge (u, v)

Without loss of generality, assume $d(w_s, u) \leq d(w_s, v)$.

For center w_s to cut (u, v), it must be that (see Figure):

(a) $d(w_s, u) \leq b_i \leq d(w_s, v)$ for some i.

(b) w_s settles edge e at level i.



Calculating the Expected Contribution

The contribution to $d_T^{w_s}(u, v)$ when this happens is at most $2^{i+2} \leq 8b_i$.

Consider a particular $x \in [d(w_s, u), d(w_s, v))$.

Probability Calculations:

- From Observation 1, the probability that some b_i lies in $[x, x + dx)$ is at most $\frac{1}{x \ln 2} dx$.
- Conditioned on $b_i = x$, any of $w_1, w_2, \ldots, w_s$ can settle (u, v) at level i.
- The first one among these in the permutation p will settle (u, v).
- Thus, the probability of event (b), conditioned on (a), is at most $\frac{1}{s}$.

Bounding the Expected Value

Expected Cost of $d_T^{w_s}(u, v)$:

$$
\begin{aligned}
\mathbb{E}[d_T^{w_s}(u, v)] &\leq \int_{d(w_s,u)}^{d(w_s,v)} \frac{1}{x \ln 2} \cdot 8x \cdot \frac{1}{s} \, dx \\
&= \frac{8}{s \ln 2} \int_{d(w_s,u)}^{d(w_s,v)} dx \\
&= \frac{8}{s \ln 2} (d(w_s, v) - d(w_s, u)) \\
&\leq \frac{8d(u, v)}{s \ln 2},
\end{aligned}
$$

where the last inequality follows from the triangle inequality.

Summing Over All Vertices

Using linearity of expectation, we get:

$$\mathbb{E}[d_T(u, v)] \leq \sum_{s=1}^{n} \frac{8d(u, v)}{s \ln 2} = \frac{8d(u, v)}{\ln 2} \cdot H_n,$$

where H_n is the n-th harmonic number.

Since $H_n \leq \ln n + 1$, we have:

$$\mathbb{E}[d_T(u, v)] \leq \frac{8d(u, v)}{\ln 2}(\ln n + 1) = O(\log n) \cdot d(u, v).$$

Conclusion

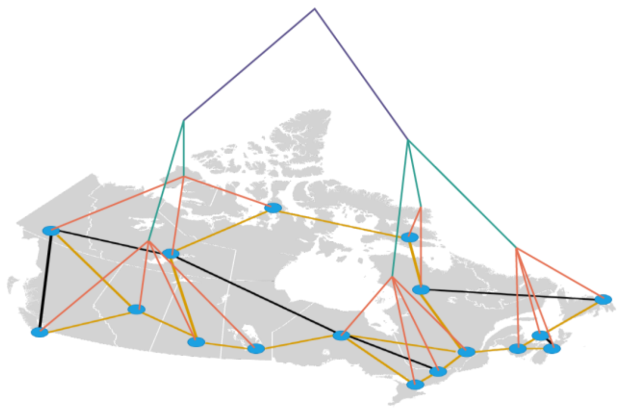
We have shown that for any edge (u, v), the expected value of $d_T(u, v)$ is $O(\log n) \cdot d(u, v)$.

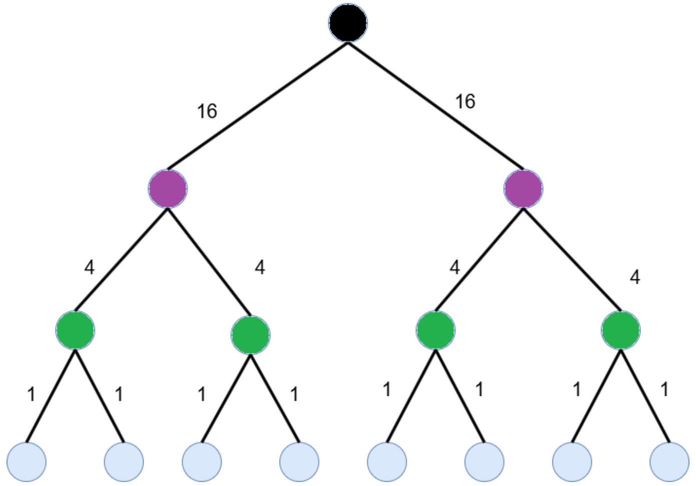
Hence, the expected distortion of the tree metric is $O(\log n)$.

With this result, approximation ratios for various problems are improved:

- Metric Labeling: $O(\log k \log \log k) \rightarrow O(\log k)$
- Earthmover LP: $O(\min(\log k, \log n))$
- Min. cost comm. network: $O(\log n)$
- Group Steiner Tree: $O(\lambda \log n \log k) \rightarrow O(\log^2 n \log k)$
- Metrical Task System: $O(\lambda \log n \log \log n) \rightarrow O(\log^2 n \log \log n)$

Where $\lambda = O(\min(\log n \log \log \log n, \log \Delta \log \log \Delta))$





References

[1] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer And System Sciences*, 69:485–497, 2004.

[2] A. Maheshwari. Notes on Algorithm Design. 2024.