Metric Techniques and Approximation Algorithms

Anupam Gupta
Carnegie Mellon University

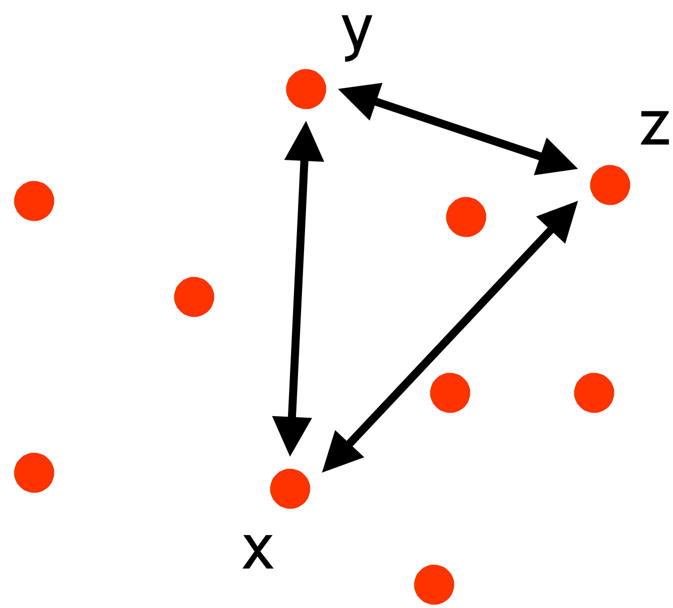
Metric space $M = (V, d)$

set V of points

distances $d(x,y)$

triangle inequality

$$d(x,y) \le d(x,z) + d(z,y)$$



why metric spaces?

Metric spaces are inputs to problems

- TSP
- round trip delays between machines
- distances between strings

but also,

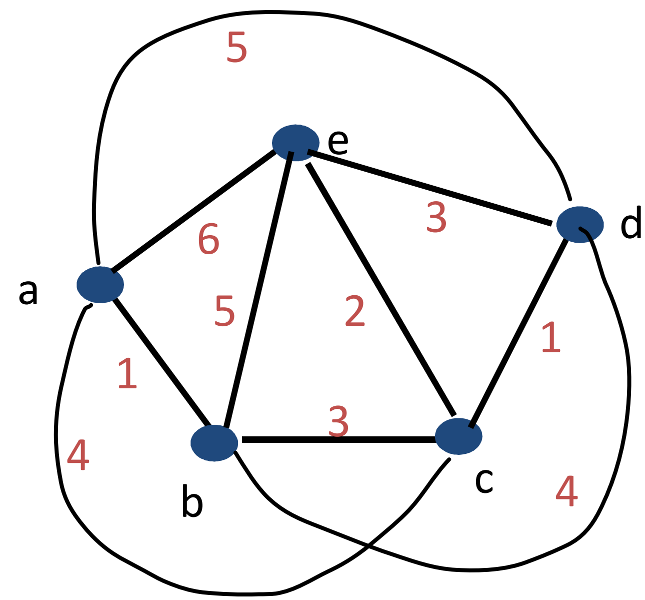
Metric spaces are useful abstractions for various problems

and interesting mathematical objects in their own right

Metrics

Distances d

$$\begin{pmatrix} 1 & 4 & 5 & 6 \\ & 3 & 4 & 5 \\ & & 1 & 2 \\ & & & 3 \end{pmatrix}$$



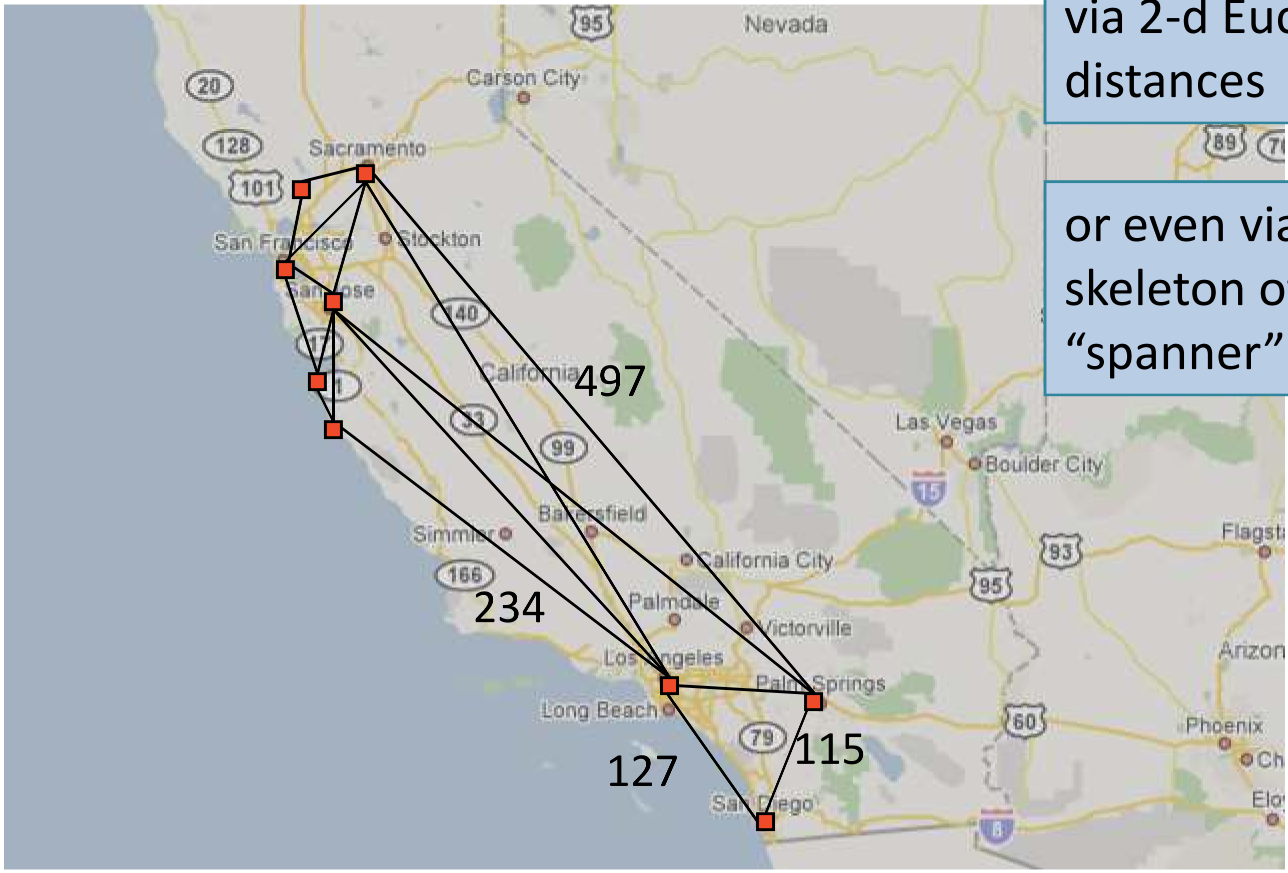
Choose your representation

	A	B	C	D	E	F	G	H	I
A	-	234	94	244	331	282	208	348	170
B	234	-	327	404	115	388	387	127	347
C	94	327	-	151	436	188	114	450	69
D	244	404	151	-	513	58	46	527	85
E	331	115	436	513	-	497	493	137	454
F	282	388	188	58	497	-	90	509	126
G	208	387	114	46	493	90	-	514	44
H	348	127	450	527	137	509	514	-	468
I	170	347	69	85	454	126	44	468	-

Choose your representation

	H.C.	LA	Mntr	Napa	PS	Sac	SF	SD	SJ
Hearst Castle	-	234	94	244	331	282	208	348	170
LA	234	-	327	404	115	388	387	127	347
Monterey	94	327	-	151	436	188	114	450	69
Napa	244	404	151	-	513	58	46	527	85
Palm Springs	331	115	436	513	-	497	493	137	454
Sacramento	282	388	188	58	497	-	90	509	126
San Francisco	208	387	114	46	493	90	-	514	44
San Diego	348	127	450	527	137	509	514	-	468
San Jose	170	347	69	85	454	126	44	468	-

Choose your representation



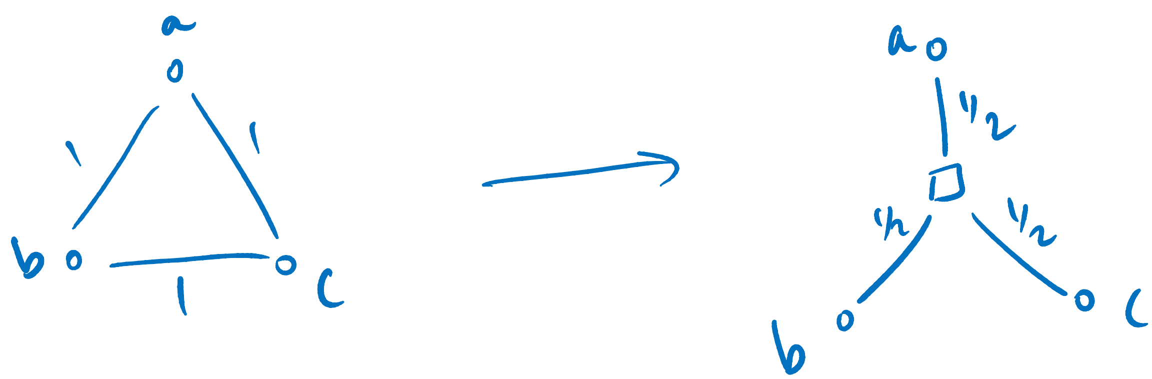
Good approximation via 2-d Euclidean distances

or even via a sparse skeleton of edges: a "spanner"

Representations

- Just a distance matrix
- Shortest-path metric of a (*simple*) graph
- Points in R^k with the ℓ_p metric
- low-dimensional geometric representations

Tree metric



Tree metric

A metric M = (V,d) is a *tree metric*

if there exists a tree $T = (V \cup X, E)$

(with edge-lengths)

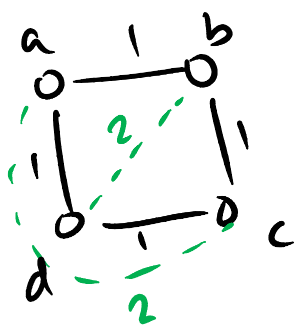
such that

$d = d_T |_{V \times V}$

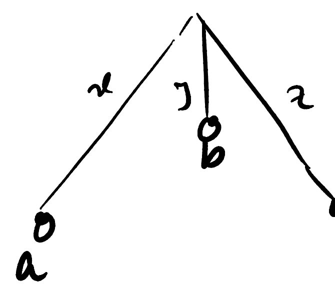
Is every metric a tree metric?

No:

Consider



C_4

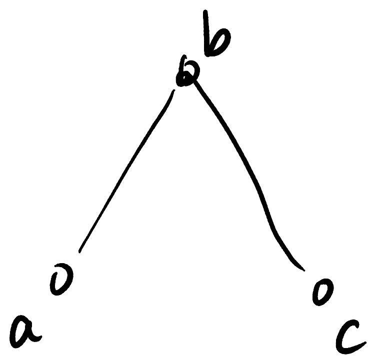


$x+y=1$

$y+z=1$

$x+z=2$

$\Rightarrow y=0$



Where would d go?

… "close" to a tree metric?

What is "closeness" between metric spaces?

$M = (V, d)$ $\qquad$ $M' = (V', d')$

$$f : V \rightarrow V'$$

$$\text{contraction}(f) = \max_{x,y \in V} \frac{d(x,y)}{d'(f(x), f(y))}$$

$$\text{expansion}(f) = \max_{x,y \in V} \frac{d'(f(x), f(y))}{d(x,y)}$$

$$\text{distortion}(f) = \text{contraction}(f) \times \text{expansion}(f)$$

Properties of **distortion**

- invariant under scaling

  ~~if~~

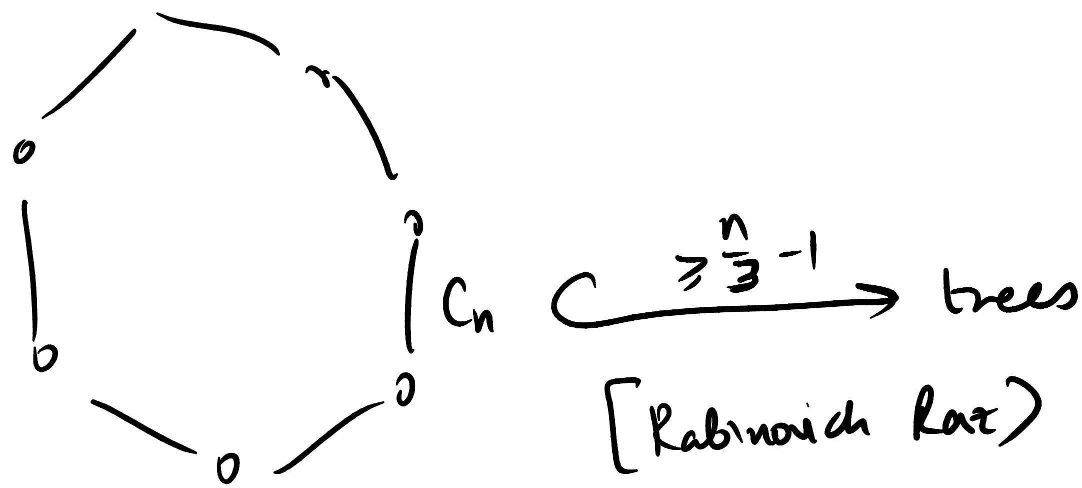
- <u>Notation</u>: $f: M \to M'$ has distortion D

  we write $M \xhookrightarrow{D} M'$

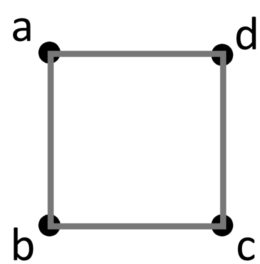
… close to a tree metric?

So, does every metric admit a low-distortion embedding into a tree metric?

No.



what do we do now?



Here's a solution... [Karp 84]

$$E[d_T(a,d)] = \frac{1}{n}\cdot(n-1) + \frac{n-1}{n}\cdot 1 = 2\left(1-\frac{1}{n}\right) \le 2\cdot d(a,d)$$

“dominating trees”

Given a metric M = (V,d)

let $\mathcal{T}$ = { tree T | $d_T \geq d$ }

distances in T “dominate” distances in d

random tree embeddings

Given $M = (V, d)$

want a probability distribution $\mathcal{D}$ over $\mathcal{T}(M)$

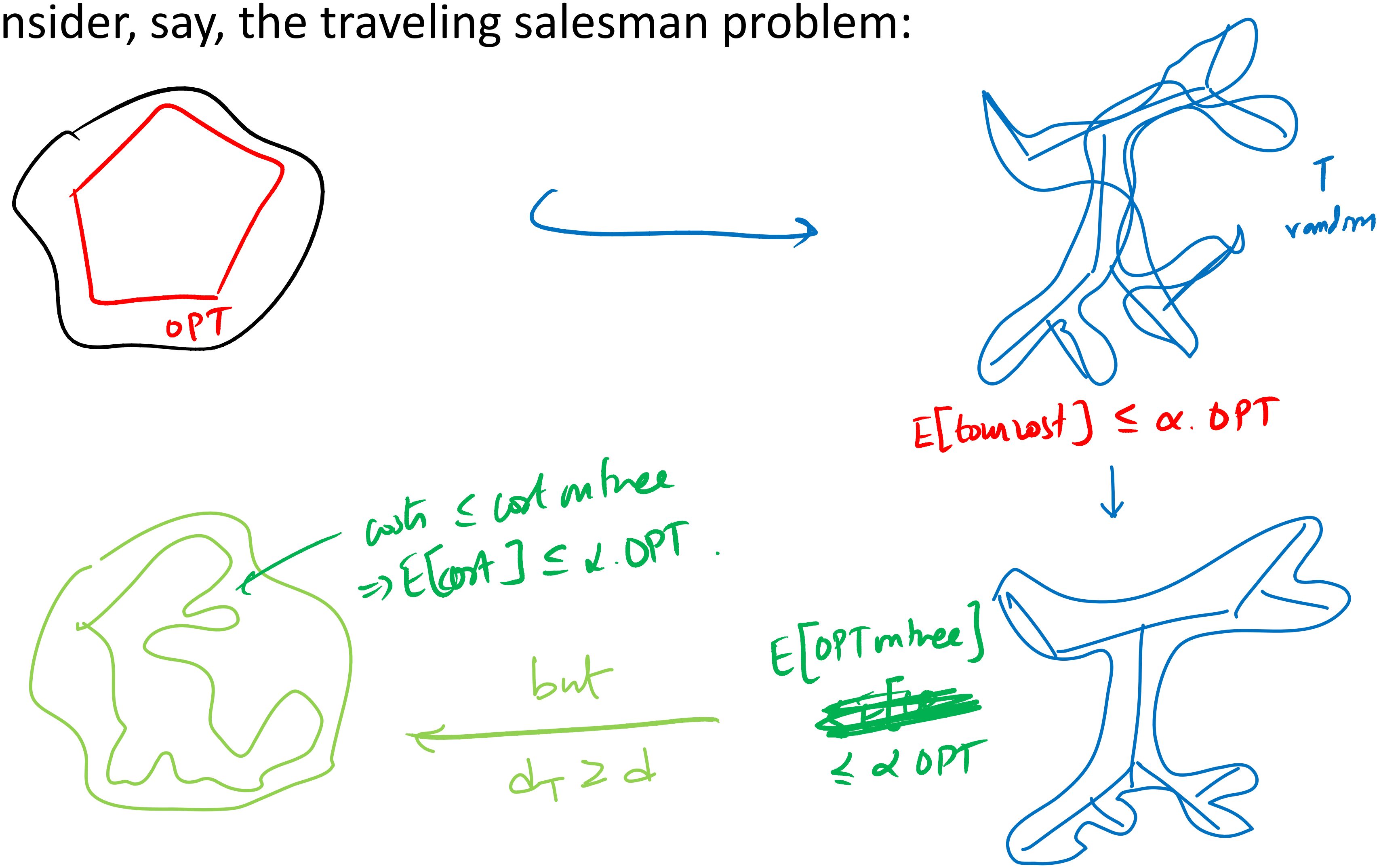
$$\left(\Rightarrow \sum_T \mathcal{D}(T) = 1\right)$$

st $\forall (x, y) \in V \times V$

$$E_{T \leftarrow \mathcal{D}} \left[d_T(x, y) \right] \leq \alpha \cdot d(x, y).$$

why are these useful?

Consider, say, the traveling salesman problem:



quick recap of goals

Given a metric $M = (V,d)$
find a distribution $\mathcal{D}$ over trees such that

1. $d(x,y) \leq d_T(x,y)$ for all trees in $\mathcal{D}$

2. $E_T[\, d_T(x,y) \,] \leq \alpha \times d(x,y)$

first results

[Alon Karp Peleg West '94]

$2^{O(\sqrt{\log n \log\log n})}$

[Bartal '96, '98]

$O(\log^2 n)$

$O(\log n \log\log n)$

current world record

[Fakcharoenphol Rao Talwar ’03]

$O(\log n)$

best possible

$\Omega(\log n)$ lower bound for

- square grid
- hypercube
- diamond graphs

Time for some proofs…

useful notation

Given a metric M = (V, d)

- Diameter(S) for S $\subseteq$ V
- Ball B(x,r) $= \{ y \in V \mid d(x,y) \leq r \}$

"padded" decompositions

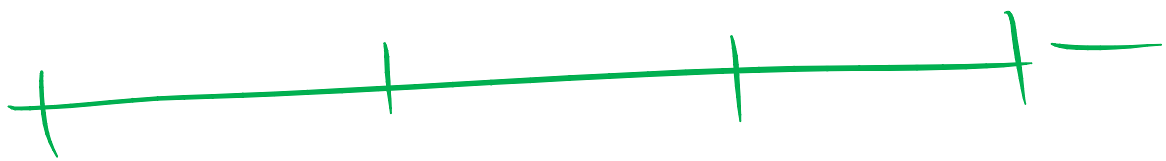
A metric (V,d) admits β-padded decompositions, if for every Δ, we can output a random partition

$$V = V_1 \uplus V_2 \uplus \ldots \uplus V_k$$

1. each V_j has diameter ≤ Δ
2. Pr[x and y in different clusters] ≤ $\frac{d(x,y)}{\Delta} \cdot \beta$

2'. Pr[ball B(x,ρ) split] ≤ $\frac{\rho}{\Delta} \cdot \beta$

why this expression?



on line, best prob of x, y separated $\approx \frac{d(x,y)}{\Delta}$

this is β fraction worse

“padded” decompositions

A metric (V,d) admits β-padded decompositions, if for every Δ, we can output a random partition

$$V = V_1 \uplus V_2 \uplus \ldots \uplus V_k$$

1. each V_j has diameter $\leq \Delta$
2. Pr[B(x,ρ) split] $\leq \frac{\rho}{\Delta} \cdot \beta$

(weaker) theorems

Theorem 1.

Every n-point metric admits an

β = O(log n)-padded decomposition

Theorem 2.

Embedding into distribution over trees with

α = O(log n × log diameter)

assume min-distance = 1

(stronger) theorems

Theorem 3.

Every n-point metric admits an $\beta(x, \Delta)$-padded decomposition with $\beta = \log \frac{|B(x, \Delta)|}{|B(x, \Delta/8)|}$

Theorem 4.

Embedding into distribution over trees with

$\alpha = O(\log n)$

we'll prove the weaker results

Theorem 1.

Every n-point metric admits an

β = O(log n)-padded decomposition

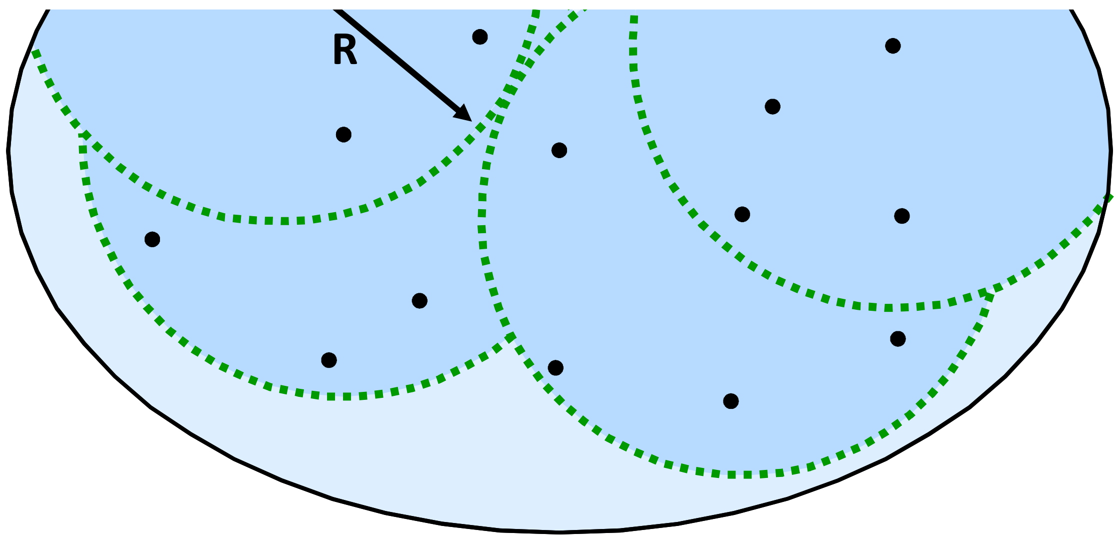
Theorem 2.

Embedding into distribution over trees with

α = O(log n log diameter)

decomposition algorithm

Given a metric M = (V,d) and a parameter Δ



decomposition algorithm

Given a metric M = (V,d) and a parameter Δ

1. Pick a random permutation π on V.
2. Pick a random radius R uniformly from the interval $(\Delta/4, \Delta/2]$.
3. Create a "cluster" C_v for each $v \in V$: assign $x \in V$ to C_v if v is the *first* vertex (according to π) such that $d(v, x) \leq R$.
4. Output all the non-empty clusters C_v.

1. each V_j has diameter ≤ Δ
2. Pr[B(x,ρ) split] ≤ $\frac{\rho}{\Delta} \cdot O(\log n)$

Now to show

Theorem 1.

Every n-point metric admits an

β = O(log n)-padded decomposition

Theorem 2.

Embedding into distribution over trees with

α = O(log n log diameter)

tree-building

Procedure FRT(X, i) (Invariant: $X \subseteq V$, diameter$(X) \leq 2^i$.)

tree-building

Procedure FRT(X, i) (Invariant: $X \subseteq V$, diameter$(X) \leq 2^i$.)

1. If $|V| = 1$, return X.
2. Use β-padded decomposition procedure on X with diameter bound 2^{i-1} to get random partition $X_1, X_2, \ldots, X_k$.
3. For each j, recursively call FRT$(X_j, i-1)$ to get tree T_j with root v_j.
4. For each $j \geq 2$, attach edges (r_1, r_j) of length 2^i to get connected tree T.
5. Return resulting tree T with root $r = r_1$.

Initially call with ***FRT(V, log(diameter))***

1. Distances in tree are at least d(x,y)
2. E[distance(x,y) in tree] ≤ O(log n log diam) d(x,y)

have seen

Theorem 1.

Every n-point metric admits an

β = O(log n)-padded decomposition

Theorem 2.

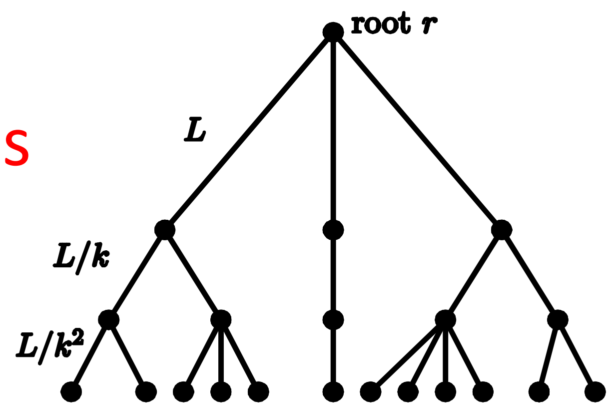
Embedding into distribution over trees with

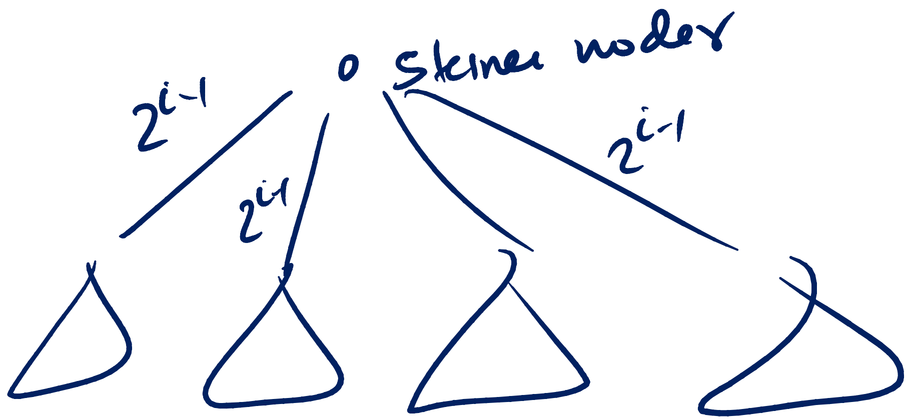
α = O(log n log diameter)

extensions

- embedding into
  Hierarchically well-**S**eparated **T**ree**s**

(HSTs)





extensions

- embedding special classes of graphs

Outerplanar graphs

extensions

- embedding graph metrics into
  distributions over their sub-trees

$2^{\sqrt{\log n \log\log n}}$ [AKPW]

$O(\log^2 n \log\log n)$ [EEST]

$O(\log n \text{ poly} \log\log n)$ [ABN08]

Padded decompositions

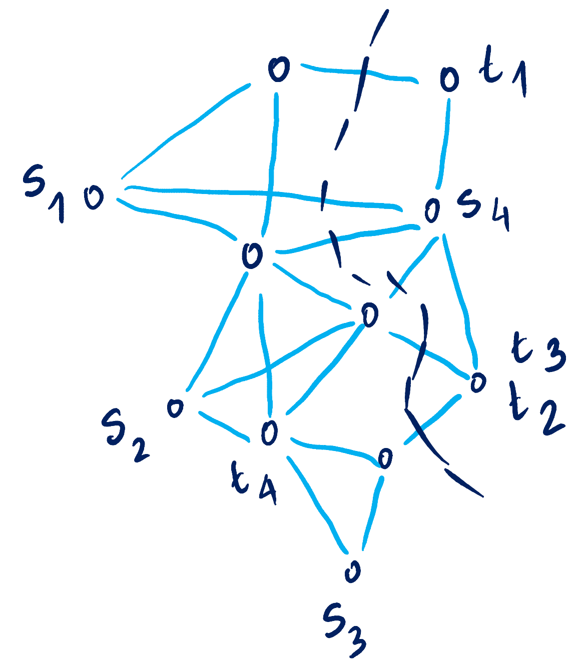
padded decomps very useful

- We'll see applications to other embeddings later
- applications to finding neighborhood covers in exercises
- here's an application to another approximation algorithm

multi-cut

Given graph G = (V,E)

with k source-sink pairs



Find the fewest edges to delete

to separate all source-sink pairs

NP-hard, APX-hard for k ≥ 3.

Best known: O(log k) approximation

[Garg Vazirani Yannakakis]

relaxation of multi-cut

Suppose we want lengths on edges

such that shortest-path-distance(s_p, t_p) ≥ 1 for all p.

One possible setting:

length of cut edges in OPT = 1, all others = 0

total length = OPT.

So, find (fractional) setting that *minimizes total length*

⇒ at most OPT.

and can be found by linear programming.

algorithm idea

Given such fractional edge-lengths
(with total length $L \leq OPT$)

Use these lengths to figure out which edges to cut

and E[number of edges cut] $\leq O(\log n) \times L$
$\leq O(\log n) \times OPT$

$\Rightarrow$ we'd have a logarithmic approximation !

randomized algorithm for multi-cut

Given lengths on edges

shortest-path-distance(s_p, t_p) ≥ 1 for all p.

Take a O(log n)-padded decomposition of this metric with Δ = 1/3.

↑ even 1-ε would do!

Facts:

1. Each terminal pair separated.
2. Pr[edge e cut] ≤ length(e) × O(log n)

(1/3)

$$E[\#\text{ edges cut}] \le \frac{\sum \text{len}(e)}{1/3} O(\log n) \le OPT \cdot O(\log n)$$

embeddings into trees

used for these problems

k-median
Group Steiner tree
min-sum clustering
metric labeling
minimum communication spanning tree
vehicle routing problems
metrical task systems and k-server
buy-at-bulk network design
oblivious network design
oblivious routing
demand-cut problem
…

app: oblivious routing

We've seen: given a metric, output a random tree

maintains distances to within expected O(log n) factor

[Räcke 08]

Given an undirected flow network G with edge-capacities

output a random tree T (with edge-capacities)

any multicommodity flow in G routable in T exactly.

any flow in T (almost) routable in G

(edge-capacities exceeded by expected O(log n) factor.)

app: "universal" TSP

Given a metric (V,d),

you find single permutation π

adversary gives you subset S $\subseteq$ V

you use order given by π to visit cities in S.

How close are you to the optimal tour on S?

If adversary does not look at actual permutation π when choosing S $\Rightarrow$ O(log n) factor worse in expectation.

What if adversary can look at π and then choose S?

can use variant of tree embeddings to get $O(\log^2 n)$

open: randomized k-server problem

Given a HST

have k servers located at some nodes

Requests come one-by-one at nodes

must move one of the servers to the requested node

Minimize total movement

Deterministic algorithm: k-“competitive” (and this is tight!)

Better randomized algorithm: ????

[Cote Meyerson Poplawski]: O(log diameter)-competitive on *binary* HSTs

recap…

Two general techniques:

Padded decompositions.

Embeddings into random trees.

Coming up:

Embeddings into geometric space

Dimension reduction and other dimensionality issues