Outline of the talk

Metric Embeddings into Trees

Arnold Filtser

Bar-Ilan University

July 02, 2024

Metric space

A metric space is an ordered pair (X, d_X), where X is a set and $d_X : X \times X \to \mathbb{R}_{\geq 0}$ is a function such that:

1. **Identity**: $\forall x, y \in X,\ d_X(x, y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x, y) = d_X(y, x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x, y) \leq d_X(x, z) + d_X(z, y)$.

Metric space

A metric space is an ordered pair (X, d_X), where X is a set and $d_X : X \times X \to \mathbb{R}_{\geq 0}$ is a function such that:

1. **Identity**: $\forall x, y \in X$, $d_X(x, y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X$, $d_X(x, y) = d_X(y, x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X$, $d_X(x, y) \leq d_X(x, z) + d_X(z, y)$.

Examples:

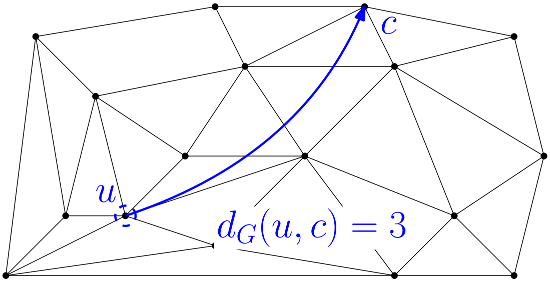
- Weighted graph $G = (V, E, w)$ with shortest path distance.

Metric space

A metric space is an ordered pair (X, d_X), where X is a set and $d_X : X \times X \to \mathbb{R}_{\geq 0}$ is a function such that:

1. **Identity**: $\forall x, y \in X,\ d_X(x, y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x, y) = d_X(y, x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x, y) \leq d_X(x, z) + d_X(z, y)$.

- Weighted graph $G = (V, E, w)$ with shortest path distance.



Metric space

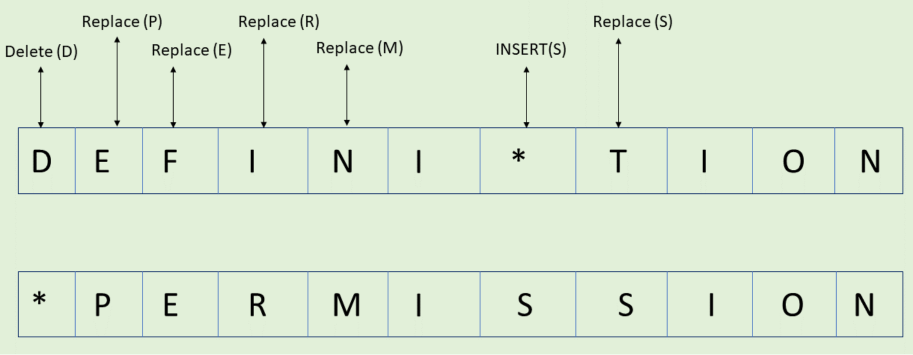
A metric space is an ordered pair (X, d_X), where X is a set and $d_X : X \times X \to \mathbb{R}_{\geq 0}$ is a function such that:

1. **Identity**: $\forall x, y \in X,\ d_X(x, y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x, y) = d_X(y, x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x, y) \leq d_X(x, z) + d_X(z, y)$.

- Weighted graph $G = (V, E, w)$ with shortest path distance.
- Edit distance: given two strings A, B how many edit operations (insert, delete, substitute) are required to transform A to B?

1. **Identity**: $\forall x, y \in X,\ d_X(x,y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x,y) = d_X(y,x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x,y) \leq d_X(x,z) + d_X(z,y)$.

- Weighted graph $G = (V, E, w)$ with shortest path distance.
- Edit distance: given two strings A, B how many edit operations (insert, delete, substitute) are required to transform A to B?



DEFINITION
EFINITION
PFINITION
PRINITION
PRRNITION
PRRMI-TION
PRRMISTION
PRRMISSION

1. **Identity**: $\forall x, y \in X,\ d_X(x,y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x,y) = d_X(y,x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x,y) \leq d_X(x,z) + d_X(z,y)$.

- Weighted graph $G = (V, E, w)$ with shortest path distance.
- Edit distance: given two strings A, B how many edit operations (insert, delete, substitute) are required to transform A to B?
- Manhattan distance ℓ_1 in $\mathbb{R}^d$: $d_{\ell_1}(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_1 = \sum_{i=1}^{d} |x_i - y_i|$.

1. **Identity**: $\forall x, y \in X,\ d_X(x, y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x, y) = d_X(y, x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x, y) \leq d_X(x, z) + d_X(z, y)$.

- Weighted graph $G = (V, E, w)$ with shortest path distance.
- Edit distance: given two strings A, B how many edit operations (insert, delete, substitute) are required to transform A to B?
- Manhattan distance ℓ_1 in $\mathbb{R}^d$: $d_{\ell_1}(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_1 = \sum_{i=1}^{d} |x_i - y_i|$.

$$\left\| \begin{pmatrix} 5 \\ 8 \\ -3 \\ 4 \\ 1 \end{pmatrix} - \begin{pmatrix} 2 \\ 10 \\ 1 \\ 1 \\ 3 \end{pmatrix} \right\|_1 = \underbrace{|5-1|}_{4} + \underbrace{|8-10|}_{2} + \underbrace{|(-3)-1|}_{4} + \underbrace{|4-1|}_{3} + \underbrace{|1-3|}_{2} = 17$$

1. **Identity**: $\forall x, y \in X,\ d_X(x,y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x,y) = d_X(y,x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x,y) \leq d_X(x,z) + d_X(z,y)$.

- Weighted graph $G = (V, E, w)$ with shortest path distance.
- Edit distance: given two strings A, B how many edit operations (insert, delete, substitute) are required to transform A to B?
- Manhattan distance ℓ_1 in $\mathbb{R}^d$: $d_{\ell_1}(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_1 = \sum_{i=1}^{d} |x_i - y_i|$.
- Euclidean space ℓ_2 in $\mathbb{R}^d$: $d_{\ell_2}(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_2 = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$.

$$\left\| \begin{pmatrix} 5 \\ 8 \\ -3 \\ 4 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 10 \\ 1 \\ 1 \\ 3 \end{pmatrix} \right\|_2 = \sqrt{\underbrace{|5-1|^2}_{16} + \underbrace{|8-10|^2}_{4} + \underbrace{|(-3)-1|^2}_{16} + \underbrace{|4-1|^2}_{9} + \underbrace{|1-3|^2}_{4}} = 7$$

Metric space

1. **Identity**: $\forall x, y \in X,\ d_X(x, y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x, y) = d_X(y, x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x, y) \leq d_X(x, z) + d_X(z, y)$.

Many problems are defined w.r.t. metric spaces. Examples:

- Metric TSP.
- k-center.
- Steiner tree.

Metric space

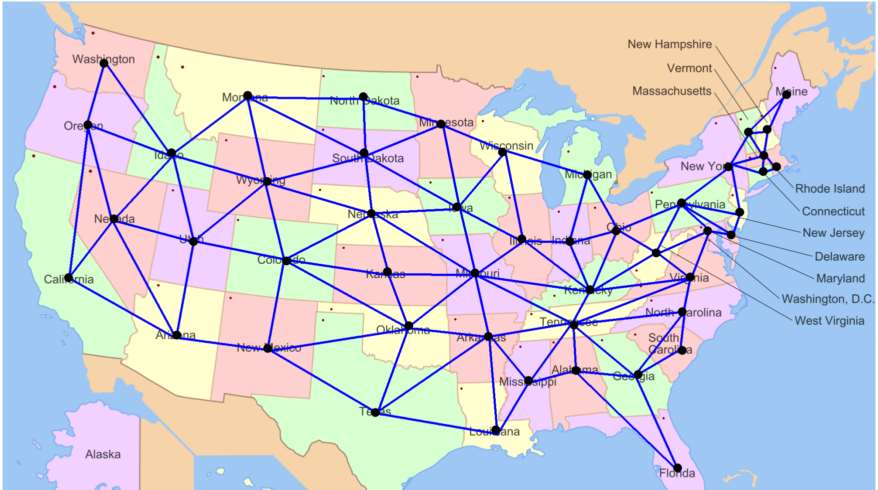
1. **Identity**: $\forall x, y \in X,\ d_X(x, y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x, y) = d_X(y, x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x, y) \leq d_X(x, z) + d_X(z, y)$.

Many problems are defined w.r.t. metric spaces. Examples:

- **Metric TSP**
- k-center.
- Steiner tree.

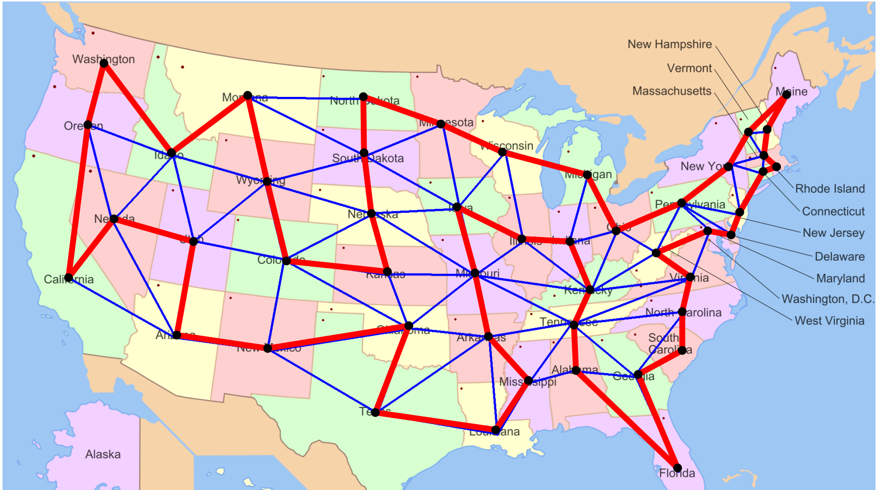
Definition (Travelling salesman problem (TSP))

Given a metric space (X, d_X) find a permutation $x_0, x_1, \ldots, x_{n-1}$ of the points in X minimizing $\sum_{i=0}^{n-1} d_X(x_i, x_{i+1})$ (i.e. a cycle of minimum weight).



Definition (Travelling salesman problem (TSP))

Given a metric space (X, d_X) find a permutation $x_0, x_1, \ldots, x_{n-1}$ of the points in X minimizing $\sum_{i=0}^{n-1} d_X(x_i, x_{i+1})$ (i.e. a cycle of minimum weight).



Metric space

1. **Identity**: $\forall x, y \in X,\ d_X(x, y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x, y) = d_X(y, x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x, y) \leq d_X(x, z) + d_X(z, y)$.

Many problems are defined w.r.t. metric spaces. Examples:

- **Metric TSP**
- k-center.
- Steiner tree.

Often these problems are NP-hard.

Metric space

1. **Identity**: $\forall x, y \in X,\ d_X(x, y) = 0 \iff x = y$.
2. **Symmetry**: $\forall x, y \in X,\ d_X(x, y) = d_X(y, x)$.
3. **Triangle-inequality:** $\forall x, y, z \in X,\ d_X(x, y) \leq d_X(x, z) + d_X(z, y)$.

Many problems are defined w.r.t. metric spaces. Examples:

- **Metric TSP**
- k-center.
- Steiner tree.

Often these problems are NP-hard.

NP-hard: a large class of equivalent problems (i.e. if you solved one-you solved all) for which we don't know of any efficient algorithms. It is generally believed that there are no efficient algorithms for these problems.

Theorem (Karp's list of 21 problems [Karp72])

The following problems are NP-Complete:

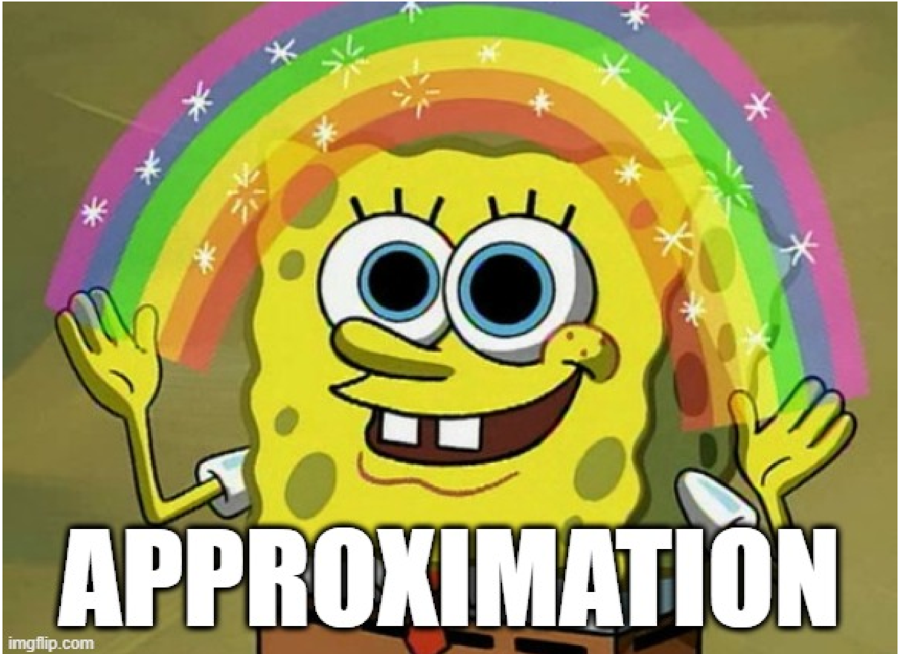
1. *SAT*
2. *0–1 integer programming*
3. *Clique*
4. *Set packing*
5. *Vertex cover*
6. *Set covering*
7. *Feedback node set*
8. *Feedback arc set*
9. *Directed Hamilton circuit*
10. *Undirected Hamilton circuit*
11. *3-SAT*
12. *Chromatic number*
13. *Clique cover*
14. *Exact cover*
15. *Hitting set*
16. *Steiner tree*
17. *3-dimensional matching*
18. *Knapsack*
19. *Job sequencing*
20. *Partition*
21. *Max cut*

Theorem (Karp's list of 21 problems [Karp72])

The following problems are NP-Complete:

1. *SAT*
2. *0–1 integer programming*
3. *Clique*
4. *Set packing*
5. *Vertex cover*
6. *Set covering*
7. *Feedback node set*
8. *Feedback arc set*
9. *Directed Hamilton circuit*
10. *Undirected Hamilton circuit*
11. *3-SAT*
12. *Chromatic number*
13. *Clique cover*
14. *Exact cover*
15. *Hitting set*
16. *Steiner tree*
17. *3-dimensional matching*
18. *Knapsack*
19. *Job sequencing*
20. *Partition*
21. *Max cut*

How should we cope with NP-hard problems?

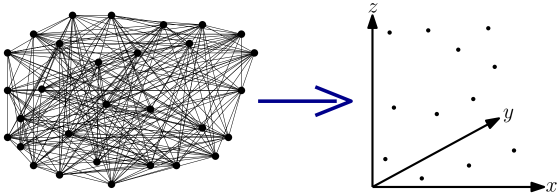

APPROXIMATION
imgflip.com

Metric Embeddings

Embedding

$(X, d_X), (Y, d_Y)$ metric spaces. $f : (X, d_X) \to (Y, d_Y)$ is called an **embedding**.

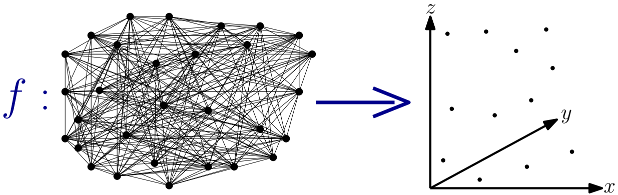
$f :$



Metric Embeddings

Embedding

$(X, d_X), (Y, d_Y)$ metric spaces. $f : (X, d_X) \to (Y, d_Y)$ is called an **embedding**.



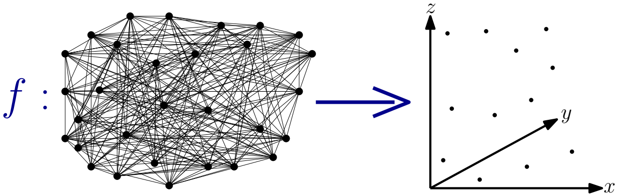
Preserve (approxierly) properties of the original space:

- **Distances**
- Cuts, Flows
- Commute time
- Effective resistance
- Clustering statistics.
- etc.

Metric Embeddings

Embedding

$(X, d_X), (Y, d_Y)$ metric spaces. $f : (X, d_X) \to (Y, d_Y)$ is called an **embedding**.



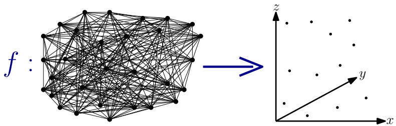
f has **distortion** t if:

$$\forall x, y \in X, \qquad d_X(x, y) \leq d_Y(f(x), f(y)) \leq t \cdot d_X(x, y) \ .$$

Metric Embeddings

Embedding

$(X, d_X), (Y, d_Y)$ metric spaces. $f : (X, d_X) \to (Y, d_Y)$ is called an **embedding**.



f has **distortion** t if:

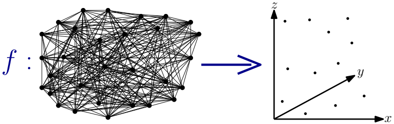
$$\forall x, y \in X, d_X(x, y) \le d_Y(f(x), f(y)) \le t \cdot d_X(x, y) \ .$$

It is highly desirable that the target space Y will have **simple structure**.

Metric Embeddings

Embedding

$(X, d_X), (Y, d_Y)$ metric spaces. $f : (X, d_X) \to (Y, d_Y)$ is called an **embedding**.



f has **distortion** t if:

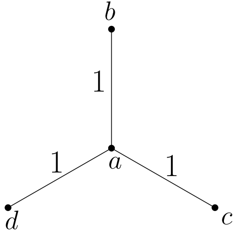
$$\forall x, y \in X, d_X(x, y) \le d_Y(f(x), f(y)) \le t \cdot d_X(x, y) \ .$$

It is highly desirable that the target space Y will have **simple structure**.

So that we could run efficient algorithms on it...

$f : (X, d_X) \to (Y, d_Y)$ has **distortion** t if:

$$\forall x, y \in X, d_X(x, y) \le d_Y(f(x), f(y)) \le t \cdot d_X(x, y) \ .$$

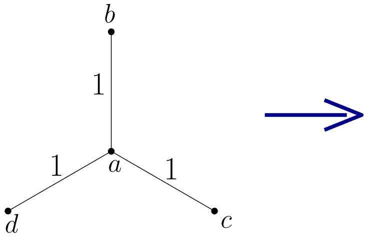


	a	b	c	d
a		1	1	1
b	1		2	2
c	1	2		2
d	1	2	2	

$f : (X, d_X) \to (Y, d_Y)$ has **distortion** t if:

$$\forall x, y \in X, d_X(x,y) \leq d_Y(f(x), f(y)) \leq t \cdot d_X(x,y) \ .$$

$f :$

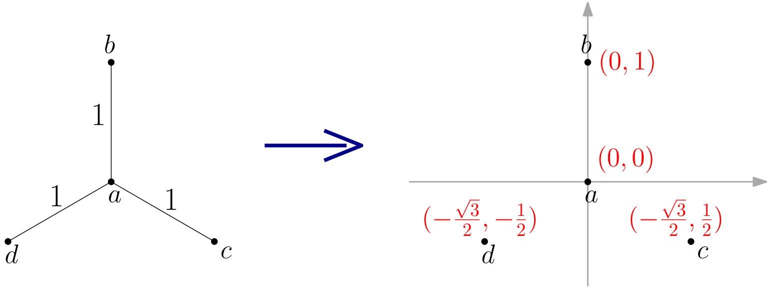


	a	b	c	d
a		1	1	1
b	1		2	2
c	1	2		2
d	1	2	2	

$f : (X, d_X) \to (Y, d_Y)$ has **distortion** t if:

$$\forall x, y \in X, d_X(x,y) \leq d_Y(f(x), f(y)) \leq t \cdot d_X(x,y) \ .$$

f :

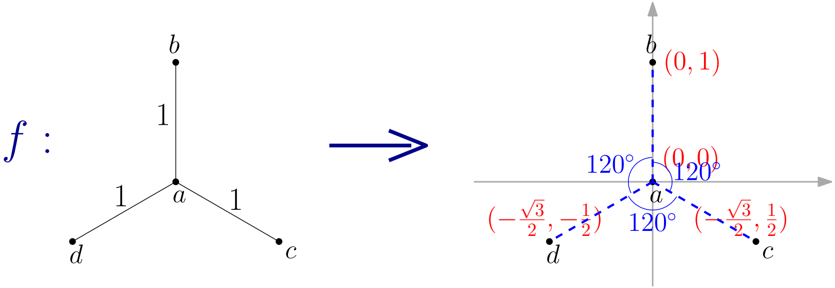


	a	b	c	d
a		1	1	1
b	1		2	2
c	1	2		2
d	1	2	2	

	a	b	c	d
a		1	1	1
b	1		$\sqrt{3}$	$\sqrt{3}$
c	1	$\sqrt{3}$		$\sqrt{3}$
d	1	$\sqrt{3}$	$\sqrt{3}$	

$f : (X, d_X) \to (Y, d_Y)$ has **distortion** t if:

$$\forall x, y \in X, d_X(x, y) \leq d_Y(f(x), f(y)) \leq t \cdot d_X(x, y)\ .$$

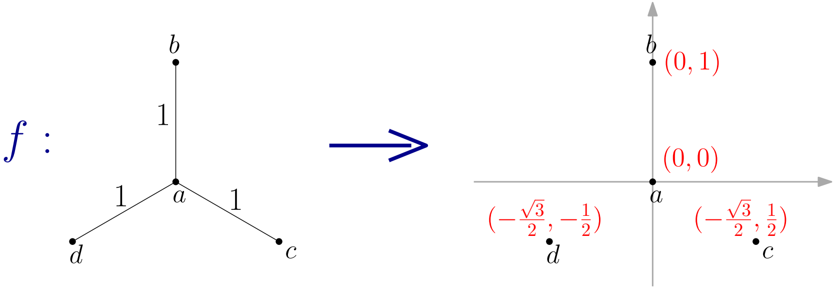


	a	b	c	d
a		1	1	1
b	1		2	2
c	1	2		2
d	1	2	2	

	a	b	c	d
a		1	1	1
b	1		$\sqrt{3}$	$\sqrt{3}$
c	1	$\sqrt{3}$		$\sqrt{3}$
d	1	$\sqrt{3}$	$\sqrt{3}$	

$f : (X, d_X) \to (Y, d_Y)$ has **distortion** t if:

$$\forall x, y \in X, d_X(x, y) \le d_Y(f(x), f(y)) \le t \cdot d_X(x, y) \ .$$

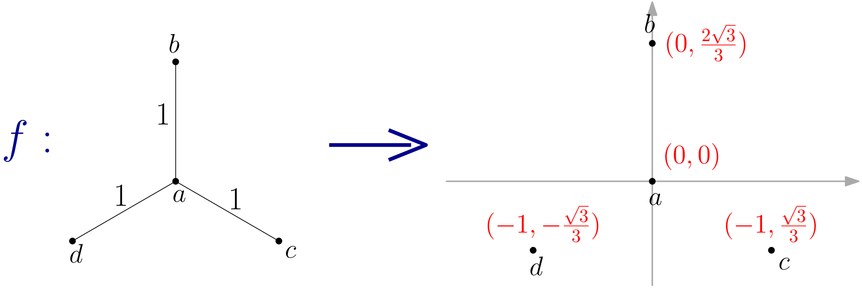


	a	b	c	d
a		1	1	1
b	1		2	2
c	1	2		2
d	1	2	2	

	a	b	c	d
a		1	1	1
b	1		$\sqrt{3}$	$\sqrt{3}$
c	1	$\sqrt{3}$		$\sqrt{3}$
d	1	$\sqrt{3}$	$\sqrt{3}$	

$f : (X, d_X) \to (Y, d_Y)$ has **distortion** t if:

$$\forall x, y \in X, d_X(x, y) \le d_Y(f(x), f(y)) \le t \cdot d_X(x, y) \ .$$

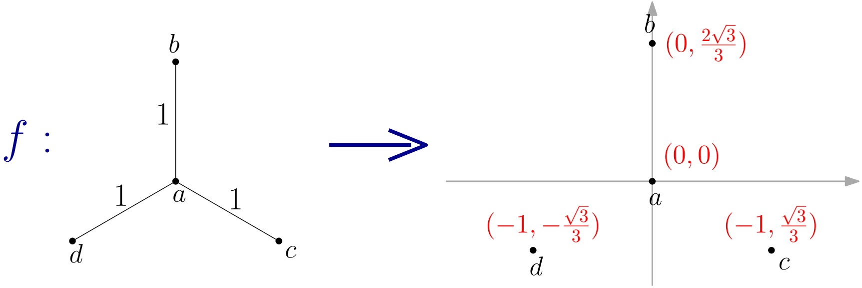


	a	b	c	d
a		1	1	1
b	1		2	2
c	1	2		2
d	1	2	2	

	a	b	c	d
a		$2/\sqrt{3}$	$2/\sqrt{3}$	$2/\sqrt{3}$
b	$2/\sqrt{3}$		2	2
c	$2/\sqrt{3}$	2		2
d	$2/\sqrt{3}$	2	2	

$f : (X, d_X) \to (Y, d_Y)$ has **distortion** t if:

$$\forall x, y \in X, d_X(x,y) \leq d_Y(f(x), f(y)) \leq t \cdot d_X(x,y) \ .$$

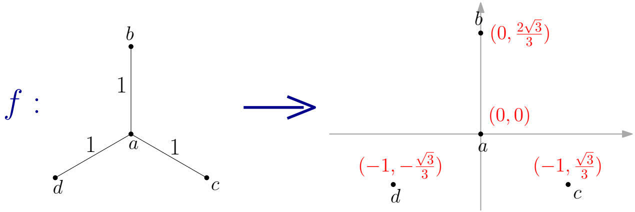


	a	b	c	d
a		1	1	1
b	1		2	2
c	1	2		2
d	1	2	2	

	a	b	c	d
a		$2/\sqrt{3}$	$2/\sqrt{3}$	$2/\sqrt{3}$
b	$2/\sqrt{3}$		2	2
c	$2/\sqrt{3}$	2		2
d	$2/\sqrt{3}$	2	2	

$f : (X, d_X) \to (Y, d_Y)$ has **distortion** t if:

$$\forall x, y \in X, d_X(x,y) \le d_Y(f(x), f(y)) \le t \cdot d_X(x,y) \ .$$



	a	b	c	d
a		1	1	1
b	1		2	2
c	1	2		2
d	1	2	2	

	a	b	c	d
a		$2/\sqrt{3}$	$2/\sqrt{3}$	$2/\sqrt{3}$
b	$2/\sqrt{3}$		2	2
c	$2/\sqrt{3}$	2		2
d	$2/\sqrt{3}$	2	2	

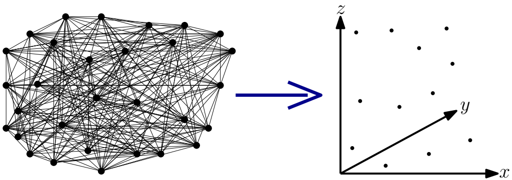
The distortion of the embedding is $\frac{2}{\sqrt{3}} \approx 1.1547$.

Metric Embeddings

Embedding

$(X, d_X), (Y, d_Y)$ metric spaces. $f : (X, d_X) \to (Y, d_Y)$ is called an **embedding**.

f :

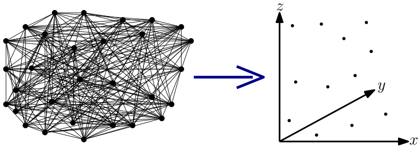


Theorem ([Bourgain 85])

Every n-point metric (X, d_X) is ***embeddable*** *into* <u>*Euclidean*</u> *space $(\mathbb{R}^d, \|\cdot\|_2)$ with* ***distortion*** *$O(\log n)$.*

Embedding

$(X, d_X), (Y, d_Y)$ metric spaces. $f : (X, d_X) \to (Y, d_Y)$ is called an **embedding**.

f : 

Theorem ([Bourgain 85])

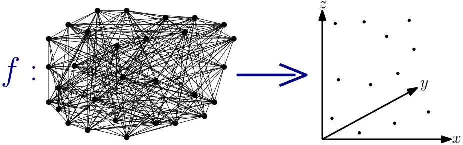
*Every n-point metric (X, d_X) is **embeddable** into Euclidean space $(\mathbb{R}^d, \|\cdot\|_2)$ with **distortion** $O(\log n)$.*

Theorem ([Linial, London, Rabinovich 95])

[Bou85] is tight.

Theorem ([Bourgain 85])

*Every n-point metric (X, d_X) is **embeddable** into Euclidean space $(\mathbb{R}^d, \|\cdot\|_2)$ with **distortion** $O(\log n)$.*

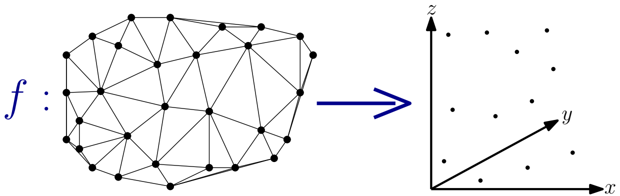


Applications:

- Approximation algorithms (e.g. **sparsest cut**, min graph bandwidth)
- Parallel computation (e.g. SSSP in MPC)
- Computational Biology (e.g. clustering and detecting protein seq.)
- etc.

Theorem ([Rao 99])

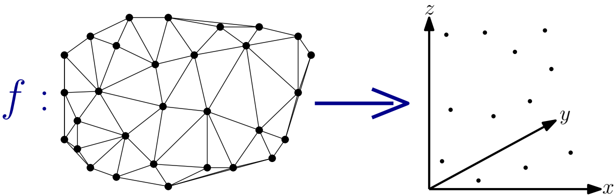
*Every n-point **planar** metric (X, d_X) is **embeddable** into Euclidean space $(\mathbb{R}^d, \|\cdot\|_2)$ with **distortion** $O(\sqrt{\log n})$.*



Planar metric- the shortest path metric of a planar graph.

Theorem ([Rao 99])

*Every n-point **planar** metric (X, d_X) is **embeddable** into Euclidean space $(\mathbb{R}^d, \|\cdot\|_2)$ with **distortion** $O(\sqrt{\log n})$.*



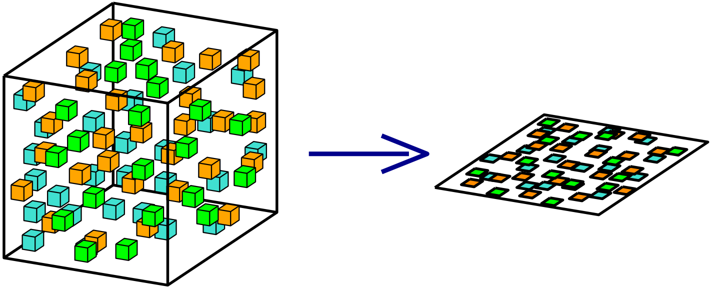
Planar metric- the shortest path metric of a planar graph.

Theorem ([Newman, Rabinovich 03])

[Rao99] is tight.

Embedding

$(X, d_X), (Y, d_Y)$ metric spaces. $f : (X, d_X) \to (Y, d_Y)$ is called an **embedding**.

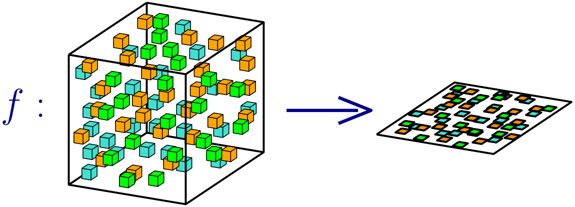
$f :$ 

Theorem ([Johnson, Lindenstrauss 84], Dimension Reduction)

$X \subset (\mathbb{R}^d, \|\cdot\|_2)$ set of size n. Then X embeds into $O(\log n/\epsilon^2)$ dimensional Euclidean space with distortion $1+\epsilon$.

Embedding

$(X, d_X), (Y, d_Y)$ metric spaces. $f : (X, d_X) \to (Y, d_Y)$ is called an **embedding**.



Theorem ([Johnson, Lindenstrauss 84], Dimension Reduction)

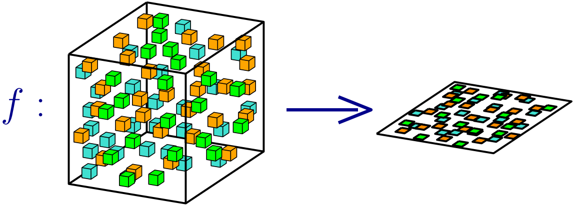
$X \subset (\mathbb{R}^d, \|\cdot\|_2)$ set of size n. Then X embeds into $O(\log n/\epsilon^2)$ dimensional Euclidean space with distortion $1 + \epsilon$.

Theorem ([Green Larsen, Nelson 17])

[JL84] is tight.

Theorem ([Johnson, Lindenstrauss 84], Dimension Reduction)

$X \subset (\mathbb{R}^d, \|\cdot\|_2)$ set of size n. Then X embeds into $O(\log n/\epsilon^2)$ dimensional Euclidean space with distortion $1+\epsilon$.



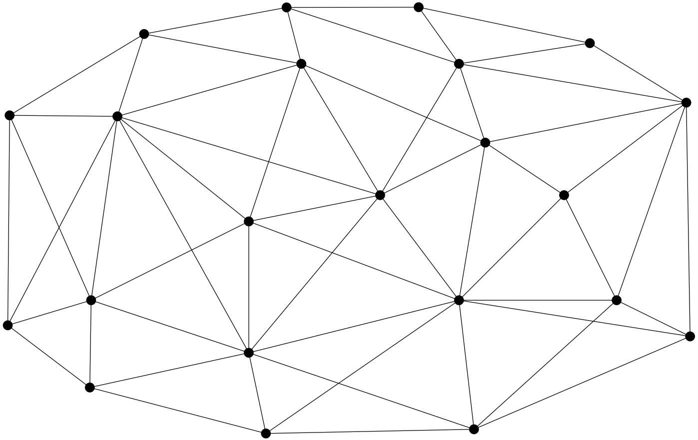
Applications:

- Speeding up-computation
- Clustering
- Nearest Neighbor Search
- Machine Learning
- etc.

Graph Spanners

$G = (V, E, w)$ weighted graph, a t-**spanner** is a **subgraph** $H = (V, E_H)$

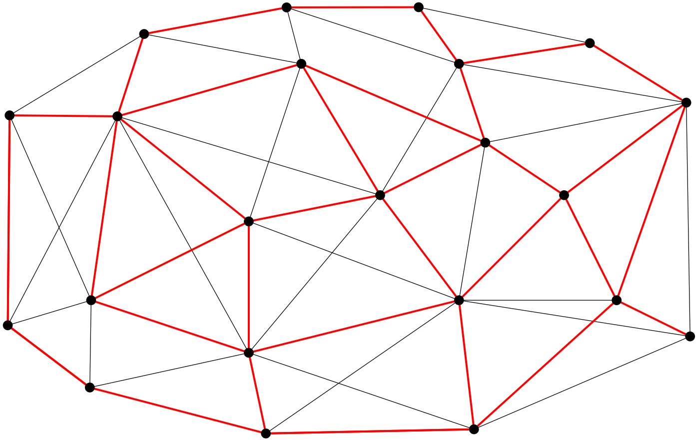
s.t. $\forall u, v \in V, \quad d_H(u, v) \leq t \cdot d_G(u, v)$



Graph Spanners

$G = (V, E, w)$ weighted graph, a t-**spanner** is a **subgraph** $H = (V, E_H)$

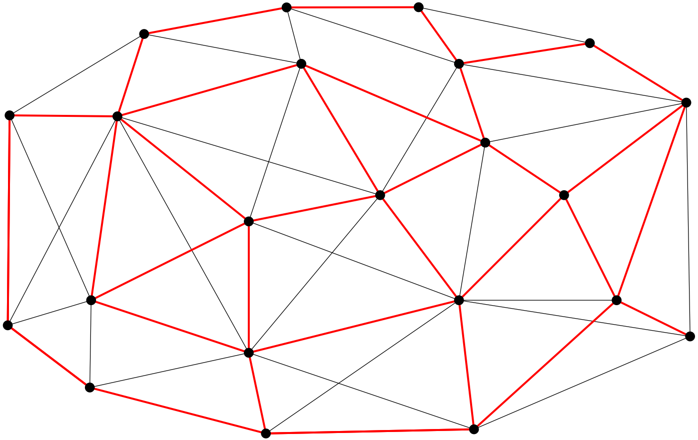
s.t. $\forall u, v \in V, \quad d_H(u, v) \leq t \cdot d_G(u, v)$



Graph Spanners

$G = (V, E, w)$ weighted graph, a t-**spanner** is a **subgraph** $H = (V, E_H)$

s.t. $\forall u, v \in V, \quad d_H(u, v) \leq t \cdot d_G(u, v)$



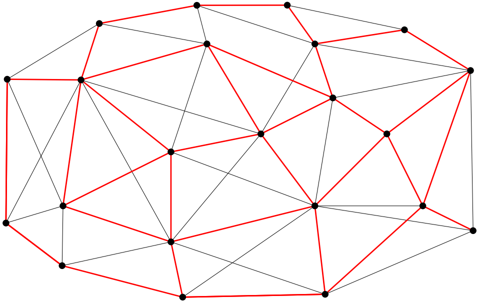
Stretch t

Sparsity $|H|$

Graph Spanners

$G = (V, E, w)$ weighted graph, a t-**spanner** is a **subgraph** $H = (V, E_H)$

s.t. $\forall u, v \in V, \quad d_H(u, v) \leq t \cdot d_G(u, v)$



Stretch t

Sparsity $|H|$

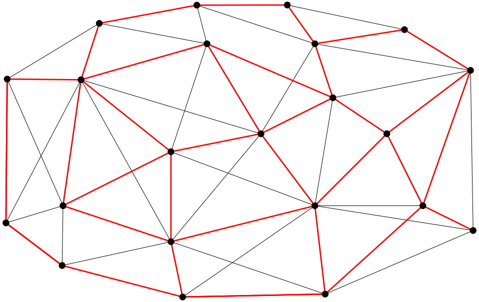
[Althofer, Das, Dobkin, Joseph, Soares 93]:

For $k \geq 1$, every graph admits $2k-1$ spanner with $O(n^{1+\frac{1}{k}})$ edges.

Graph Spanners

$G = (V, E, w)$ weighted graph, a t-**spanner** is a **subgraph** $H = (V, E_H)$

s.t. $\forall u, v \in V, \quad d_H(u, v) \leq t \cdot d_G(u, v)$



Stretch t

Sparsity $|H|$

[Althofer, Das, Dobkin, Joseph, Soares 93]:

For $k \geq 1$, every graph admits $2k - 1$ spanner with $O(n^{1+\frac{1}{k}})$ edges.

Tight. (assuming Erdös' girth conjecture).

Graph Spanners

[Althofer, Das, Dobkin, Joseph, Soares 93]:

For $k \geq 1$, every graph admits $2k-1$ spanner with $O(n^{1+\frac{1}{k}})$ edges.

Tight. (assuming Erdös' girth conjecture).

Applications:

- Approximation Algorithms (e.g. PTAS for TSP)
- Distributed Computing
- Network Routing
- Computational Biology (e.g. measure genetic distance)
- etc.

Outline of the talk

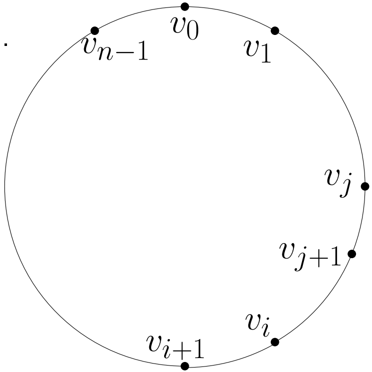
Embedding into Trees

Tree is very **simple** and **desirable** target space.

Many NP-hard problems are easy on trees (using dynamic programming).

Embedding into Trees

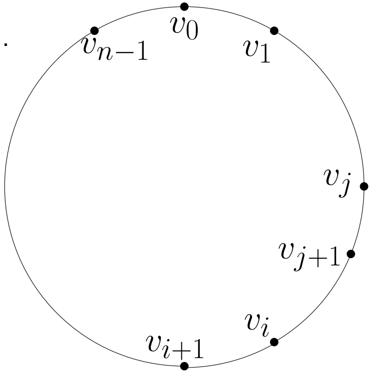
Tree is very **simple** and **desirable** target space.



Embedding C_n **requires** distortion $\Omega(n)$.

Embedding into Trees

Tree is very **simple** and **desirable** target space.

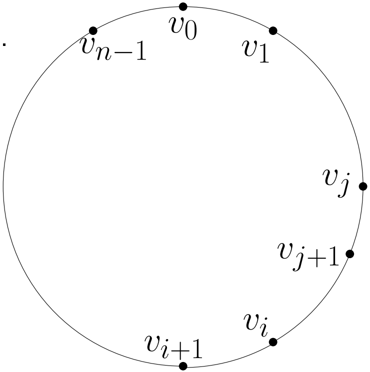


Embedding C_n **requires** distortion $\Omega(n)$.

What if we delete a **random** edge $\tilde{e}$?

Embedding into Trees

Tree is very **simple** and **desirable** target space.



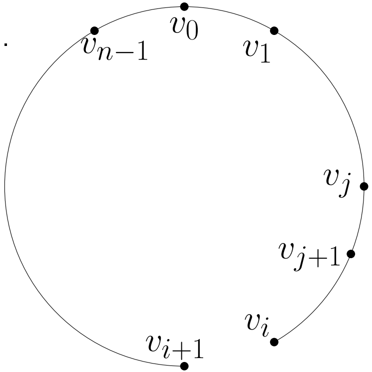
Embedding C_n **requires** distortion $\Omega(n)$.

What if we delete a **random** edge $\tilde{e}$?

$$\mathbb{E}_{T\sim\mathcal{D}}[d_T(v_i, v_{i+1})]$$

Embedding into Trees

Tree is very **simple** and **desirable** target space.



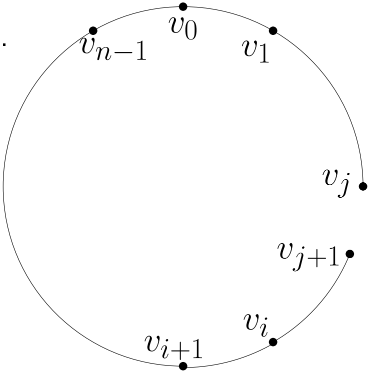
Embedding C_n **requires** distortion $\Omega(n)$.

What if we delete a **random** edge $\tilde{e}$?

$$\mathbb{E}_{T\sim\mathcal{D}}[d_T(v_i, v_{i+1})] = \Pr[\tilde{e} = \{v_i, v_{i+1}\}] \cdot (n-1) + \Pr[\tilde{e} \neq \{v_i, v_{i+1}\}] \cdot 1$$

Embedding into Trees

Tree is very **simple** and **desirable** target space.



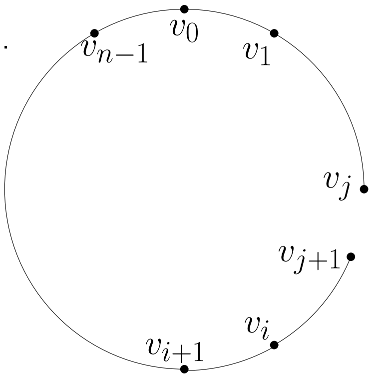
Embedding C_n **requires** distortion $\Omega(n)$.

What if we delete a **random** edge $\tilde{e}$?

$$\mathbb{E}_{T\sim\mathcal{D}}[d_T(v_i, v_{i+1})] = \Pr[\tilde{e} = \{v_i, v_{i+1}\}] \cdot (n-1) + \Pr[\tilde{e} \neq \{v_i, v_{i+1}\}] \cdot 1$$

Embedding into Trees

Tree is very **simple** and **desirable** target space.



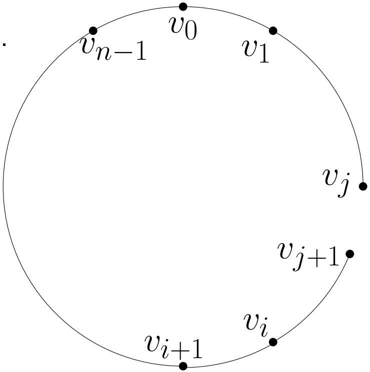
Embedding C_n **requires** distortion $\Omega(n)$.

What if we delete a **random** edge $\tilde{e}$?

$$\mathbb{E}_{T\sim\mathcal{D}}[d_T(v_i, v_{i+1})] = \Pr[\tilde{e} = \{v_i, v_{i+1}\}] \cdot (n-1) + \Pr[\tilde{e} \neq \{v_i, v_{i+1}\}] \cdot 1$$
$$= \frac{1}{n} \cdot (n-1) + \frac{n-1}{n} \cdot 1$$

Embedding into Trees

Tree is very **simple** and **desirable** target space.

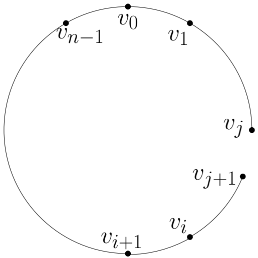


Embedding C_n **requires** distortion $\Omega(n)$.

What if we delete a **random** edge $\tilde{e}$?

$$\mathbb{E}_{T\sim\mathcal{D}}[d_T(v_i, v_{i+1})] = \Pr[\tilde{e} = \{v_i, v_{i+1}\}] \cdot (n-1) + \Pr[\tilde{e} \neq \{v_i, v_{i+1}\}] \cdot 1$$
$$= \frac{1}{n} \cdot (n-1) + \frac{n-1}{n} \cdot 1 \quad = \quad \frac{2(n-1)}{n} \quad < 2 \quad .$$

Embedding into Trees



Embedding C_n **requires** distortion $\Omega(n)$.

What if we delete a **random** edge $\tilde{e}$?

$$\begin{aligned}\mathbb{E}_{T\sim\mathcal{D}}[d_T(v_i,v_{i+1})] &= \Pr[\tilde{e}=\{v_i,v_{i+1}\}]\cdot(n-1)+\Pr[\tilde{e}\neq\{v_i,v_{i+1}\}]\cdot 1\\ &= \frac{1}{n}\cdot(n-1)+\frac{n-1}{n}\cdot 1 \quad = \quad \frac{2(n-1)}{n} \quad < 2 \quad .\end{aligned}$$

By triangle inequality and linearity of expectation

$$\forall v_i,v_j,\quad \mathbb{E}_{T\sim\mathcal{D}}[d_T(v_i,v_j)] = \sum_{q=i}^{j-1}\mathbb{E}_{T\sim\mathcal{D}}[d_T(v_q,v_{q+1 \pmod{n}})] \le 2\cdot d_{C_n}(v_i,v_j)\ .$$

Stochastic Embedding into Trees

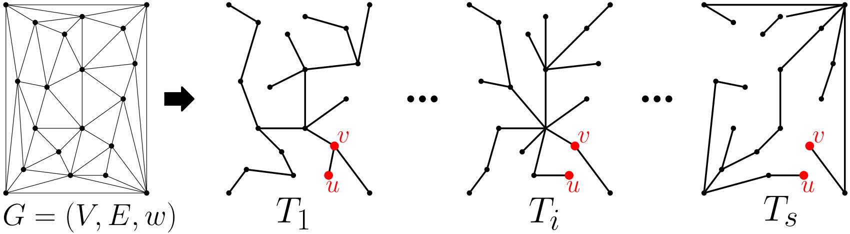
Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log n)$.

Stochastic Embedding into Trees

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

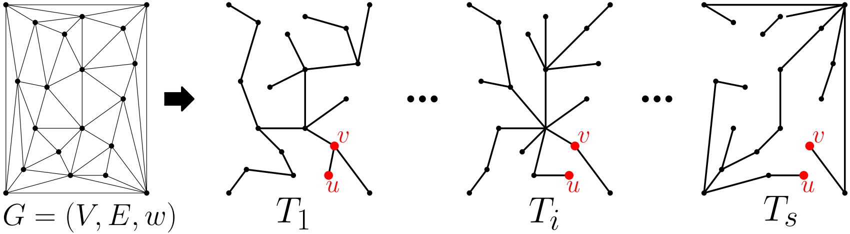
*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log n)$.*



Stochastic Embedding into Trees

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log n)$.

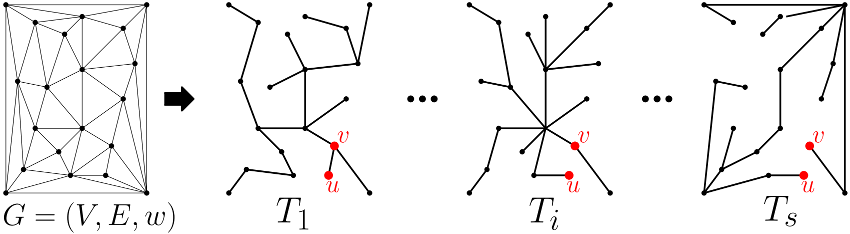


For every $u, v \in X$ and $T \in \mathrm{supp}(\mathcal{D})$, $d_X(u, v) \leq d_T(f(u), f(v))$.

Stochastic Embedding into Trees

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log n)$.



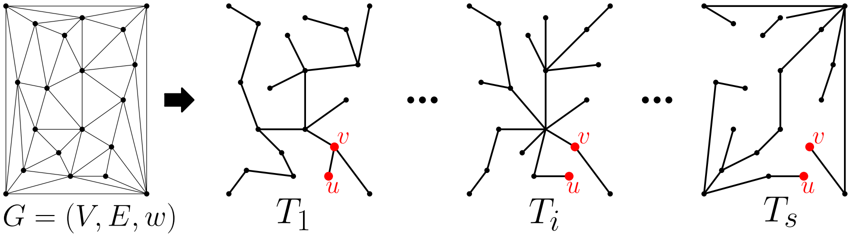
For every $u, v \in X$ and $T \in \mathrm{supp}(\mathcal{D})$, $d_X(u, v) \leq d_T(f(u), f(v))$.

For every $u, v \in X$ $\quad \mathbb{E}_{T\sim\mathcal{D}}[d_T(f(u), f(v))] \leq O(\log n) \cdot d_X(u, v)$.

Stochastic Embedding into Trees

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

Every n-point metric space (X, d) *embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log n)$.*



For every $u, v \in X$ and $T \in \mathrm{supp}(\mathcal{D})$, $d_X(u, v) \le d_T(f(u), f(v))$.

For every $u, v \in X$ $\quad \mathbb{E}_{T\sim\mathcal{D}}[d_T(f(u), f(v))] \le O(\log n) \cdot d_X(u, v)$.

[Alon, Karp, Peleg, West 95]: Tight!

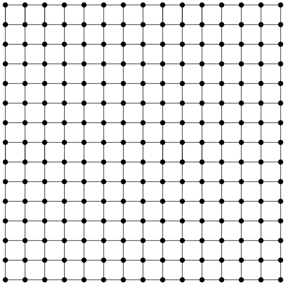
Stochastic Embedding into Trees

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log n)$.*

[Alon, Karp, Peleg, West 95]: Tight!

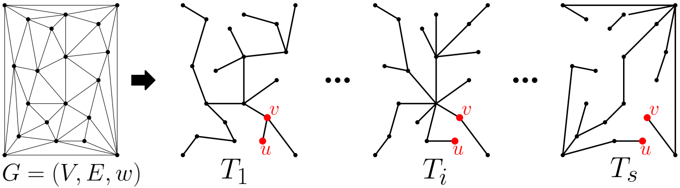
In fact, tight already for the $n \times n$ grid graph!



Stochastic Embedding into Trees

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log n)$.



A useful hammer

Transforms arbitrary metric into a tree!

Stochastic Embedding into Trees

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

Every n-point metric space (X, d) *embeds into* ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log n)$.

A useful hammer

Transforms arbitrary metric into a tree!

Applications:

- Approximation Algorithms.
- Online Algorithms.
- Distributed Computing.
- etc.

Outline of the talk

Distance Oracle

A **succinct** data structure that **approximately** answers distance queries.

Distance Oracle

A **succinct** data structure that **approximately** answers distance queries.

The properties of interest are size, distortion and query time.

Distance Oracle

A **succinct** data structure that **approximately** answers distance queries.

The properties of interest are size, distortion and query time.

Given an n point metric space one can store all distances pairwise distances.

Distance Oracle

A **succinct** data structure that **approximately** answers distance queries.

The properties of interest are size, distortion and query time.

Given an n point metric space one can store all distances pairwise distances.
Space: $O(n^2)$, **query time**: $O(1)$

Distance Oracle

A **succinct** data structure that **approximately** answers distance queries.

The properties of interest are size, distortion and query time.

Given an n point metric space one can store all distances pairwise distances.
Space: $O(n^2)$, **query time**: $O(1)$

Best possible for **exact** distance oracle.

Distance Oracle

A **succinct** data structure that **approximately** answers distance queries.

The properties of interest are size, distortion and query time.

Given an n point metric space one can store all distances pairwise distances.
Space: $O(n^2)$, **query time**: $O(1)$

Best possible for **exact** distance oracle.

Could we do better by allowing the oracle to returned approximated distances?

Distance Oracle

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log n)$.

Distance Oracle

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

Every n-point metric space (X, d) embeds into ***distribution*** *$\mathcal{D}$ over* ***dominating trees*** *with* ***expected distortion*** *$O(\log n)$.*

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$

$$\mathbb{E}_{T\sim\mathcal{D}}[d_T(x, y)] = O(\log n) \cdot d_X(x, y) \ .$$

Distance Oracle

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log n)$.*

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$. $\mathbb{E}_{T\sim\mathcal{D}}[d_T(x, y)] = O(\log n) \cdot d_X(x, y)$.

Distance Oracle construction

Sample $s = 4 \log n$ trees $T_1, \ldots, T_s$. Given x, y return $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y)$.

Distance Oracle

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log n)$.

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$. $\mathbb{E}_{T\sim\mathcal{D}}[d_T(x, y)] = O(\log n) \cdot d_X(x, y)$.

Distance Oracle construction

Sample $s = 4 \log n$ trees $T_1, \ldots, T_s$. Given x, y return $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y) \geq d_X(x, y)$.

Distance Oracle

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log n)$.*

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$. $\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)] = O(\log n) \cdot d_X(x,y)$.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \dots, T_s$. Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y) \geq d_X(x,y)$.

Markov inequality: $\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$. $\qquad \mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)] = O(\log n) \cdot d_X(x,y)$.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \dots, T_s$. $\quad$ Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Clearly, as the trees are **dominating**, $\qquad \mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y) \ge d_X(x,y)$.

Markov inequality: $\Pr[X \ge a] \le \frac{\mathbb{E}[X]}{a}$

$$\Pr_{T\sim\mathcal{D}}[d_T(x,y) \ge 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]] \le \frac{\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]}{2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]} = \frac{1}{2} .$$

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$. $\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)] = O(\log n) \cdot d_X(x,y)$.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \dots, T_s$. Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y) \geq d_X(x,y)$.

Markov inequality: $\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$

$$\Pr_{T\sim\mathcal{D}}\left[d_T(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] \leq \frac{\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]}{2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]} = \frac{1}{2} .$$

$$\Pr_{T\sim\mathcal{D}}\left[\mathrm{DO}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right]$$

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$. $\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)] = O(\log n) \cdot d_X(x,y)$.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \dots, T_s$. Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y) \geq d_X(x,y)$.

Markov inequality: $\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$

$$\Pr_{T\sim\mathcal{D}}\left[d_T(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] \leq \frac{\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]}{2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]} = \frac{1}{2} .$$

$$\Pr_{T\sim\mathcal{D}}\left[\mathrm{DO}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] = \Pr_{T\sim\mathcal{D}}\left[\forall i \quad d_{T_i}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right]$$

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$. $\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)] = O(\log n)\cdot d_X(x,y)$.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \dots, T_s$. Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y) \geq d_X(x,y)$.

Markov inequality: $\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$

$$\Pr_{T\sim\mathcal{D}}\left[d_T(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] \leq \frac{\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]}{2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]} = \frac{1}{2} .$$

$$\begin{aligned}\Pr_{T\sim\mathcal{D}}\left[\mathrm{DO}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] &= \Pr_{T\sim\mathcal{D}}\left[\forall i \quad d_{T_i}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] \\ &= \prod_{i=1}^{s}\Pr_{T\sim\mathcal{D}}\left[d_{T_i}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right]\end{aligned}$$

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$. $\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)] = O(\log n) \cdot d_X(x,y)$.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \dots, T_s$. Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y) \geq d_X(x,y)$.

Markov inequality: $\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$

$$\Pr_{T\sim\mathcal{D}}[d_T(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]] \leq \frac{\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]}{2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]} = \frac{1}{2}\ .$$

$$\begin{aligned}\Pr_{T\sim\mathcal{D}}[\mathrm{DO}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]] &= \Pr_{T\sim\mathcal{D}}[\forall i \quad d_{T_i}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]] \\ &= \prod_{i=1}^{s}\Pr_{T\sim\mathcal{D}}[d_{T_i}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]] \\ &\leq \left(\frac{1}{2}\right)^s\end{aligned}$$

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$. $\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)] = O(\log n) \cdot d_X(x,y)$.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \dots, T_s$. Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y) \geq d_X(x,y)$.

Markov inequality: $\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$

$$\Pr_{T\sim\mathcal{D}}[d_T(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]] \leq \frac{\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]}{2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]} = \frac{1}{2} \,.$$

$$\begin{aligned}\Pr_{T\sim\mathcal{D}}[\mathrm{DO}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]] &= \Pr_{T\sim\mathcal{D}}[\forall i \quad d_{T_i}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]] \\ &= \prod_{i=1}^{s} \Pr_{T\sim\mathcal{D}}[d_{T_i}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]] \\ &\leq \left(\frac{1}{2}\right)^{s} = \left(\frac{1}{2}\right)^{4\log n}\end{aligned}$$

Fix $x, y \in X$, and sample a tree $T \sim \mathcal{D}$. $\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)] = O(\log n)\cdot d_X(x,y)$.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1,\dots,T_s$. Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y) \geq d_X(x,y)$.

Markov inequality: $\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$

$$\Pr_{T\sim\mathcal{D}}\left[d_T(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] \leq \frac{\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]}{2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]} = \frac{1}{2} .$$

$$\begin{aligned}\Pr_{T\sim\mathcal{D}}\left[\mathrm{DO}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] &= \Pr_{T\sim\mathcal{D}}\left[\forall i \quad d_{T_i}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] \\ &= \prod_{i=1}^{s}\Pr_{T\sim\mathcal{D}}\left[d_{T_i}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] \\ &\leq \left(\frac{1}{2}\right)^{s} = \left(\frac{1}{2}\right)^{4\log n} = \frac{1}{n^4} .\end{aligned}$$

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \ldots, T_s$. Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y) \geq d_X(x,y)$.

$$\Pr_{T\sim\mathcal{D}}[\mathrm{DO}(x,y) \geq 2\cdot \mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]] \leq \frac{1}{n^4}\ .$$

By union bound, the probability that some pair x, y got too large distortion is at most

$$\binom{n}{2}\cdot\frac{1}{n^4} \leq \frac{1}{n^2}\ .$$

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \ldots, T_s$. Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y) \geq d_X(x,y)$.

$$\Pr_{T\sim\mathcal{D}}\left[\mathrm{DO}(x,y) \geq 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)]\right] \leq \frac{1}{n^4}\ .$$

By union bound, the probability that some pair x, y got too large distortion is at most

$$\binom{n}{2}\cdot\frac{1}{n^4} \leq \frac{1}{n^2}\ .$$

Thus with high probability, for every $x, y \in X$

$$\mathrm{DO}(x,y) < 2\cdot\mathbb{E}_{T\sim\mathcal{D}}[d_T(x,y)] = O(\log n)\cdot d_G(x,y)\ .$$

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \ldots, T_s$. Given x, y return $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y) \geq d_X(x, y)$.

Thus with high probability, for every $x, y \in X$

$$\mathrm{DO}(x, y) < 2 \cdot \mathbb{E}_{T\sim\mathcal{D}}[d_T(x, y)] = O(\log n) \cdot d_G(x, y) \ .$$

Space: storing $O(\log n)$ trees. Total space is $O(n \log n)$ (machine words).

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \ldots, T_s$. Given x, y return $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y) \geq d_X(x, y)$.

Thus with high probability, for every $x, y \in X$

$$\mathrm{DO}(x, y) < 2 \cdot \mathbb{E}_{T\sim\mathcal{D}}[d_T(x, y)] = O(\log n) \cdot d_G(x, y) \ .$$

Space: storing $O(\log n)$ trees. Total space is $O(n\log n)$ (machine words).

Query time: computing $d_{T_i}(x, y)$ for $i \in [1, s]$.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \dots, T_s$. Given x, y return $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y) \geq d_X(x, y)$.

Thus with high probability, for every $x, y \in X$

$$\mathrm{DO}(x, y) < 2 \cdot \mathbb{E}_{T\sim\mathcal{D}}[d_T(x, y)] = O(\log n) \cdot d_G(x, y) \ .$$

Space: storing $O(\log n)$ trees. Total space is $O(n\log n)$ (machine words).

Query time: computing $d_{T_i}(x, y)$ for $i \in [1, s]$.

There is a data structure computing $O(1)$ distance approximation in trees in $O(1)$ time. Overall $O(\log n)$ query time.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \ldots, T_s$. Given x, y return $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y) \geq d_X(x, y)$.

Thus with high probability, for every $x, y \in X$

$$\mathrm{DO}(x, y) < 2 \cdot \mathbb{E}_{T\sim\mathcal{D}}[d_T(x, y)] = O(\log n) \cdot d_G(x, y) \ .$$

Space: storing $O(\log n)$ trees. Total space is $O(n\log n)$ (machine words).

Query time: computing $d_{T_i}(x, y)$ for $i \in [1, s]$.

There is a data structure computing $O(1)$ distance approximation in trees in $O(1)$ time.
Overall $O(\log n)$ query time.

Overall we obtained distance approximation $O(\log n)$ with $O(\log n)$ query time and $O(n\log n)$ space.

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \ldots, T_s$. Given x, y return $\mathrm{DO}(x,y) = \min_{i\in[1,s]} d_{T_i}(x,y)$.

Space: storing $O(\log n)$ trees. Total space is $O(n\log n)$ (machine words).

Query time: computing $d_{T_i}(x,y)$ for $i \in [1,s]$.

There is a data structure computing $O(1)$ distance approximation in trees in $O(1)$ time. Overall $O(\log n)$ query time.

Overall we obtained distance approximation $O(\log n)$ with $O(\log n)$ query time and $O(n\log n)$ space.

Theorem ([Chechik 15])

Distance oracle with approximation $O(\log n)$, space $O(n)$, and query time $O(1)$.

Distance Oracle construction

Sample $s = 4 \log n$ trees $T_1, \ldots, T_s$. Given x, y return $\mathrm{DO}(x, y) = \min_{i \in [1,s]} d_{T_i}(x, y)$.

Space: storing $O(\log n)$ trees. Total space is $O(n \log n)$ (machine words).

Query time: computing $d_{T_i}(x, y)$ for $i \in [1, s]$.

There is a data structure computing $O(1)$ distance approximation in trees in $O(1)$ time.
Overall $O(\log n)$ query time.

Overall we obtained distance approximation $O(\log n)$ with $O(\log n)$ query time and $O(n \log n)$ space.

Theorem ([Chechik 15])

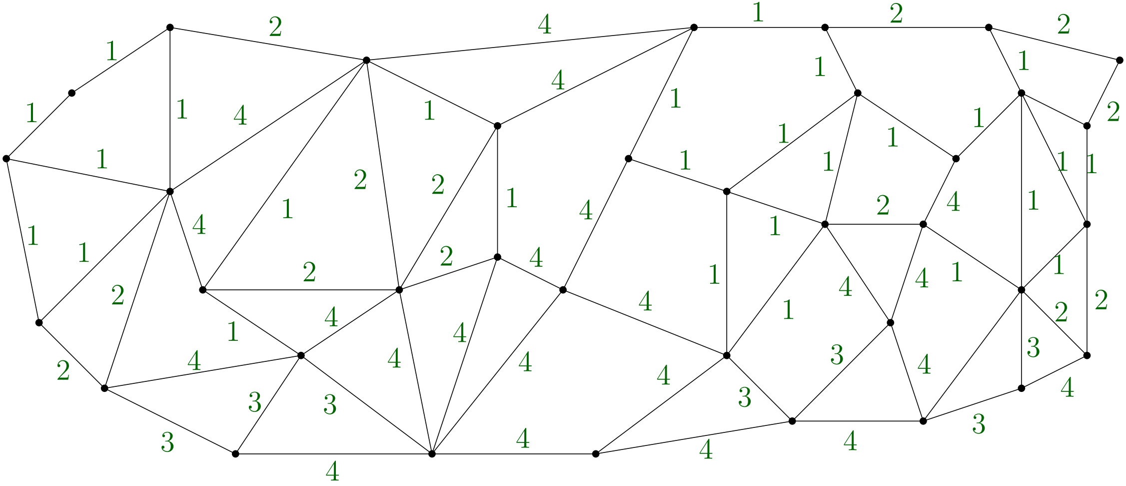
Distance oracle with approximation $O(\log n)$, space $O(n)$, and query time $O(1)$.

Distance oracle with approximation $2k - 1$, space $O(n^{1+\frac{1}{k}})$, and query time $O(1)$.

Outline of the talk

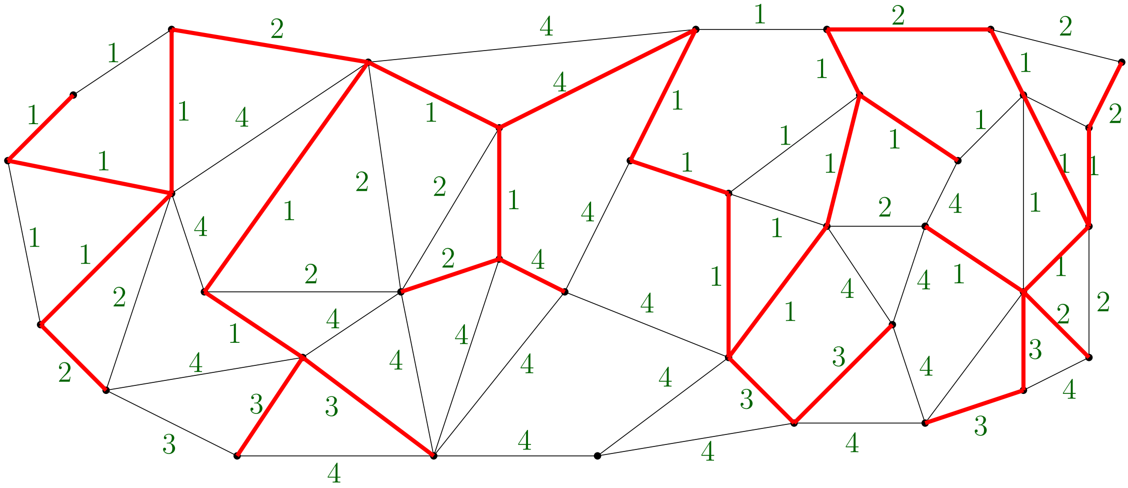
Minimum Spanning Tree (MST)

Given a weighted graph $G = (V, E, w)$ find a spanning tree if minimum total weight.



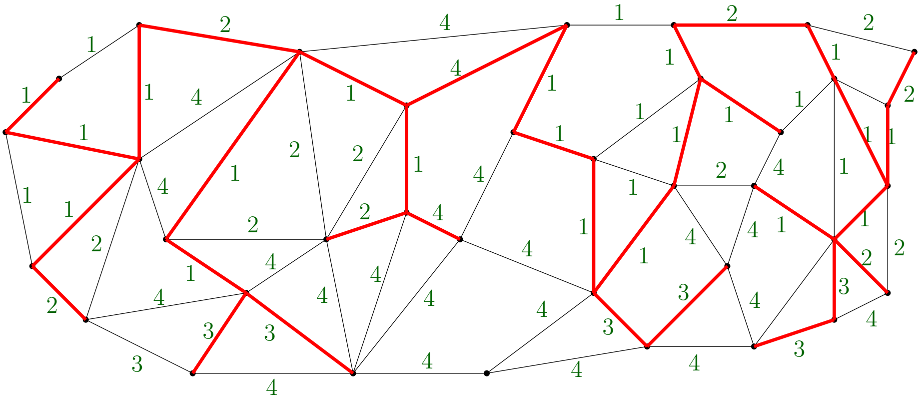
Minimum Spanning Tree (MST)

Given a weighted graph $G = (V, E, w)$ find a spanning tree if minimum total weight.



Minimum Spanning Tree (MST)

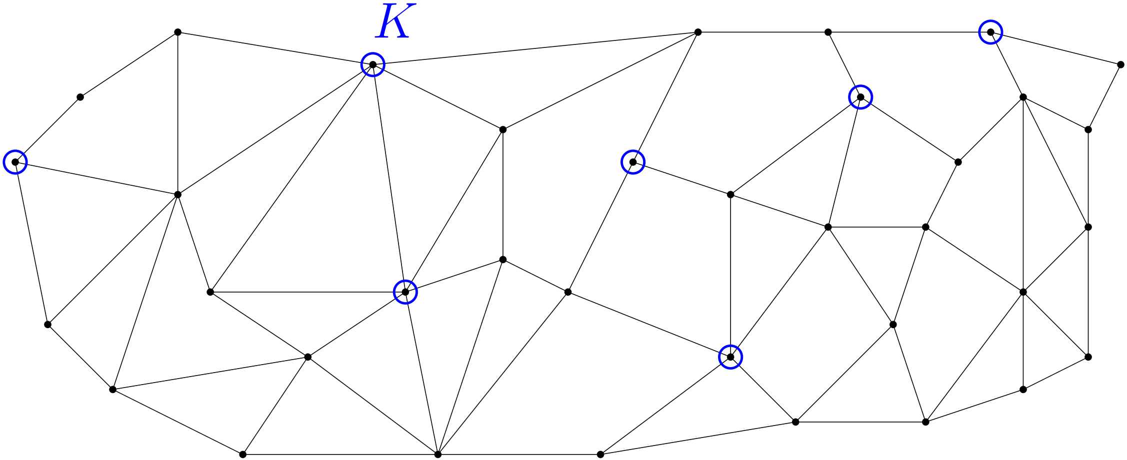
Given a weighted graph $G = (V, E, w)$ find a spanning tree if minimum total weight.



Classic problem, admits efficient poly-time solution.

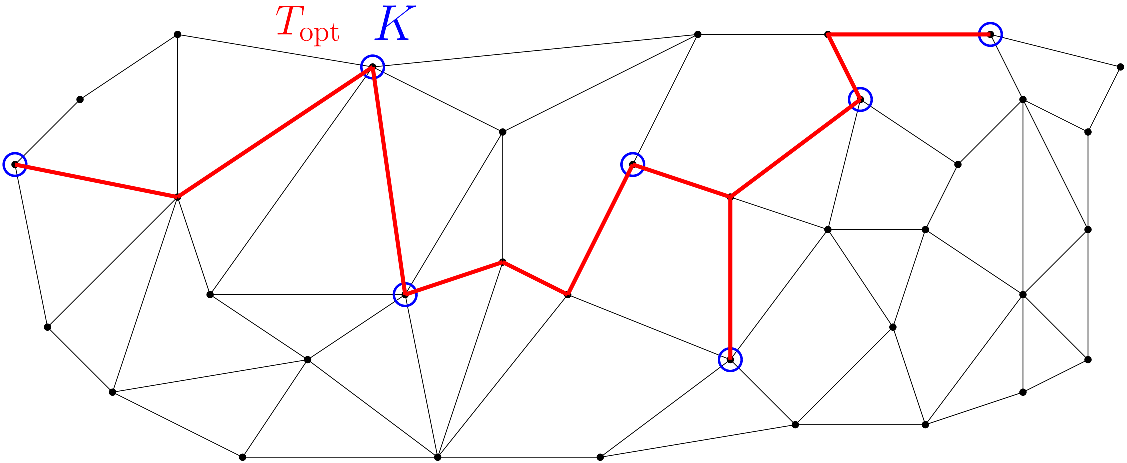
Steiner Tree

Given set of terminals K, find minimum weight tree T spanning K



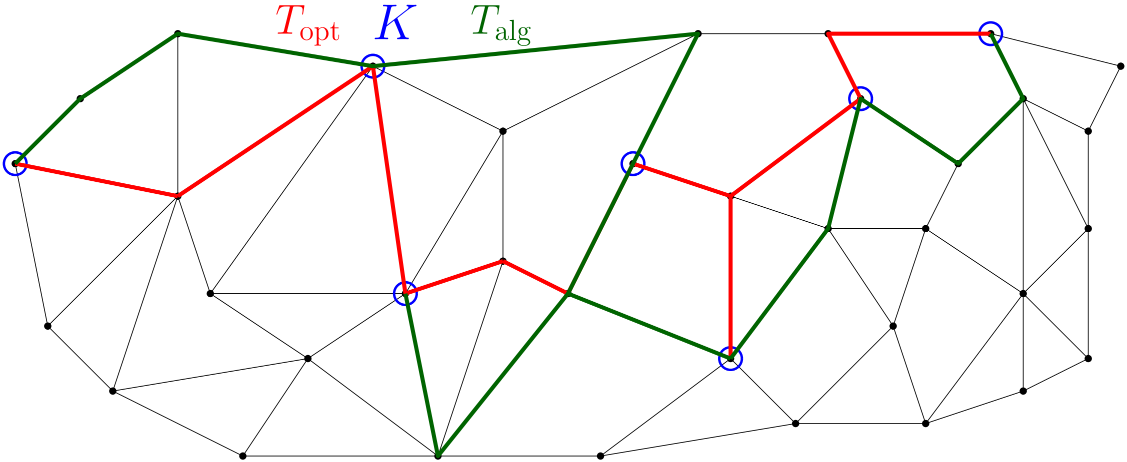
Steiner Tree

Given set of terminals K, find minimum weight tree T spanning K



Steiner Tree

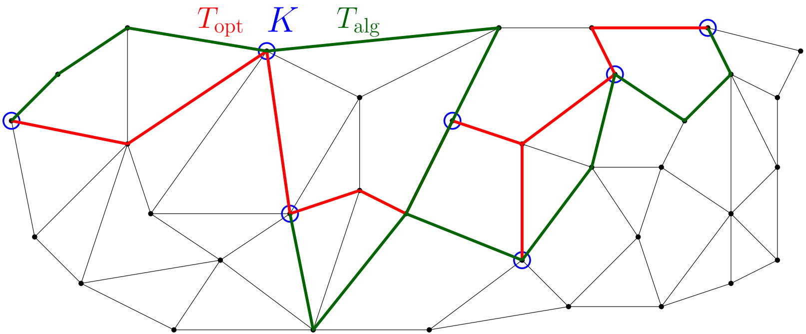
Given set of terminals K, find minimum weight tree T spanning K



NP-hard. There is a simple 2-approximation algorithm for the Steiner tree problem.

Steiner Tree

Given set of terminals K, find minimum weight tree T spanning K

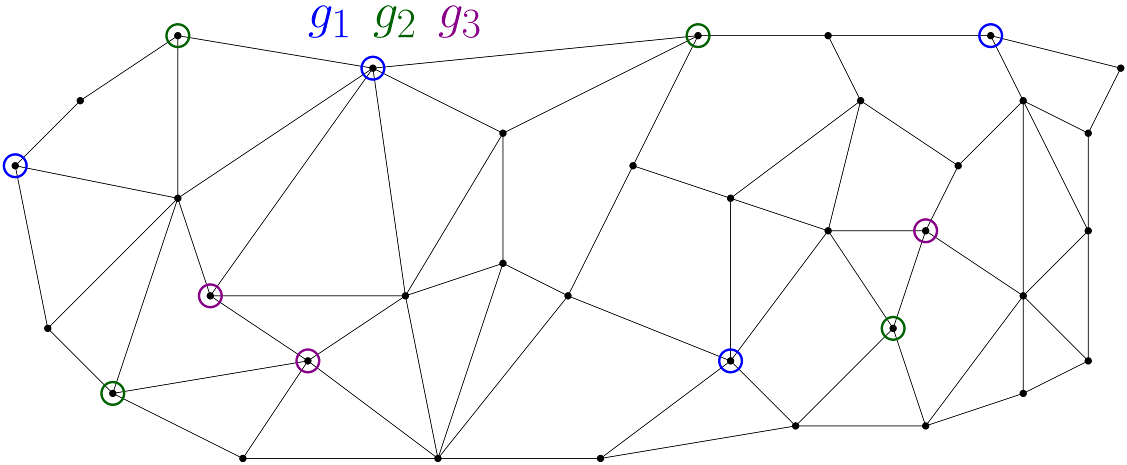


NP-hard. There is a simple 2-approximation algorithm for the Steiner tree problem.

That is, there is a polynomial time algorithm that returns a tree T_{alg} of weight at most $w(T_{\mathrm{alg}}) \leq 2 \cdot w(T_{\mathrm{opt}})$.

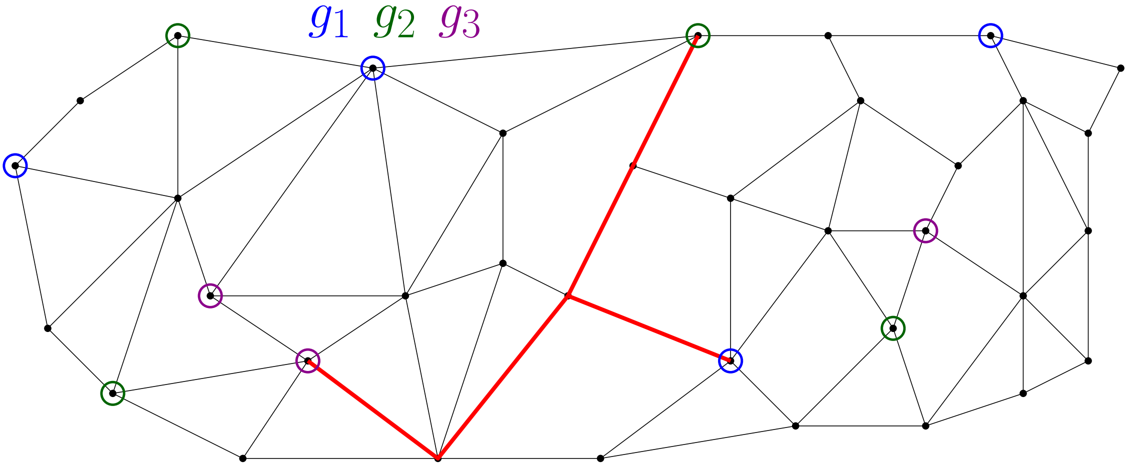
Group Steiner Tree (GST)

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i



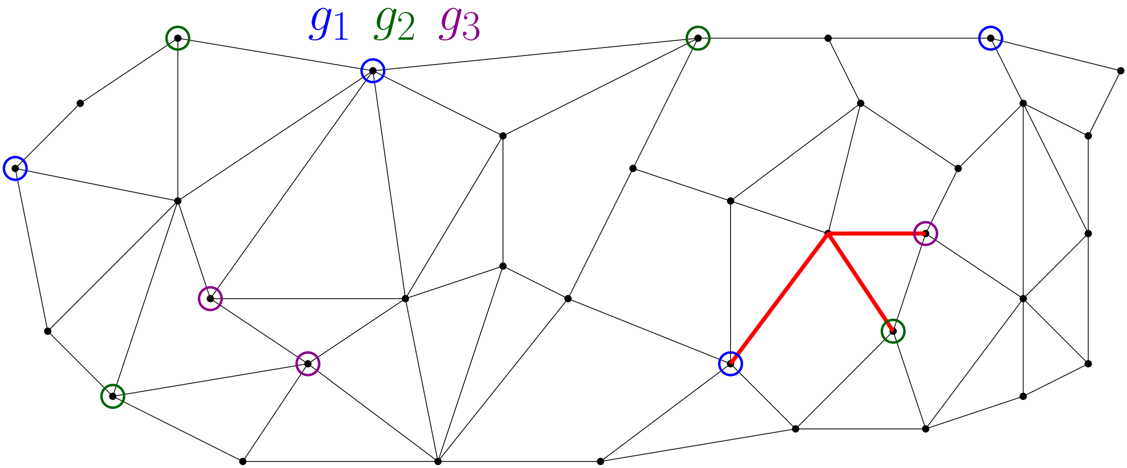
Group Steiner Tree (GST)

Given subsets $g_1, g_2, \dots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i



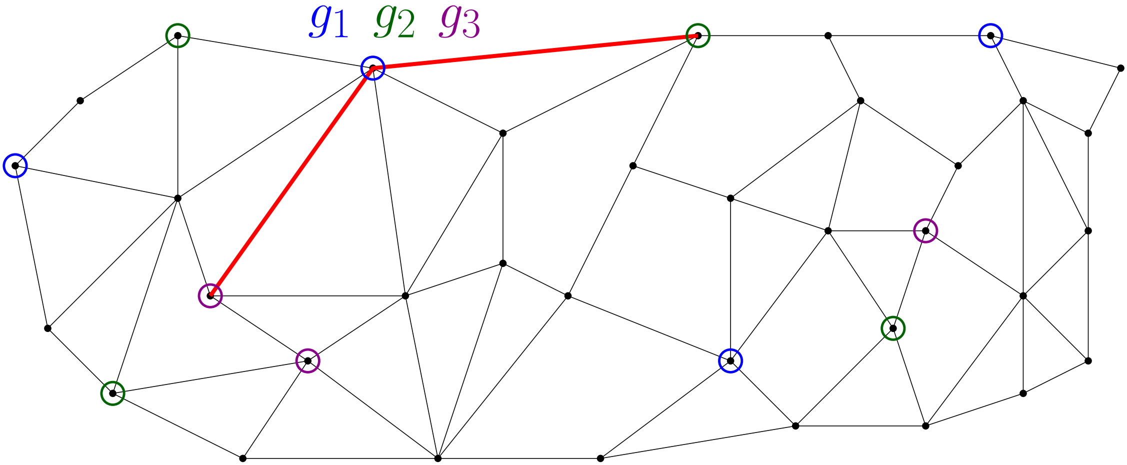
Group Steiner Tree (GST)

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i



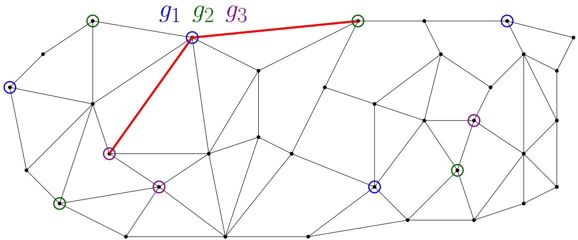
Group Steiner Tree (GST)

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i



Group Steiner Tree (GST)

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i

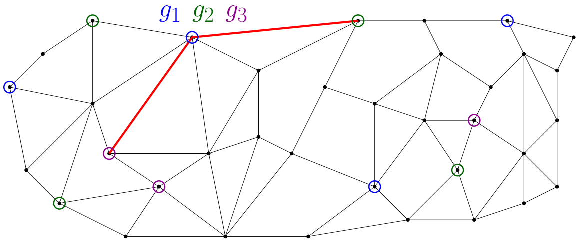


Note that Steiner tree is a special case of GST where all group sizes are 1.

Even this case is hard!

Group Steiner Tree (GST)

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i



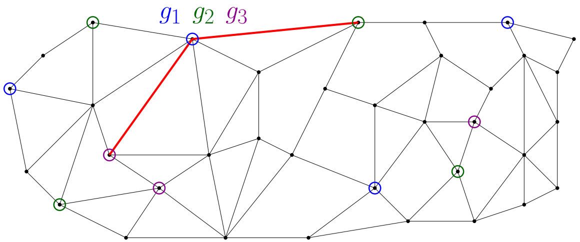
Note that Steiner tree is a special case of GST where all group sizes are 1.

Even this case is hard!

The GST problem is much harder - we first need to choose which vertex to span from each group, and then to decide how to connect them.

Group Steiner Tree (GST)

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i



Note that Steiner tree is a special case of GST where all group sizes are 1.

Even this case is hard!

The GST problem is much harder - we first need to choose which vertex to span from each group, and then to decide how to connect them.

In fact, the problem is hard already on trees! (reduction to set cover)

Group Steiner Tree (GST)

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i

Note that Steiner tree is a special case of GST where all group sizes are 1.

Even this case is hard!

The GST problem is much harder - we first need to choose which vertex to span from each group, and then to decide how to connect them.

In fact, the problem is hard already on trees! (reduction to set cover)

Theorem ([Garg, Konjevod, Ravi 00])

$O(\log n \cdot \log k)$-approximation algorithm for the GST problem on trees.

Group Steiner Tree (GST)

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i

Theorem ([Garg, Konjevod, Ravi 00])

$O(\log n \cdot \log k)$-approximation algorithm for the GST problem on trees.

That is given a tree $T = (V, E)$ and groups $g_1, g_2, \ldots, g_k \subseteq V$, there is an efficient algorithm that finds a sub-tree T_{alg} spanning a subset A of vertices such that:

- For every group g_i, $A \cap g_i \neq \emptyset$.
- $w(T_{\mathrm{alg}}) \leq O(\log n \cdot \log k) \cdot w(T_{\mathrm{opt}})$ (where T_{opt} is the optimal solution).

Group Steiner Tree (GST)

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i

Theorem ([Garg, Konjevod, Ravi 00])

$O(\log n \cdot \log k)$-approximation algorithm for the GST problem on trees.

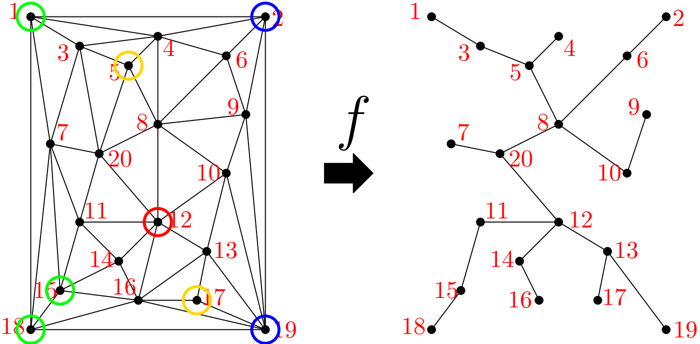
That is given a tree $T = (V, E)$ and groups $g_1, g_2, \ldots, g_k \subseteq V$, there is an efficient algorithm that finds a sub-tree T_{alg} spanning a subset A of vertices such that:

- For every group g_i, $A \cap g_i \neq \emptyset$.
- $w(T_{\mathrm{alg}}) \leq O(\log n \cdot \log k) \cdot w(T_{\mathrm{opt}})$ (where T_{opt} is the optimal solution).

We will use stochastic tree embeddings to generalize [GKR00] to general graphs.

GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Embedding f with expected distortion $O(\log n)$.



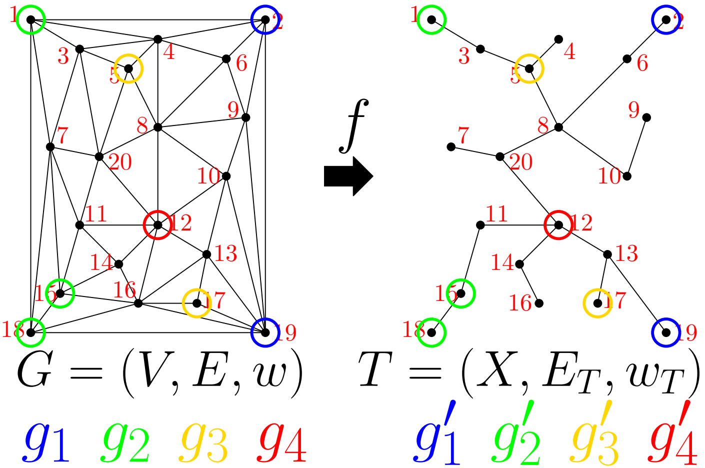
$G = (V, E, w)$ $\quad$ $T = (X, E_T, w_T)$

g_1 $\quad$ g_2 $\quad$ g_3 $\quad$ g_4

GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Embedding f with expected distortion $O(\log n)$.

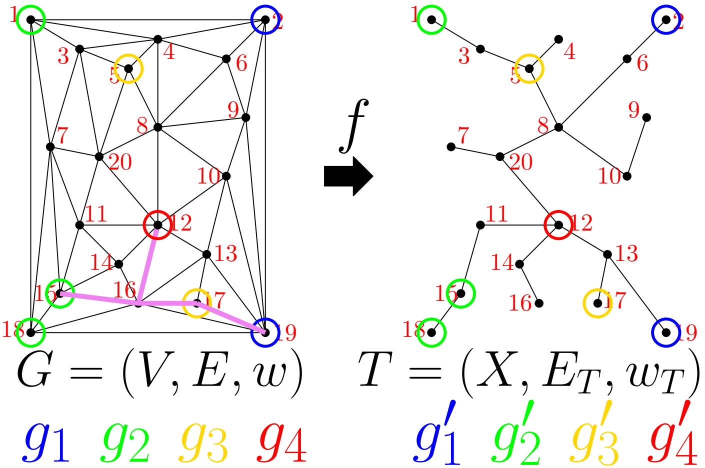
$g_i' = f(g_i)$



GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Embedding f with expected distortion $O(\log n)$. $g_i' = f(g_i)$

$S^\star$ optimal solution.

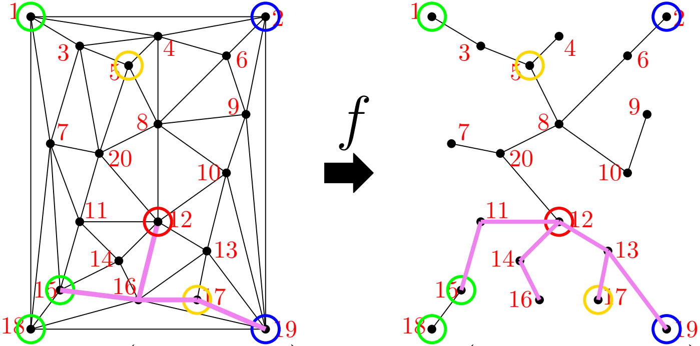


GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Embedding f with expected distortion $O(\log n)$. $\quad g_i' = f(g_i)$

$S^\star$ optimal solution.

$S^\star_T$: $\forall (u,v) \in S^\star$ add the path from $f(u)$ to $f(v)$.
(valid solution)



$G = (V, E, w)$ $\quad$ $T = (X, E_T, w_T)$

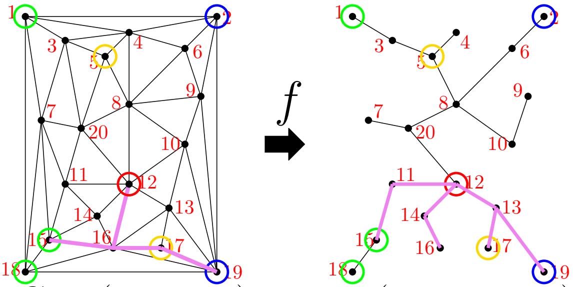
g_1 g_2 g_3 g_4 $\quad$ g_1' g_2' g_3' g_4'

GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Embedding f with expected distortion $O(\log n)$. $\quad g_i' = f(g_i)$

$S^\star$ optimal solution.

$S^\star_T$: $\forall (u,v) \in S^\star$ add the path from $f(u)$ to $f(v)$.



$G = (V, E, w)$ $\quad T = (X, E_T, w_T)$

$g_1 \;\; g_2 \;\; g_3 \;\; g_4 \qquad g_1' \;\; g_2' \;\; g_3' \;\; g_4'$

$$\mathbb{E}[w_T(S^\star_T)] \le \sum_{(u,v)\in S^\star} \mathbb{E}\left[d_T(f(u), f(v))\right] = O(\log n) \cdot \sum_{(u,v)\in S^\star} d_T(f(u), f(v))$$

$$= O(\log n) \cdot \sum_{(u,v)\in S^\star} w_G((u,v)) = O(\log n) \cdot w_G(S^\star)$$

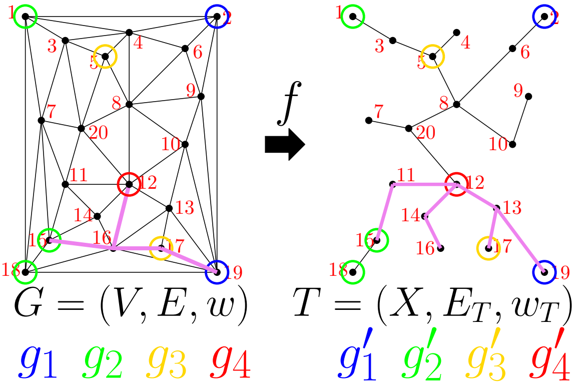
GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Embedding f with expected distortion $O(\log n)$. $\quad g_i' = f(g_i)$

$S^\star$ optimal solution.

$S^\star_T$: $\forall (u,v) \in S^\star$ add the path from $f(u)$ to $f(v)$.

$\mathbb{E}[w_T(S^\star_T)] = O(\log n) \cdot w(S^\star)$



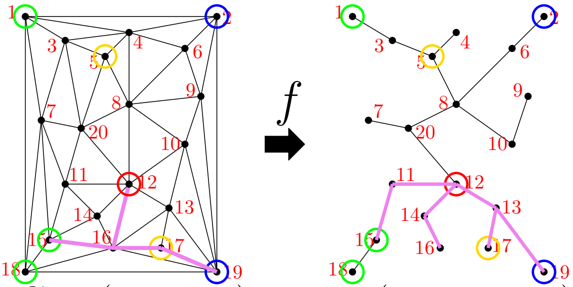
GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Embedding f with expected distortion $O(\log n)$. $\quad g_i' = f(g_i)$

$S^\star$ optimal solution.

$S^\star_T$: $\forall (u,v) \in S^\star$ add the path from $f(u)$ to $f(v)$.

$\mathbb{E}[w_T(S^\star_T)] = O(\log n) \cdot w(S^\star)$



$G = (V, E, w) \qquad T = (X, E_T, w_T)$

$g_1 \quad g_2 \quad g_3 \quad g_4 \qquad g_1' \quad g_2' \quad g_3' \quad g_4'$

Theorem ([Garg, Konjevod, Ravi 00])

$O(\log n \cdot \log k)$-approximation algorithm for the GST problem on trees.

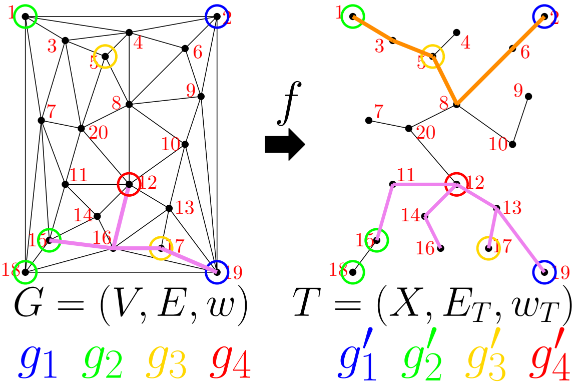
GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Embedding f with expected distortion $O(\log n)$. $\quad g_i' = f(g_i)$

$S^\star$ optimal solution.

$S^\star_T$: $\forall (u,v) \in S^\star$ add the path from $f(u)$ to $f(v)$.
$\mathbb{E}[w_T(S^\star_T)] = O(\log n) \cdot w(S^\star)$

$\tilde{S}_T$ solution by [GKR00],
$w(\tilde{S}_T) \leq O(\log n \cdot \log k) \cdot w(S^\star_T)$.



Theorem ([Garg, Konjevod, Ravi 00])

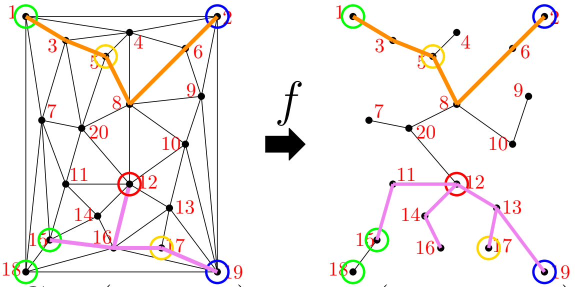
$O(\log n \cdot \log k)$-approximation algorithm for the GST problem on trees.

GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

$S^\star$ optimal solution.

$S^\star_T$: $\forall (u, v) \in S^\star$ add the path from $f(u)$ to $f(v)$.
$\mathbb{E}[w_T(S^\star_T)] = O(\log n) \cdot w(S^\star)$

$\tilde{S}_T$ solution by [GKR00],
$w(\tilde{S}_T) \leq O(\log n \cdot \log k) \cdot w(S^\star_T)$.



$G = (V, E, w)$ $\quad T = (X, E_T, w_T)$

$g_1 \; g_2 \; g_3 \; g_4 \qquad g'_1 \; g'_2 \; g'_3 \; g'_4$

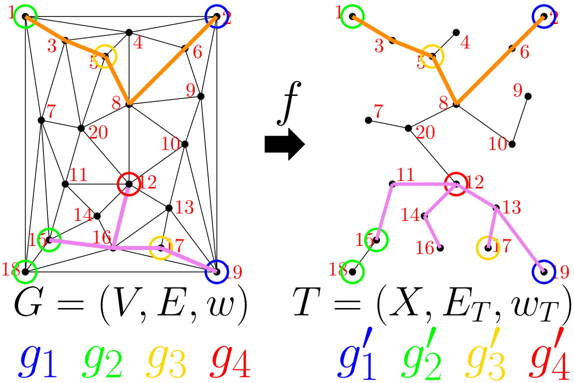
$\tilde{S} = \cup_{\{v', u'\} \in \tilde{S}_T} P^T_{v', u'}$ is a union of shortest paths in G Return $\tilde{S}$.

GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

$S^\star$ optimal solution.

$S^\star_T$: $\forall (u,v) \in S^\star$ add the path from $f(u)$ to $f(v)$.
$\mathbb{E}[w_T(S^\star_T)] = O(\log n) \cdot w(S^\star)$

$\tilde{S}_T$ solution by [GKR00], $w(\tilde{S}_T) \leq O(\log n \cdot \log k) \cdot w(S^\star_T)$.



$\tilde{S} = \cup_{\{v',u'\} \in \tilde{S}_T} P^T_{v',u'}$ is a union of shortest paths in G Return $\tilde{S}$.

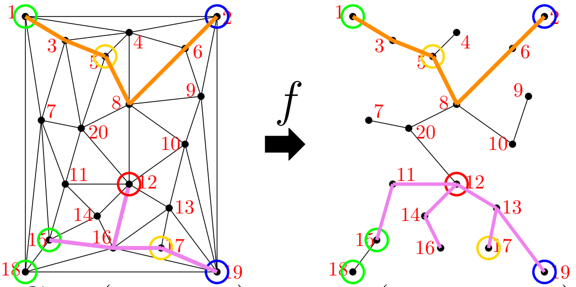
$$\mathbb{E}[w_T(\tilde{S})] \leq \mathbb{E}[w_T(\tilde{S}_T)]$$
$$\leq O(\log n \cdot \log k) \cdot \mathbb{E}[w(S^\star_T)] \leq O(\log^2 n \cdot \log k) \cdot w(S^\star)$$

GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

$S^\star$ optimal solution.

$S^\star_T$: $\forall (u,v) \in S^\star$ add the path from $f(u)$ to $f(v)$.
$\mathbb{E}[w_T(S^\star_T)] = O(\log n) \cdot w(S^\star)$

$\tilde{S}_T$ solution by [GKR00],
$w(\tilde{S}_T) \leq O(\log n \cdot \log k) \cdot w(S^\star_T)$.



$G = (V, E, w)$ $\quad$ $T = (X, E_T, w_T)$

g_1 g_2 g_3 g_4 $\quad$ g'_1 g'_2 g'_3 g'_4

$\tilde{S} = \cup_{\{v',u'\} \in \tilde{S}_T} P^T_{v',u'}$ is a union of shortest paths in G $\quad$ Return $\tilde{S}$.

$\mathbb{E}[w_T(\tilde{S})] \leq O(\log^2 n \cdot \log k) \cdot w(S^\star)$.

We got an $O(\log^2 n \cdot \log k)$ **approximation** (in expectation)

GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

$\tilde{S} = \cup_{\{v',u'\} \in \tilde{S}_T} P^T_{v',u'}$ is a union of shortest paths in G Return $\tilde{S}$.

$\mathbb{E}[w_T(\tilde{S})] \leq O(\log^2 n \cdot \log k) \cdot w(S^\star)$.

We got an $O(\log^2 n \cdot \log k)$ **approximation** (in expectation)

How to get the approximation guarantee with high probability?

GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

$\tilde{S} = \cup_{\{v',u'\} \in \tilde{S}_T} P^T_{v',u'}$ is a union of shortest paths in G Return $\tilde{S}$.

$\mathbb{E}[w_T(\tilde{S})] \leq O(\log^2 n \cdot \log k) \cdot w(S^\star)$.

We got an $O(\log^2 n \cdot \log k)$ **approximation** (in expectation)

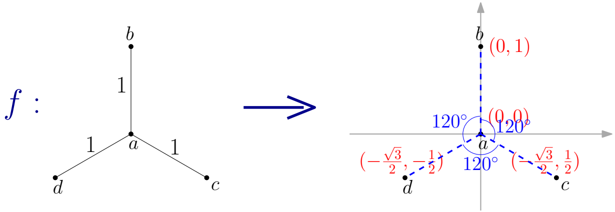
How to get the approximation guarantee with high probability?

Repeat the process $O(\log n)$ times, and return the observed solution of minimum weight.

Outline of the talk

$f : (X, d_X) \to (Y, d_Y)$ has **distortion** t if:

$$\forall x, y \in X, d_X(x, y) \leq d_Y(f(x), f(y)) \leq t \cdot d_X(x, y) \ .$$



	a	b	c	d
a		1	1	1
b	1		2	2
c	1	2		2
d	1	2	2	

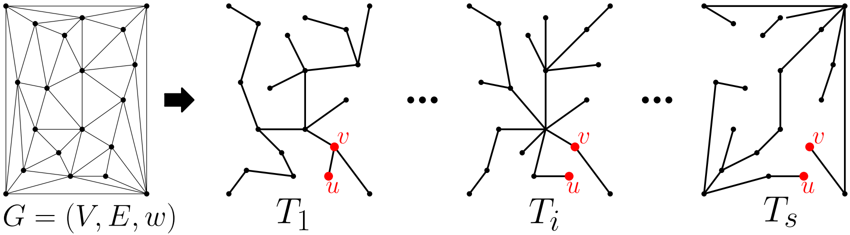
	a	b	c	d
a		$2/\sqrt{3}$	$2/\sqrt{3}$	$2/\sqrt{3}$
b	$2/\sqrt{3}$		2	2
c	$2/\sqrt{3}$	2		2
d	$2/\sqrt{3}$	2	2	

The distortion of the embedding is $\frac{2}{\sqrt{3}} \approx 1.1547$.

Stochastic Embedding into Trees

Theorem ([Fakcharoenphol, Rao, Talwar 04], improving [Bartal 96+98])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log n)$.



For every $u, v \in X$ and $T \in \mathrm{supp}(\mathcal{D})$, $d_X(u, v) \leq d_T(f(u), f(v))$.

For every $u, v \in X$ $\quad \mathbb{E}_{T\sim\mathcal{D}}[d_T(f(u), f(v))] \leq O(\log n) \cdot d_X(u, v)$.

[Alon, Karp, Peleg, West 95]: Tight!

Distance Oracle construction

Sample $s = 4\log n$ trees $T_1, \ldots, T_s$. Given x, y return $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y)$.

Clearly, as the trees are **dominating**, $\mathrm{DO}(x, y) = \min_{i\in[1,s]} d_{T_i}(x, y) \geq d_X(x, y)$.

Thus with high probability, for every $x, y \in X$

$$\mathrm{DO}(x, y) < 2 \cdot \mathbb{E}_{T\sim\mathcal{D}}[d_T(x, y)] = O(\log n) \cdot d_G(x, y) \ .$$

Space: storing $O(\log n)$ trees. Total space is $O(n \log n)$ (machine words).

Query time: computing $d_{T_i}(x, y)$ for $i \in [1, s]$.

There is a data structure computing distance in (some) trees in $O(1)$ time.

Overall $O(\log n)$ query time.

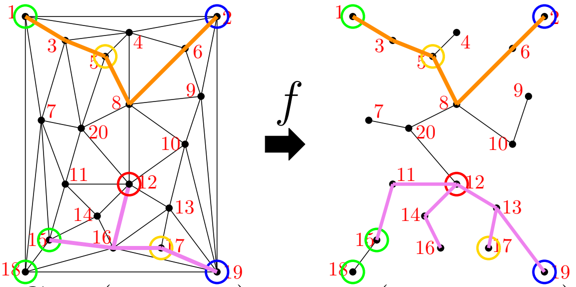
Overall we obtained distance approximation $O(\log n)$ with $O(\log n)$ query time and $O(n \log n)$ space.

GST: Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

$S^\star$ optimal solution.

$S^\star_T$: $\forall (u,v) \in S^\star$ add the path from $f(u)$ to $f(v)$.
$\mathbb{E}[w_T(S^\star_T)] = O(\log n) \cdot w(S^\star)$

$\tilde{S}_T$ solution by [GKR00],
$w(\tilde{S}_T) \le O(\log n \cdot \log k) \cdot w(S^\star_T)$.



$G = (V, E, w)$ $\quad T = (X, E_T, w_T)$

$g_1 \; g_2 \; g_3 \; g_4 \qquad g'_1 \; g'_2 \; g'_3 \; g'_4$

$\tilde{S} = \cup_{\{v',u'\} \in \tilde{S}_T} P^T_{v',u'}$ is a union of shortest paths in G Return $\tilde{S}$.

$\mathbb{E}[w_T(\tilde{S})] \le O(\log^2 n \cdot \log k) \cdot w(S^\star)$.

We got an $O(\log^2 n \cdot \log k)$ **approximation** (in expectation)

I am looking for master students!

Did you enjoyed the lecture? Do you like designing and analyzing algorithms?

I am looking for master students!

Did you enjoyed the lecture? Do you like designing and analyzing algorithms?

There are many open problems in the field waiting to be solved!

I am looking for master students!

Did you enjoyed the lecture? Do you like designing and analyzing algorithms?

There are many open problems in the field waiting to be solved!

Come and talk to me, there are fun and challenging problems (and a generous scholarship) waiting for you.

I am looking for master students!

Did you enjoyed the lecture? Do you like designing and analyzing algorithms?

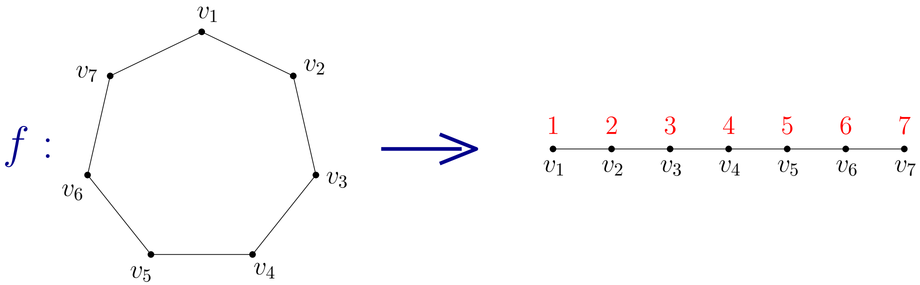
There are many open problems in the field waiting to be solved!

Come and talk to me, there are fun and challenging problems (and a generous scholarship) waiting for you.

You can learn about my research from many different videos in my home-page.

Quiz.

Q0: Consider an embedding of the circle graph C_7 into the line, such that the vertices $v_1, v_2, v_3, v_4, v_5, v_6, v_7$ are mapped to $\{1,2,3,4,5,6,7\}$ respectively.
What is the distortion?

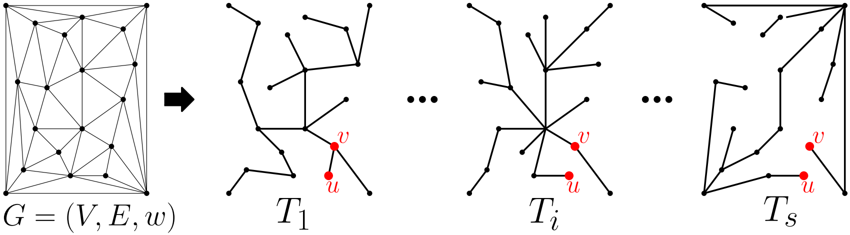


Quiz. Consider a graph family called *Graphica Prime*, such that every n-graph G in the family embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $\boldsymbol{t}$.

Stochastic Embedding into Trees

Theorem (Stochastic embedding for graph family $\mathcal{G}$)

*Every graph $G = (V, E, w)$ in Graphica Prime embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $\boldsymbol{t}$.*



For every $u, v \in X$ and $T \in \mathrm{supp}(\mathcal{D})$, $d_X(u, v) \leq d_T(f(u), f(v))$.

For every $u, v \in X$ $\quad \mathbb{E}_{T\sim\mathcal{D}}[d_T(f(u), f(v))] \leq \boldsymbol{t} \cdot d_X(u, v)$.

Quiz.

Consider a graph family called *Graphica Prime*, such that every n-graph G in the family embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $\boldsymbol{t}$. Using similar techniques to what we did in class:

Q1: What **distance oracle** can you achieve for graphs in *Graphica Prime*?

Q2: What **approximation factor** can you obtain for graphs in *Graphica Prime* for the group Steiner tree problem?

Quiz.

Consider a graph family called *Graphica Prime*, such that every n-graph G in the family embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** t.

Using similar techniques to what we did in class:

Q1: What **distance oracle** can you achieve for graphs in *Graphica Prime*?

Q2: What **approximation factor** can you obtain for graphs in *Graphica Prime* for the group Steiner tree problem?

Link to quiz:

Can also be found in my homepage:

arnold.filtser.com

Or just google Arnold Filtser.

Link to slides:

Outline of the talk - Appendix

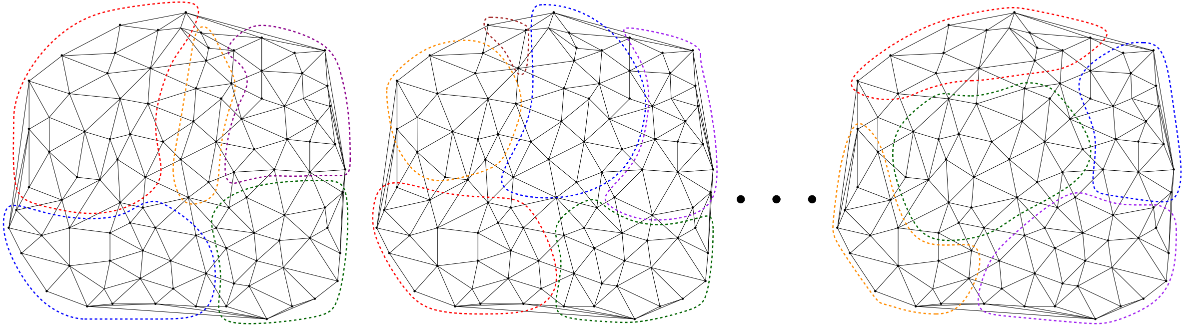
We will begin our tour of metric embeddings into trees with the classics: [Bartal 96]

We will begin our tour of metric embeddings into trees with the classics: [Bartal 96]

This one is based on random partitions of metric spaces.

Definition (Padded Decomposition)

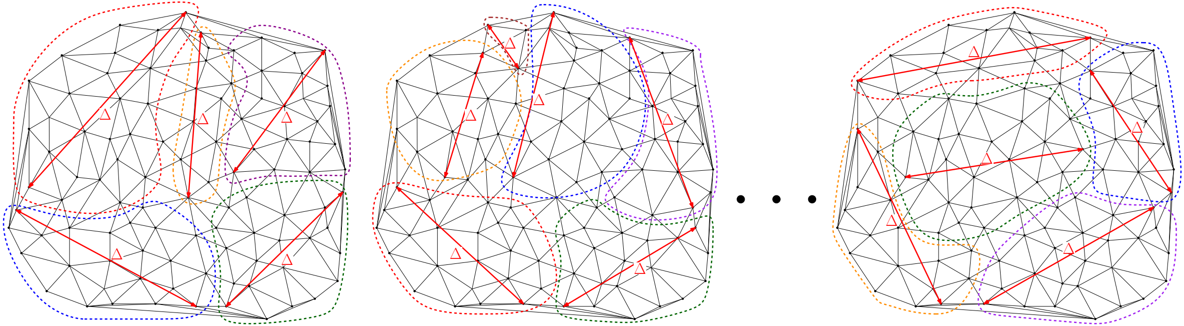
Given a metric space (X, d_X) (or a weight graph $G = (V, E, w)$).
Distribution $\mathcal{D}$ over partitions of G is (β, Δ)-padded decomposition if:



Definition (Padded Decomposition)

Given a metric space (X, d_X) (or a weight graph $G = (V, E, w)$).
Distribution $\mathcal{D}$ over partitions of G is (β, Δ)-padded decomposition if:

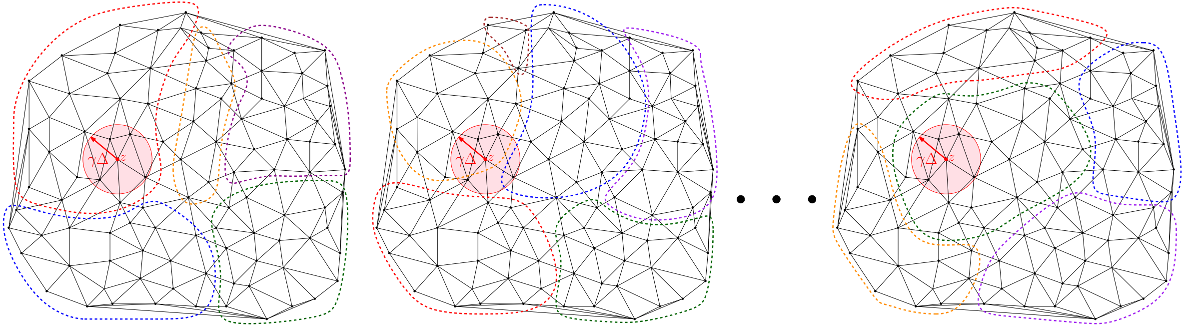
- For every cluster $C \in \mathcal{P} \sim \mathcal{D}$, $\mathbf{diam(C)} \leq \Delta$.



Definition (Padded Decomposition)

Given a metric space (X, d_X) (or a weight graph $G = (V, E, w)$).
Distribution $\mathcal{D}$ over partitions of G is (β, Δ)-padded decomposition if:

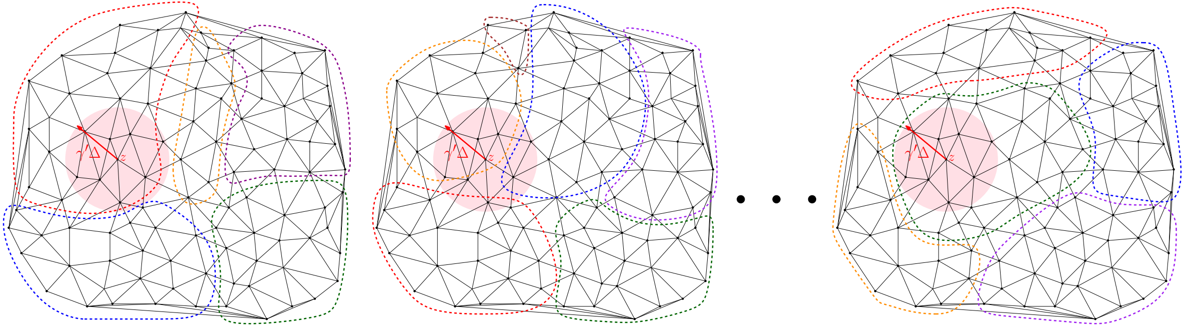
- For every cluster $C \in \mathcal{P} \sim \mathcal{D}$, $\mathbf{diam}(C) \leq \Delta$.
- For every small $0 \leq \gamma$, and $z \in V$, $\Pr[B(z, \gamma\Delta) \subseteq P(z)] \geq e^{-\beta\gamma}$.



Definition (Padded Decomposition)

Given a metric space (X, d_X) (or a weight graph $G = (V, E, w)$).
Distribution $\mathcal{D}$ over partitions of G is (β, Δ)-padded decomposition if:

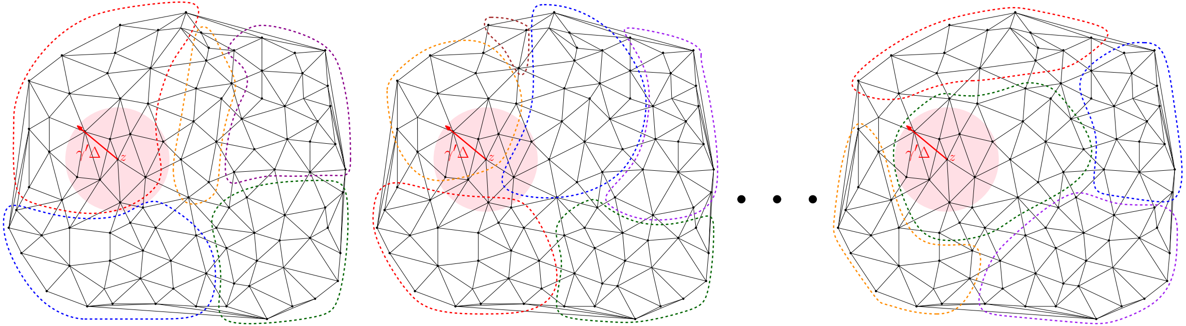
- For every cluster $C \in \mathcal{P} \sim \mathcal{D}$, $\mathbf{diam}(C) \le \Delta$.
- For every small $0 \le \gamma$, and $z \in V$, $\Pr[B(z, \gamma\Delta) \subseteq P(z)] \ge e^{-\beta\gamma}$.



Definition (Padded Decomposition)

Given a metric space (X, d_X) (or a weight graph $G = (V, E, w)$).
Distribution $\mathcal{D}$ over partitions of G is (β, Δ)-padded decomposition if:

- For every cluster $C \in \mathcal{P} \sim \mathcal{D}$, $\mathbf{diam(C)} \leq \Delta$.
- For every small $0 \leq \gamma$, and $z \in V$, $\Pr[B(z, \gamma\Delta) \subseteq P(z)] \geq e^{-\beta\gamma}$.



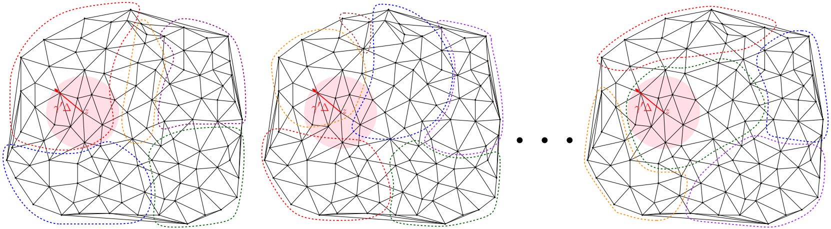
G admits a β-padded decomposition **scheme**:

$\forall \Delta > 0$, G admits (β, Δ)-padded decomposition.

Definition (Padded Decomposition)

Given a metric space (X, d_X) (or a weight graph $G = (V, E, w)$).
Distribution $\mathcal{D}$ over partitions of G is (β, Δ)-padded decomposition if:

- For every cluster $C \in \mathcal{P} \sim \mathcal{D}$, $\mathbf{diam}(C) \leq \Delta$.
- For every small $0 \leq \gamma$, and $z \in V$, $\Pr[B(z, \gamma\Delta) \subseteq P(z)] \geq e^{-\beta\gamma}$.

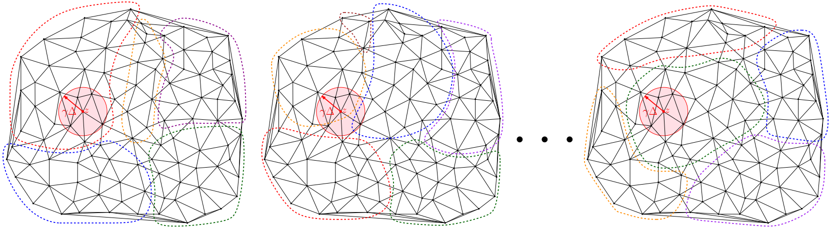


Note: $\Pr[B(z, \frac{1}{\beta} \cdot \Delta) \subseteq P(z)] \geq \Omega(1)$.

Definition (Padded Decomposition)

Given a metric space (X, d_X) (or a weight graph $G = (V, E, w)$).
Distribution $\mathcal{D}$ over partitions of G is (β, Δ)-padded decomposition if:

- For every cluster $C \in \mathcal{P} \sim \mathcal{D}$, $\mathbf{diam}(C) \leq \Delta$.
- For every small $0 \leq \gamma$, and $z \in V$, $\quad \Pr[B(z, \gamma\Delta) \subseteq P(z)] \geq e^{-\beta\gamma}$.



Note: $\Pr[B(z, \frac{1}{\beta} \cdot \Delta) \subseteq P(z)] \geq \Omega(1)$.

For small enough γ, cut probability: $\Pr[B(z, \gamma\Delta) \not\subseteq P(z)] \leq 1 - e^{-\beta\gamma} \approx \beta\gamma$.

Definition (Padded Decomposition)

Given a metric space (X, d_X) (or a weight graph $G = (V, E, w)$).
Distribution $\mathcal{D}$ over partitions of G is (β, Δ)-padded decomposition if:

- For every cluster $C \in \mathcal{P} \sim \mathcal{D}$, $\mathbf{diam}(C) \le \Delta$.
- For every small $0 \le \gamma$, and $z \in V$, $\quad \Pr[B(z, \gamma\Delta) \subseteq P(z)] \ge e^{-\beta\gamma}$.

Theorem ([Bartal 96])

Every n-point metric space admits an $O(\log n)$-padded decomposition scheme.

Definition (Padded Decomposition)

Given a metric space (X, d_X) (or a weight graph $G = (V, E, w)$).
Distribution $\mathcal{D}$ over partitions of G is (β, Δ)-padded decomposition if:

- For every cluster $C \in \mathcal{P} \sim \mathcal{D}$, $\mathbf{diam}(C) \leq \Delta$.
- For every small $0 \leq \gamma$, and $z \in V$, $\quad \Pr[B(z, \gamma\Delta) \subseteq P(z)] \geq e^{-\beta\gamma}$.

Theorem ([Bartal 96])

Every n-point metric space admits an $O(\log n)$-padded decomposition scheme.

This is also tight! [Bartal 96]

Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

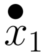
1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \mathrm{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
    1. Sample $r_i \sim \mathrm{Exp}(1)$.
    2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

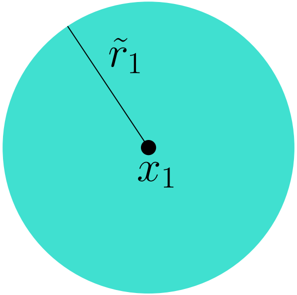


Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \text{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

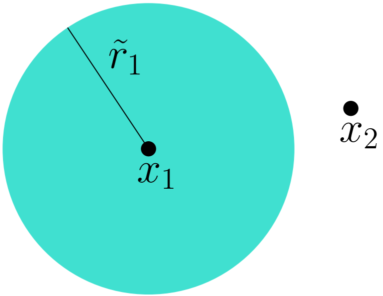


Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \mathrm{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

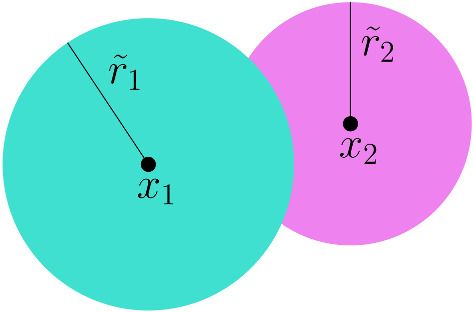


Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \mathrm{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

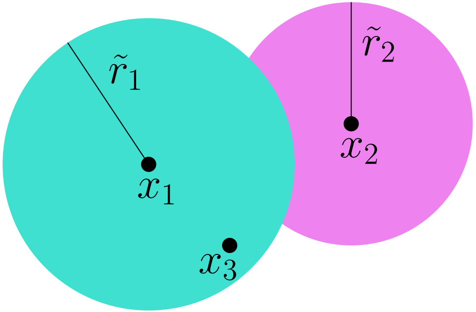


Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \text{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

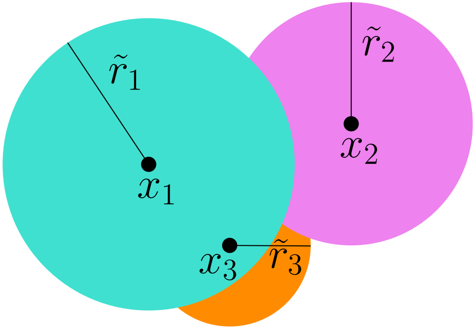


Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
    1. Sample $r_i \sim \mathrm{Exp}(1)$.
    2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

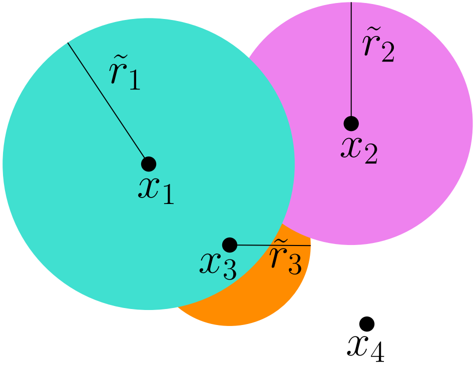


Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
    1. Sample $r_i \sim \text{Exp}(1)$.
    2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

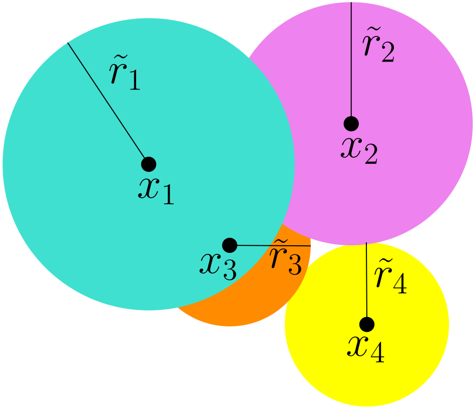


Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
    1. Sample $r_i \sim \mathrm{Exp}(1)$.
    2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

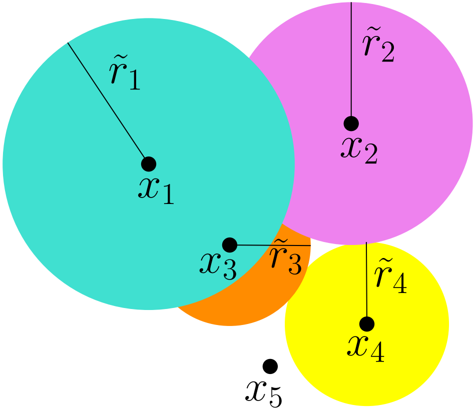


Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \mathrm{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

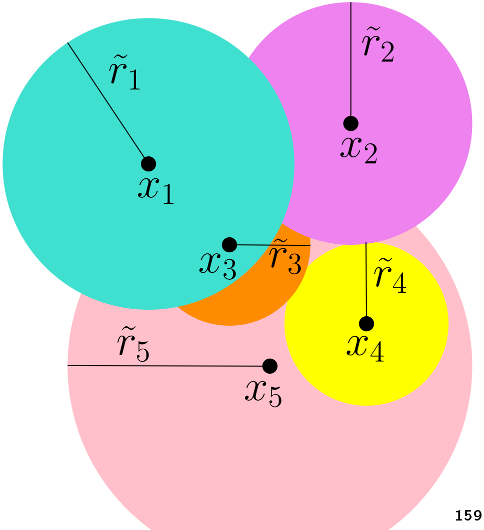


Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \text{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.



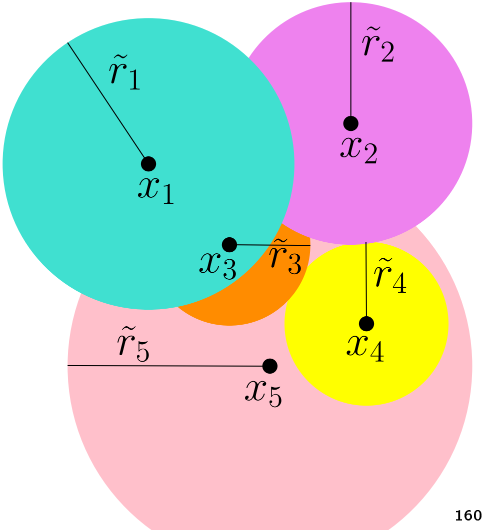
Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \mathrm{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

W.h.p. $\forall i,\ r_i \leq \frac{c}{2} \cdot \log n$



Theorem ([Bartal 96])

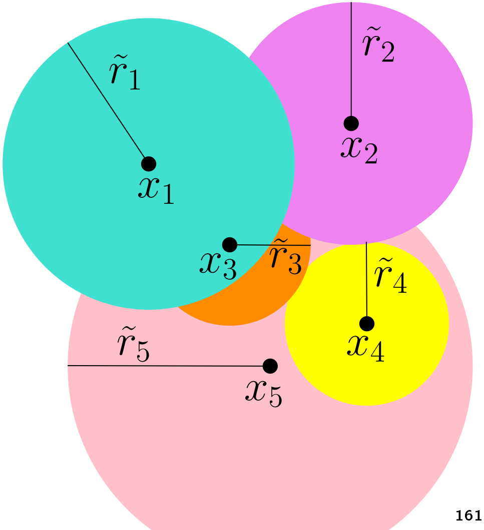
Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
    1. Sample $r_i \sim \mathrm{Exp}(1)$.
    2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

W.h.p. $\forall i,\ r_i \le \frac{c}{2} \cdot \log n$

Thus all the sampled radii $\le \frac{\Delta}{2}$.



Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

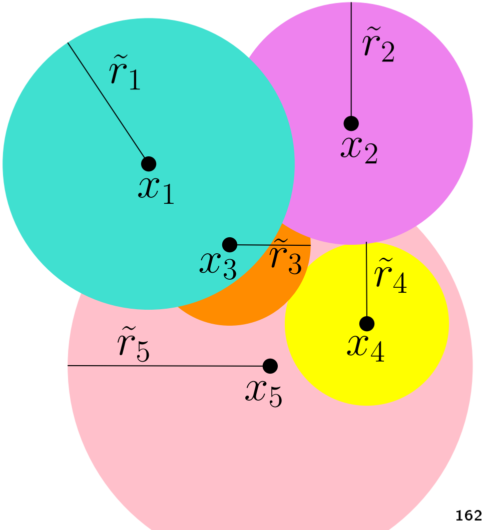
Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
    1. Sample $r_i \sim \text{Exp}(1)$.
    2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

W.h.p. $\forall i,\ r_i \leq \frac{c}{2} \cdot \log n$

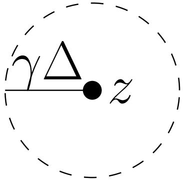
Thus all the sampled radii $\leq \frac{\Delta}{2}$.

$\Rightarrow$ all clusters have diameter $\leq \Delta$.



Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.



Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \mathrm{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

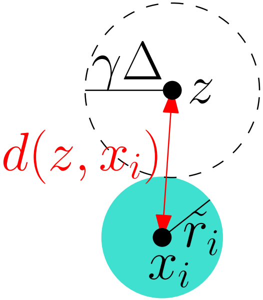
$\Pr\left[B(z, \gamma\Delta) \subseteq P(z)\right] \geq ??$

Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \mathrm{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.



$\Pr\left[B(z, \gamma\Delta) \subseteq P(z)\right] \geq ??$

i is the first s.t. $C_i \cap B(z, \gamma\Delta) \neq \emptyset$

Theorem ([Bartal 96])

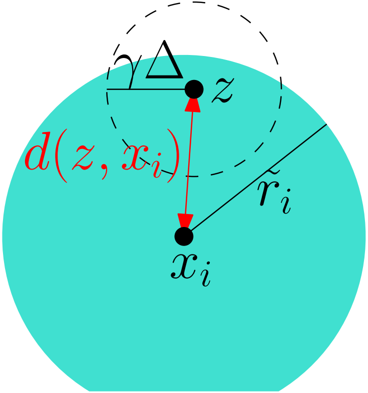
Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \mathrm{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.

$\Pr\left[B(z, \gamma\Delta) \subseteq P(z)\right] \geq ??$

i is the first s.t. $C_i \cap B(z, \gamma\Delta) \neq \emptyset$

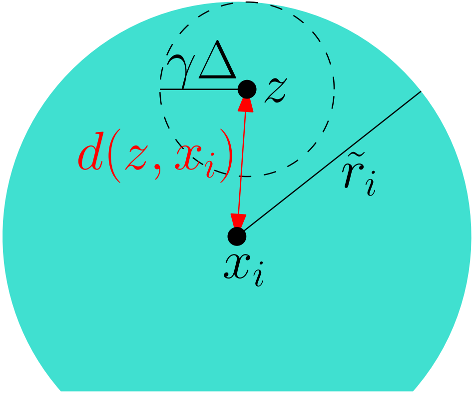


Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
    1. Sample $r_i \sim \text{Exp}(1)$.
    2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.



$\Pr\left[B(z, \gamma\Delta) \subseteq P(z)\right] \geq ??$

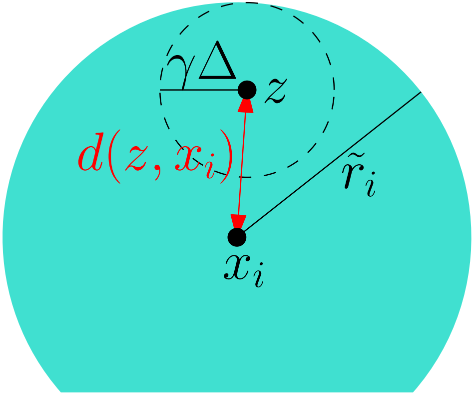
i is the first s.t. $C_i \cap B(z, \gamma\Delta) \neq \emptyset$

Theorem ([Bartal 96])

Every n-point metric space (X, d_X) admits an $O(\log n)$-padded decomposition scheme.

Algorithm:

1. Arbitrarily order X: $x_1, x_2, \ldots, x_n$.
2. For $i = 1$ to n
   1. Sample $r_i \sim \text{Exp}(1)$.
   2. $C_i = B\left(x_i, \tilde{r}_i = r_i \cdot \frac{\Delta}{c \cdot \log n}\right) \setminus \cup_{j<i} C_j$
3. Return $(C_1, C_2 \ldots, C_n)$.



$\Pr\left[B(z, \gamma\Delta) \subseteq P(z)\right] \geq ??$

i is the first s.t. $C_i \cap B(z, \gamma\Delta) \neq \emptyset$

By **Memorylessness**,

$$\Pr\left[B(z, \gamma\Delta) \subseteq C_i \mid B(z, \gamma\Delta) \cap C_i \neq \emptyset\right] \geq \Pr\left[\tilde{r}_i \geq 2\gamma\Delta\right] = e^{-\gamma \cdot 2c \log n}$$

Theorem ([Bartal 96])

Every n-point metric space (X, d) embeds into ***distribution*** *$\mathcal{D}$ over* ***dominating trees*** *with* ***expected distortion*** *$O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

Theorem ([Bartal 96])

*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

Theorem ([Bartal 96])

*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \ldots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \ldots, \mathcal{P}_{\log \Phi}$.

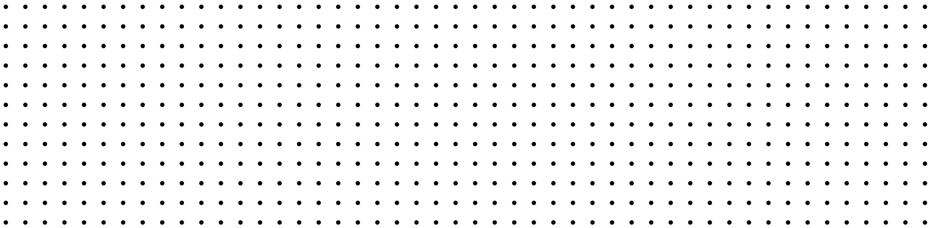
Theorem ([Bartal 96])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log^2 n)$.

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.



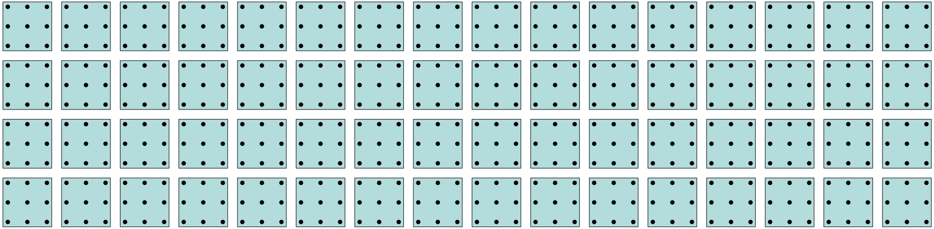
Theorem ([Bartal 96])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log^2 n)$.

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.



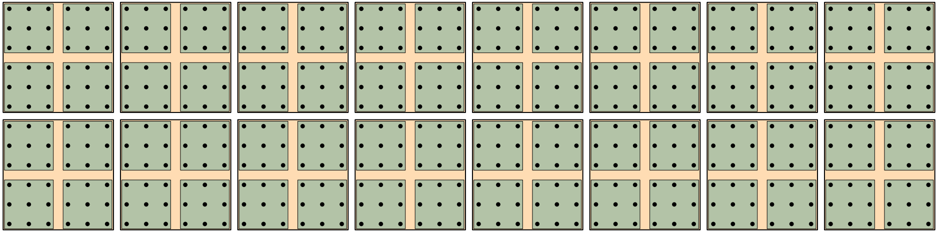
Theorem ([Bartal 96])

*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.



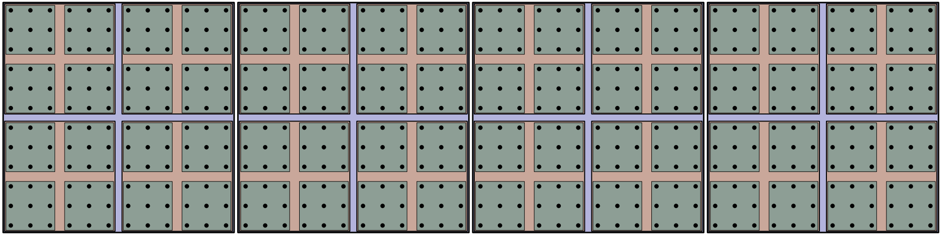
Theorem ([Bartal 96])

*Every n-point metric space (X,d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1,\Phi=n^{10}]$.

For every $i\in[1,\log\Phi]$ sample an $\left(O(\log n),2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1,\tilde{\mathcal{P}}_2,\dots,\tilde{\mathcal{P}}_{\log\Phi}$.
Here x,y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i,\mathcal{P}_{i+1},\dots,\mathcal{P}_{\log\Phi}$.



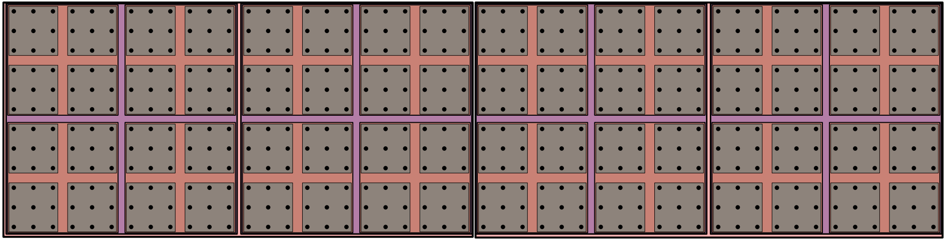
Theorem ([Bartal 96])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log^2 n)$.

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.



Theorem ([Bartal 96])

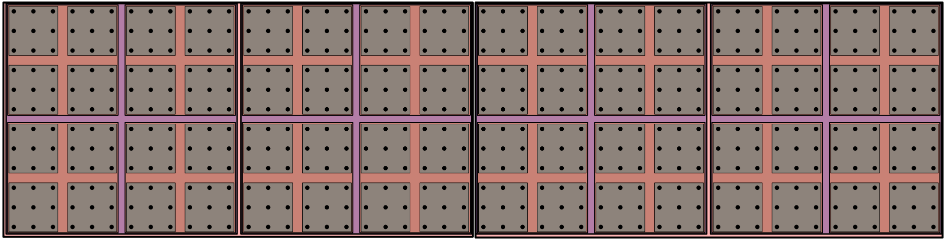
*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

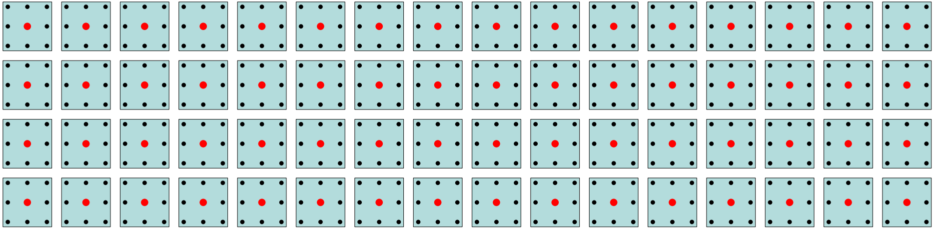
*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

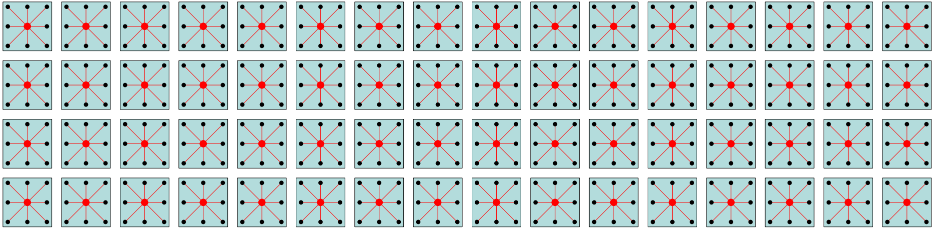
Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log^2 n)$.

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

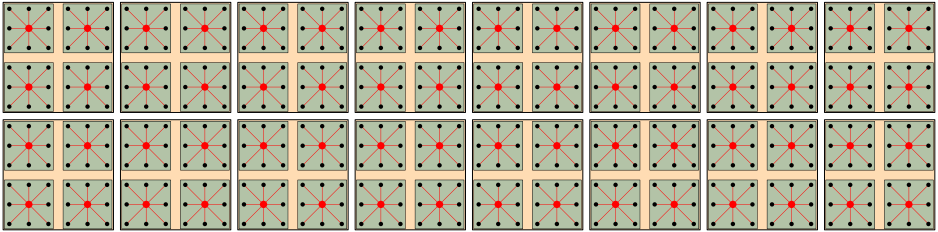
*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

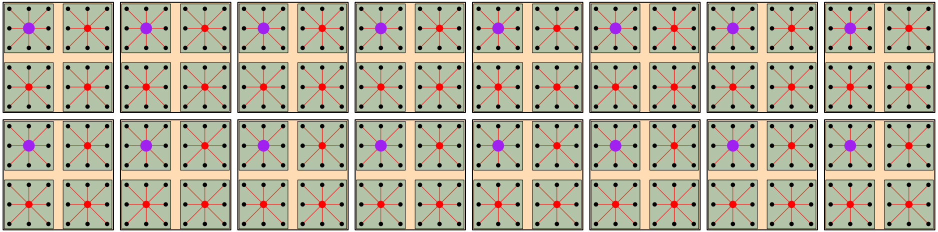
*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log\Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log\Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

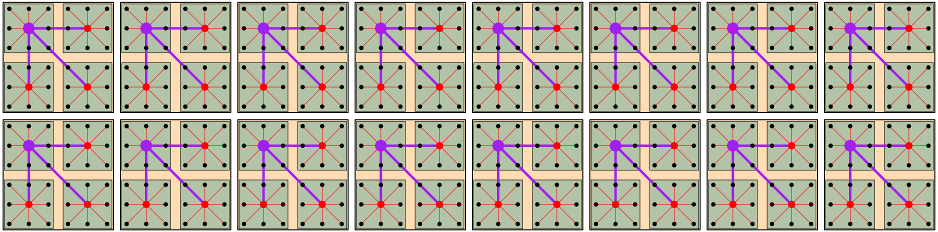
*Every n-point metric space (X,d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1,\Phi=n^{10}]$.

For every $i\in[1,\log\Phi]$ sample an $\left(O(\log n),2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1,\tilde{\mathcal{P}}_2,\dots,\tilde{\mathcal{P}}_{\log\Phi}$.
Here x,y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i,\mathcal{P}_{i+1},\dots,\mathcal{P}_{\log\Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

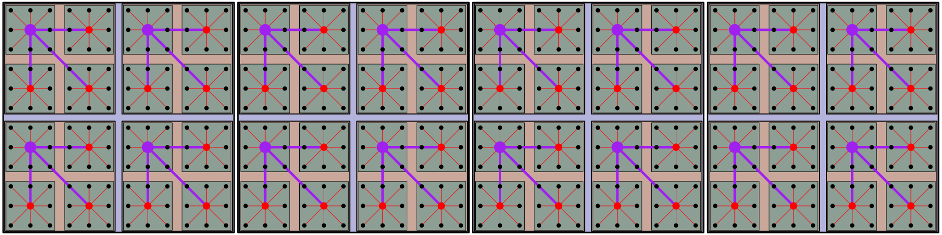
*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

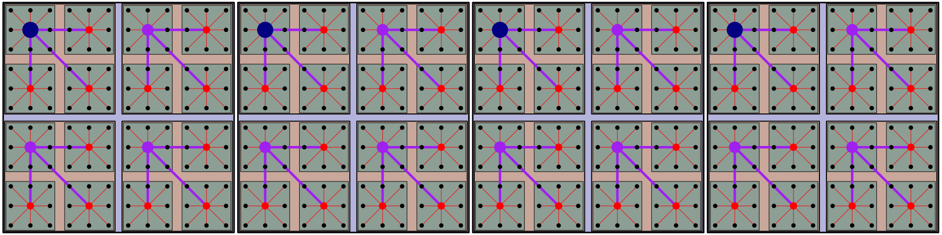
Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log^2 n)$.

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

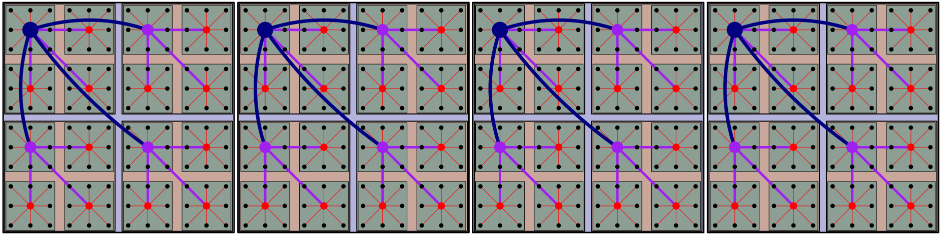
*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

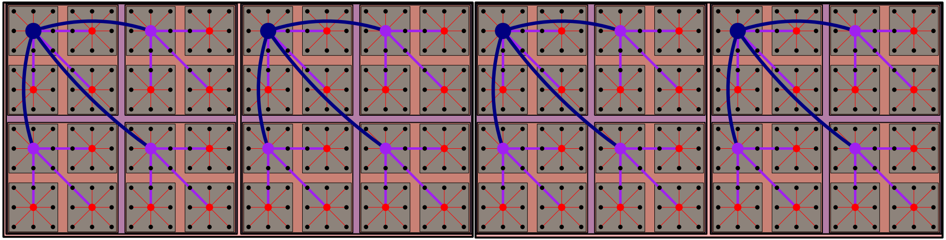
Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log^2 n)$.

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

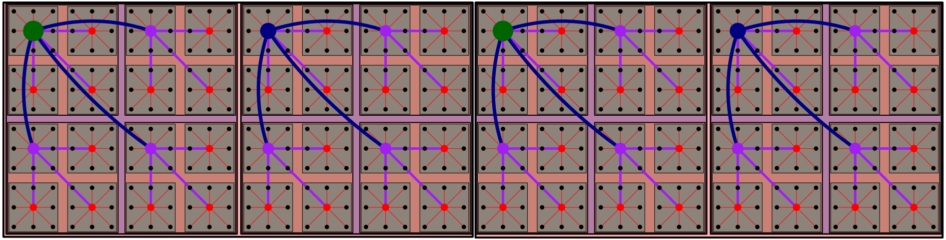
*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log\Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log\Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log\Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

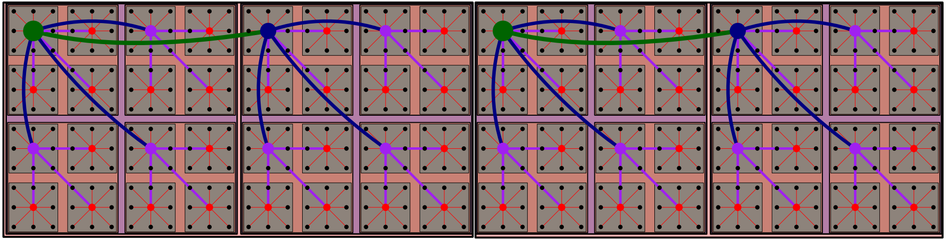
Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log^2 n)$.

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

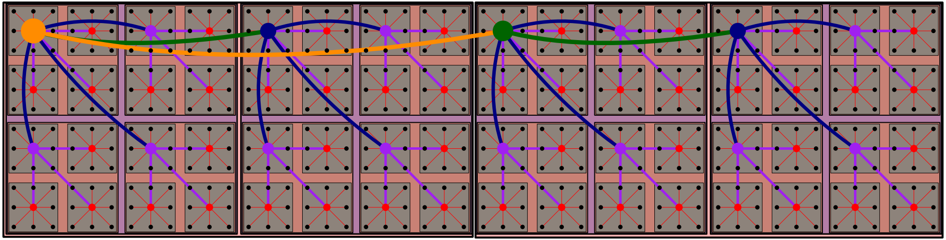
Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log^2 n)$.

For simplicity, we will assume that all the pairwise distances are in $[1, \Phi = n^{10}]$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \dots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Theorem ([Bartal 96])

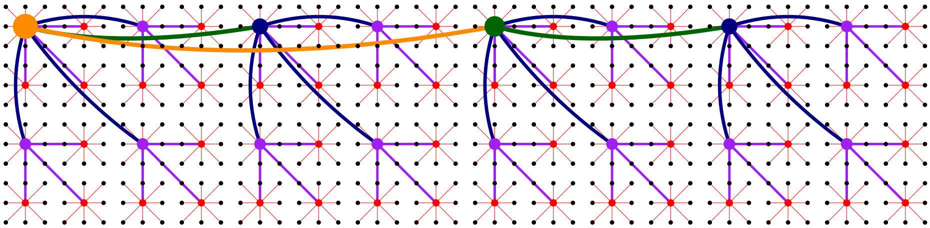
*Every n-point metric space (X,d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

For simplicity, we will assume that all the pairwise distances are in $[1,\Phi=n^{10}]$.

For every $i\in[1,\log\Phi]$ sample an $\left(O(\log n),2^i\right)$ padded decomposition $\mathcal{P}_i$.

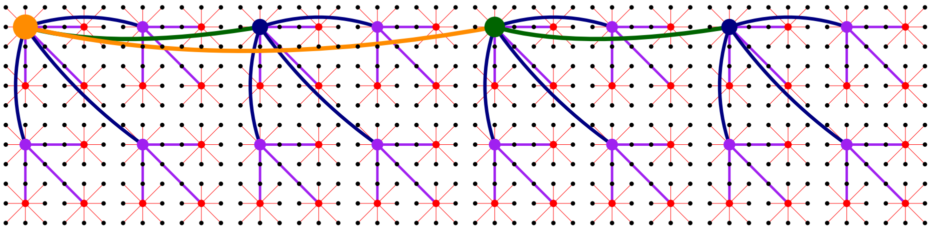
"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1,\tilde{\mathcal{P}}_2,\dots,\tilde{\mathcal{P}}_{\log\Phi}$.
Here x,y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i,\mathcal{P}_{i+1},\dots,\mathcal{P}_{\log\Phi}$.

Put a "tree structure" on top of the laminar partition.



"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \ldots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \ldots, \mathcal{P}_{\log \Phi}$.

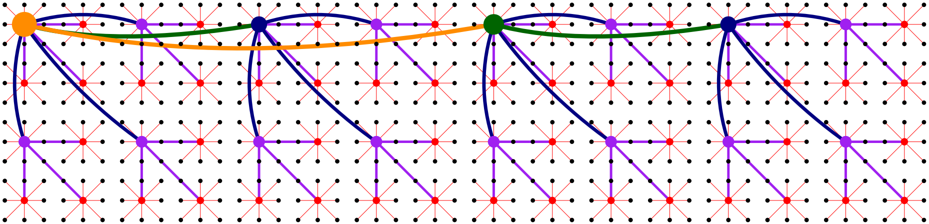
Put a "tree structure" on top of the laminar partition.



The weight of the edges between the $\tilde{\mathcal{P}}_{i-1}$ representatives to their respective $\tilde{\mathcal{P}}_i$ representatives will be 2^i.

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \ldots, \tilde{\mathcal{P}}_{\log\Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \ldots, \mathcal{P}_{\log\Phi}$.

Put a "tree structure" on top of the laminar partition.

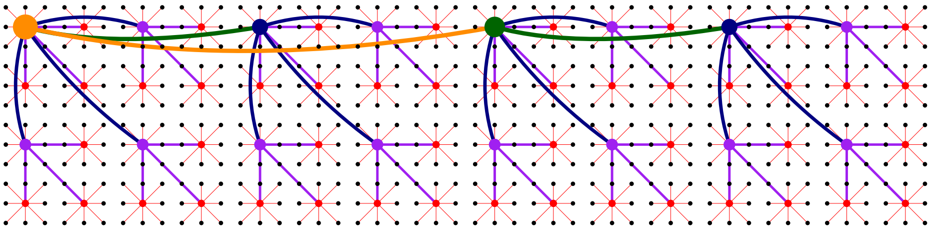


The weight of the edges between the $\tilde{\mathcal{P}}_{i-1}$ representatives to their respective $\tilde{\mathcal{P}}_i$ representatives will be 2^i.

Observation 1: For every $x, y \in X$, $d_X(x, y) \leq d_T(x, y)$

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \ldots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \ldots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



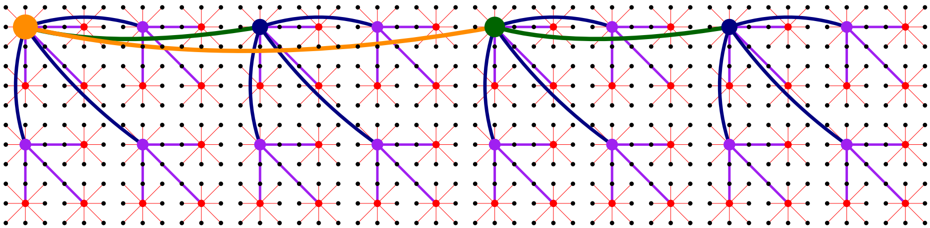
The weight of the edges between the $\tilde{\mathcal{P}}_{i-1}$ representatives to their respective $\tilde{\mathcal{P}}_i$ representatives will be 2^i.

Observation 1: For every $x, y \in X$, $d_X(x, y) \leq d_T(x, y)$

Observation 2: If $\mathcal{P}_i(x) = \mathcal{P}_i(y)$, then $d_T(x, y) \leq O(2^i)$

"Force" them into a laminar partition: $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \ldots, \tilde{\mathcal{P}}_{\log \Phi}$.
Here x, y clustered together in $\tilde{\mathcal{P}}_i$ iff they are clustered together in $\mathcal{P}_i, \mathcal{P}_{i+1}, \ldots, \mathcal{P}_{\log \Phi}$.

Put a "tree structure" on top of the laminar partition.



Observation 1: For every $x, y \in X$, $d_X(x, y) \leq d_T(x, y)$

Observation 2: If $\mathcal{P}_i(x) = \mathcal{P}_i(y)$, then $d_T(x, y) \leq O(2^i)$

$d_T(x, y) = O(2^{i_{x,y}})$ where $i_{x,y}$ is the maximum index such that $\mathcal{P}_i(x) \neq \mathcal{P}_i(y)$.

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

Observation 1: For every $x, y \in X$, $d_X(x,y) \le d_T(x,y)$

Observation 2: If $\mathcal{P}_i(x) = \mathcal{P}_i(y)$, then $d_T(x,y) \le O(2^i)$

$d_T(x,y) = O(2^{i_{x,y}})$ where $i_{x,y}$ is the maximum index such that $\mathcal{P}_i(x) \neq \mathcal{P}_i(y)$.

$$\mathbb{E}[d_T(x,y)] \le \sum_{i=0}^{\log \Phi} O(2^i) \cdot \Pr\left[\mathcal{P}_i(x) \neq \mathcal{P}_i(y)\right]$$

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

Observation 1: For every $x, y \in X$, $d_X(x,y) \leq d_T(x,y)$

Observation 2: If $\mathcal{P}_i(x) = \mathcal{P}_i(y)$, then $d_T(x,y) \leq O(2^i)$

$d_T(x,y) = O(2^{i_{x,y}})$ where $i_{x,y}$ is the maximum index such that $\mathcal{P}_i(x) \neq \mathcal{P}_i(y)$.

$$\begin{aligned}\mathbb{E}[d_T(x,y)] &\leq \sum_{i=0}^{\log \Phi} O(2^i) \cdot \Pr\left[\mathcal{P}_i(x) \neq \mathcal{P}_i(y)\right] \\ &\leq \sum_{i=0}^{\log \Phi} O(2^i) \cdot \frac{d_X(x,y)}{2^i} \cdot O(\log n)\end{aligned}$$

For every $i \in [1, \log \Phi]$ sample an $\big(O(\log n), 2^i\big)$ padded decomposition $\mathcal{P}_i$.

Observation 1: For every $x, y \in X$, $d_X(x, y) \le d_T(x, y)$

Observation 2: If $\mathcal{P}_i(x) = \mathcal{P}_i(y)$, then $d_T(x, y) \le O(2^i)$

$d_T(x, y) = O(2^{i_{x,y}})$ where $i_{x,y}$ is the maximum index such that $\mathcal{P}_i(x) \neq \mathcal{P}_i(y)$.

$$\begin{aligned}
\mathbb{E}[d_T(x, y)] &\le \sum_{i=0}^{\log \Phi} O(2^i) \cdot \Pr\left[\mathcal{P}_i(x) \neq \mathcal{P}_i(y)\right] \\
&\le \sum_{i=0}^{\log \Phi} O(2^i) \cdot \frac{d_X(x, y)}{2^i} \cdot O(\log n) \\
&= O(\log^2 n) \cdot d_X(x, y) \ .
\end{aligned}$$

For every $i \in [1, \log \Phi]$ sample an $\left(O(\log n), 2^i\right)$ padded decomposition $\mathcal{P}_i$.

Observation 1: For every $x, y \in X$, $d_X(x, y) \le d_T(x, y)$

Observation 2: If $\mathcal{P}_i(x) = \mathcal{P}_i(y)$, then $d_T(x, y) \le O(2^i)$

$d_T(x, y) = O(2^{i_{x,y}})$ where $i_{x,y}$ is the maximum index such that $\mathcal{P}_i(x) \ne \mathcal{P}_i(y)$.

$$
\begin{aligned}
\mathbb{E}[d_T(x, y)] &\le \sum_{i=0}^{\log \Phi} O(2^i) \cdot \Pr\left[\mathcal{P}_i(x) \ne \mathcal{P}_i(y)\right] \\
&\le \sum_{i=0}^{\log \Phi} O(2^i) \cdot \frac{d_X(x, y)}{2^i} \cdot O(\log n) \\
&= O(\log^2 n) \cdot d_X(x, y) \ .
\end{aligned}
$$

Theorem ([Bartal 96])

Every n-point metric space (X, d) embeds into ***distribution*** *$\mathcal{D}$ over* ***dominating trees*** *with* ***expected distortion*** *$O(\log^2 n)$.*

Theorem ([Bartal 96])

*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

Theorem ([Bartal 96])

*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log^2 n)$.*

Theorem ([Fakcharoenphol, Rao, Talwar 04] , [Bartal 04])

*Every n-point metric space (X, d) embeds into **distribution** $\mathcal{D}$ over **dominating trees** with **expected distortion** $O(\log n)$.*

Theorem ([Bartal 96])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log^2 n)$.

Theorem ([Fakcharoenphol, Rao, Talwar 04] , [Bartal 04])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log n)$.

The improvement is achieved by sampling the padded decomposition in various levels in a **correlated** fashion.

Theorem ([Bartal 96])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log^2 n)$.

Theorem ([Fakcharoenphol, Rao, Talwar 04] , [Bartal 04])

Every n-point metric space (X, d) embeds into ***distribution*** $\mathcal{D}$ *over* ***dominating trees*** *with* ***expected distortion*** $O(\log n)$.

The improvement is achieved by sampling the padded decomposition in various levels in a **correlated** fashion.

Specifically, the probability to cut x, y at scale Δ is

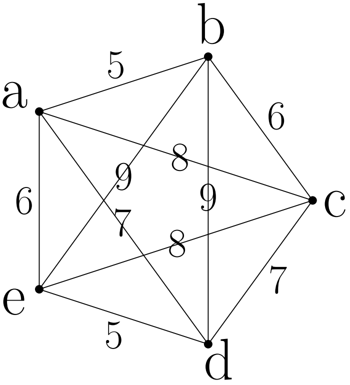
$$\approx \frac{d_X(x,y)}{\Delta} \cdot \log \frac{|B(x, c \cdot 2^i)|}{|B(x, 2^i/c)|}$$

for some constant c, instead of $\approx \frac{d_X(x,y)}{\Delta} \cdot \log n$. Then the sum "telescopes".

Outline of the talk - Appendix

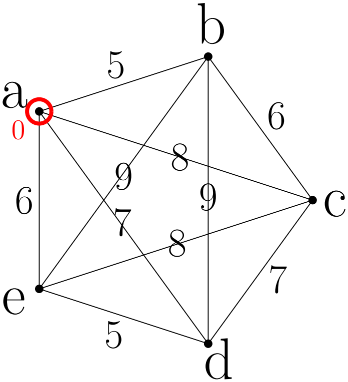
Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.



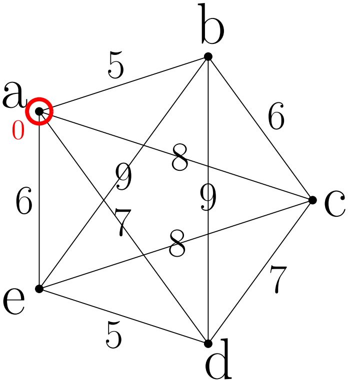
Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.



Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

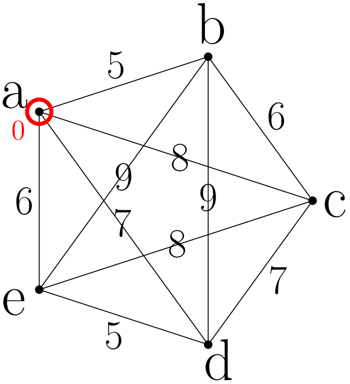


$$f_1(a) = 3$$
$$f_1(b) = 1$$
$$f_1(c) = 2$$
$$f_1(d) = 4$$
$$f_1(e) = 1$$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.



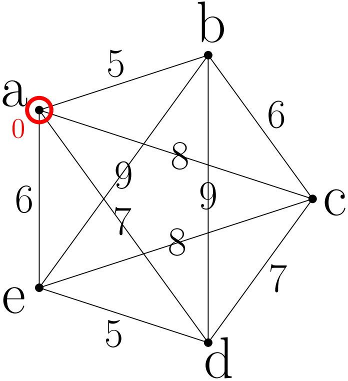
$$f_1(a) = 3$$
$$f_1(b) = 1$$
$$f_1(c) = 2$$
$$f_1(d) = 4$$
$$f_1(e) = 1$$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



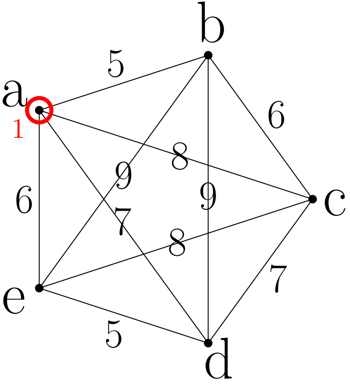
$$f_1(a) = 3$$
$$f_1(b) = 1$$
$$f_1(c) = 2$$
$$f_1(d) = 4$$
$$f_1(e) = 1$$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_1(a) = 3$

$f_1(b) = 1$

$f_1(c) = 2$

$f_1(d) = 4$

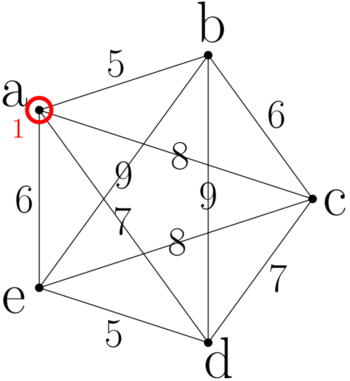
$f_1(e) = 1$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_1(a) = 3$

$f_1(b) = 1$

$f_1(c) = 2$

$f_1(d) = 4$

$f_1(e) = 1$

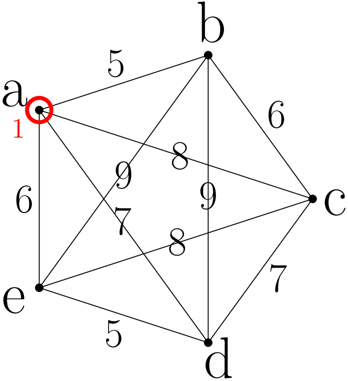
Cost alg(1)=3

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.

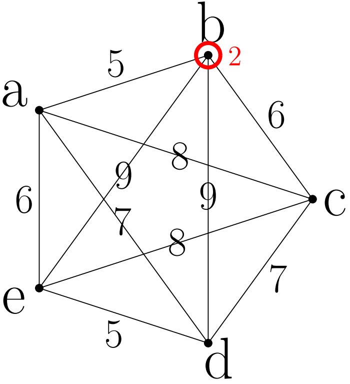


$f_2(a) = 3$
$f_2(b) = 1$
$f_2(c) = 3$
$f_2(d) = 2$
$f_2(e) = 1$

Cost alg(1)=3

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.
Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.
Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_2(a) = 3$$
$$f_2(b) = 1$$
$$f_2(c) = 3$$
$$f_2(d) = 2$$
$$f_2(e) = 1$$

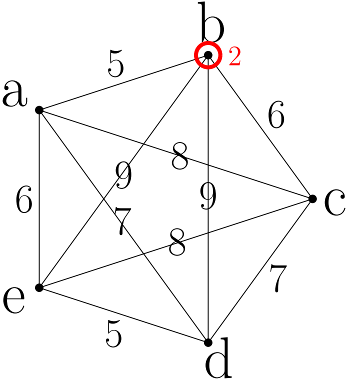
Cost alg(1)=3

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_2(a) = 3$
$f_2(b) = 1$
$f_2(c) = 3$
$f_2(d) = 2$
$f_2(e) = 1$

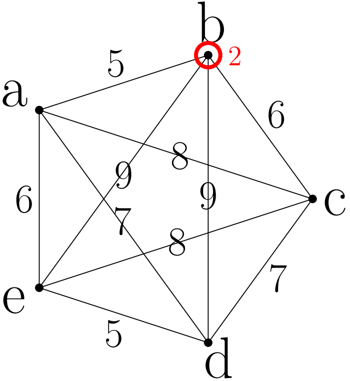
Cost alg(2)$=3+5+1=9$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_3(a) = 1$
$f_3(b) = 4$
$f_3(c) = 3$
$f_3(d) = 2$
$f_3(e) = 1$

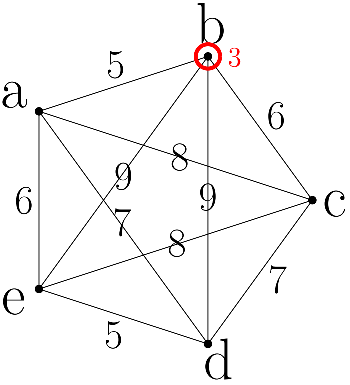
Cost alg(2)$=3 + 5 + 1 = 9$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_3(a) = 1$$
$$f_3(b) = 4$$
$$f_3(c) = 3$$
$$f_3(d) = 2$$
$$f_3(e) = 1$$

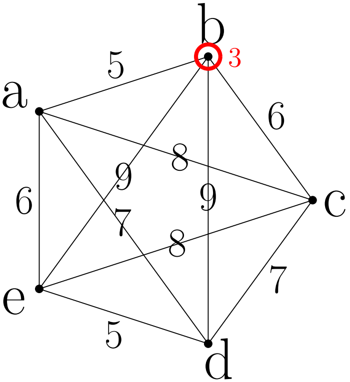
Cost alg(2)$=3 + 5 + 1 = 9$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_3(a) = 1$$
$$f_3(b) = 4$$
$$f_3(c) = 3$$
$$f_3(d) = 2$$
$$f_3(e) = 1$$

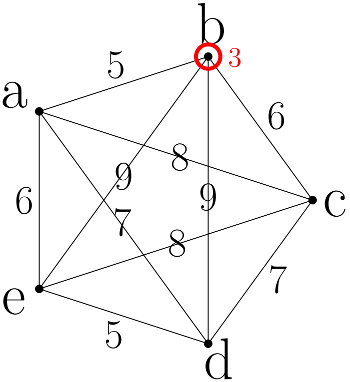
Cost alg(3)$=9 + 4 = 13$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_4(a) = 5$
$f_4(b) = 10$
$f_4(c) = 4$
$f_4(d) = 0$
$f_4(e) = 1$

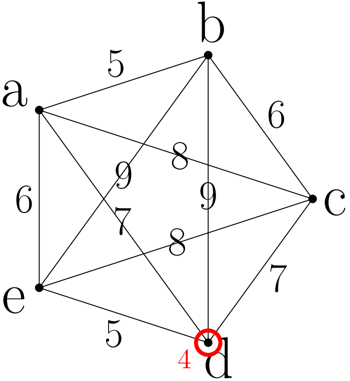
Cost alg(3)$=9 + 4 = 13$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_4(a) = 5$
$f_4(b) = 10$
$f_4(c) = 4$
$f_4(d) = 0$
$f_4(e) = 1$

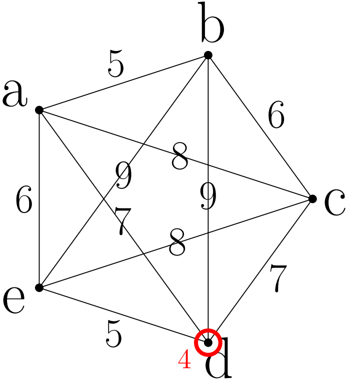
Cost alg(3)$=9 + 4 = 13$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_4(a) = 5$

$f_4(b) = 10$

$f_4(c) = 4$

$f_4(d) = 0$

$f_4(e) = 1$

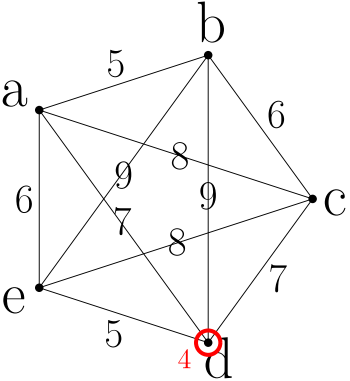
Cost alg(4)$=13 + 9 + 0 = 22$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_5(a) = 3$
$f_5(b) = 4$
$f_5(c) = 4$
$f_5(d) = 5$
$f_5(e) = 1$

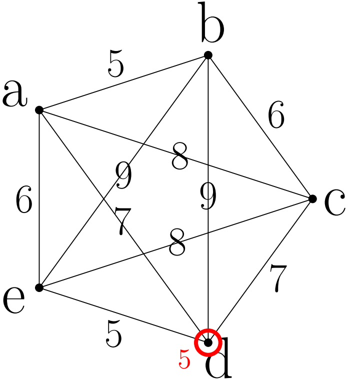
Cost alg(4)=$13 + 9 + 0 = 22$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_5(a) = 3$$
$$f_5(b) = 4$$
$$f_5(c) = 4$$
$$f_5(d) = 5$$
$$f_5(e) = 1$$

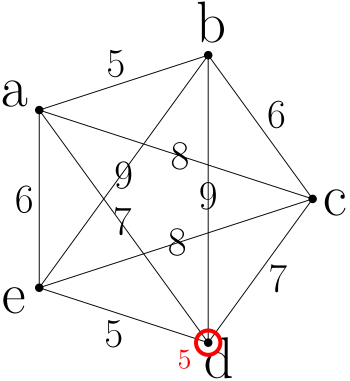
Cost alg(4)$=13 + 9 + 0 = 22$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_5(a) = 3$
$f_5(b) = 4$
$f_5(c) = 4$
$f_5(d) = 5$
$f_5(e) = 1$

Cost alg(5)=$22 + 5 = 27$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.

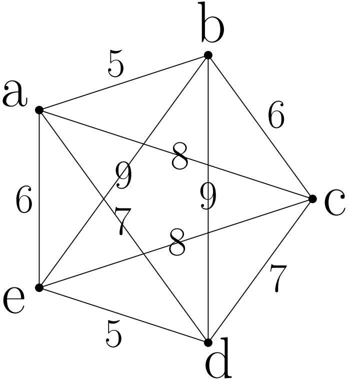
What about Opt?

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.

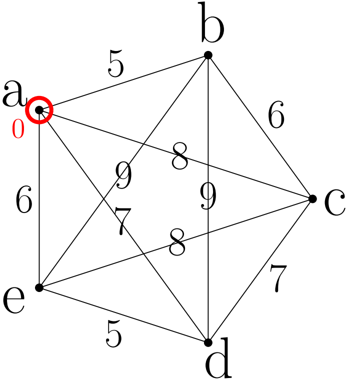


Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.

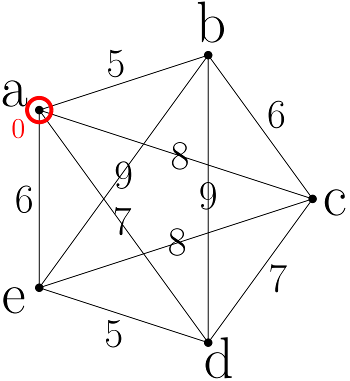


Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



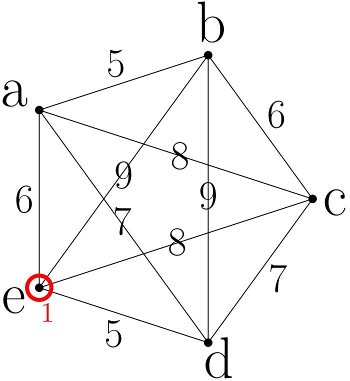
$$f_1(a) = 3$$
$$f_1(b) = 1$$
$$f_1(c) = 2$$
$$f_1(d) = 4$$
$$f_1(e) = 1$$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



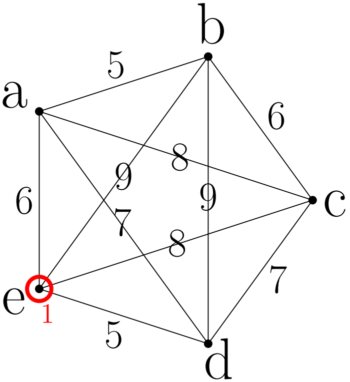
$$f_1(a) = 3$$
$$f_1(b) = 1$$
$$f_1(c) = 2$$
$$f_1(d) = 4$$
$$f_1(e) = 1$$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_1(a) = 3$$
$$f_1(b) = 1$$
$$f_1(c) = 2$$
$$f_1(d) = 4$$
$$f_1(e) = 1$$

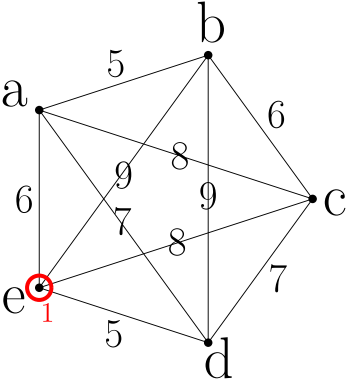
Cost opt(1)$=6+1=7$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_2(a) = 3$
$f_2(b) = 1$
$f_2(c) = 3$
$f_2(d) = 2$
$f_2(e) = 1$

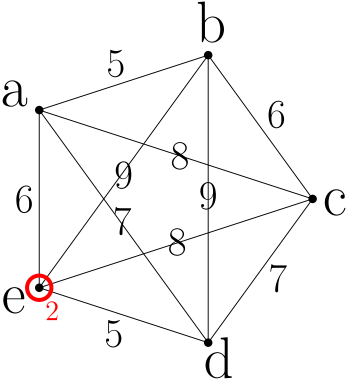
Cost opt(1)$=6+1=7$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_2(a) = 3$$
$$f_2(b) = 1$$
$$f_2(c) = 3$$
$$f_2(d) = 2$$
$$f_2(e) = 1$$

Cost opt(1)$=6 + 1 = 7$

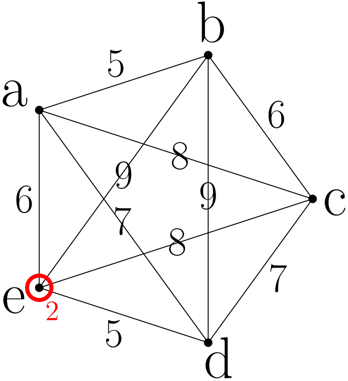
Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.

At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_2(a) = 3$

$f_2(b) = 1$

$f_2(c) = 3$

$f_2(d) = 2$

$f_2(e) = 1$

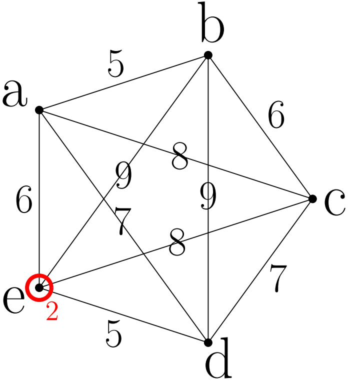
Cost opt(2)$=7+1=8$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_3(a) = 1$
$f_3(b) = 4$
$f_3(c) = 3$
$f_3(d) = 2$
$f_3(e) = 1$

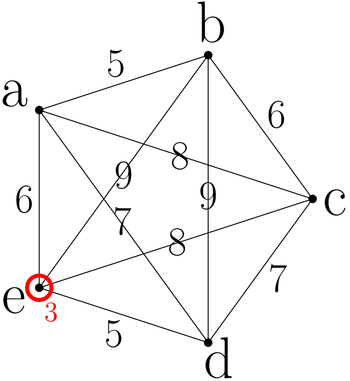
Cost opt(2)$=7 + 1 = 8$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_3(a) = 1$$
$$f_3(b) = 4$$
$$f_3(c) = 3$$
$$f_3(d) = 2$$
$$f_3(e) = 1$$

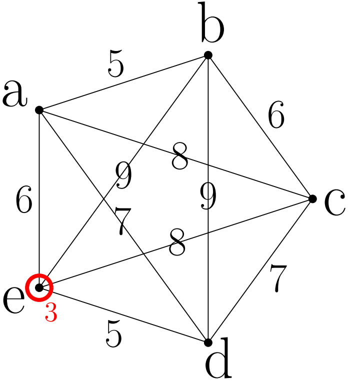
Cost opt(2)$=7+1=8$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.

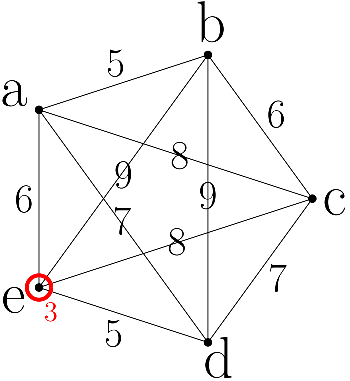


$f_3(a) = 1$
$f_3(b) = 4$
$f_3(c) = 3$
$f_3(d) = 2$
$f_3(e) = 1$

Cost opt(3)$=8 + 1 = 9$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.
Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.
Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_4(a) = 5$$
$$f_4(b) = 10$$
$$f_4(c) = 4$$
$$f_4(d) = 0$$
$$f_4(e) = 1$$

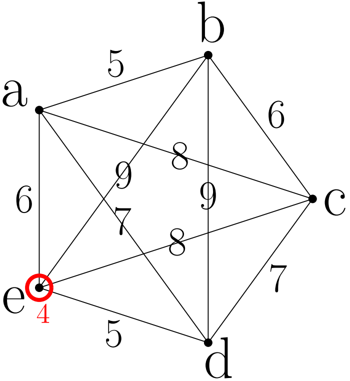
Cost opt(3)$=8+1=9$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_4(a) = 5$
$f_4(b) = 10$
$f_4(c) = 4$
$f_4(d) = 0$
$f_4(e) = 1$

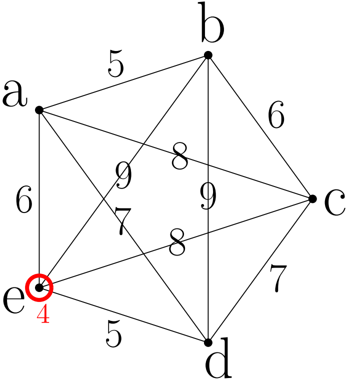
Cost opt(3)$=8 + 1 = 9$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$f_4(a) = 5$
$f_4(b) = 10$
$f_4(c) = 4$
$f_4(d) = 0$
$f_4(e) = 1$

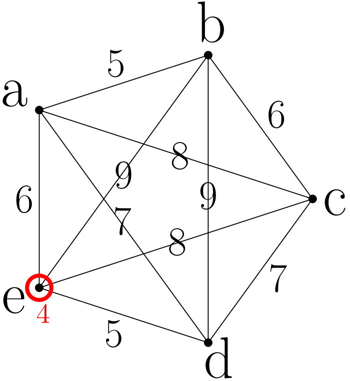
Cost opt(4)=$9 + 1 = 10$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_5(a) = 3$$
$$f_5(b) = 4$$
$$f_5(c) = 4$$
$$f_5(d) = 5$$
$$f_5(e) = 1$$

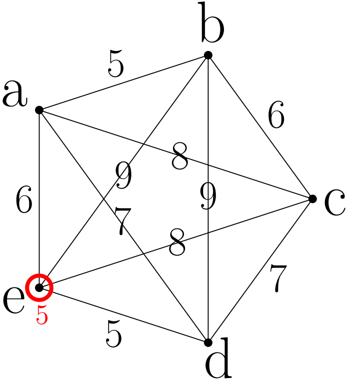
Cost opt(4)=$9 + 1 = 10$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_5(a) = 3$$
$$f_5(b) = 4$$
$$f_5(c) = 4$$
$$f_5(d) = 5$$
$$f_5(e) = 1$$

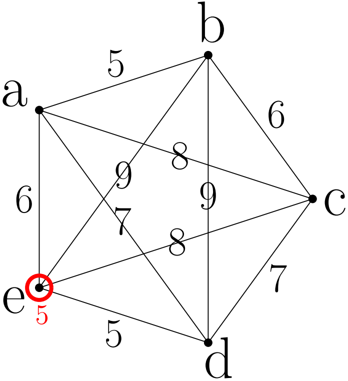
Cost opt(4)$=9 + 1 = 10$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.

Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.

Goal: Minimize the competitive ratio between our algorithm to opt.



$$f_5(a) = 3$$
$$f_5(b) = 4$$
$$f_5(c) = 4$$
$$f_5(d) = 5$$
$$f_5(e) = 1$$

Cost opt(5)=$10 + 1 = 11$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.
Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.
Goal: Minimize the competitive ratio between our algorithm to opt.

$$\text{Competitive ratio} = \max_{\text{input } I} \frac{\mathrm{Alg}(I)}{\mathrm{opt}(I)}.$$

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.
Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.
Goal: Minimize the competitive ratio between our algorithm to opt.

$$\text{Competitive ratio} = \max_{\text{input } I} \frac{\mathrm{Alg}(I)}{\mathrm{opt}(I)}.$$

Here $\frac{\mathrm{Alg}(I)}{\mathrm{opt}(I)} = \frac{27}{11}$.

Online problem - Metrical Task System (MTS)

Input: Metric space (X, d_X). Initial configuration $x_0 \in X$.
At step i, a task arrives with cost function $f_i : X \to \mathbb{R}_{\geq 0}$.
Output: Point $x_i \in X$ s.t. the task performed at x_i at cost $d_X(x_{i-1}, x_i) + f_i(x_i)$.
Goal: Minimize the competitive ratio between our algorithm to opt.

$$\text{Competitive ratio} = \max_{\text{input } I} \frac{\mathrm{Alg}(I)}{\mathrm{opt}(I)}.$$

Here $\frac{\mathrm{Alg}(I)}{\mathrm{opt}(I)} = \frac{27}{11}$.

Competitive ratio against oblivious adversary is

$$\max_{\text{input } I} \frac{\mathbb{E}[\mathrm{Alg}(I)]}{\mathrm{opt}(I)}.$$

Online problem - Metrical Task System (MTS)

Approach: embed into a tree, and then make all the decisions based on the tree.

Online problem - Metrical Task System (MTS)

Approach: embed into a tree, and then make all the decisions based on the tree.

Theorem ([Fiat, Mendel 2000])

Given an n point tree * T*, there is an online algorithm for MTS with competitive ratio* $O(\log n \cdot \log\log n)$ *against oblivious adversary.*

* Actually on an HST, which is a special kind of tree. [FRT04] is into HST's.

Online problem - Metrical Task System (MTS)

Approach: embed into a tree, and then make all the decisions based on the tree.

Theorem ([Fiat, Mendel 2000])

Given an n point tree * T*, there is an online algorithm for MTS with competitive ratio* $O(\log n \cdot \log\log n)$ *against oblivious adversary.*

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

Online problem - Metrical Task System (MTS)

Approach: embed into a tree, and then make all the decisions based on the tree.

Theorem ([Fiat, Mendel 2000])

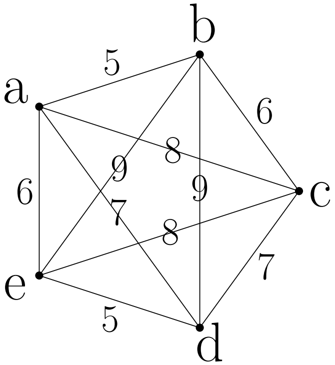
Given an n point tree * T, *there is an online algorithm for MTS with competitive ratio* $O(\log n \cdot \log\log n)$ *against oblivious adversary.*

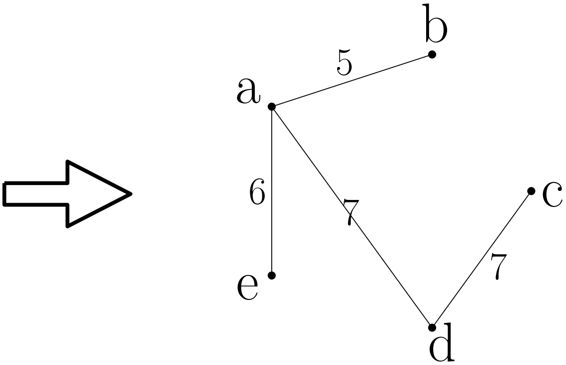
Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].
2. Run [FM00] on T with the same cost functions.
Make the same decisions as [FM00].

Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

2. Run [FM00] on T with the same cost functions.
   Make the same decisions as [FM00].



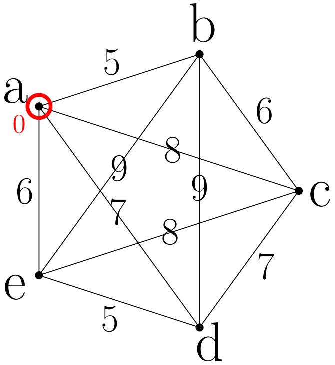


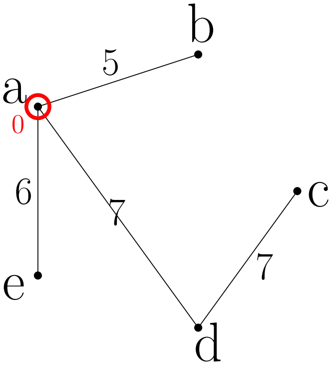
Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

2. Run [FM00] on T with the same cost functions.

Make the same decisions as [FM00].



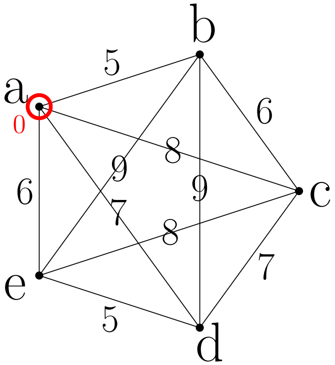


Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

2. Run [FM00] on T with the same cost functions.

Make the same decisions as [FM00].



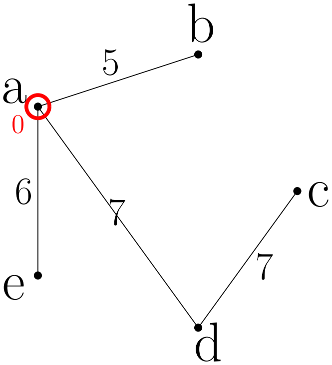
$f_1(a) = 3$

$f_1(b) = 1$

$f_1(c) = 2$

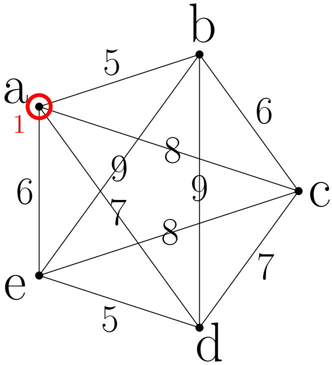
$f_1(d) = 4$

$f_1(e) = 1$

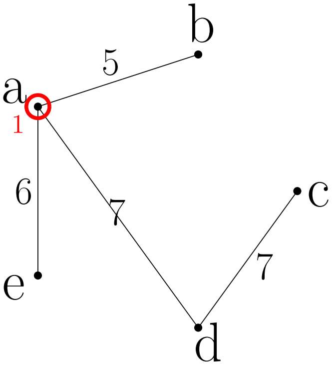


Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].
2. Run [FM00] on T with the same cost functions.
Make the same decisions as [FM00].



$$f_1(a) = 3$$
$$f_1(b) = 1$$
$$f_1(c) = 2$$
$$f_1(d) = 4$$
$$f_1(e) = 1$$

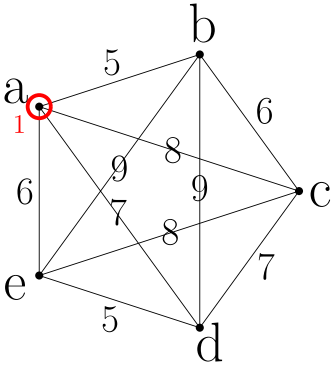


Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

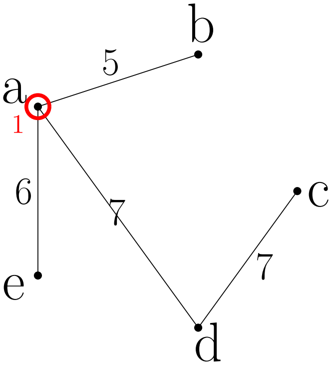
2. Run [FM00] on T with the same cost functions.

Make the same decisions as [FM00].



Cost alg(1)=3

$$f_1(a) = 3$$
$$f_1(b) = 1$$
$$f_1(c) = 2$$
$$f_1(d) = 4$$
$$f_1(e) = 1$$



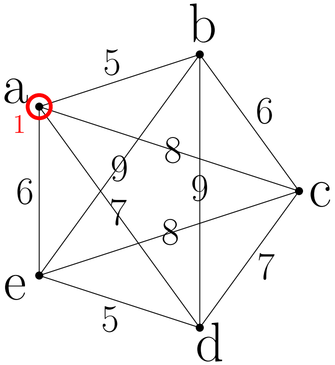
Cost $\text{alg}_T(1)$=3

Online problem - Metrical Task System (MTS)

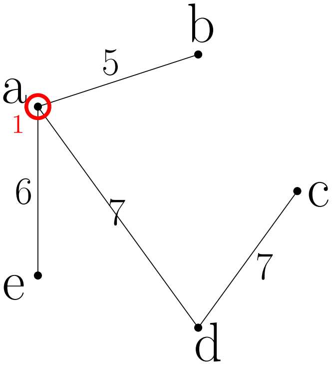
Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

2. Run [FM00] on T with the same cost functions.

Make the same decisions as [FM00].



$$f_2(a) = 3$$
$$f_2(b) = 1$$
$$f_2(c) = 3$$
$$f_2(d) = 2$$
$$f_2(e) = 1$$



Cost alg(1)=3

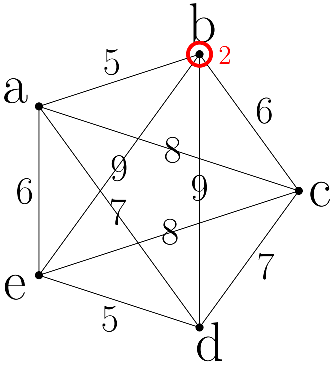
Cost $\mathrm{alg}_T(1)$=3

Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

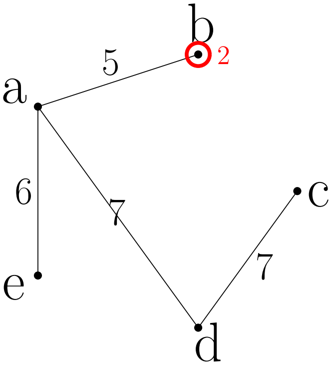
2. Run [FM00] on T with the same cost functions.

Make the same decisions as [FM00].



$f_2(a) = 3$
$f_2(b) = 1$
$f_2(c) = 3$
$f_2(d) = 2$
$f_2(e) = 1$

Cost alg(1)=3



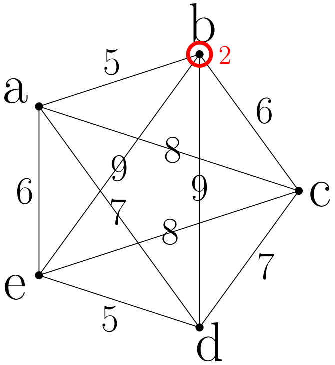
Cost $\mathrm{alg}_T(1)$=3

Online problem - Metrical Task System (MTS)

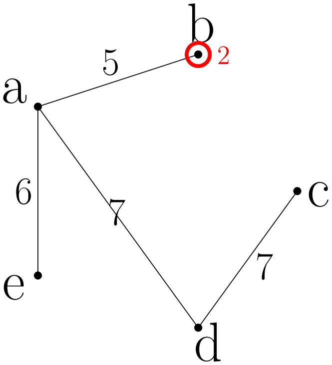
Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

2. Run [FM00] on T with the same cost functions.

Make the same decisions as [FM00].



$$f_2(a) = 3$$
$$f_2(b) = 1$$
$$f_2(c) = 3$$
$$f_2(d) = 2$$
$$f_2(e) = 1$$



Cost $\mathrm{alg}(2){=}3 + 5 + 1 = 9$

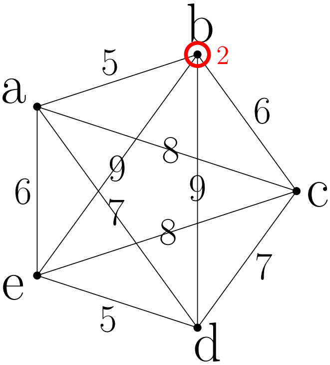
Cost $\mathrm{alg}_T(2){=}3 + 5 + 1 = 9$

Online problem - Metrical Task System (MTS)

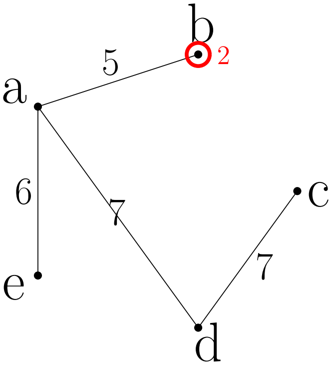
Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

2. Run [FM00] on T with the same cost functions.

Make the same decisions as [FM00].



$$f_3(a) = 1$$
$$f_3(b) = 4$$
$$f_3(c) = 3$$
$$f_3(d) = 2$$
$$f_3(e) = 1$$



Cost alg(2)=$3 + 5 + 1 = 9$

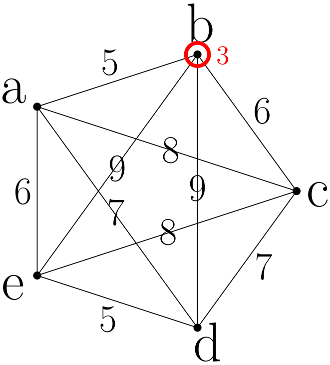
Cost $\mathrm{alg}_T(2)$=$3 + 5 + 1 = 9$

Online problem - Metrical Task System (MTS)

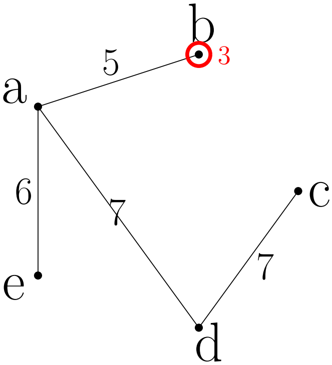
Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

2. Run [FM00] on T with the same cost functions.

Make the same decisions as [FM00].



$$f_3(a) = 1$$
$$f_3(b) = 4$$
$$f_3(c) = 3$$
$$f_3(d) = 2$$
$$f_3(e) = 1$$

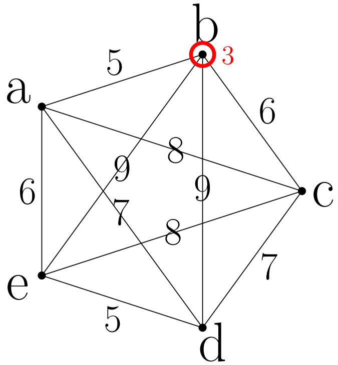


Cost alg(2)=$3 + 5 + 1 = 9$

Cost $\mathrm{alg}_T(2)$=$3 + 5 + 1 = 9$

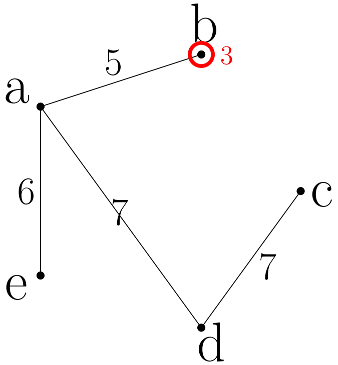
Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].
2. Run [FM00] on T with the same cost functions.
Make the same decisions as [FM00].



Cost alg(3)=$9 + 4 = 13$

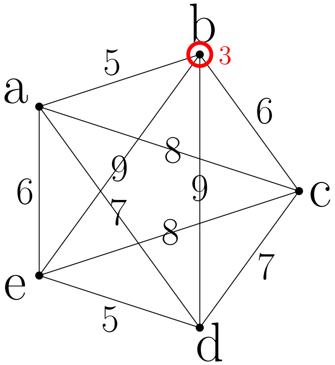
$f_3(a) = 1$
$f_3(b) = 4$
$f_3(c) = 3$
$f_3(d) = 2$
$f_3(e) = 1$



Cost $\text{alg}_T(3)$=$9 + 4 = 13$

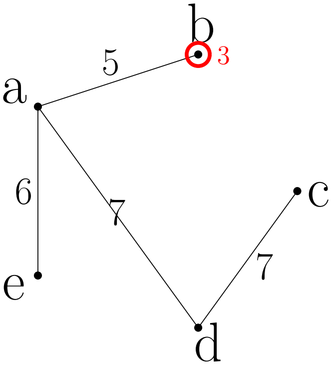
Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].
2. Run [FM00] on T with the same cost functions.
Make the same decisions as [FM00].



Cost alg(3)=$9 + 4 = 13$

$f_4(a) = 5$
$f_4(b) = 10$
$f_4(c) = 4$
$f_4(d) = 0$
$f_4(e) = 1$



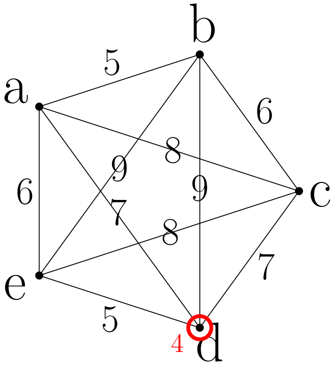
Cost $\text{alg}_T(3)$=$9 + 4 = 13$

Online problem - Metrical Task System (MTS)

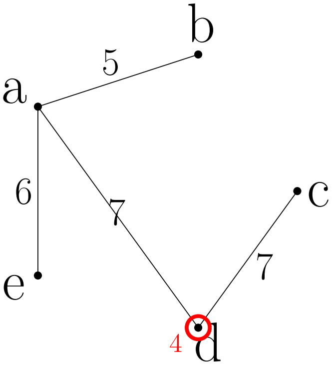
Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

2. Run [FM00] on T with the same cost functions.

Make the same decisions as [FM00].



$$f_4(a) = 5$$
$$f_4(b) = 10$$
$$f_4(c) = 4$$
$$f_4(d) = 0$$
$$f_4(e) = 1$$

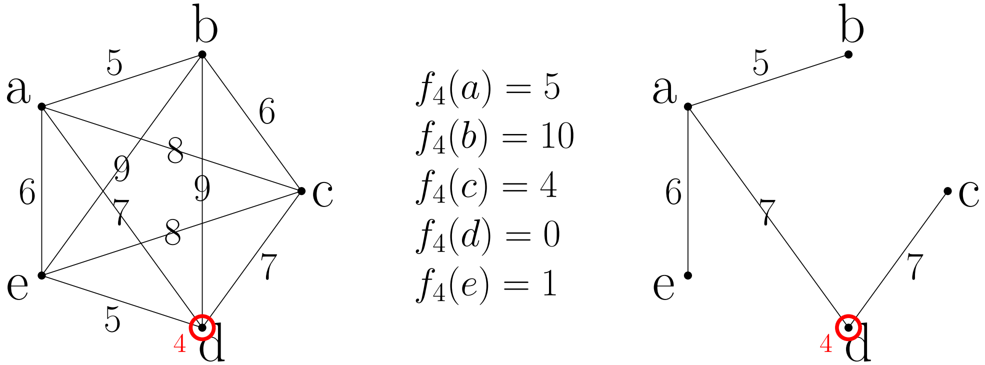


Cost alg(3)=$9 + 4 = 13$

Cost $\text{alg}_T(3)$=$9 + 4 = 13$

Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].
2. Run [FM00] on T with the same cost functions.
Make the same decisions as [FM00].



$$f_4(a) = 5$$
$$f_4(b) = 10$$
$$f_4(c) = 4$$
$$f_4(d) = 0$$
$$f_4(e) = 1$$

Cost $\mathrm{alg}(4) = 13 + 9 + 0 = 22$

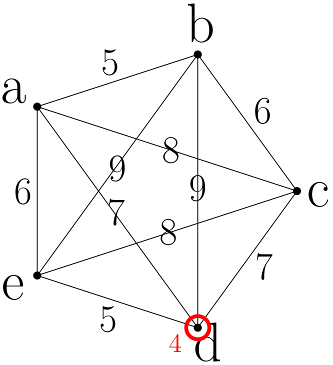
Cost $\mathrm{alg}_T(4) = 13 + 12 + 0 = 25$

Online problem - Metrical Task System (MTS)

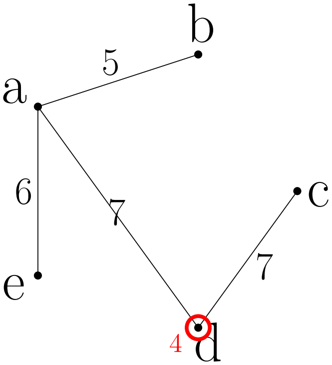
Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].

2. Run [FM00] on T with the same cost functions.

Make the same decisions as [FM00].



$$f_5(a) = 3$$
$$f_5(b) = 4$$
$$f_5(c) = 4$$
$$f_5(d) = 5$$
$$f_5(e) = 1$$

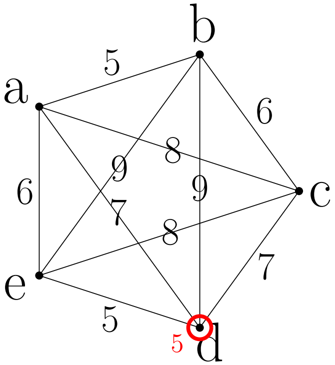


Cost alg(4)$=13 + 9 + 0 = 22$

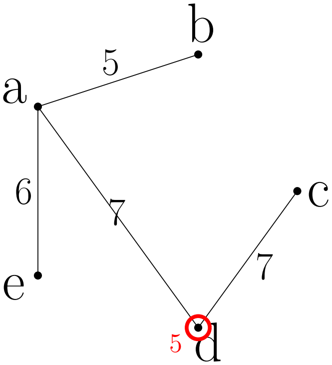
Cost $\text{alg}_T(4)$$=13 + 12 + 0 = 25$

Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].
2. Run [FM00] on T with the same cost functions.
Make the same decisions as [FM00].



$f_5(a) = 3$
$f_5(b) = 4$
$f_5(c) = 4$
$f_5(d) = 5$
$f_5(e) = 1$

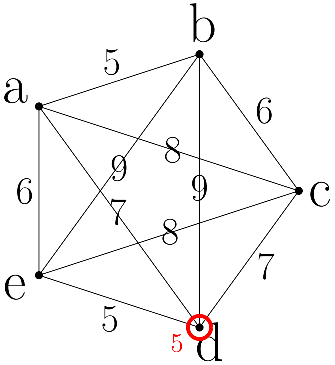


Cost $\mathrm{alg}(4)$=$13 + 9 + 0 = 22$

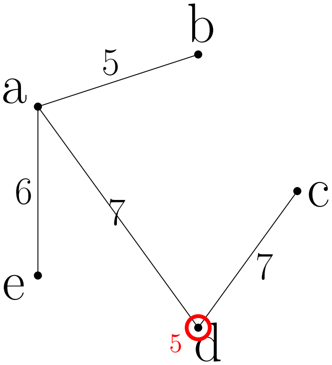
Cost $\mathrm{alg}_T(4)$=$13 + 12 + 0 = 25$

Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].
2. Run [FM00] on T with the same cost functions.
Make the same decisions as [FM00].



$f_5(a) = 3$
$f_5(b) = 4$
$f_5(c) = 4$
$f_5(d) = 5$
$f_5(e) = 1$



Cost alg(5)=$22 + 5 = 27$

Cost $\mathrm{alg}_T(5)$=$25 + 5 = 30$

Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].
2. Run [FM00] on T with the same cost functions.
Make the same decisions as [FM00].

Analysis. Let $x_1, x_2, \ldots, x_k$ be the decisions of opt. Thus
$\mathrm{opt} = \sum_{i=1}^{k} f_i(x_i) + \sum_{i=1}^{k} d_X(x_{i-1}, x_i)$.

Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].
2. Run [FM00] on T with the same cost functions.
Make the same decisions as [FM00].

Analysis. Let $x_1, x_2, \ldots, x_k$ be the decisions of opt. Thus
$\mathrm{opt} = \sum_{i=1}^{k} f_i(x_i) + \sum_{i=1}^{k} d_X(x_{i-1}, x_i)$.

We've sampled a tree T, $x_1, x_2, \ldots, x_k$ is also a valid decisions for T. Hence
$\mathrm{opt}_T \leq \sum_{i=1}^{k} f_i(x_i) + \sum_{i=1}^{k} d_T(x_{i-1}, x_i)$.

Online problem - Metrical Task System (MTS)

Algorithm: 1. Sample a tree T over (X, d_X) using [FRT04].
2. Run [FM00] on T with the same cost functions.
Make the same decisions as [FM00].

Analysis. Let $x_1, x_2, \ldots, x_k$ be the decisions of opt. Thus
$\mathrm{opt} = \sum_{i=1}^{k} f_i(x_i) + \sum_{i=1}^{k} d_X(x_{i-1}, x_i)$.

We've sampled a tree T, $x_1, x_2, \ldots, x_k$ is also a valid decisions for T. Hence
$\mathrm{opt}_T \leq \sum_{i=1}^{k} f_i(x_i) + \sum_{i=1}^{k} d_T(x_{i-1}, x_i)$.

[FM00] is $O(\log n \cdot \log\log n)$-competitive on T. Hence it choose points $y_1, \ldots, y_k$ such that $\mathbb{E}[\mathrm{alg}_T] = \mathbb{E}[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_T(y_{i-1}, y_i)] \leq O(\log n \cdot \log\log n) \cdot \mathrm{opt}_T$.

We made the same decisions, so overall:

$$\mathbb{E}\left[\text{alg}\right] = \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_X(y_{i-1}, y_i)\right]$$

We made the same decisions, so overall:

$$\begin{aligned}
\mathbb{E}\left[\mathrm{alg}\right] &= \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_X(y_{i-1}, y_i)\right] \\
&\leq \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_T(y_{i-1}, y_i)\right]
\end{aligned}$$

We made the same decisions, so overall:

$$
\begin{aligned}
\mathbb{E}\left[\mathrm{alg}\right] &= \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_X(y_{i-1}, y_i)\right] \\
&\leq \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_T(y_{i-1}, y_i)\right] \\
&\leq O(\log n \cdot \log\log n) \cdot \mathbb{E}\left[\sum_{i=1}^{k} f_i(x_i) + \sum_{i=1}^{k} d_T(x_{i-1}, x_i)\right]
\end{aligned}
$$

We made the same decisions, so overall:

$$\begin{aligned}
\mathbb{E}\left[\mathrm{alg}\right] &= \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_X(y_{i-1}, y_i)\right] \\
&\leq \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_T(y_{i-1}, y_i)\right] \\
&\leq O(\log n \cdot \log\log n) \cdot \mathbb{E}\left[\sum_{i=1}^{k} f_i(x_i) + \sum_{i=1}^{k} d_T(x_{i-1}, x_i)\right] \\
&= O(\log n \cdot \log\log n) \cdot \left(\sum_{i=1}^{k} f_i(x_i) + O(\log n) \cdot \sum_{i=1}^{k} d_T(x_{i-1}, x_i)\right)
\end{aligned}$$

We made the same decisions, so overall:

$$
\begin{aligned}
\mathbb{E}\left[\mathrm{alg}\right] &= \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_X(y_{i-1}, y_i)\right] \\
&\leq \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_T(y_{i-1}, y_i)\right] \\
&\leq O(\log n \cdot \log\log n) \cdot \mathbb{E}\left[\sum_{i=1}^{k} f_i(x_i) + \sum_{i=1}^{k} d_T(x_{i-1}, x_i)\right] \\
&= O(\log n \cdot \log\log n) \cdot \left(\sum_{i=1}^{k} f_i(x_i) + O(\log n) \cdot \sum_{i=1}^{k} d_T(x_{i-1}, x_i)\right) \\
&\leq O(\log^2 n \cdot \log\log n) \cdot \mathrm{Opt}\ .
\end{aligned}
$$

We made the same decisions, so overall:

$$
\begin{aligned}
\mathbb{E}\left[\text{alg}\right] &= \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_X(y_{i-1}, y_i)\right] \\
&\le \mathbb{E}\left[\sum_{i=1}^{k} f_i(y_i) + \sum_{i=1}^{k} d_T(y_{i-1}, y_i)\right] \\
&\le O(\log n \cdot \log\log n) \cdot \mathbb{E}\left[\sum_{i=1}^{k} f_i(x_i) + \sum_{i=1}^{k} d_T(x_{i-1}, x_i)\right] \\
&= O(\log n \cdot \log\log n) \cdot \left(\sum_{i=1}^{k} f_i(x_i) + O(\log n) \cdot \sum_{i=1}^{k} d_T(x_{i-1}, x_i)\right) \\
&\le O(\log^2 n \cdot \log\log n) \cdot \text{Opt} \ .
\end{aligned}
$$

Theorem

MTS has an $O(\log^2 n \cdot \log\log n)$ competitive algorithm against oblivious adversary.

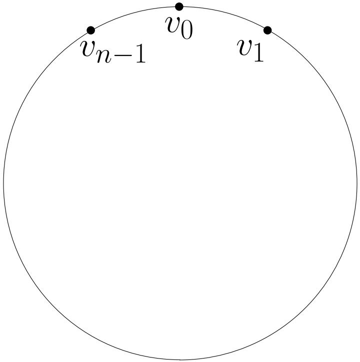
Outline of the talk - Appendix

Ramsey type Embeddings

Ramsey type theorem: Every **big** enough object, contains a **structured subset**.

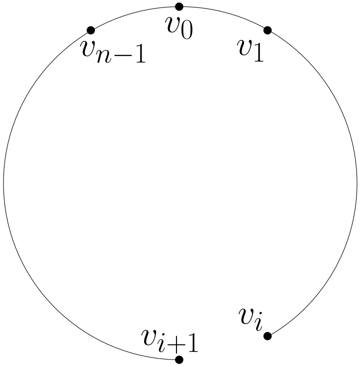
Ramsey type Embeddings

Ramsey type theorem: Every **big** enough object, contains a **structured subset**.



Ramsey type Embeddings

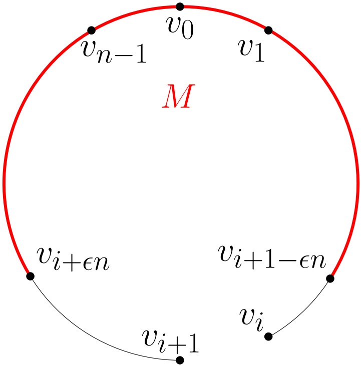
Ramsey type theorem: Every **big** enough object, contains a **structured subset**.



Suppose we delete $\{v_i, v_{i+1}\}$.

Ramsey type Embeddings

Ramsey type theorem: Every **big** enough object, contains a **structured subset**.

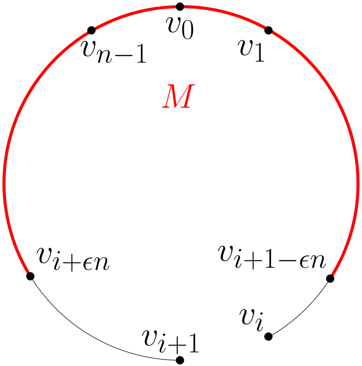


Suppose we delete $\{v_i, v_{i+1}\}$.

Set $M = \{v_{i+1-\epsilon n}, v_{i+2-\epsilon n}, \ldots, v_{i+\epsilon n}\}$.

Ramsey type Embeddings

Ramsey type theorem: Every **big** enough object, contains a **structured subset**.



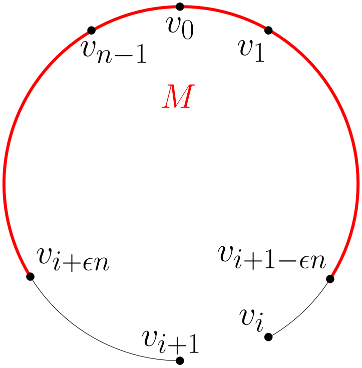
Suppose we delete $\{v_i, v_{i+1}\}$.

Set $M = \{v_{i+1-\epsilon n}, v_{i+2-\epsilon n}, \ldots, v_{i+\epsilon n}\}$.

For every $x, y \in M$, $\frac{d_T(x,y)}{d_{C_n}(x,y)} < \frac{1}{2\epsilon}$.

Ramsey type Embeddings

Ramsey type theorem: Every **big** enough object, contains a **structured subset**.



Suppose we delete $\{v_i, v_{i+1}\}$.

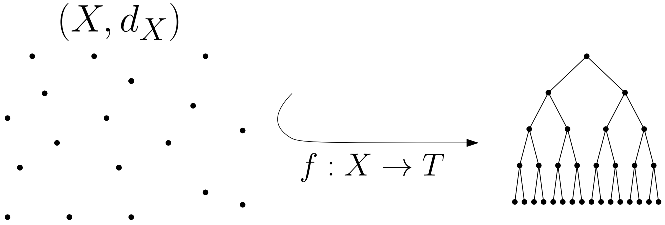
Set $M = \{v_{i+1-\epsilon n}, v_{i+2-\epsilon n}, \ldots, v_{i+\epsilon n}\}$.

For every $x, y \in M$, $\frac{d_T(x,y)}{d_{C_n}(x,y)} < \frac{1}{2\epsilon}$.

Choose i u.a.r., then $\Pr[v \in M] = 1 - 2\epsilon$.

Ramsey type Embeddings

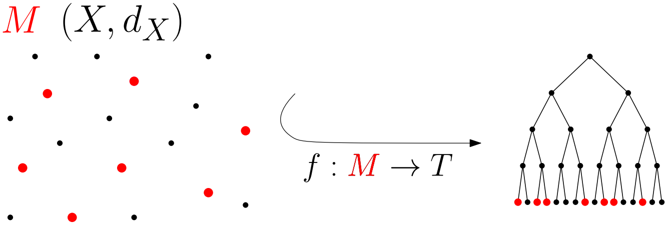
Fix $k > 1$, what is the **largest subset** $M \subset X$, s.t. (M, d_X) embeds into a tree with **distortion** k?



Ramsey type Embeddings

Fix $k > 1$, what is the **largest subset** $M \subset X$,

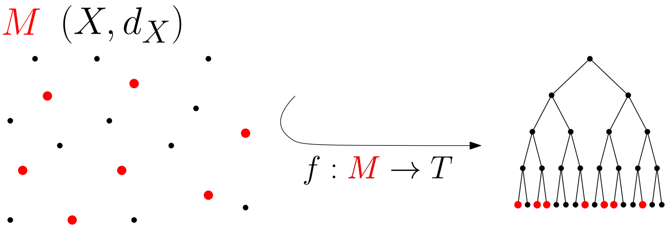
s.t. (M, d_X) embeds into a tree with **distortion** k?



$$\forall x, y \in M, \qquad d_X(x,y) \le d_T(f(x), f(y)) \le \boldsymbol{k} \cdot d_X(x,y)$$

Ramsey type Embeddings

Fix $\boldsymbol{k} > 1$, what is the **largest subset** $M \subset X$,
s.t. (M, d_X) embeds into a tree with **distortion** $\boldsymbol{k}$?



$\forall x, y \in M, \qquad d_X(x, y) \le d_T(f(x), f(y)) \le \boldsymbol{k} \cdot d_X(x, y)$

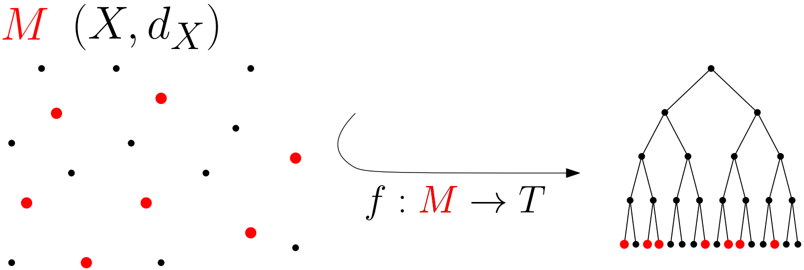
Theorem ([Mendel, Naor 07], following [BFM86, BLMN05])

$\forall$ *n-point metric space and* $\boldsymbol{k} \ge 1$, $\exists$ ***subset*** M *of size* $n^{1-1/k}$
that ***embeds*** *into a* ***tree*** *with distortion* $O(\boldsymbol{k})$.

Ramsey type Embeddings

Theorem ([Mendel, Naor 07], following [BFM86, BLMN05])

$\forall$ *n-point metric space and* $\boldsymbol{k} \geq 1$, $\exists$ ***subset*** M *of size* $n^{1-1/k}$ *that* ***embeds*** *into a* ***tree*** *with distortion* $O(\boldsymbol{k})$.

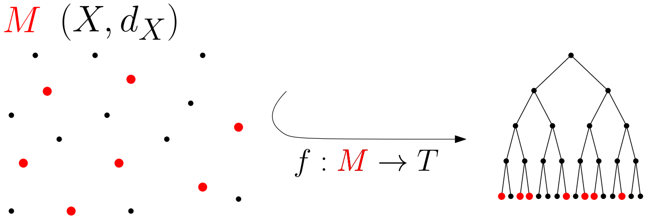


Asymptotically tight.

Ramsey type Embeddings

Theorem ([Mendel, Naor 07], following [BFM86, BLMN05])

∀ *n-point metric space and* $\boldsymbol{k} \geq 1$, ∃ ***subset*** M *of size* $n^{1-1/k}$ *that* ***embeds*** *into a* ***tree*** *with distortion* $O(\boldsymbol{k})$.



Asymptotically tight.

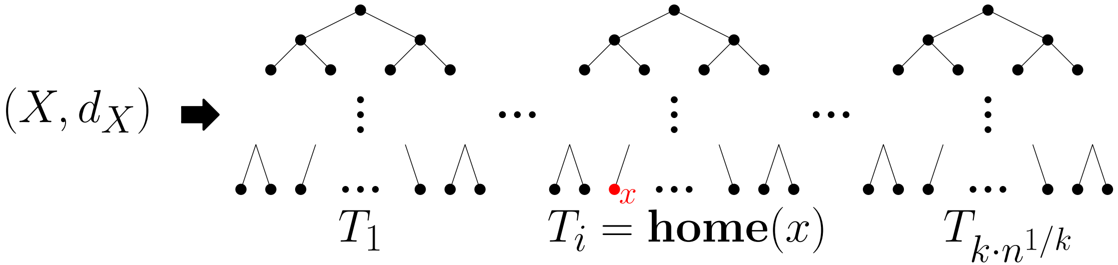
[Naor, Tao 12]: distortion $2e \cdot k$.

Ramsey type Embeddings

Corollary

For every n-point metric space and $\boldsymbol{k} \geq 1$, *there is a **set*** $\mathcal{T}$ *of* $k \cdot n^{\frac{1}{k}}$ ***trees** and a* *mapping* **home** : $X \to \mathcal{T}$, *such that for every* $x, y \in X$,

$$d_{\mathbf{home}(x)}(x, y) \leq O(\boldsymbol{k}) \cdot d_X(x, y)$$



Ramsey type Embeddings

Corollary

For every n-point metric space and $\boldsymbol{k} \geq 1$*, there is a* ***set*** $\mathcal{T}$ *of* $k \cdot n^{\frac{1}{k}}$ ***trees*** *and a* *mapping* **home** $: X \to \mathcal{T}$*, such that for every* $x, y \in X$*,*

$$d_{\mathbf{home}(x)}(x, y) \leq O(\boldsymbol{k}) \cdot d_X(x, y)$$

Applications:

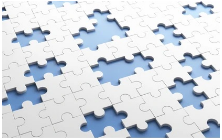
- **Distance oracle**
- Compact routing scheme
- Online algorithms
- Approximate ranking
- etc.

Ramsey type Embeddings

Theorem ([Mendel, Naor 07], following [BFM86, BLMN05])

$\forall$ *n-point metric space and* $\boldsymbol{k} \geq 1$, $\exists$ ***subset*** M *of size* $n^{1-1/k}$ *that* ***embeds*** *into a* ***tree*** *with distortion* $O(\boldsymbol{k})$.

Compromises: only partial guarantees

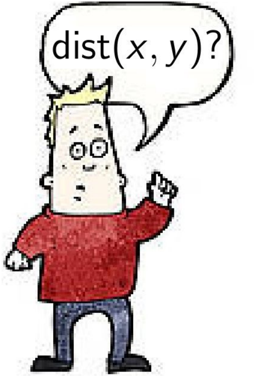


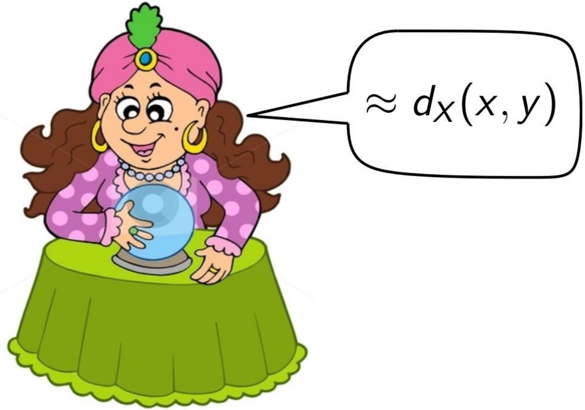
Distance Oracle

A **succinct** data structure that **approximately** answers distance queries.

Distance Oracle

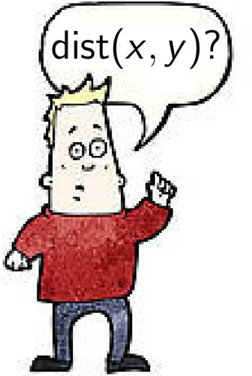
A **succinct** data structure that **approximately** answers distance queries.

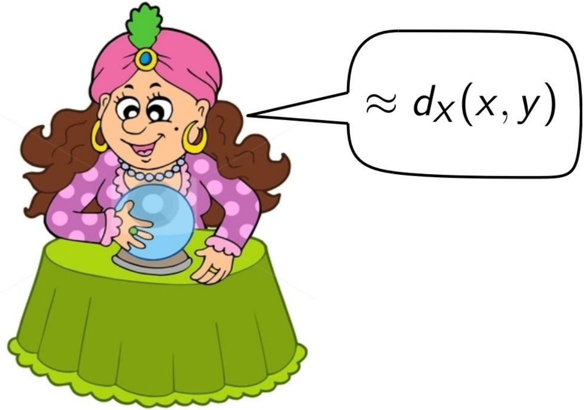




Distance Oracle

A **succinct** data structure that **approximately** answers distance queries.





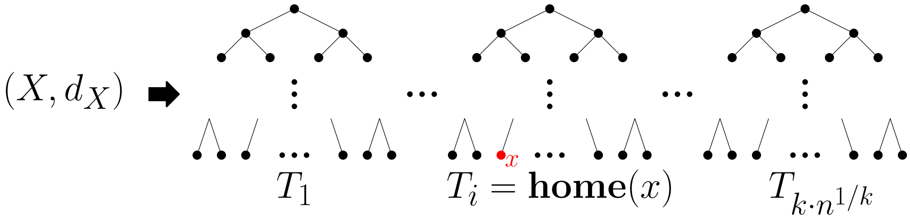
The properties of interest are size, distortion and query time.

Distance Oracles

Corollary

For every n-point metric space and $\boldsymbol{k} \geq 1$, *there is a* ***set*** $\mathcal{T}$ *of* $k \cdot n^{\frac{1}{k}}$ ***trees*** *and a* *mapping* **home** : $X \to \mathcal{T}$, *such that for every* $x, y \in X$,

$$d_{\mathbf{home}(x)}(x, y) \leq O(\boldsymbol{k}) \cdot d_X(x, y)$$

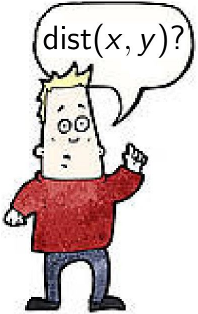


Distance Oracles

Corollary

For every n-point metric space and $\mathbf{k} \geq 1$*, there is a* ***set*** $\mathcal{T}$ *of* $k \cdot n^{\frac{1}{k}}$ ***trees*** *and a* *<u>mapping</u>* **home** $: X \to \mathcal{T}$*, such that for every* $x, y \in X$*,*

$$d_{\mathbf{home}(x)}(x, y) \leq O(\mathbf{k}) \cdot d_X(x, y)$$



Distance Oracles

Corollary

For every n-point metric space and $\boldsymbol{k} \geq 1$*, there is a* ***set*** $\mathcal{T}$ *of* $k \cdot n^{\frac{1}{k}}$ ***trees*** *and a* *mapping* **home** $: X \to \mathcal{T}$*, such that for every* $x, y \in X$*,*

$$d_{\mathbf{home}(x)}(x, y) \leq O(\boldsymbol{k}) \cdot d_X(x, y)$$

Theorem (Tree Distance Oracle [Harel, Tarjan 84], [Bender, Farach-Colton 00])

For every ***tree metric****, there is an exact distance oracle of* ***linear size*** *and* ***constant*** *query* ***time****.*

Distance Oracles

Corollary

For every n-point metric space and $\boldsymbol{k} \geq 1$*, there is a* ***set*** $\mathcal{T}$ *of* $k \cdot n^{\frac{1}{k}}$ ***trees*** *and a* *mapping* **home** $: X \to \mathcal{T}$*, such that for every* $x, y \in X$*,*

$$d_{\mathbf{home}(x)}(x, y) \leq O(\boldsymbol{k}) \cdot d_X(x, y)$$

Theorem (Tree Distance Oracle [Harel, Tarjan 84], [Bender, Farach-Colton 00])

For every ***tree metric*****, there is an exact distance oracle of* ***linear size*** *and* ***constant*** *query* ***time****.*

Theorem (Ramsey based Deterministic Distance Oracle)

For any n-point metric space, there is a distance oracle with :

Distortion	***Size***	***Query time***
$O(k)$	$O(k \cdot n^{1+1/k})$	$O(1)$

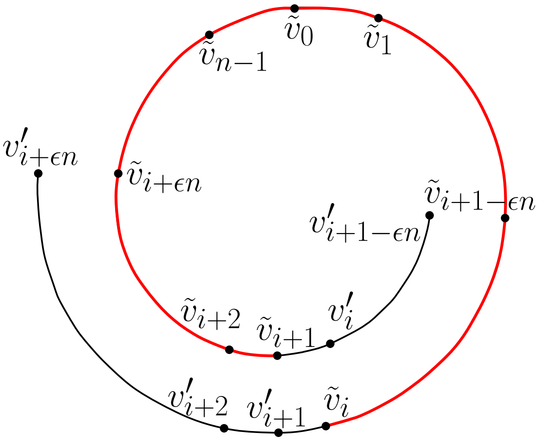
Outline of the talk - Appendix

Clan Embedding

Idea: **duplicate** vertices to meet all guarantees!

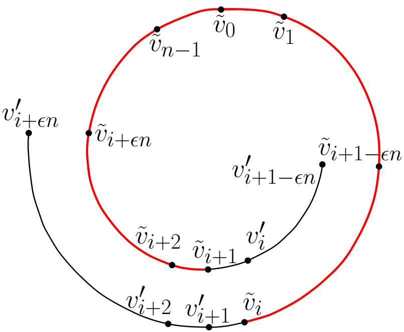
Clan Embedding

Idea: **duplicate** vertices to meet all guarantees!



Clan Embedding

Idea: **duplicate** vertices to meet all guarantees!

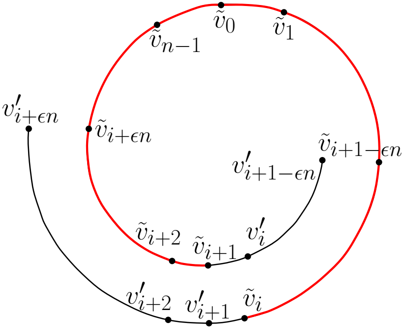


One-to-many embedding from (X, d_X) to (Y, d_Y): A map $f : X \to 2^Y$ where:

1) $\forall x,\ f(x) \neq \emptyset$.
2) $\forall x, y,\ f(x) \cap f(y) = \emptyset$.

Clan Embedding

Idea: **duplicate** vertices to meet all guarantees!



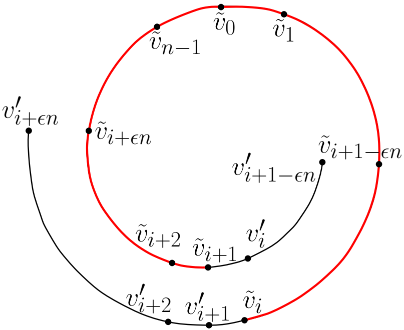
One-to-many embedding from (X, d_X) to (Y, d_Y): A map $f : X \to 2^Y$ where:

1) $\forall x,\ f(x) \neq \emptyset$.
2) $\forall x, y,\ f(x) \cap f(y) = \emptyset$.

$f(x)$ is the **clan** of x. Each $x' \in f(x)$ is a **copy** of x.

Clan Embedding

Idea: **duplicate** vertices to meet all guarantees!



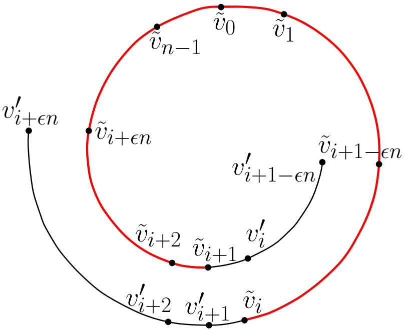
One-to-many embedding from (X, d_X) to (Y, d_Y): A map $f : X \to 2^Y$ where:
1) $\forall x,\ f(x) \neq \emptyset$.
2) $\forall x, y,\ f(x) \cap f(y) = \emptyset$.

$f(x)$ is the **clan** of x. Each $x' \in f(x)$ is a **copy** of x.

Dominating: $d_X(x, y) \leq \min_{x' \in f(x), y' \in f(y)} d_Y(x', y')$

Clan Embedding

Idea: **duplicate** vertices to meet all guarantees!



One-to-many embedding from (X, d_X) to (Y, d_Y): A map $f : X \to 2^Y$ where:
1) $\forall x,\ f(x) \neq \emptyset$.
2) $\forall x, y,\ f(x) \cap f(y) = \emptyset$.

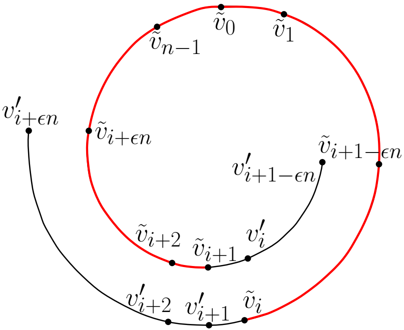
$f(x)$ is the **clan** of x. Each $x' \in f(x)$ is a **copy** of x.

Dominating: $d_X(x, y) \leq \min_{x' \in f(x), y' \in f(y)} d_Y(x', y')$

Each clan $f(x)$ has a **chief**: $\chi(x) \in f(x)$

Clan Embedding

Idea: **duplicate** vertices to meet all guarantees!



One-to-many embedding from (X, d_X) to (Y, d_Y): A map $f : X \to 2^Y$ where:
1) $\forall x,\ f(x) \neq \emptyset$.
2) $\forall x, y,\ f(x) \cap f(y) = \emptyset$.

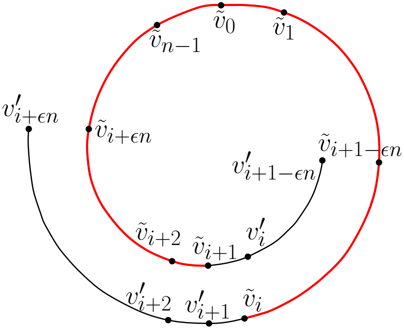
$f(x)$ is the **clan** of x. Each $x' \in f(x)$ is a **copy** of x.

Dominating: $d_X(x, y) \leq \min_{x' \in f(x), y' \in f(y)} d_Y(x', y')$

Each clan $f(x)$ has a **chief**: $\chi(x) \in f(x)$

$f : C_n \to 2^{P_{2n}}$, $\qquad f(v_i) \subseteq \{\tilde{v}_i, v'_i\}$, $\qquad \chi(v_i) = \tilde{v}_i$

Clan Embedding



One-to-many embedding from (X, d_X) to (Y, d_Y): A map $f : X \to 2^Y$ where:
1) $\forall x,\ f(x) \neq \emptyset$.
2) $\forall x, y,\ f(x) \cap f(y) = \emptyset$.

$f(x)$ is the **clan** of x. Each $x' \in f(x)$ is a **copy** of x.

Dominating: $d_X(x, y) \leq \min_{x' \in f(x), y' \in f(y)} d_Y(x', y')$

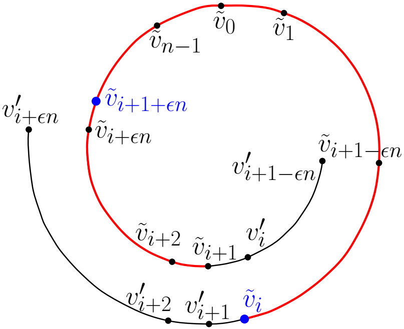
Each clan $f(x)$ has a **chief**: $\chi(x) \in f(x)$

$f : C_n \to 2^{P_{2n}}$, $\quad f(v_i) \subseteq \{\tilde{v}_i, v'_i\}$, $\quad \chi(v_i) = \tilde{v}_i$

Clan embedding is a pair (f, χ), where f is **dominating** one-to-many embedding.
(f, χ) has **distortion** t, if $\forall x, y \in X$, $\quad \min_{y' \in f(y)} d_Y(\chi(x), y') \leq t \cdot d_X(x, y)$

Clan Embedding

Clan embedding is a pair (f,χ), where f is **dominating** one-to-many embedding.
(f,χ) has **distortion** t, if $\forall x,y\in X$, $\min_{y'\in f(y)} d_Y(\chi(x),y') \le t\cdot d_X(x,y)$



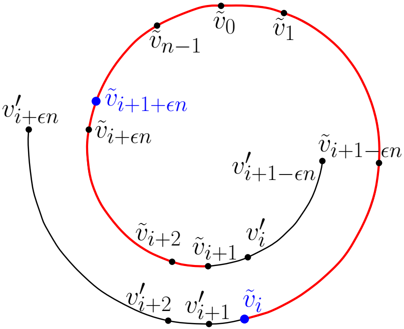
$$f : C_n \to 2^{P_{2n}}, \qquad f(v_i)\subseteq\{\tilde{v}_i, v'_i\}, \qquad \chi(v_i)=\tilde{v}_i$$

$$\min_{y'\in f(v_{i+1+\epsilon n})} d_{P_{2n}}(\chi(v_i),y') \le (1-\epsilon)n$$

$$< \frac{1}{\epsilon}\cdot d_{C_n}(v_i, v_{i+\epsilon n+1})$$

Clan Embedding

Clan embedding is a pair (f, χ), where f is **dominating** one-to-many embedding.
(f, χ) has **distortion** t, if $\forall x, y \in X$, $\quad \min_{y' \in f(y)} d_Y(\chi(x), y') \leq t \cdot d_X(x, y)$



$$f : C_n \to 2^{P_{2n}}, \qquad f(v_i) \subseteq \{\tilde{v}_i, v'_i\}, \qquad \chi(v_i) = \tilde{v}_i$$

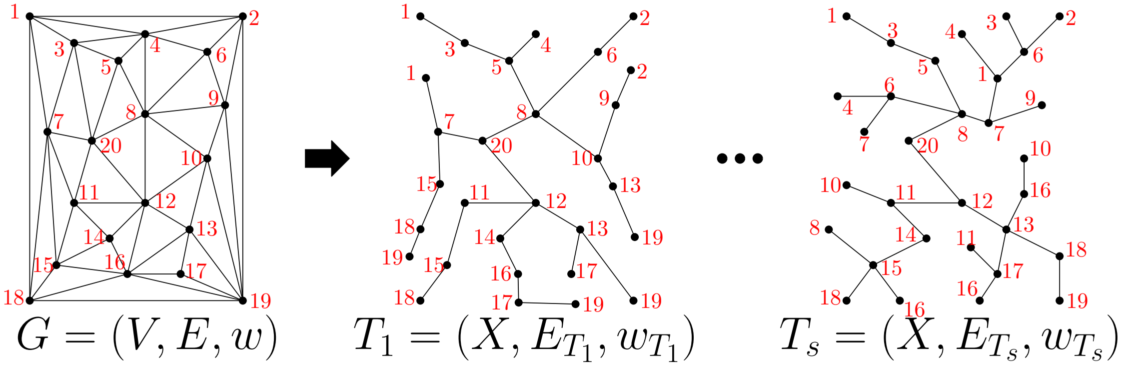
$$\min_{y' \in f(v_{i+1+\epsilon n})} d_{P_{2n}}(\chi(v_i), y') \leq (1 - \epsilon)n$$
$$< \frac{1}{\epsilon} \cdot d_{C_n}(v_i, v_{i+\epsilon n+1})$$

Choose i u.a.r., then $\qquad \mathbb{E}[|f(v_i)|] = 1 + 2\epsilon.$

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) *n point metric space,* $\forall \epsilon \in (0,1)$*, there is* ***distribution*** $\mathcal{D}$ *over* ***dominating clan embeddings*** *into trees such that:*

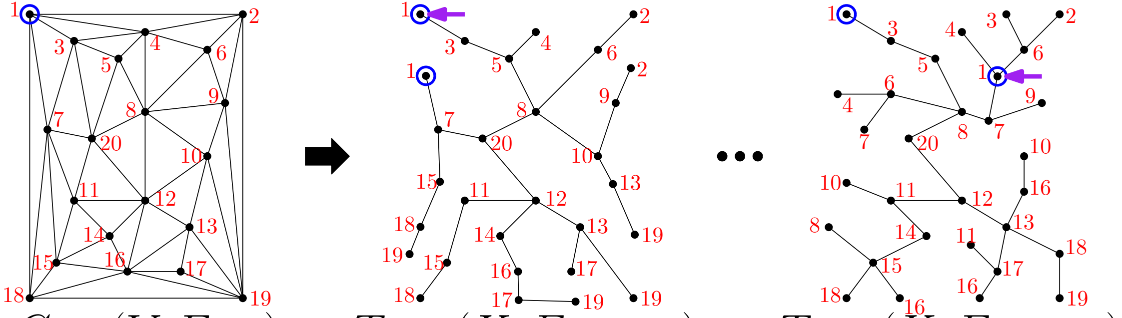
- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion* $O(\frac{\log n}{\epsilon})$.
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \le 1 + \epsilon$.



Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) n point metric space, $\forall \epsilon \in (0,1)$, there is ***distribution*** *$\mathcal{D}$ over* ***dominating clan embeddings*** *into trees such that:*

- $\forall$ $(f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion* $O(\frac{\log n}{\epsilon})$.
- $\forall$ $x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \le 1 + \epsilon$.

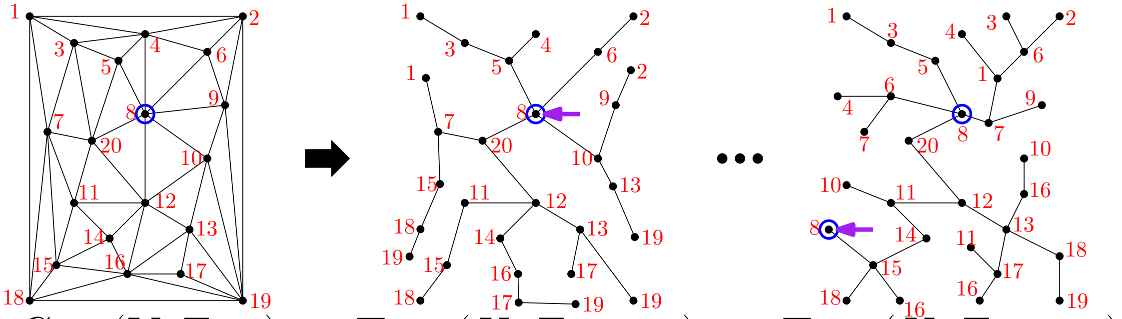


$G = (V, E, w)$ $\quad$ $T_1 = (X, E_{T_1}, w_{T_1})$ $\quad$ $T_s = (X, E_{T_s}, w_{T_s})$

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) n point metric space, $\forall \epsilon \in (0,1)$, there is ***distribution*** *$\mathcal{D}$ over* ***dominating clan embeddings*** *into trees such that:*

- *$\forall\ (f,\chi) \in \mathrm{supp}(\mathcal{D})$ has distortion $O(\frac{\log n}{\epsilon})$.*
- *$\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \le 1+\epsilon$.*

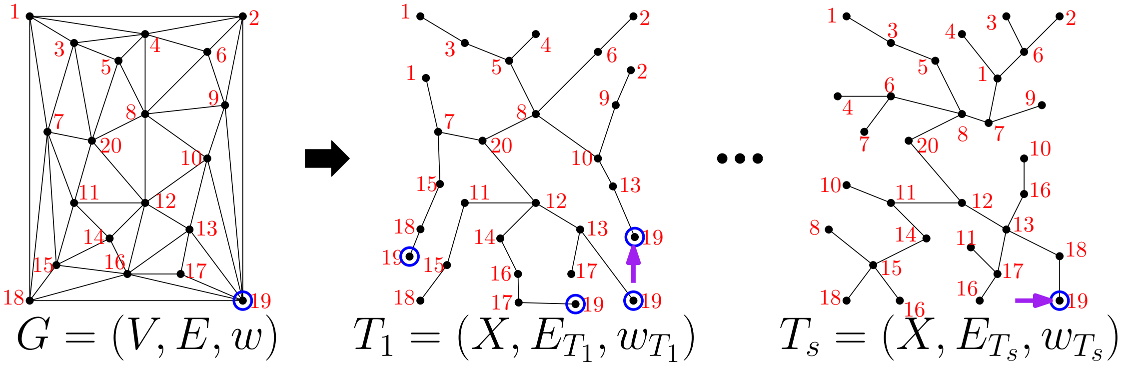


$G = (V, E, w)$ $\quad T_1 = (X, E_{T_1}, w_{T_1})$ $\quad T_s = (X, E_{T_s}, w_{T_s})$

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) *n point metric space,* $\forall \epsilon \in (0,1)$, *there is* ***distribution*** $\mathcal{D}$ *over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f,\chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion* $O(\frac{\log n}{\epsilon})$.
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq 1+\epsilon$.



Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) *n point metric space*, $\forall\epsilon \in (0, 1)$, *there is* ***distribution*** $\mathcal{D}$ *over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion* $O(\frac{\log n}{\epsilon})$.
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq 1 + \epsilon$.

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) *n point metric space*, $\forall k \in \mathbb{N}$, *there is* ***distribution*** $\mathcal{D}$ *over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion* $O(k)$.
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq O(n^{\frac{1}{k}})$

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) *n point metric space,* $\forall\epsilon \in (0,1)$, *there is* ***distribution*** $\mathcal{D}$ *over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion* $O(\frac{\log n}{\epsilon})$.
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq 1 + \epsilon$.

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) *n point metric space,* $\forall k \in \mathbb{N}$, *there is* ***distribution*** $\mathcal{D}$ *over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion* $O(k)$.
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq O(n^{\frac{1}{k}})$

(both) Tight!

Clan Embedding

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) n point metric space, $\forall k \in \mathbb{N}$, there is ***distribution*** *$\mathcal{D}$ over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion $O(k)$.*
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq O(n^{\frac{1}{k}})$

Compromises: Not a real classic embedding

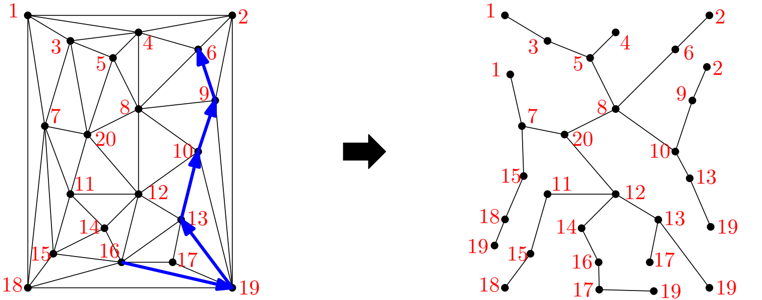
Path Distortion

Originally appeared in [Bartal, Mendel 04] in the context of multi-embeddings.

Definition

One-to-many embedding $f : X \to 2^Y$ has *path-distortion* t if for every sequence $(x_0, x_1, \ldots, x_m)$ in X there is a sequence $y_0, \ldots, y_m$ where $y_i \in f(x_i)$, and

$$\sum_{i=0}^{m-1} d_Y(y_i, y_{i+1}) \leq t \cdot \sum_{i=1}^{m-1} d_X(x_i, x_{i+1}).$$



$G = (V, E, w)$ $\qquad$ $T = (X, E_T, w_T)$

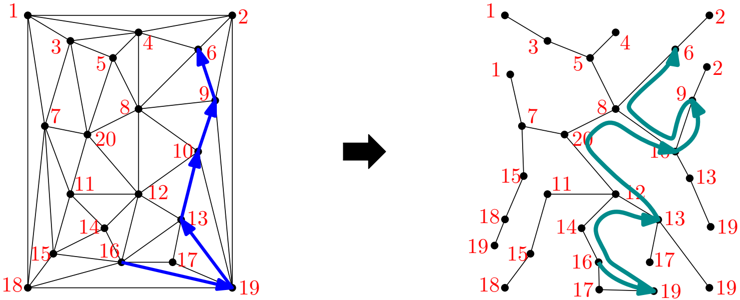
Path Distortion

Originally appeared in [Bartal, Mendel 04] in the context of multi-embeddings.

Definition

One-to-many embedding $f : X \to 2^Y$ has *path-distortion* t if for every sequence $(x_0, x_1, \ldots, x_m)$ in X there is a sequence $y_0, \ldots, y_m$ where $y_i \in f(x_i)$, and

$$\sum_{i=0}^{m-1} d_Y(y_i, y_{i+1}) \leq t \cdot \sum_{i=1}^{m-1} d_X(x_i, x_{i+1}).$$



$G = (V, E, w)$ $\quad$ $T = (X, E_T, w_T)$

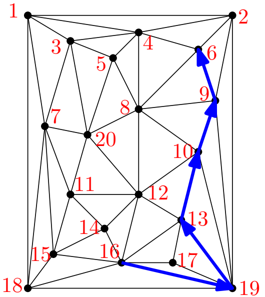
Path Distortion

Originally appeared in [Bartal, Mendel 04] in the context of multi-embeddings.

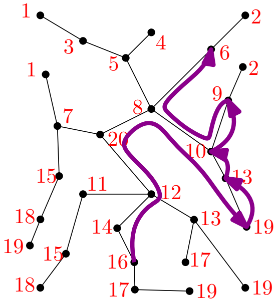
Definition

One-to-many embedding $f : X \to 2^Y$ has *path-distortion* t if for every sequence $(x_0, x_1, \ldots, x_m)$ in X there is a sequence $y_0, \ldots, y_m$ where $y_i \in f(x_i)$, and

$$\sum_{i=0}^{m-1} d_Y(y_i, y_{i+1}) \leq t \cdot \sum_{i=1}^{m-1} d_X(x_i, x_{i+1}).$$



$G = (V, E, w)$

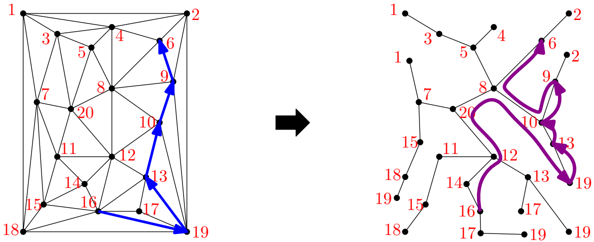


$T = (X, E_T, w_T)$

Definition

One-to-many embedding $f : X \to 2^Y$ has *path-distortion* t if for every sequence $(x_0, x_1, \dots, x_m)$ in X there is a sequence $y_0, \dots, y_m$ where $y_i \in f(x_i)$, and

$$\sum_{i=0}^{m-1} d_Y(y_i, y_{i+1}) \leq t \cdot \sum_{i=1}^{m-1} d_X(x_i, x_{i+1}).$$



$G = (V, E, w)$ $\qquad$ $T = (X, E_T, w_T)$

Theorem (Clan embedding into trees, [Filtser, Le 21])

Our clan embeddings with distortion $O(k)$ have path distortion $O(k \cdot \log n)$.

Theorem (Clan embedding into trees, [Filtser, Le 21])

Our clan embeddings with distortion $O(k)$ have path distortion $O(k \cdot \log n)$.

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) n point metric space, $\forall k \in \mathbb{N}$, there is
distribution *$\mathcal{D}$ over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion* $O(k)$.
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq O(n^{\frac{1}{k}})$

Theorem (Clan embedding into trees, [Filtser, Le 21])

Our clan embeddings with distortion $O(k)$ have path distortion $O(k \cdot \log n)$.

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) n point metric space, $\forall k \in \mathbb{N}$, there is ***distribution*** *$\mathcal{D}$ over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion $O(k)$.*
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq O(n^{\frac{1}{k}})$

That is there is a one-to-many embedding with a total of $O(n^{1+\frac{1}{k}})$ copies and path distortion $O(k \cdot \log n)$.

Theorem (Clan embedding into trees, [Filtser, Le 21])

Our clan embeddings with distortion $O(k)$ have path distortion $O(k\cdot\log n)$.

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X,d_X) n point metric space, $\forall k\in\mathbb{N}$, there is ***distribution*** *$\mathcal{D}$ over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f,\chi)\in\mathrm{supp}(\mathcal{D})$ *has distortion $O(k)$.*
- $\forall\ x\in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|]\le O(n^{\frac{1}{k}})$

That is there is a one-to-many embedding with a total of $O(n^{1+\frac{1}{k}})$ copies and path distortion $O(k\cdot\log n)$.

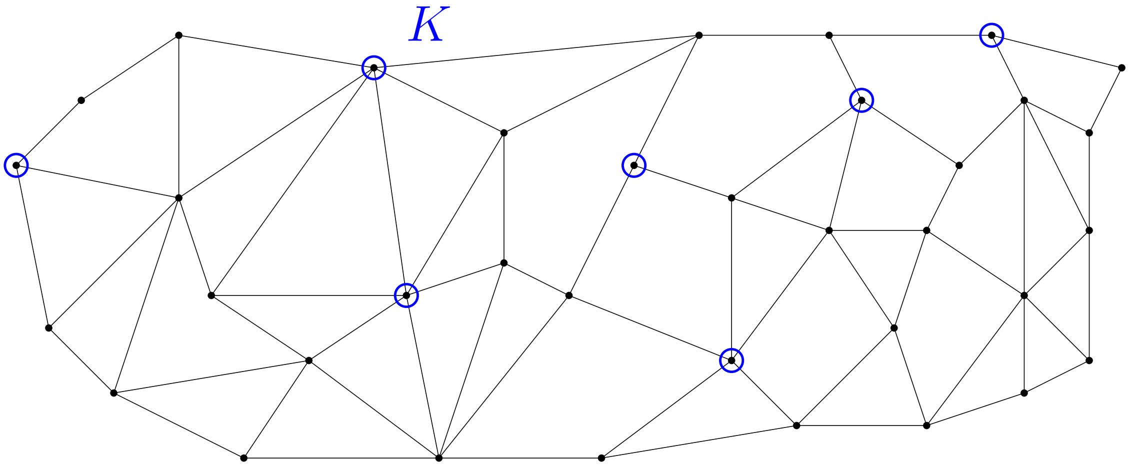
Or a total of $O(n^{1+\frac{1}{2}})$ copies and path distortion $O(\log n)$.

Outline of the talk - Appendix

7. Bartal 96 and Padded decompositions
8. Metrical Task System
9. Ramsey type embeddings
10. Clan embedding
11. Group Steiner Tree (using clan embedding)

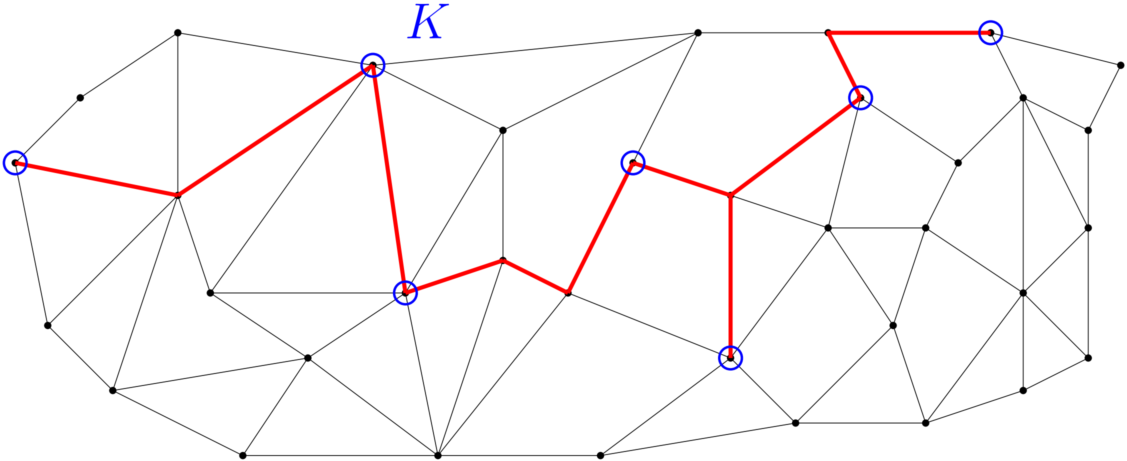
Steiner Tree

Given set of terminals K, find minimum weight tree T spanning K



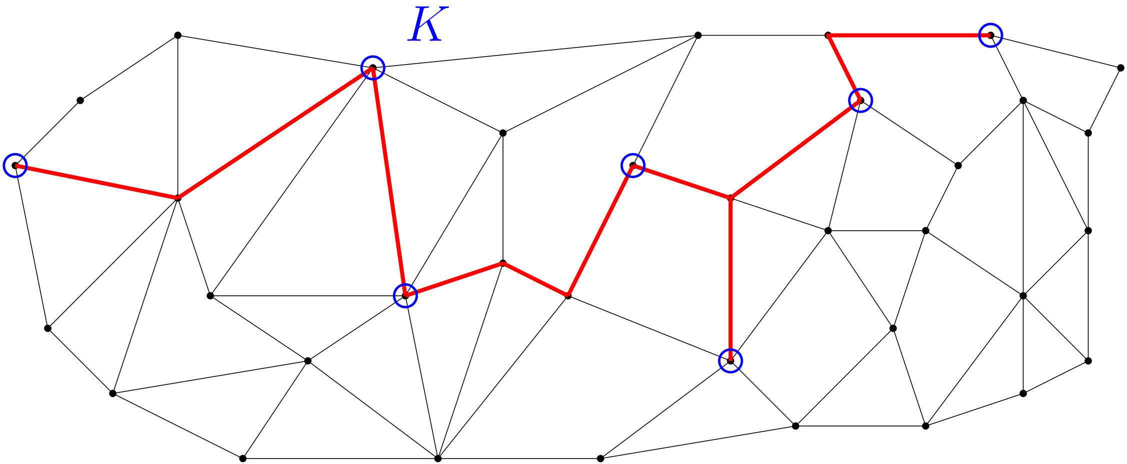
Steiner Tree

Given set of terminals K, find minimum weight tree T spanning K



Steiner Tree

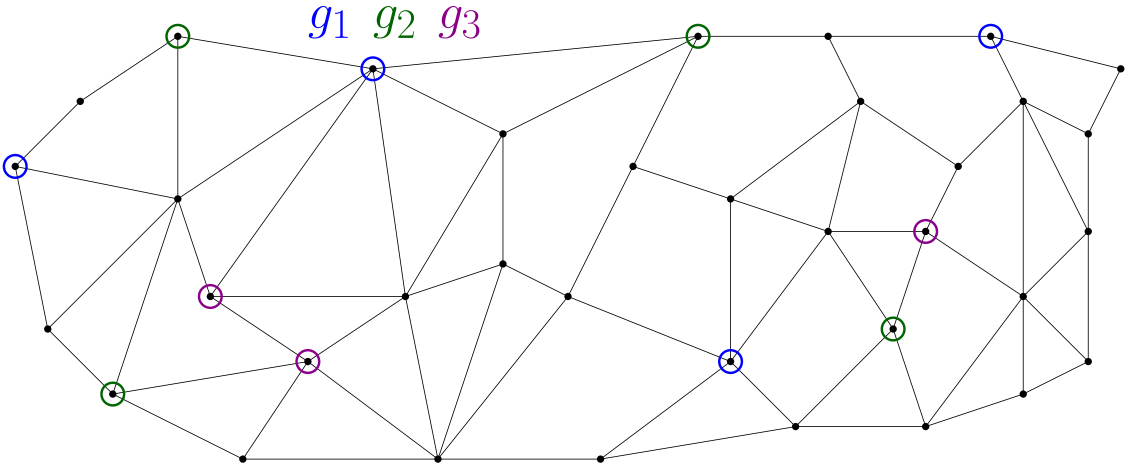
Given set of terminals K, find minimum weight tree T spanning K



In class we saw a 2-approximation algorithm for the Steiner tree problem.

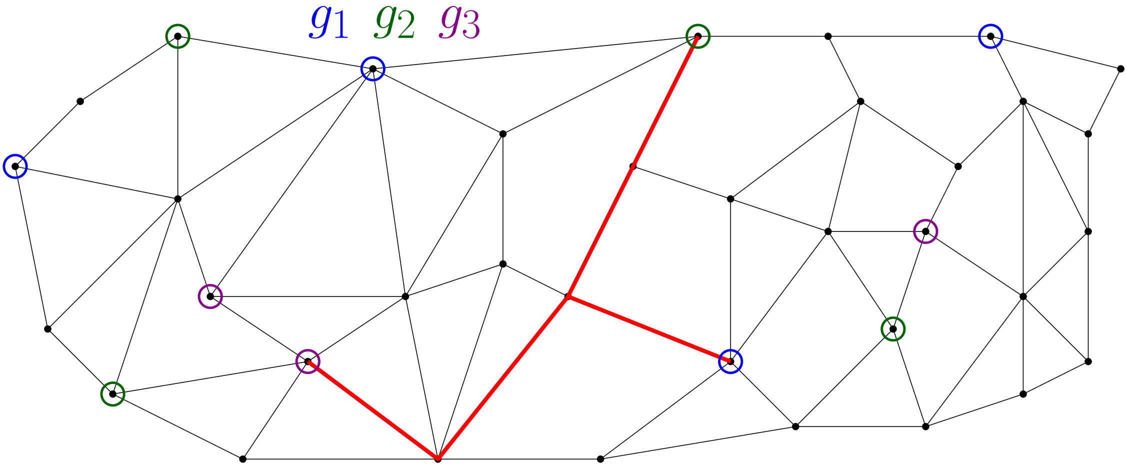
Group Steiner Tree

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i



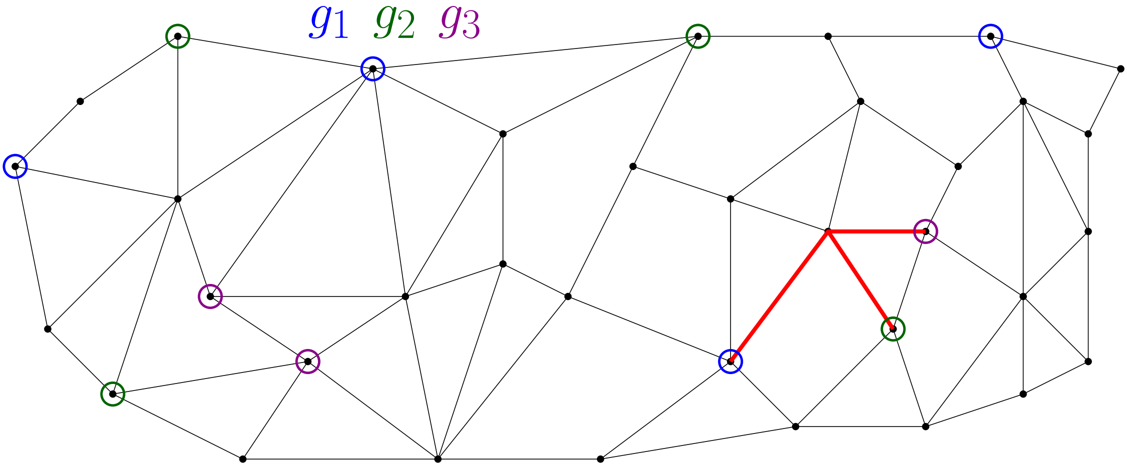
Group Steiner Tree

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i



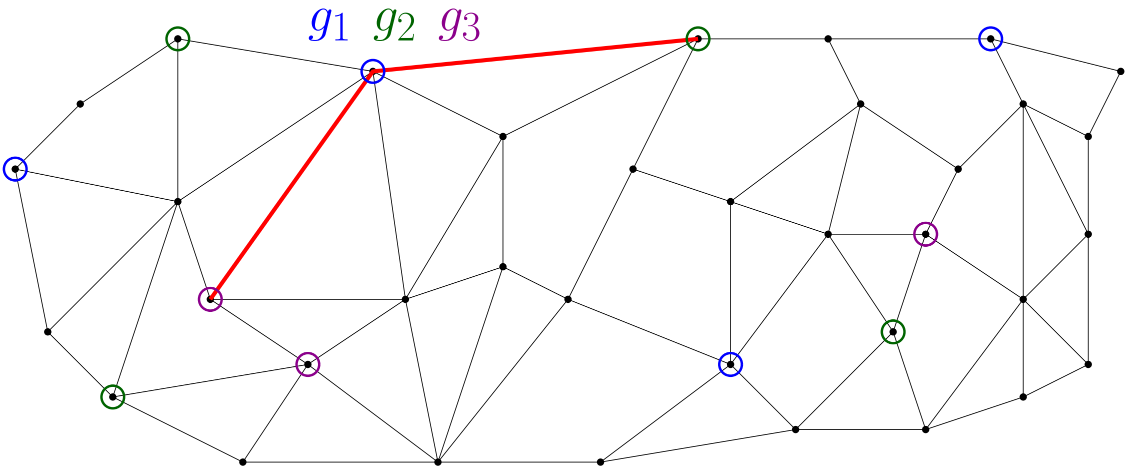
Group Steiner Tree

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i



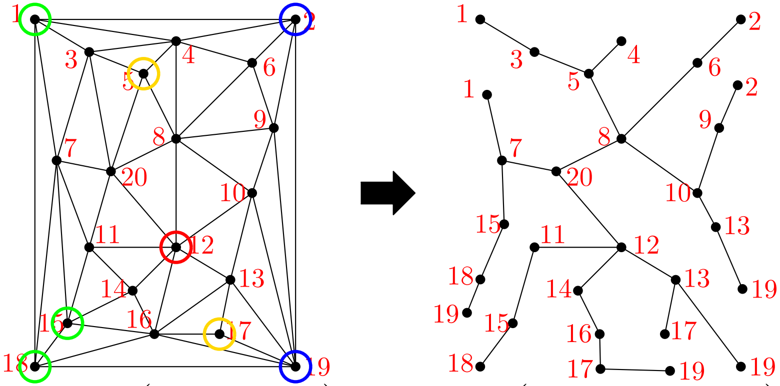
Group Steiner Tree

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i



Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Clan embedding f with path distortion $O(\log n)$.



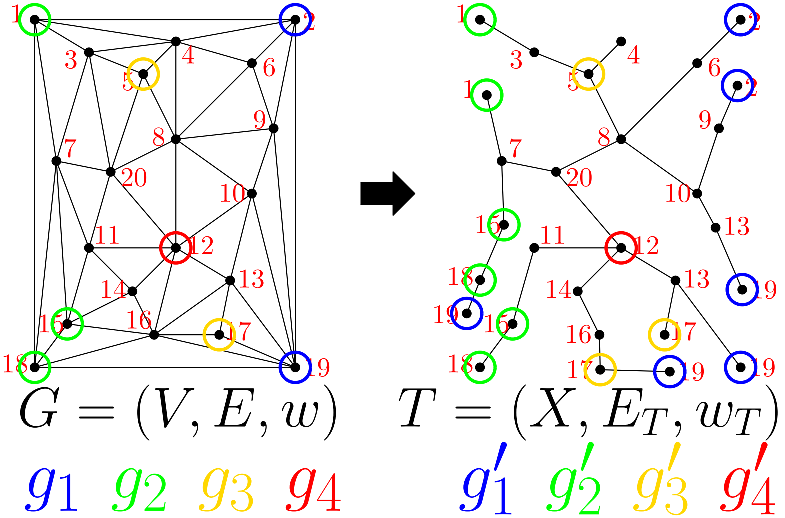
$$G = (V, E, w) \qquad T = (X, E_T, w_T)$$

$g_1 \quad g_2 \quad g_3 \quad g_4$

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Clan embedding f with path distortion $O(\log n)$.

$g_i' = f(g_i)$

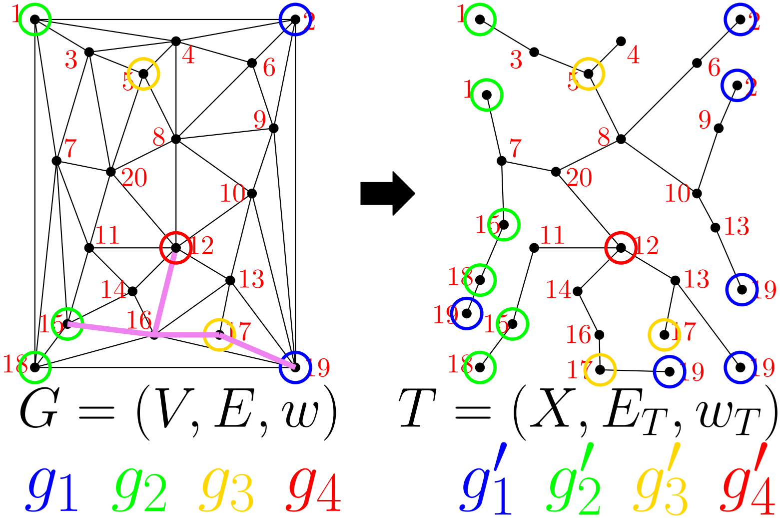


Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Clan embedding f with path distortion $O(\log n)$.

$g_i' = f(g_i)$

$S^\star$ optimal solution.



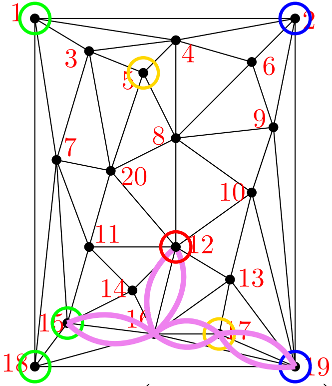
Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

Clan embedding f with path distortion $O(\log n)$.

$g_i' = f(g_i)$

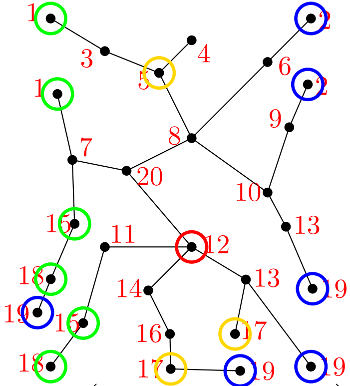
$S^\star$ optimal solution.

Double each edge: $2S^\star$



$G = (V, E, w)$

$g_1 \quad g_2 \quad g_3 \quad g_4$



$T = (X, E_T, w_T)$

$g_1' \quad g_2' \quad g_3' \quad g_4'$

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

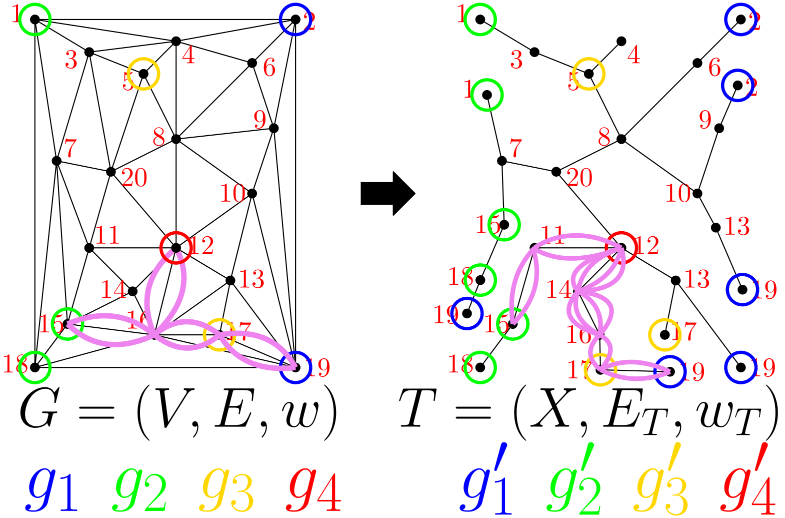
Clan embedding f with path distortion $O(\log n)$.

$g_i' = f(g_i)$

$S^\star$ optimal solution.

Double each edge: $2S^\star$

Guaranteed path: $S_T^\star$ (valid solution), $w(S_T^\star) \le O(\log n) \cdot w(S^\star)$



Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

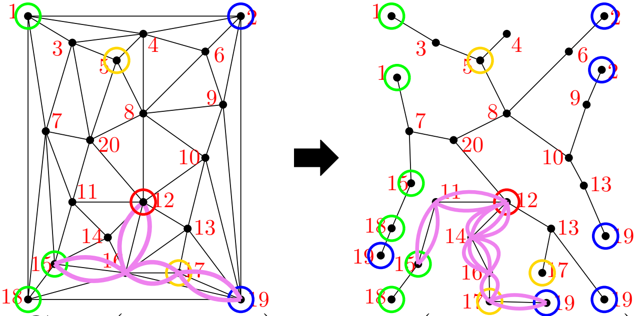
Clan embedding f with path distortion $O(\log n)$.

$g'_i = f(g_i)$

$S^\star$ optimal solution.

Double each edge: $2S^\star$

Guaranteed path: $S^\star_T$, $w(S^\star_T) \leq O(\log n) \cdot w(S^\star)$



$G = (V, E, w)$ $\quad$ $T = (X, E_T, w_T)$

g_1 g_2 g_3 g_4 $\quad$ g'_1 g'_2 g'_3 g'_4

Theorem ([Garg, Konjevod, Ravi 00])

$O(\log n \cdot \log k)$-approximation algorithm for the GST problem on trees.

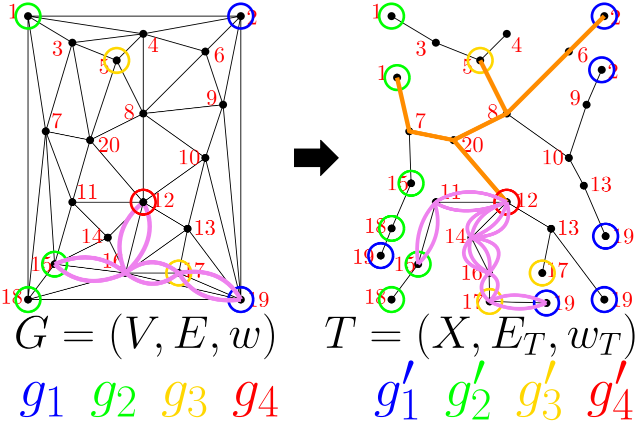
Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

$S^\star$ optimal solution.

Double each edge: $2S^\star$

Guaranteed path: $S^\star_T$, $w(S^\star_T) \leq O(\log n) \cdot w(S^\star)$

$\tilde{S}_T$ solution, $w(\tilde{S}_T) \leq O(\log n \cdot \log k) \cdot w(S^\star_T) \leq O(\log^2 n \cdot \log k) \cdot w(S^\star)$



Theorem ([Garg, Konjevod, Ravi 00])

$O(\log n \cdot \log k)$-approximation algorithm for the GST problem on trees.

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

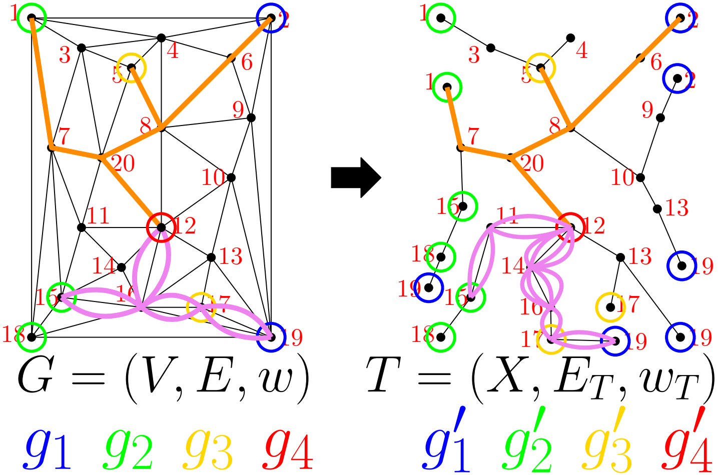
$S^\star$ optimal solution.

Double each edge: $2S^\star$

Guaranteed path: $S^\star_T$,
$w(S^\star_T) \le O(\log n) \cdot w(S^\star)$

$\tilde{S}_T$ solution, $w(\tilde{S}_T) \le O(\log n \cdot \log k) \cdot w(S^\star_T) \le O(\log^2 n \cdot \log k) \cdot w(S^\star)$

Return: $\tilde{S} = f^{-1}(\tilde{S}_T)$



$\tilde{S} = \cup_{\{v',u'\} \in \tilde{S}_T} P^T_{v',u'}$ is a union of shortest paths in G

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

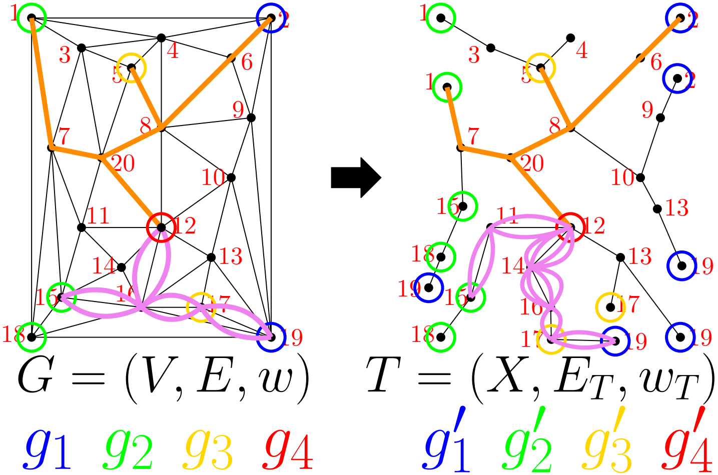
$S^\star$ optimal solution.

Double each edge: $2S^\star$

Guaranted path: $S^\star_T$,
$w(S^\star_T) \leq O(\log n) \cdot w(S^\star)$

$\tilde{S}_T$ solution, $w(\tilde{S}_T) \leq O(\log n \cdot \log k) \cdot w(S^\star_T) \leq O(\log^2 n \cdot \log k) \cdot w(S^\star)$

Return: $\tilde{S} = f^{-1}(\tilde{S}_T)$



$\tilde{S} = \cup_{\{v', u'\} \in \tilde{S}_T} P^T_{v', u'}$ is a union of shortest paths in G

$$w(\tilde{S}) \leq w(\tilde{S}_T) \leq O(\log^2 n \cdot \log k) \cdot w(S^\star).$$

Given subsets $g_1, g_2, \ldots, g_k \subseteq V$, find minimum weight tree T spanning at least one vertex from each g_i.

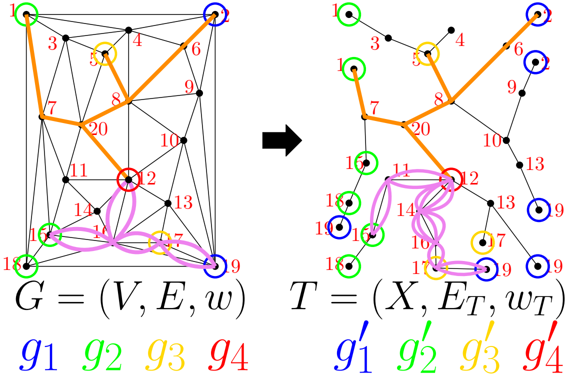
$S^\star$ optimal solution.

Double each edge: $2S^\star$

Guaranteed path: $S^\star_T$, $w(S^\star_T) \leq O(\log n) \cdot w(S^\star)$

$\tilde{S}_T$ solution, $w(\tilde{S}_T) \leq O(\log n \cdot \log k) \cdot w(S^\star_T) \leq O(\log^2 n \cdot \log k) \cdot w(S^\star)$

Return: $\tilde{S} = f^{-1}(\tilde{S}_T)$



$$w(\tilde{S}) \leq w(\tilde{S}_T) \leq O(\log^2 n \cdot \log k) \cdot w(S^\star).$$

We got an $O(\log^2 n \cdot \log k)$ **approximation**.

Clan Embeddings construction

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) n point metric space, $\forall k \in \mathbb{N}$, there is ***distribution*** *$\mathcal{D}$ over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion* $O(k)$.
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq O(n^{\frac{1}{k}})$

Clan Embeddings construction

Theorem (Clan embedding into trees, [Filtser, Le 21])

*(X, d_X) n point metric space, $\forall k \in \mathbb{N}$, there is **distribution** $\mathcal{D}$ over **dominating clan embeddings** into trees such that:*

- *$\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ has distortion $O(k)$.*
- *$\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq O(n^{\frac{1}{k}})$*

Lemma

*(X, d_X) n point metric, there is **dominating clan embedding** into a tree s.t.:*

- *(f, χ) has distortion $O(k)$.*
- *$\sum_{x\in X} |f(x)| \leq O(n^{1+\frac{1}{k}})$*

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) n point metric space, $\forall k \in \mathbb{N}$, there is ***distribution*** *$\mathcal{D}$ over* ***dominating clan embeddings*** *into trees such that:*

- *$\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ has distortion $O(k)$.*
- *$\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq O(n^{\frac{1}{k}})$*

Replace cardinality with weights

Lemma

(X, d_X) n point metric, and weights $\mu : X \to \mathbb{R}_{\geq 1}$

there is ***dominating clan embedding*** *into a tree s.t.:*

- *(f, χ) has distortion $O(k)$.*
- $\sum_{x\in X} |f(x)| \cdot \mu(x) \leq O(\mu(X)^{1+\frac{1}{k}})$

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) n point metric space, $\forall k \in \mathbb{N}$, there is ***distribution*** *$\mathcal{D}$ over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion $O(k)$.*
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq O(n^{\frac{1}{k}})$

Replace cardinality with weights ⇑⇑⇑ mini-max theorem

Lemma

(X, d_X) n point metric, and weights $\mu : X \to \mathbb{R}_{\geq 1}$

there is ***dominating clan embedding*** *into a tree s.t.:*

- *(f, χ) has distortion $O(k)$.*
- $\sum_{x\in X} |f(x)| \cdot \mu(x) \leq O(\mu(X)^{1+\frac{1}{k}})$

Theorem (Clan embedding into trees, [Filtser, Le 21])

(X, d_X) *n point metric space,* $\forall k \in \mathbb{N}$, *there is* ***distribution*** $\mathcal{D}$ *over* ***dominating clan embeddings*** *into trees such that:*

- $\forall\ (f, \chi) \in \mathrm{supp}(\mathcal{D})$ *has distortion* $O(k)$.
- $\forall\ x \in X$, $\mathbb{E}_{(f,\chi)\sim\mathcal{D}}[|f(x)|] \leq O(n^{\frac{1}{k}})$

Replace cardinality with weights ⇑⇑⇑ mini-max theorem

Lemma

(X, d_X) *n point metric, there is* ***dominating clan embedding*** *into a tree s.t.:*

- (f, χ) *has distortion* $O(k)$.
- $\sum_{x\in X} |f(x)| \leq O(n^{1+\frac{1}{k}})$

Definition (Ultrametric)

Ultrametric (X, d) is a metric space satisfying the **strong** triangle inequality:

$$\forall x, y, z \in X, \qquad d(x, z) \leq \max\{d(x, y), d(y, z)\} \ .$$

Definition (Ultrametric)

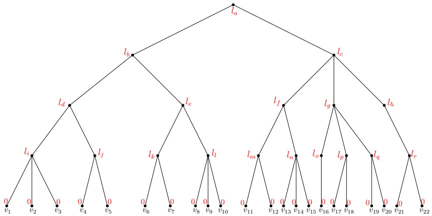
Ultrametric (X, d) is a metric space satisfying the **strong** triangle inequality:

$$\forall x, y, z \in X, \qquad d(x, z) \leq \max\{d(x, y), d(y, z)\} \ .$$

Definition (Hierarchical well-separated tree (HST))

(X, d_X) is a HST if X is mapped (by ϕ) to **leaves** of a rooted tree T where:

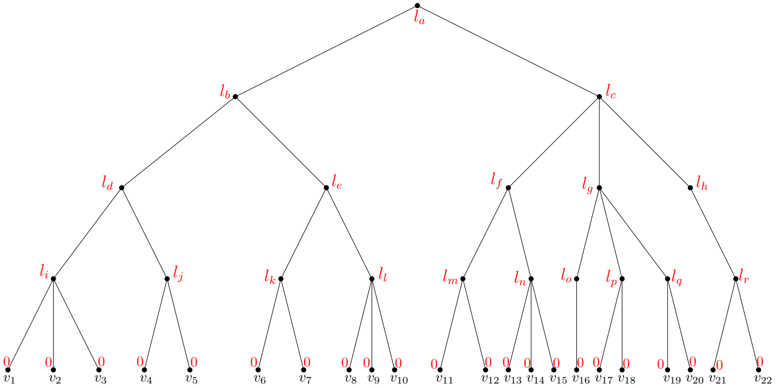
- The nodes of T associated with **monotone** labels l_v.
- $d_X(x, y) = \Gamma_{\mathrm{lca}(\varphi(x), \varphi(y))}$.



Definition (Hierarchical well-separated tree (HST))

(X, d_X) is a HST if X is mapped (by ϕ) to **leaves** of a rooted tree T where:

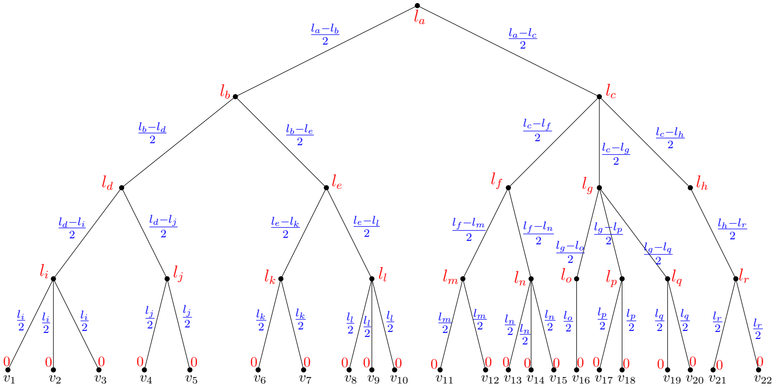
- The nodes of T associated with **monotone** labels l_v.
- $d_X(x, y) = \Gamma_{\mathrm{lca}(\varphi(x),\varphi(y))}$.



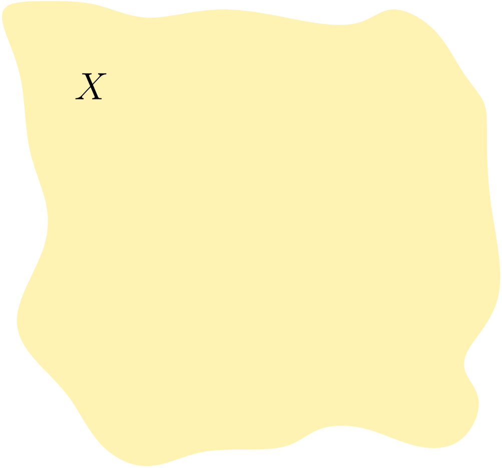
Definition (Hierarchical well-separated tree (HST))

(X, d_X) is a HST if X is mapped (by ϕ) to **leaves** of a rooted tree T where:

- The nodes of T associated with **monotone** labels l_v.
- $d_X(x, y) = \Gamma_{\mathrm{lca}(\varphi(x),\varphi(y))}$.

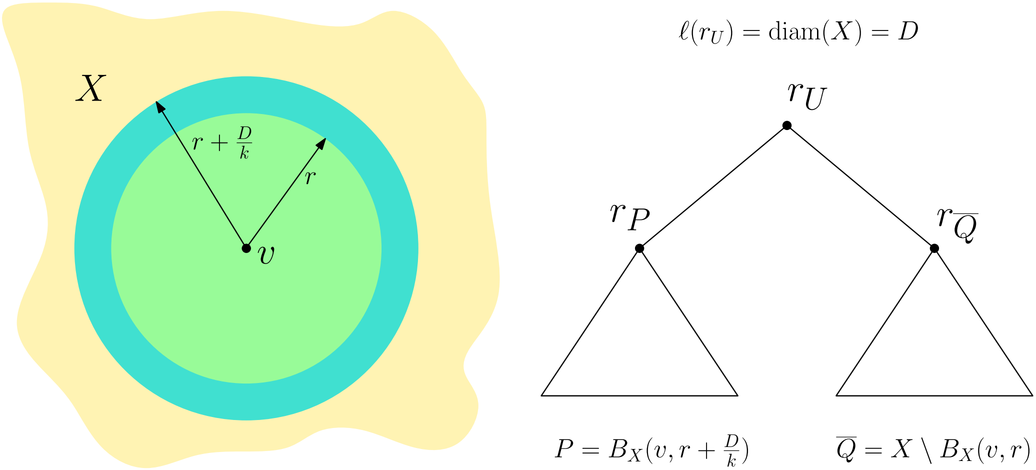


Construction

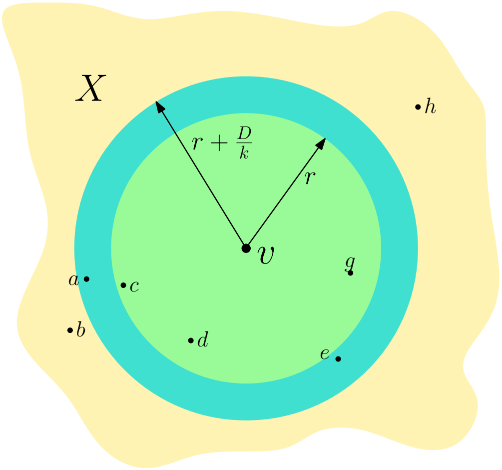


$$\ell(r_U) = \mathrm{diam}(X) = D$$

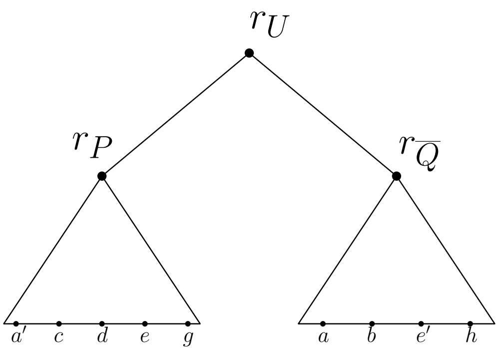
Construction



Construction

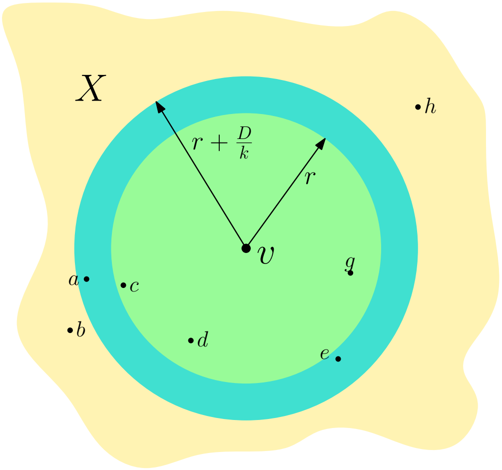


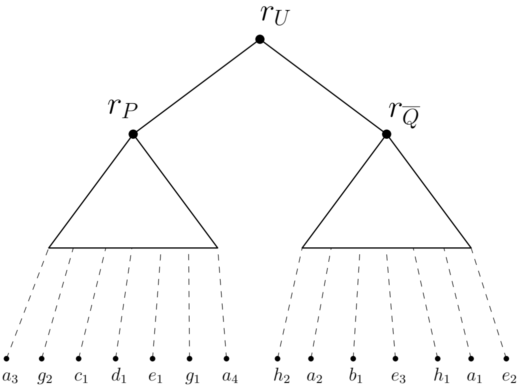
$$\ell(r_U) = \mathrm{diam}(X) = D$$



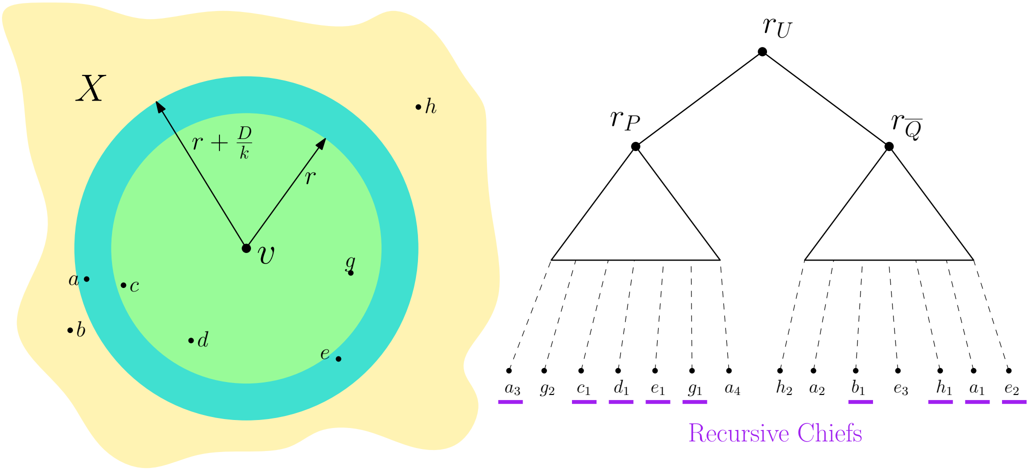
$P = B_X(v, r + \frac{D}{k})$ $\overline{Q} = X \setminus B_X(v, r)$

Construction

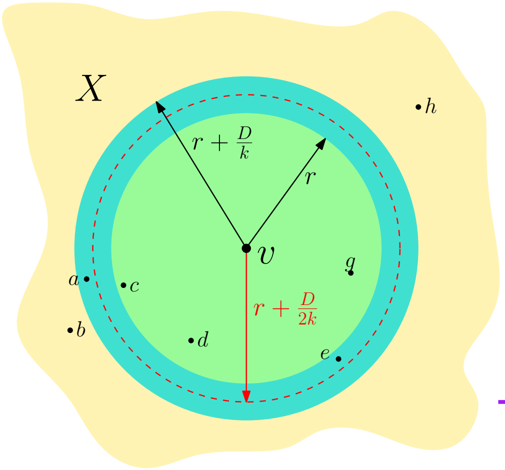


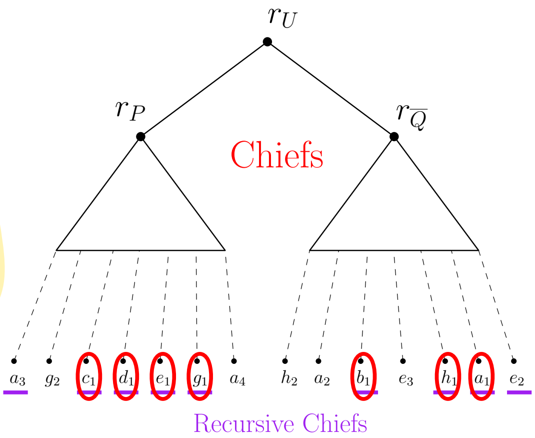


Construction

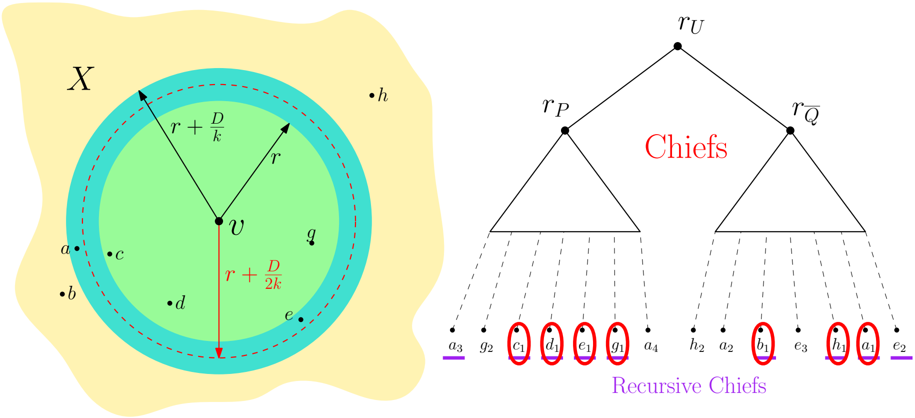


Construction



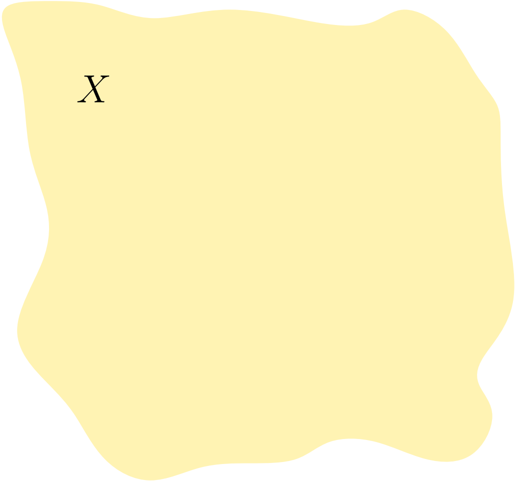


Construction - distortion bound



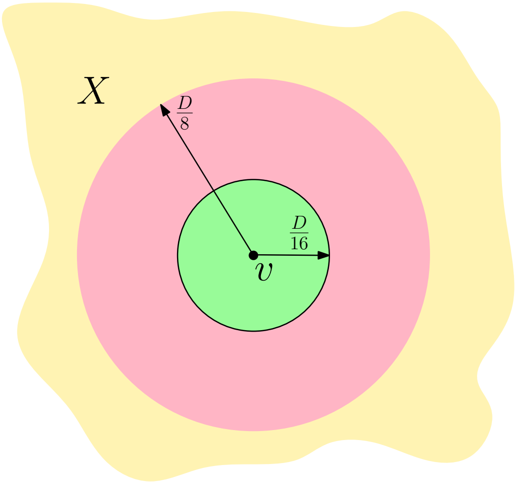
$$\min_{c' \in f(c)} d_U(c', \chi(a)) = D \le 2k \cdot d_X(c, a) \ .$$

Construction - cardinality bound



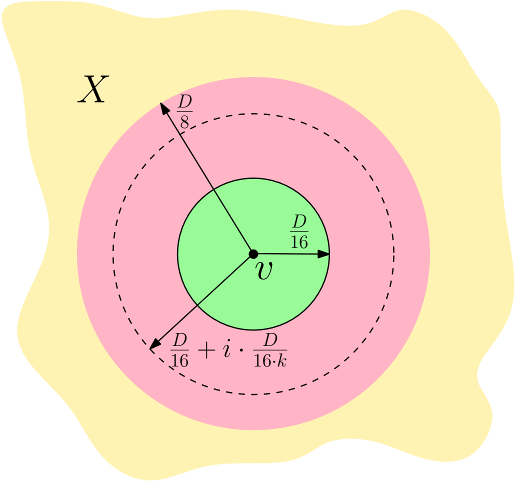
$$D = \operatorname{diam}(X)$$

Construction - cardinality bound



$$D = \mathrm{diam}(X)$$

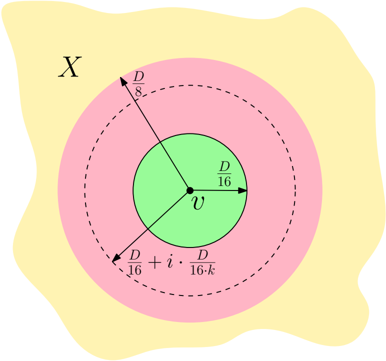
Construction - cardinality bound



$$D = \operatorname{diam}(X)$$

$$A_i = B_X(v, \tfrac{D}{16} + i \cdot \tfrac{D}{16 \cdot k})$$
$$i \in \{0, \ldots, k-1\}$$

Construction - cardinality bound

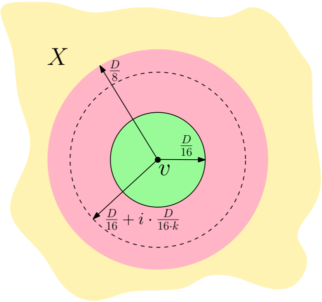


$D = \mathrm{diam}(X)$

$A_i = B_X(v, \frac{D}{16} + i \cdot \frac{D}{16 \cdot k})$
$i \in \{0, \ldots, k-1\}$

There is some i s.t. $\frac{|A_{i+1}|}{|A_i|} \leq \left(\frac{|A_k|}{|A_0|}\right)^{\frac{1}{k}}$.

Construction - cardinality bound



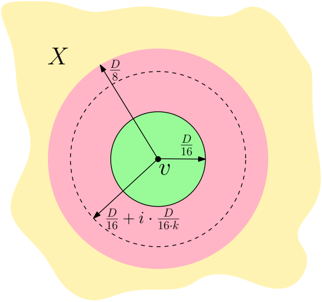
$D = \mathrm{diam}(X)$

$A_i = B_X(v, \frac{D}{16} + i \cdot \frac{D}{16 \cdot k})$
$i \in \{0, \ldots, k-1\}$

There is some i s.t. $\frac{|A_{i+1}|}{|A_i|} \leq \left(\frac{|A_k|}{|A_0|}\right)^{\frac{1}{k}}$. Otherwise

$$|A_k| > |A_{k-1}| \cdot \left(\frac{|A_k|}{|A_0|}\right)^{\frac{1}{k}} > |A_{k-2}| \cdot \left(\frac{|A_k|}{|A_0|}\right)^{\frac{2}{k}} > \cdots > |A_0| \cdot \left(\frac{|A_k|}{|A_0|}\right)^{\frac{k}{k}} = |A_k|$$

Construction - cardinality bound



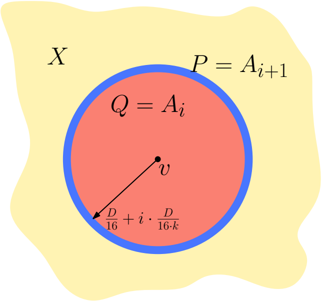
$D = \mathrm{diam}(X)$

$A_i = B_X(v, \frac{D}{16} + i \cdot \frac{D}{16 \cdot k})$
$i \in \{0, \ldots, k-1\}$

There is some i s.t. $\frac{|A_{i+1}|}{|A_i|} \leq \left(\frac{|A_k|}{|A_0|}\right)^{\frac{1}{k}}$.

Claim

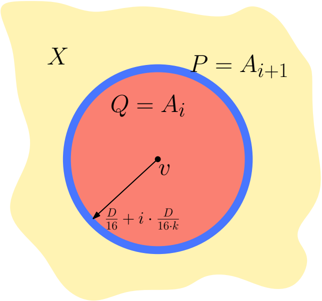
There is $v \in X$, and i, s.t. $\frac{|A_{i+1}|}{|A_i|} \leq \left(\frac{\mu^*(X)}{\mu^*(A_{i+1})}\right)^{1/k} = \left(\frac{\max_{x \in X} \left|B_X(x, \frac{\mathrm{diam(X)}}{4})\right|}{\max_{x \in A_{i+1}} \left|B_{A_{i+1}}(x, \frac{\mathrm{diam(A_{i+1})}}{4})\right|}\right)^{1/k}$.



Claim

There is $v \in X$*, and* i*, s.t.* $\frac{|A_{i+1}|}{|A_i|} \leq \left(\frac{\mu^*(X)}{\mu^*(A_{i+1})}\right)^{1/k} = \left(\frac{\max_{x \in X}\left|B_X(x, \frac{\mathrm{diam(X)}}{4})\right|}{\max_{x \in A_{i+1}}\left|B_{A_{i+1}}(x, \frac{\mathrm{diam(A_{i+1})}}{4})\right|}\right)^{1/k}$.

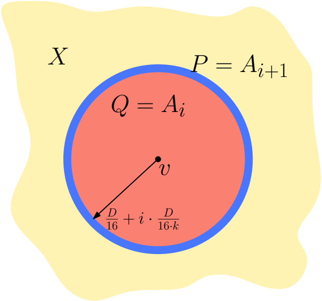
Set $P = A_{i+1}$, $Q = A_i$, and $\overline{Q} = X \setminus A_i$.



Claim

There is $v \in X$, and i, s.t. $\frac{|A_{i+1}|}{|A_i|} \leq \left(\frac{\mu^*(X)}{\mu^*(A_{i+1})}\right)^{1/k} = \left(\frac{\max_{x \in X} \left|B_X(x, \frac{\mathrm{diam(X)}}{4})\right|}{\max_{x \in A_{i+1}} \left|B_{A_{i+1}}(x, \frac{\mathrm{diam(A_{i+1})}}{4})\right|}\right)^{1/k}$.

Set $P = A_{i+1}$, $Q = A_i$, and $\overline{Q} = X \setminus A_i$.
By the claim: $|P| \cdot \mu^*(P)^{\frac{1}{k}} \leq |Q| \cdot \mu^*(X)^{\frac{1}{k}}$.



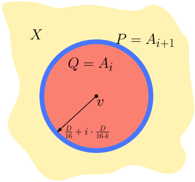
Claim

There is $v \in X$, and i, s.t. $\frac{|A_{i+1}|}{|A_i|} \leq \left(\frac{\mu^*(X)}{\mu^*(A_{i+1})}\right)^{1/k} = \left(\frac{\max_{x \in X} \left|B_X(x, \frac{\mathrm{diam}(X)}{4})\right|}{\max_{x \in A_{i+1}} \left|B_{A_{i+1}}(x, \frac{\mathrm{diam}(A_{i+1})}{4})\right|}\right)^{1/k}.$

Set $P = A_{i+1}$, $Q = A_i$, and $\overline{Q} = X \setminus A_i$.

By the claim: $|P| \cdot \mu^*(P)^{\frac{1}{k}} \leq |Q| \cdot \mu^*(X)^{\frac{1}{k}}$.

Recurse on P and $\overline{Q}$.



Claim

There is $v \in X$, and i, s.t. $\frac{|A_{i+1}|}{|A_i|} \leq \left(\frac{\mu^*(X)}{\mu^*(A_{i+1})}\right)^{1/k} = \left(\frac{\max_{x\in X}\left|B_X(x, \frac{\mathrm{diam(X)}}{4})\right|}{\max_{x\in A_{i+1}}\left|B_{A_{i+1}}(x, \frac{\mathrm{diam(A_{i+1})}}{4})\right|}\right)^{1/k}$.

Set $P = A_{i+1}$, $Q = A_i$, and $\overline{Q} = X \setminus A_i$.

By the claim: $|P| \cdot \mu^*(P)^{\frac{1}{k}} \leq |Q| \cdot \mu^*(X)^{\frac{1}{k}}$. Recurse on P and $\overline{Q}$.

We argue by induction: $|f(X)| \leq |X| \cdot \mu^*(X)^{\frac{1}{k}} \leq |X|^{1+\frac{1}{k}}$.

Here $|f(X)| = \sum_{x\in X} |f(x)|$.

Claim

There is $v \in X$, *and* i, *s.t.* $\frac{|A_{i+1}|}{|A_i|} \le \left(\frac{\mu^*(X)}{\mu^*(A_{i+1})}\right)^{1/k} = \left(\frac{\max_{x\in X}\left|B_X(x, \frac{\mathrm{diam(X)}}{4})\right|}{\max_{x\in A_{i+1}}\left|B_{A_{i+1}}(x, \frac{\mathrm{diam(A_{i+1})}}{4})\right|}\right)^{1/k}$.

Set $P = A_{i+1}$, $Q = A_i$, and $\overline{Q} = X \setminus A_i$.

By the claim: $|P| \cdot \mu^*(P)^{\frac{1}{k}} \le |Q| \cdot \mu^*(X)^{\frac{1}{k}}$. Recurse on P and $\overline{Q}$.

We argue by induction: $|f(X)| \le |X| \cdot \mu^*(X)^{\frac{1}{k}} \le |X|^{1+\frac{1}{k}}$.

$$|f(X)| = |f(P)| + |f(\overline{Q})|$$

Claim

There is $v \in X$, and i, s.t. $\frac{|A_{i+1}|}{|A_i|} \le \left(\frac{\mu^*(X)}{\mu^*(A_{i+1})}\right)^{1/k} = \left(\frac{\max_{x \in X} \left|B_X(x, \frac{\mathrm{diam(X)}}{4})\right|}{\max_{x \in A_{i+1}} \left|B_{A_{i+1}}(x, \frac{\mathrm{diam(A_{i+1})}}{4})\right|}\right)^{1/k}$.

Set $P = A_{i+1}$, $Q = A_i$, and $\overline{Q} = X \setminus A_i$.
By the claim: $|P| \cdot \mu^*(P)^{\frac{1}{k}} \le |Q| \cdot \mu^*(X)^{\frac{1}{k}}$. Recurse on P and $\overline{Q}$.

We argue by induction: $|f(X)| \le |X| \cdot \mu^*(X)^{\frac{1}{k}} \le |X|^{1+\frac{1}{k}}$.

$$\begin{aligned} |f(X)| &= |f(P)| + |f(\overline{Q})| \\ &\le |P| \cdot \mu^*(P)^{\frac{1}{k}} + \left|\overline{Q}\right| \cdot \mu^*(\overline{Q})^{\frac{1}{k}} \end{aligned}$$

Claim

There is $v \in X$, and i, s.t. $\frac{|A_{i+1}|}{|A_i|} \leq \left(\frac{\mu^*(X)}{\mu^*(A_{i+1})}\right)^{1/k} = \left(\frac{\max_{x\in X}\left|B_X(x, \frac{\mathrm{diam(X)}}{4})\right|}{\max_{x\in A_{i+1}}\left|B_{A_{i+1}}(x, \frac{\mathrm{diam(A_{i+1})}}{4})\right|}\right)^{1/k}$.

Set $P = A_{i+1}$, $Q = A_i$, and $\overline{Q} = X \setminus A_i$.

By the claim: $|P| \cdot \mu^*(P)^{\frac{1}{k}} \leq |Q| \cdot \mu^*(X)^{\frac{1}{k}}$. Recurse on P and $\overline{Q}$.

We argue by induction: $|f(X)| \leq |X| \cdot \mu^*(X)^{\frac{1}{k}} \leq |X|^{1+\frac{1}{k}}$.

$$\begin{aligned}
|f(X)| &= |f(P)| + |f(\overline{Q})| \\
&\leq |P| \cdot \mu^*(P)^{\frac{1}{k}} + \left|\overline{Q}\right| \cdot \mu^*(\overline{Q})^{\frac{1}{k}} \\
&\leq |Q| \cdot \mu^*(X)^{\frac{1}{k}} + \left|\overline{Q}\right| \cdot \mu^*(X)^{\frac{1}{k}}
\end{aligned}$$

Claim

There is $v \in X$, and i, s.t. $\frac{|A_{i+1}|}{|A_i|} \leq \left(\frac{\mu^*(X)}{\mu^*(A_{i+1})}\right)^{1/k} = \left(\frac{\max_{x\in X}\left|B_X(x,\frac{\mathrm{diam(X)}}{4})\right|}{\max_{x\in A_{i+1}}\left|B_{A_{i+1}}(x,\frac{\mathrm{diam(A_{i+1})}}{4})\right|}\right)^{1/k}$.

Set $P = A_{i+1}$, $Q = A_i$, and $\overline{Q} = X \setminus A_i$.
By the claim: $|P| \cdot \mu^*(P)^{\frac{1}{k}} \leq |Q| \cdot \mu^*(X)^{\frac{1}{k}}$. Recurse on P and $\overline{Q}$.

We argue by induction: $|f(X)| \leq |X| \cdot \mu^*(X)^{\frac{1}{k}} \leq |X|^{1+\frac{1}{k}}$.

$$\begin{aligned}
|f(X)| &= |f(P)| + |f(\overline{Q})| \\
&\leq |P| \cdot \mu^*(P)^{\frac{1}{k}} + \left|\overline{Q}\right| \cdot \mu^*(\overline{Q})^{\frac{1}{k}} \\
&\leq |Q| \cdot \mu^*(X)^{\frac{1}{k}} + \left|\overline{Q}\right| \cdot \mu^*(X)^{\frac{1}{k}} \\
&= |X| \cdot \mu^*(X)^{\frac{1}{k}} \ .
\end{aligned}$$

Claim

There is $v \in X$, and i, s.t. $\frac{|A_{i+1}|}{|A_i|} \le \left(\frac{\mu^*(X)}{\mu^*(A_{i+1})}\right)^{1/k} = \left(\frac{\max_{x \in X}\left|B_X(x, \frac{\mathrm{diam(X)}}{4})\right|}{\max_{x \in A_{i+1}}\left|B_{A_{i+1}}(x, \frac{\mathrm{diam(A_{i+1})}}{4})\right|}\right)^{1/k}$.

Set $P = A_{i+1}$, $Q = A_i$, and $\overline{Q} = X \setminus A_i$.

By the claim: $|P| \cdot \mu^*(P)^{\frac{1}{k}} \le |Q| \cdot \mu^*(X)^{\frac{1}{k}}$. Recurse on P and $\overline{Q}$.

We argue by induction: $|f(X)| \le |X| \cdot \mu^*(X)^{\frac{1}{k}} \le |X|^{1+\frac{1}{k}}$.

$$
\begin{aligned}
|f(X)| &= |f(P)| + |f(\overline{Q})| \\
&\le |P| \cdot \mu^*(P)^{\frac{1}{k}} + \left|\overline{Q}\right| \cdot \mu^*(\overline{Q})^{\frac{1}{k}} \\
&\le |Q| \cdot \mu^*(X)^{\frac{1}{k}} + \left|\overline{Q}\right| \cdot \mu^*(X)^{\frac{1}{k}} \\
&= |X| \cdot \mu^*(X)^{\frac{1}{k}} \;.
\end{aligned}
$$