Graph Spanners for Group Steiner Distances

Davide Bilò[1], Luciano Gualà[2],

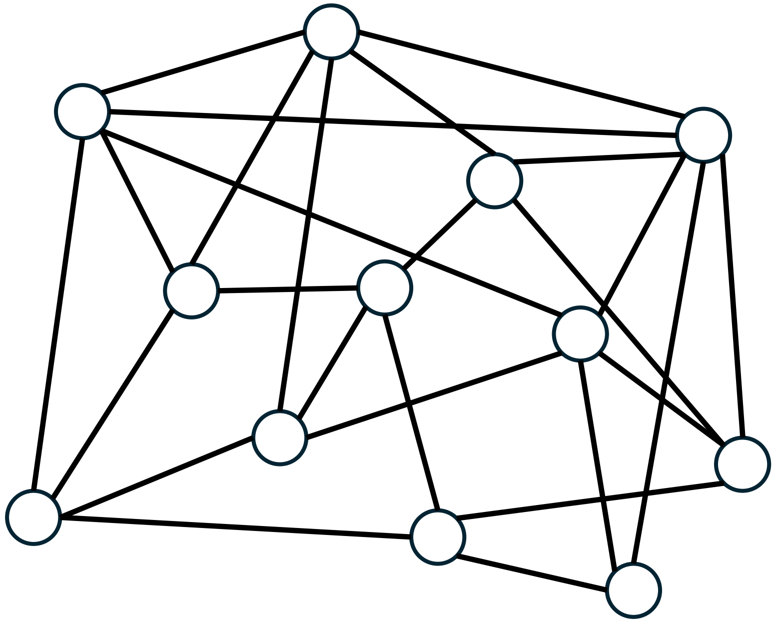
Stefano Leucci[1], **Alessandro Straziota**[2]

(1) UNIVERSITÀ DEGLI STUDI DELL'AQUILA

(2) TOR VERGATA UNIVERSITÀ DEGLI STUDI DI ROMA

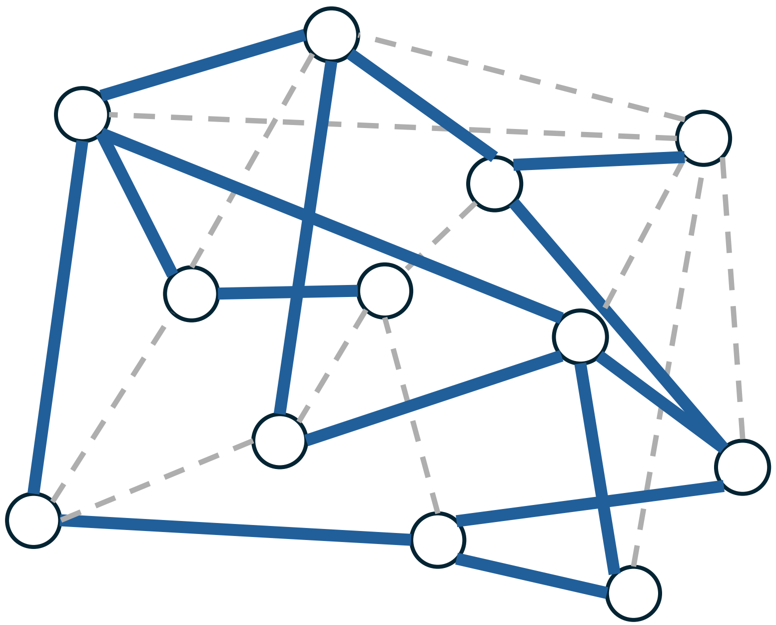
Graph Spanners

Graph Spanners



G

Graph Spanners

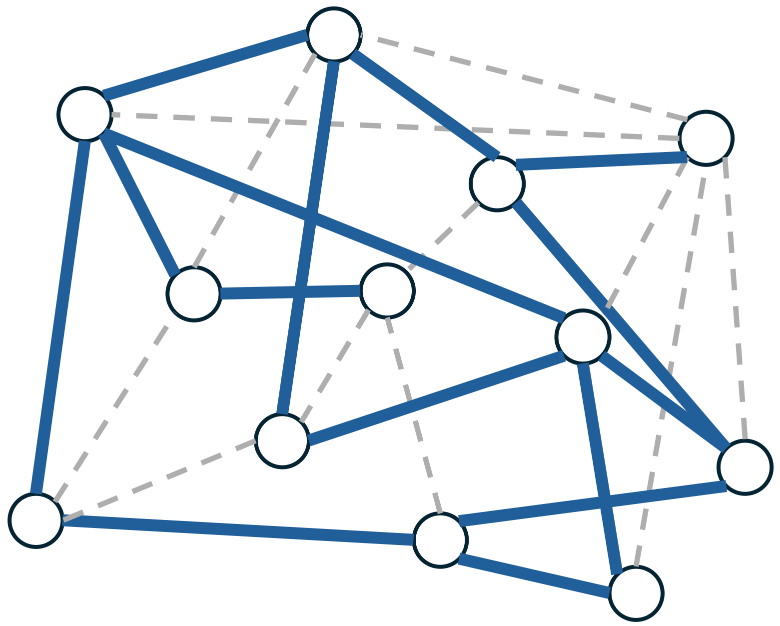


$H \subseteq G$

H is an α-spanner of G if

$$d_H(u,v) \leq \alpha \, d_G(u,v) \quad \forall u,v \in V(G)$$

Graph Spanners

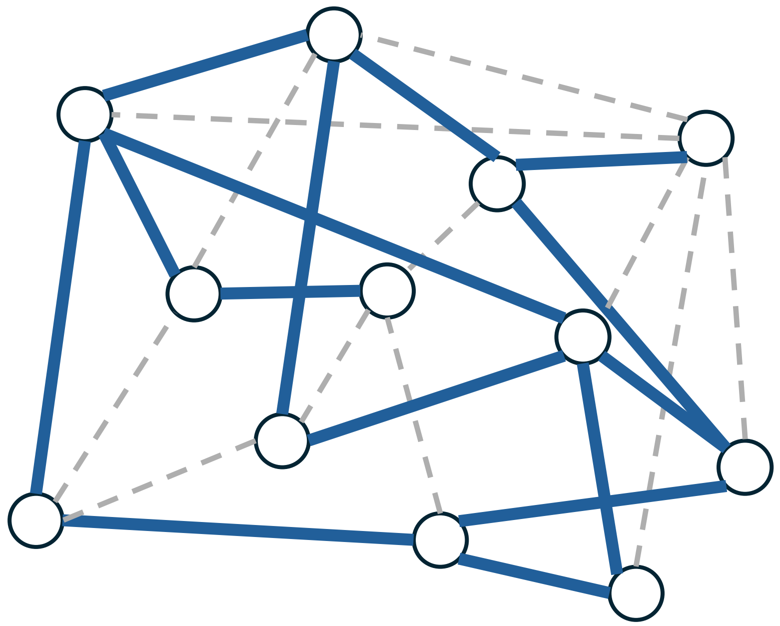


$H \subseteq G$

H is an α-spanner of G if

$$d_H(u,v) \leq \boxed{\alpha}\, d_G(u,v) \quad \forall u,v \in V(G)$$

Graph Spanners



$H \subseteq G$

H is an α-spanner of G if

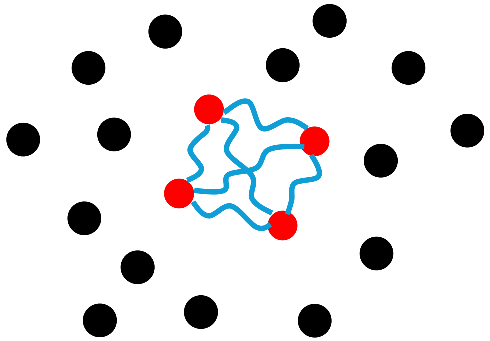
$$d_H(u,v) \leq \boxed{\alpha}\, d_G(u,v) \quad \forall u,v \in V(G)$$

stretch factor

GOAL: small size and low stretch

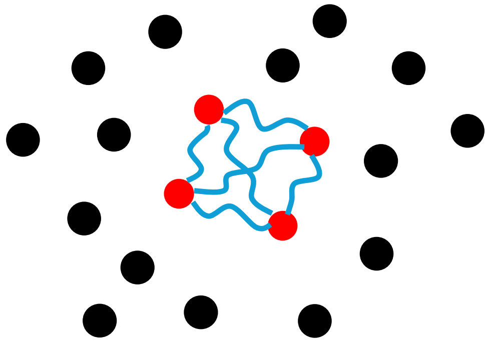
Some variants

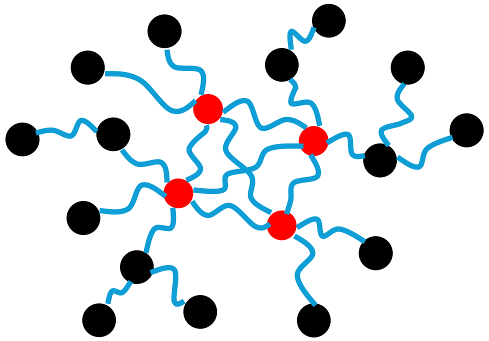
Subset-wise spanner
or $R \times R$ spanner



Some variants

Subset-wise spanner
or $R \times R$ spanner

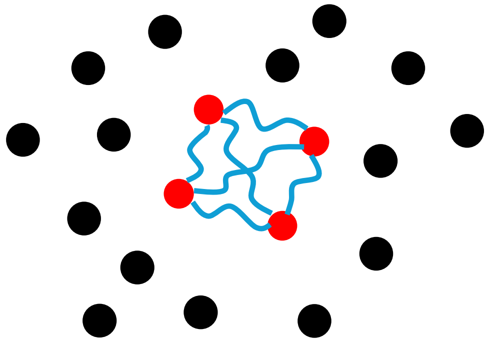


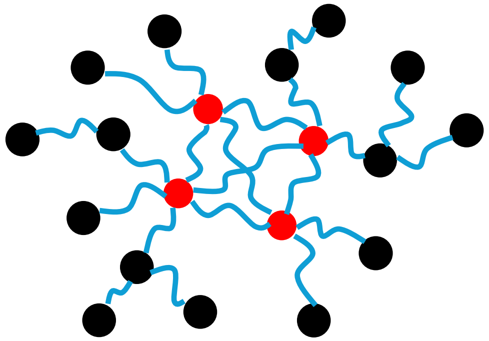


Source-wise spanner
or $R \times V$ spanner

Some variants

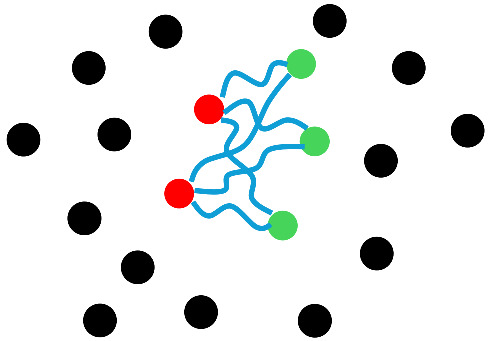
Subset-wise spanner
or $R \times R$ spanner





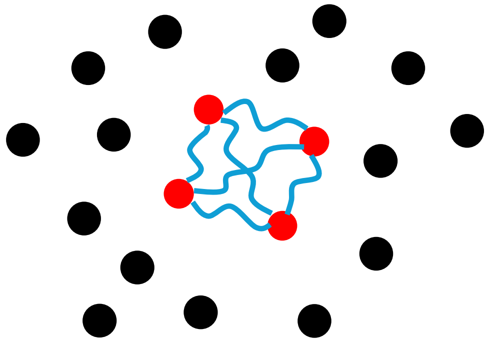
Source-wise spanner
or $R \times V$ spanner

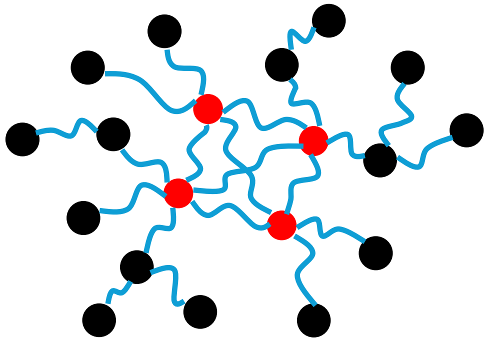
Source-Target spanner
or $S \times T$ spanner



Some variants

Subset-wise spanner
or $R \times R$ spanner



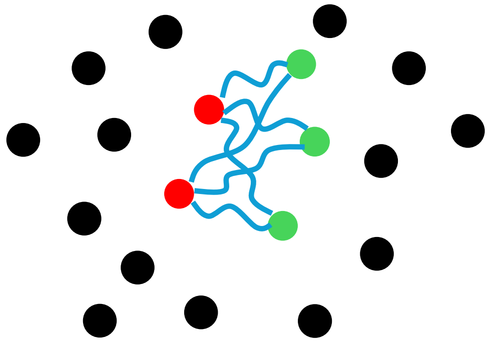


Source-wise spanner
or $R \times V$ spanner

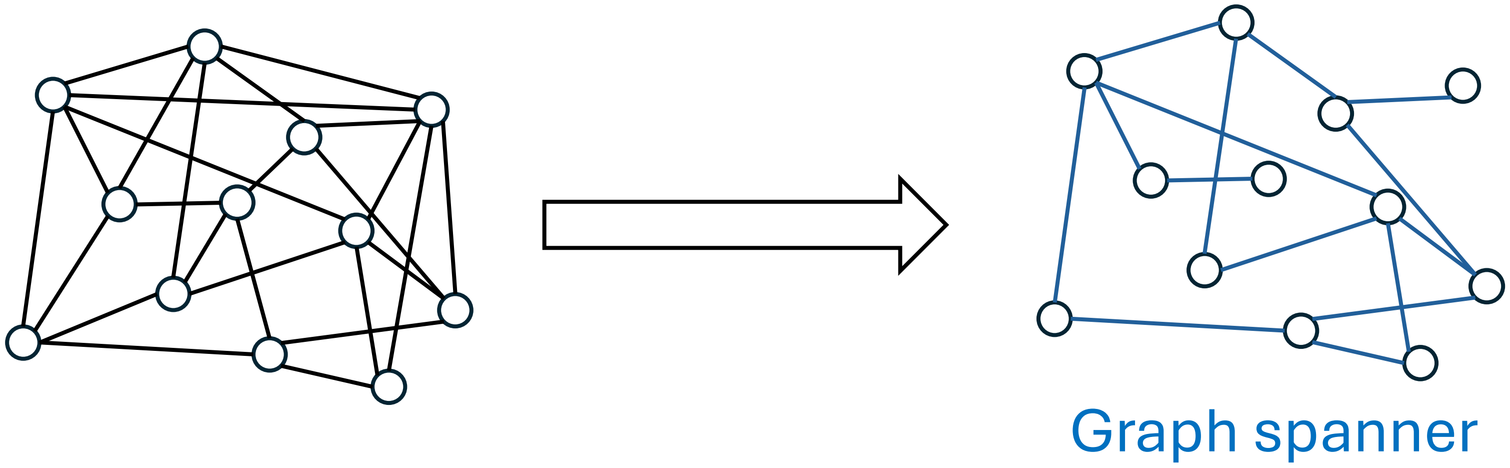
$S = T = R$

$S = R$
$T = V$

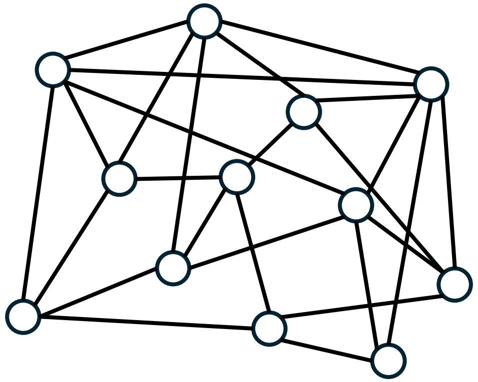
Source-Target spanner
or $S \times T$ spanner

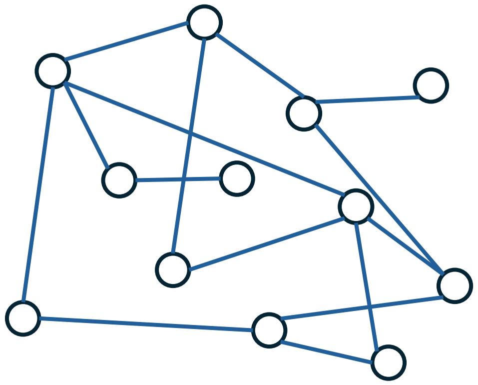


Distance oracle



Distance oracle

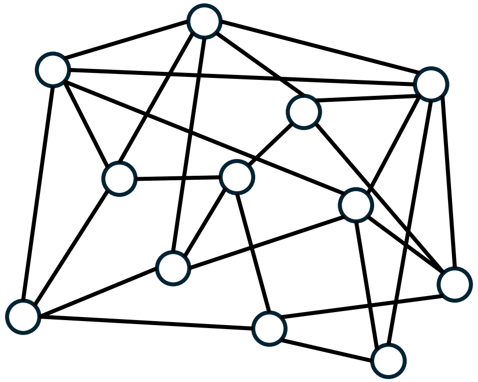


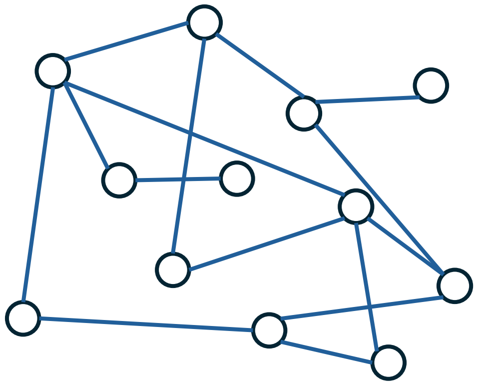


Graph spanner

Distance Oracle: **compact** data structure able to **quickly** report exact or approximated distances.

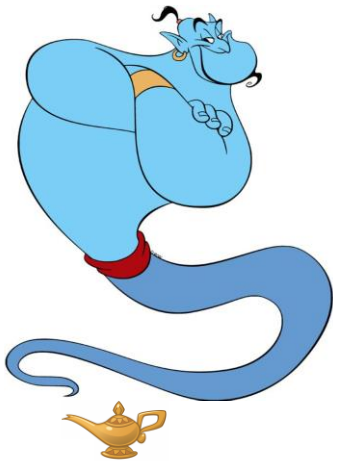
Distance oracle





Graph spanner

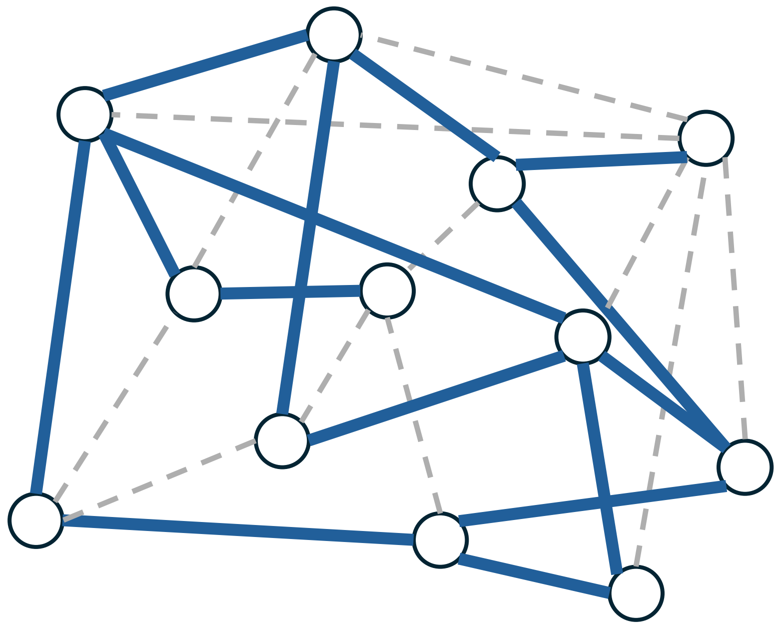
A spanner is a distance oracle
with polynomial query time



Distance Oracle: **compact** data structure able to **quickly** report exact or approximated distances.

Group Steiner Spanner

Our question



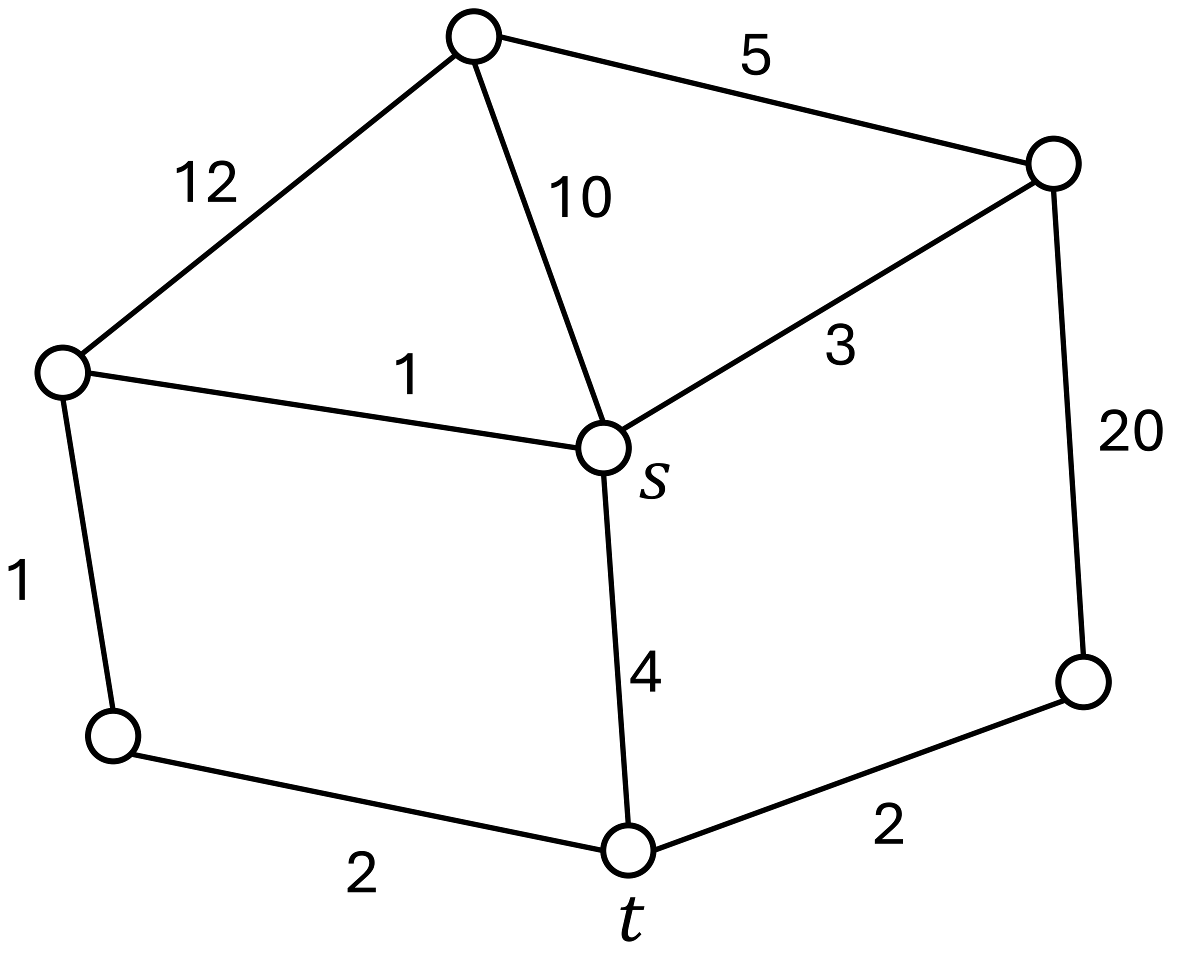
$H \subseteq G$

H is an α-spanner of G if

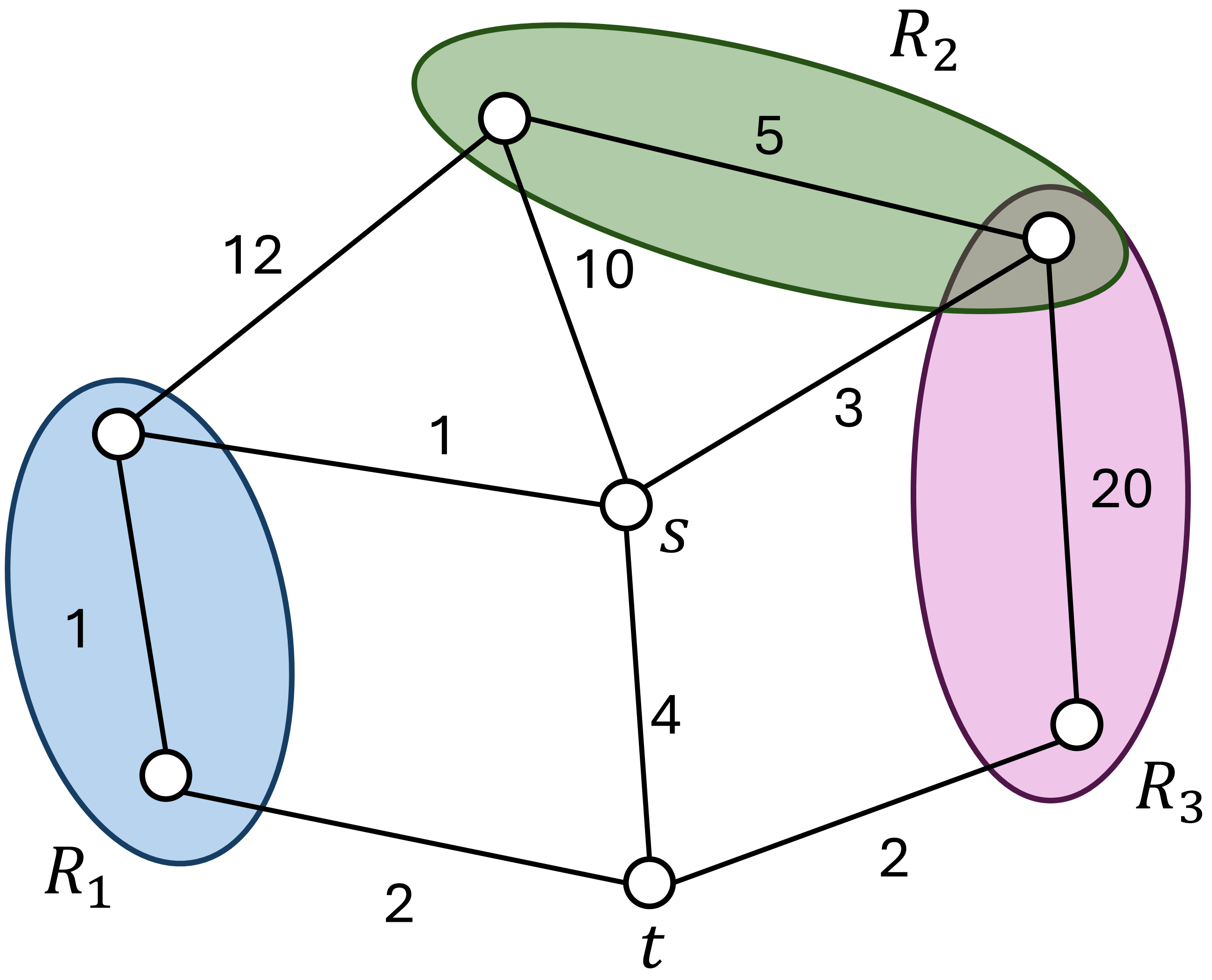
$$\underbrace{d_H(u,v) \leq \alpha\, d_G(u,v)} \quad \forall u,v \in V(G)$$

What if we consider a **different metric** then the shortest path distance?

Group Steiner Distance

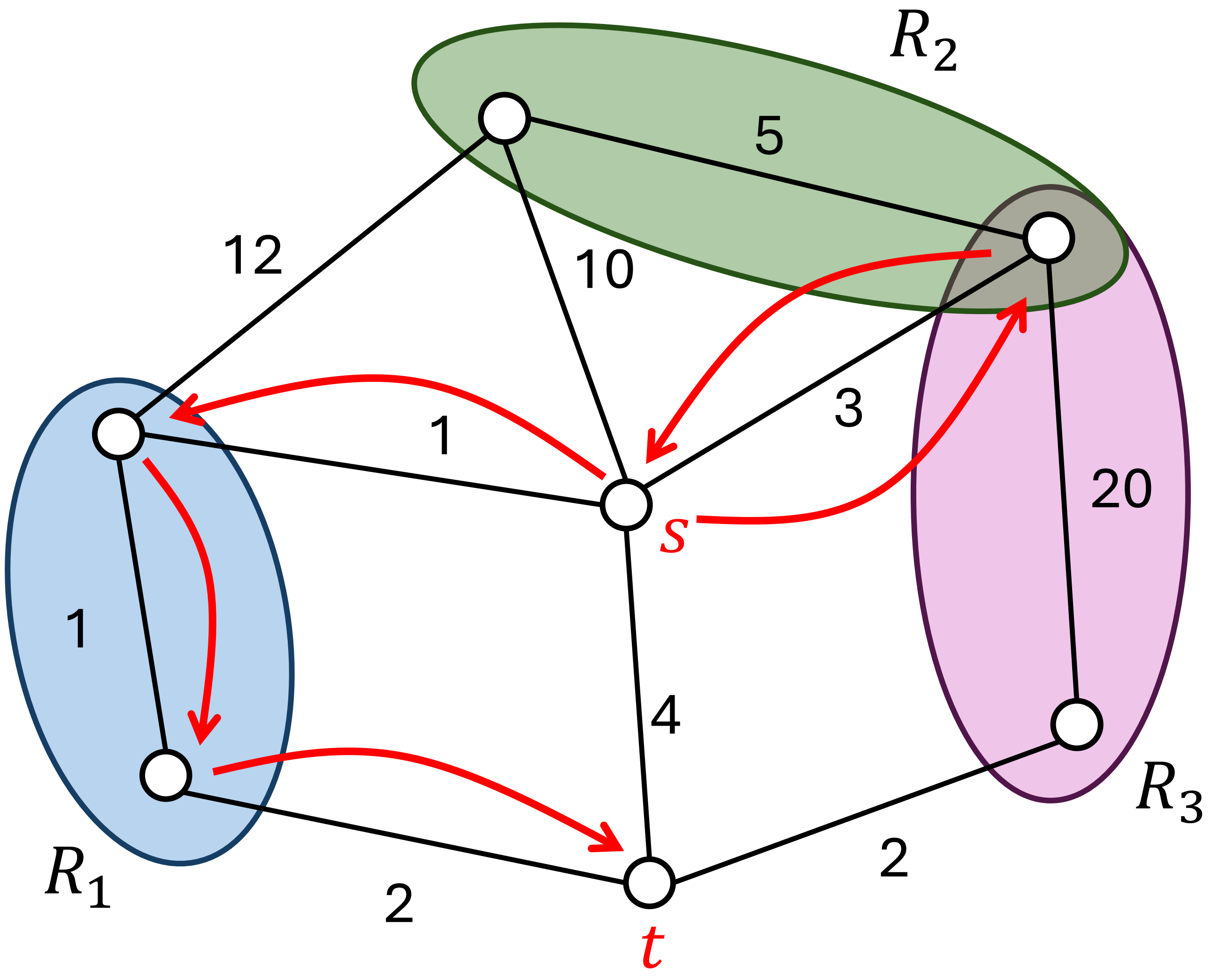


Group Steiner Distance



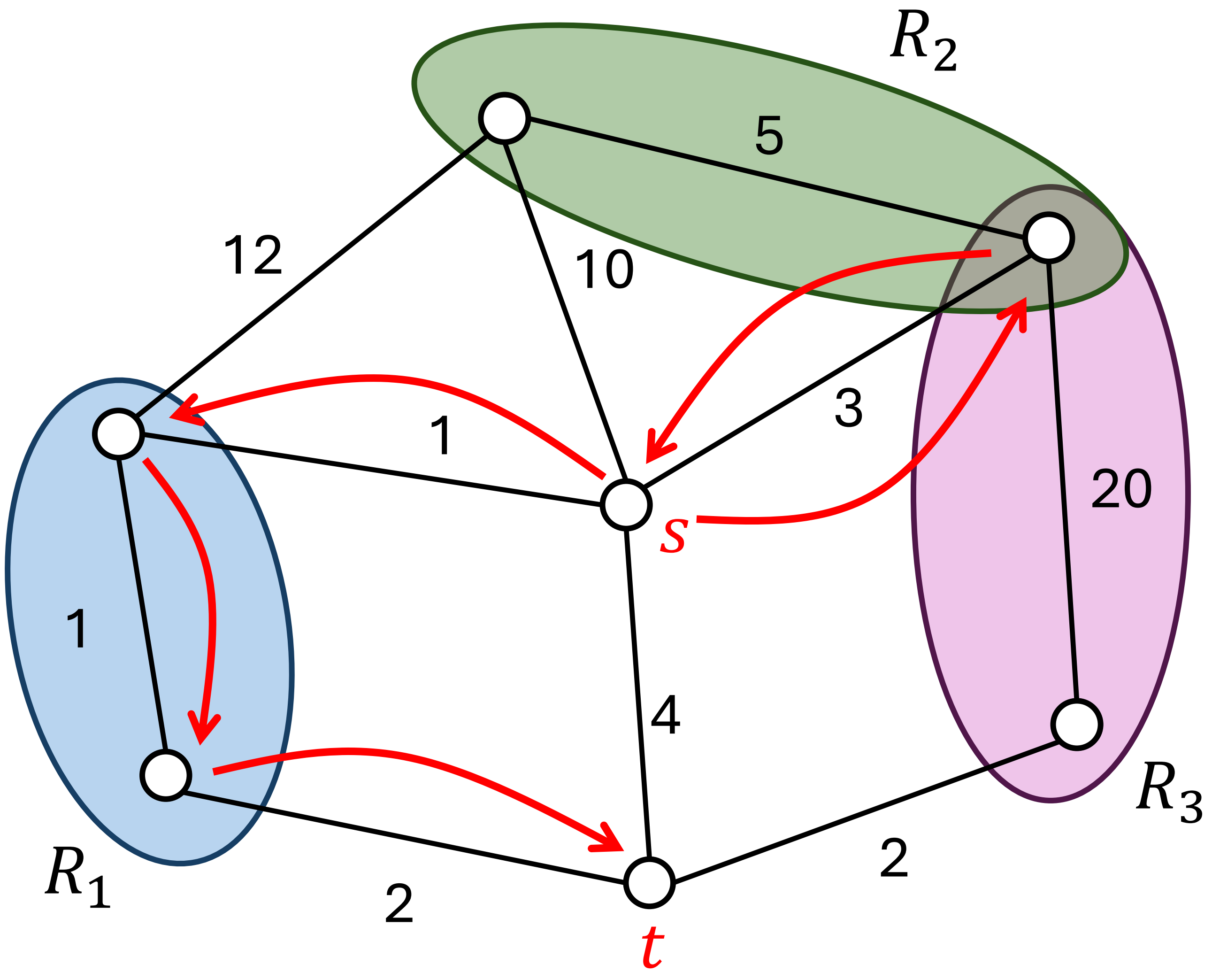
- A collection $\mathcal{R} = \{R_1, \ldots, R_k\}$ of k groups of subest of **required vertices**

Group Steiner Distance



- A collection $\mathcal{R} = \{R_1, \ldots, R_k\}$ of k groups of subest of **required vertices**
- Group Steiner path from s to t wrt $\mathcal{R}$: a walk that traverses at least one required vertice for each group, in **any order**

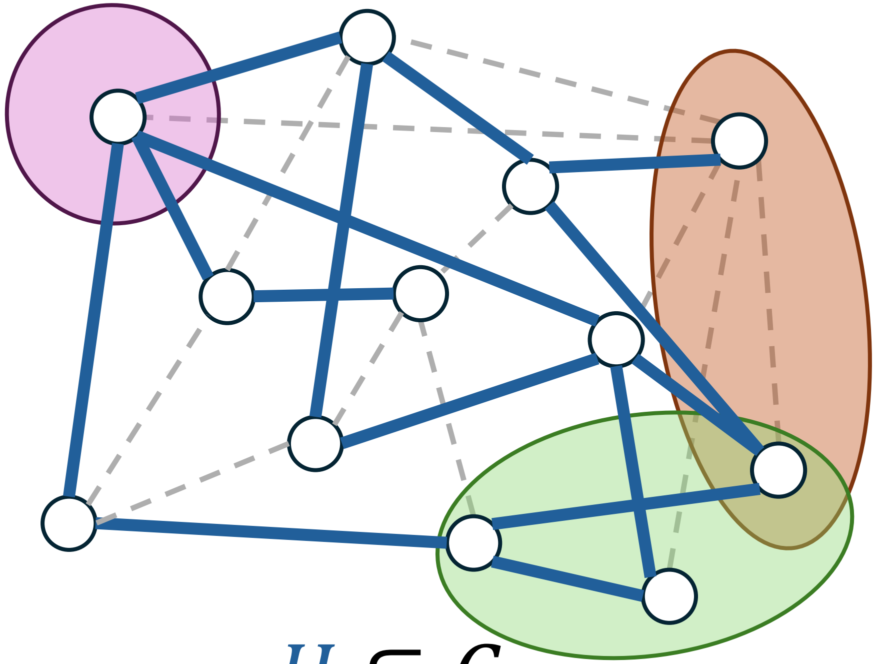
Group Steiner Distance



$\sigma_G(s,t) = 3 + 3 + 1 + 1 + 2 = 10$

- A collection $\mathcal{R} = \{R_1, \ldots, R_k\}$ of k groups of subest of **required vertices**
- Group Steiner path from s to t wrt $\mathcal{R}$: a walk that traverses at least one required vertice for each group, in **any order**
- Group Steiner distance $\sigma_G(s,t)$: the sum of weights (**with their molteplicity**) of the edges in shortest group Steiner path

Group Steiner Spanner



$H \subseteq G$

H is a **Group Steiner** α-spanner of G if

$$\sigma_H(u, v) \leq \alpha\, \sigma_G(u, v) \quad \forall u, v \in V(G)$$

Computing Group Steiner Distances

(results from minimum Group Steiner Tree problem)

- NP-hard for large k

Computing Group Steiner Distances

(results from minimum Group Steiner Tree problem)

- NP-hard for large k

- [Halperin & Krauthgamer - STOC 2003] there exists no poly-time $O(\log^{2-\varepsilon} k)$-apx, unless $NP \subseteq ZPTIME\left(n^{polylog(n)}\right)$

Computing Group Steiner Distances

(results from minimum Group Steiner Tree problem)

- NP-hard for large k
- [Halperin & Krauthgamer - STOC 2003] there exists no poly-time $O(\log^{2-\varepsilon} k)$-apx, unless $NP \subseteq ZPTIME\left(n^{polylog(n)}\right)$
- We can compute GS path and distances in **FPT** time $2^k k n^{O(1)}$, which is polynomial for small $k = O(\log n)$

Previous Results

When $k = 1$ GP-distance is equivalent to the **Beer-distance** [Bacic et. al. – ISAAC 2021]

Several works about exact beer-distance oracle for **special** classes of graphs:

- Outer planar graphs [Bacic et. al. – ISAAC 2021]
- Interval graphs [Das et. al. – ISAAC 2022]
- Bounded tree-width graphs [Gudmundsson et. al. – ISAAC 2023]

Previous Results

When $k = 1$ GP-distance is equivalent to the **Beer-distance** [Bacic et. al. – ISAAC 2

Several w asses of graphs:

- Outer p
- Interval
- Bounded tree-width graphs [Gudmundsson et. al. – ISAAC 2023]

Our focus is on:

1. General GP-distances ($k \geq 1$)
2. General graphs

Our Results

Our results

- Gruoup Steiner Spanner for
  1. General group sizes (and relation with classical source-wise)
  2. The singleton case
  3. General group sizes – single source

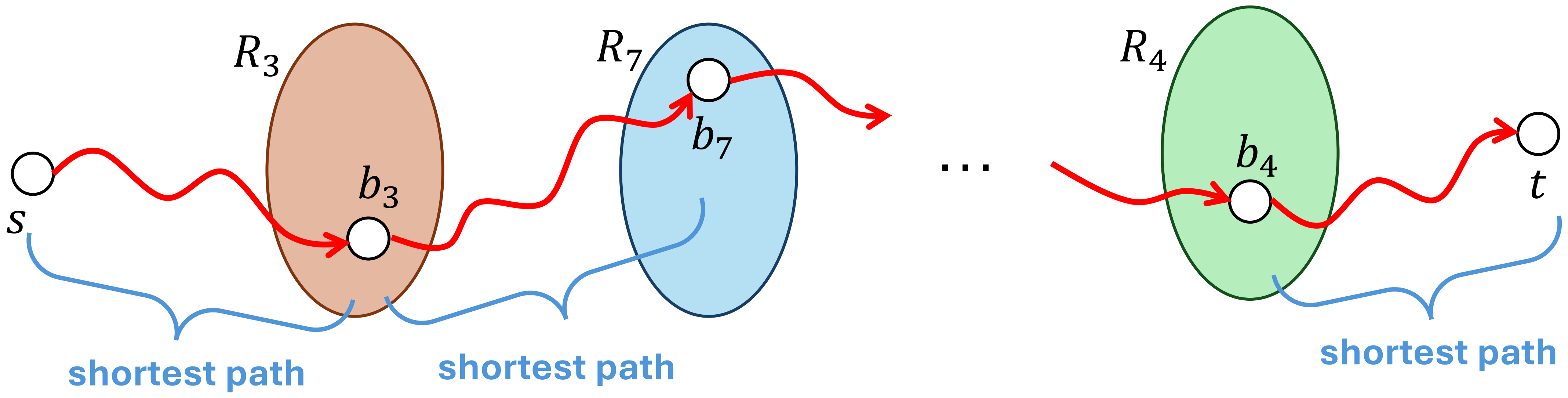
Our results

- Gruoup Steiner Spanner for
    1. General group sizes (and relation with classical source-wise)
    2. The singleton case
    3. General group sizes – single source

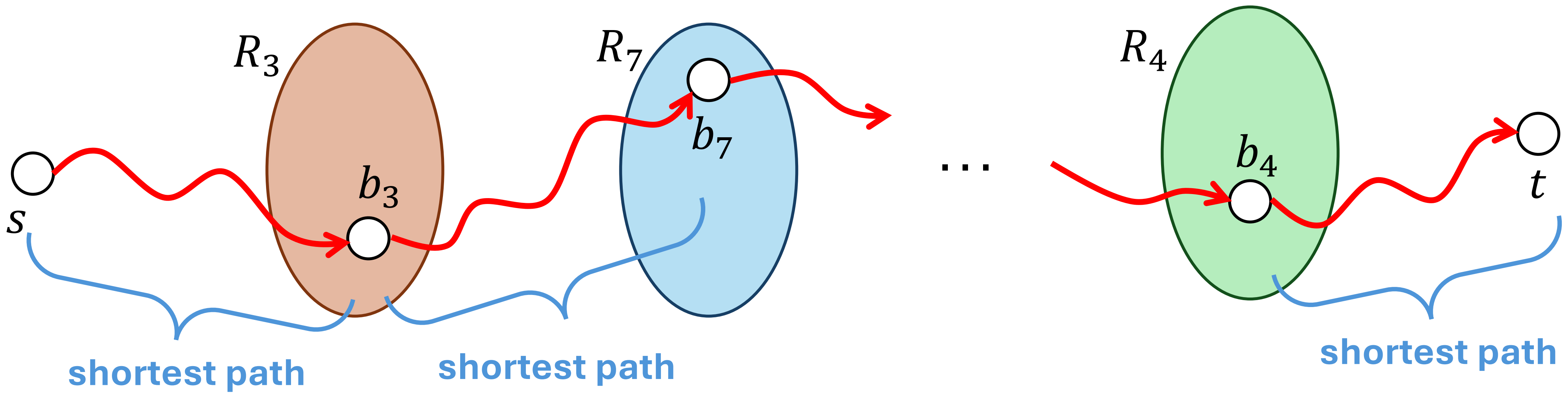
- How to turn our spanners into distance oracles with
    1. Same stretch
    2. Same (asymptotic) size
    3. Same building time
    4. Non-trivial query time

General group sizes

GS-Spanners vs Source-Wise Spanner



GS-Spanners vs Source-Wise Spanner

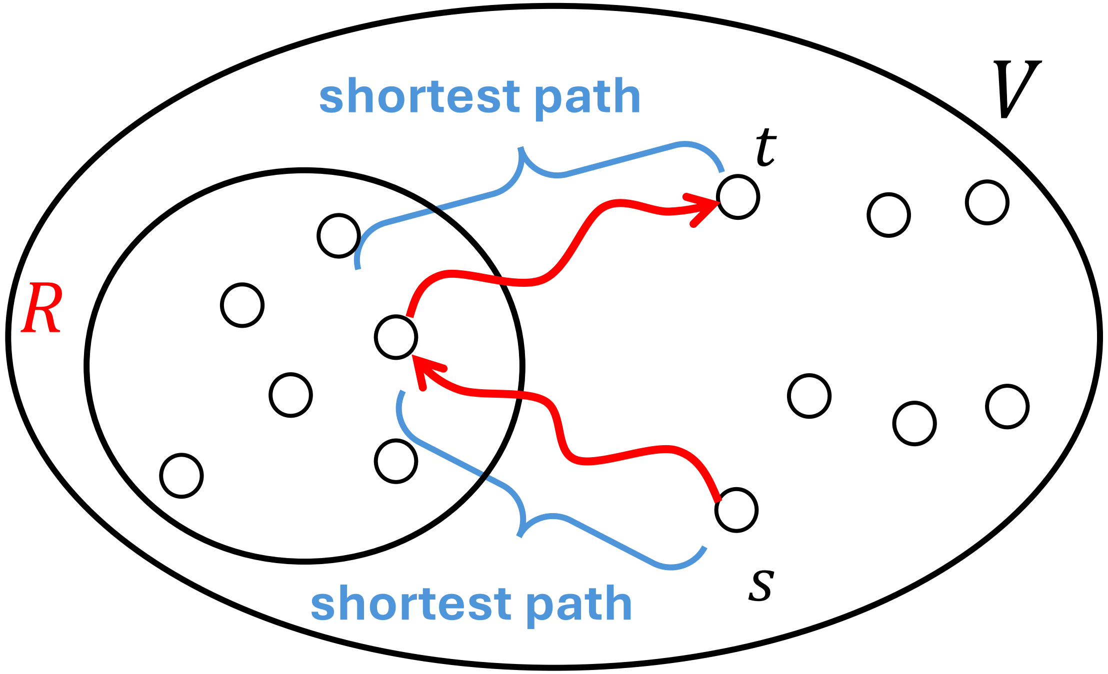


Source-Wise $(R_1 \cup \cdots \cup R_k) \times V$ spanner $\Rightarrow$ Group Steiner Spanner

Upper bounds for classical source-wise carry over to GS-spanners

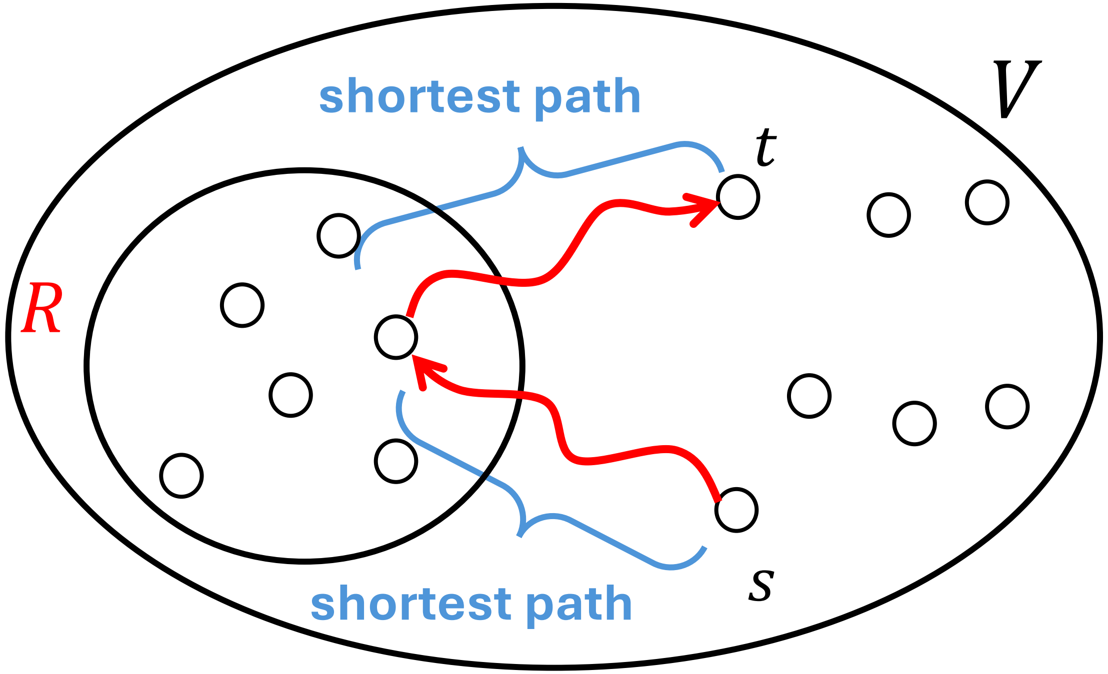
GS-Spanners vs Source-Wise Spanner

$R_1 = \cdots = R_k = R$



GS-Spanners vs Source-Wise Spanner

$R_1 = \cdots = R_k = R$



Source-Wise $R \times V$ spanner $\Leftarrow$ Group Steiner Spanner

We can't obtain better size-stretch trade-offs then the ones for source-wise

(b) All-pairs, general group sizes.

Stretch	Size	Building time	Reference	D.O. query time
$2\alpha+1$	$kn + \lvert \bigcup_i (R_i \times R_i\ \alpha\text{-spanner}) \rvert$	polynomial	Theorem 13	$O(2^k k \cdot \lvert R\rvert^2 + \lvert R\rvert^3)$
$2\alpha+1$	$n + \lvert R \times R\ \alpha\text{-spanner}\rvert$	polynomial	Theorem 14	$O(2^k k \cdot \lvert R\rvert^2 + \lvert R\rvert^3)$

(b) All-pairs, general group sizes.

Stretch	Size	Building time	Reference	D.O. query time
$2\alpha+1$	$kn + \lvert \bigcup_i (R_i \times R_i\ \alpha\text{-spanner}) \rvert$	polynomial	Theorem 13	$O(2^k k \cdot \lvert R\rvert^2 + \lvert R\rvert^3)$
$2\alpha+1$	$n + \lvert R \times R\ \alpha\text{-spanner}\rvert$	polynomial	Theorem 14	$O(2^k k \cdot \lvert R\rvert^2 + \lvert R\rvert^3)$

All-pairs Weighted

α	Size	Reference
⚂ 3	$O(n^{4/3}\lvert R\rvert^{1/3})$	Ob. 2 + [1]
3	$O(kn + kR_{\max}^2 n^{1/2})$	Th. 13 + [13]
3	$O(n + \lvert R\rvert^2 n^{1/2})$	Ob. 2 + [17]
3	$O(n^{3/2})$	Ob. 2 + [3]
⚂ 5	$O(n^{9/7}\lvert R\rvert^{2/7})$	Ob. 2 + [1]
5	$O(n^{4/3})$	Ob. 2 + [3]
✻ ⚂ 7	$O(n\lvert R\rvert^{2/3})$	Th. 14 + [1]
7	$O(n + \lvert R\rvert^{3/2} n^{1/2})$	Ob. 2 + [17]
7	$O(n^{5/4})$	Ob. 2 + [3]
✻ $7+\varepsilon$	$O(n\lvert R\rvert^{1/2}/\varepsilon)$	Th. 14 + [18]
$2h-1$	$O(n^{1+1/h})$	Ob. 2 + [3]
$4h-1$	$O(n + n^{1/2}\lvert R\rvert^{1+1/h})$	Ob. 2 + [17]

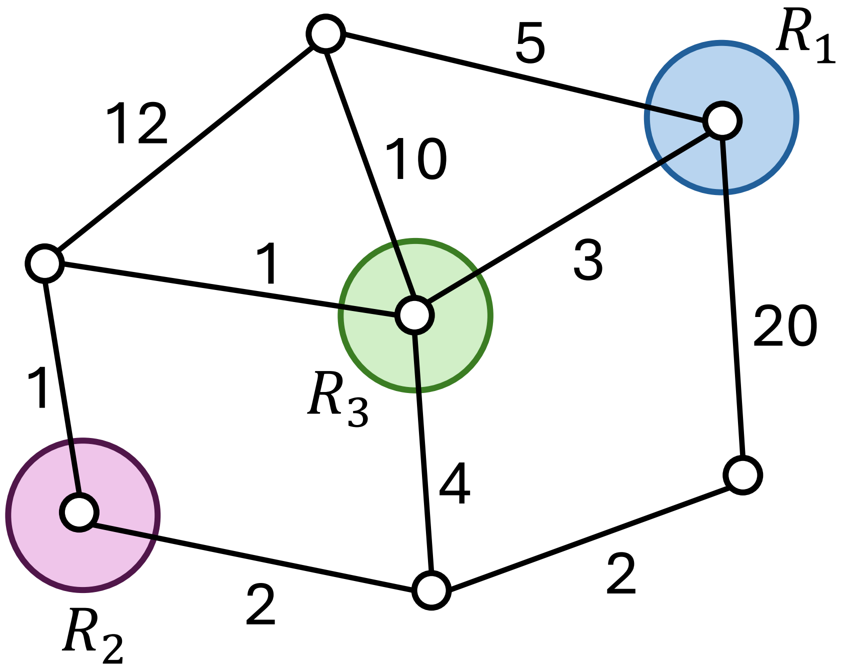
All-pairs Unweighted

α	Size	Reference
✻ 3	$O(n^{2/3}\lvert R\rvert^{4/3} + n\lvert R\rvert^{2/3})$	Th. 14 + [8]
3	$O(kn^{2/3}R_{\max}^{4/3} + knR_{\max}^{2/3})$	Th. 13 + [8]
3	$\widetilde{O}(n^{5/4}\lvert R\rvert^{1/4})$	Ob. 2 + [9]
5	$\widetilde{O}(n^{11/9}\lvert R\rvert^{2/9})$	Ob. 2 + [24]
7	$\widetilde{O}(n^{6/5}\lvert R\rvert^{1/5})$	Ob. 2 + [24]
7	$O(n + \lvert R\rvert^{3/2} n^{1/2})$	Ob. 2 + [17]
✻ 7	$O(n\lvert R\rvert^{1/2})$	Th. 14 + [14]

Singleton case

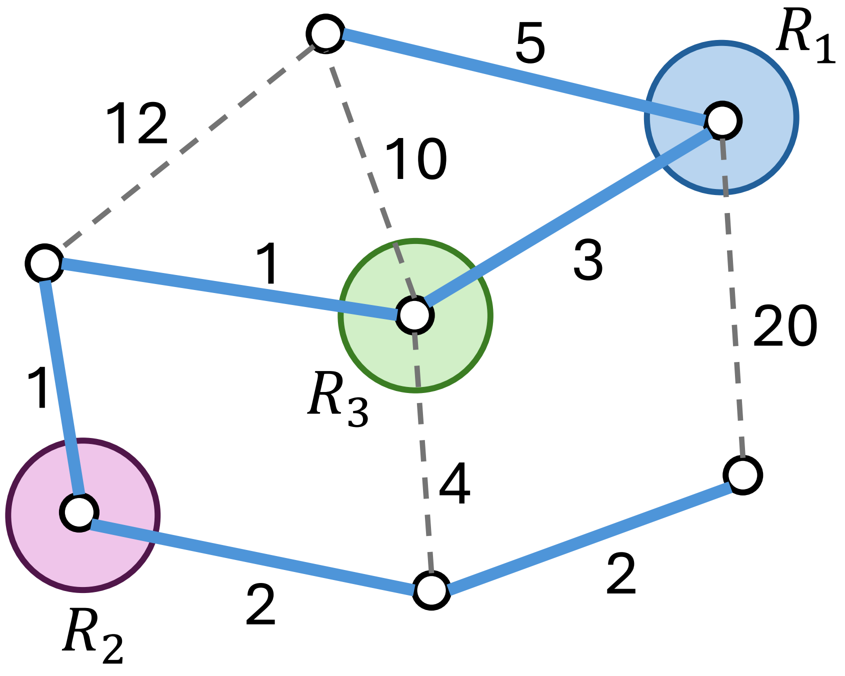
The Singleton case

$$|R_1| = \cdots = |R_k| = 1$$



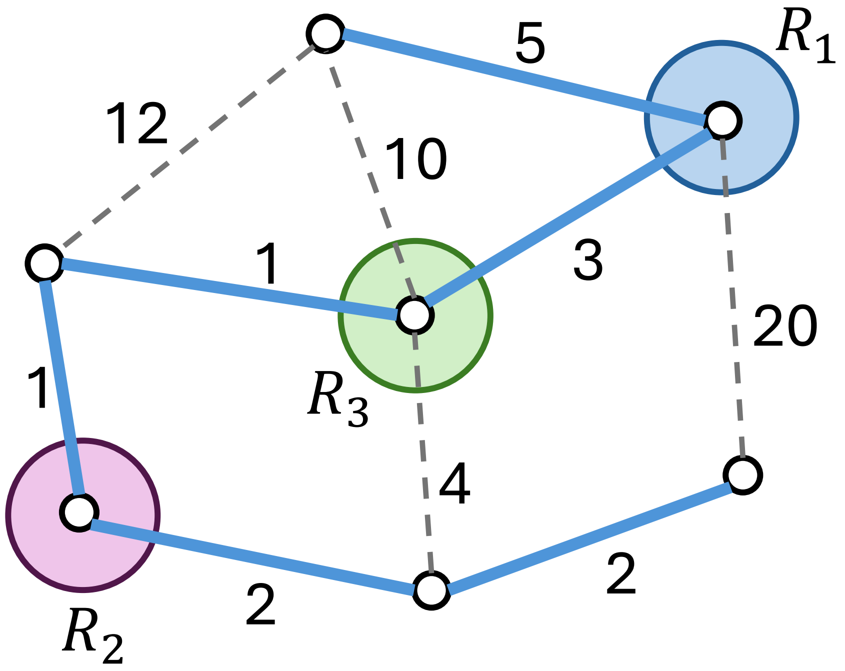
The Singleton case

$$|R_1| = \cdots = |R_k| = 1$$



The Singleton case

$$|R_1| = \cdots = |R_k| = 1$$



Singleton case ≡ minimum-cost metric Hamiltonian path

- NP-hard
- Does not admit poly-time $\frac{185}{184}$—apx, unless P = NP [Karpinski et. al – JCSS '15]
- Admits poly-time $\frac{3}{2}$—apx [Zenklusen – SODA '19]

Singleton case – all pairs

(a) All-pairs, singleton case ($|R_i| = 1\ \forall i = 1, \ldots, k$).

	Stretch	Size	Building time	Reference	D.O. query time
☆	1	$O(kn)$	polynomial	Theorem 7	$O(2^k \cdot k^3)$
	$1+\varepsilon$	$O(n/\varepsilon^2)$	FPT	Theorem 8	$O(1/\varepsilon^2)$
	$\gamma+\varepsilon$	$O(n/\varepsilon^2)$	polynomial	Theorem 9	$O(1/\varepsilon^2)$
☆	2	$n-1$	polynomial	Theorem 11	$O(1)$

Singleton case – all pairs

(a) All-pairs, singleton case ($|R_i| = 1\ \forall i = 1, \ldots, k$).

	Stretch	Size	Building time	Reference	D.O. query time
☆	1	$O(kn)$	polynomial	Theorem 7	$O(2^k \cdot k^3)$
	$1+\varepsilon$	$O(n/\varepsilon^2)$	FPT	Theorem 8	$O(1/\varepsilon^2)$
	$\gamma+\varepsilon$	$O(n/\varepsilon^2)$	polynomial	Theorem 9	$O(1/\varepsilon^2)$
☆	2	$n-1$	polynomial	Theorem 11	$O(1)$

Tight bound: stretch 1 $\Rightarrow$ $\Omega(kn)$ edges are required

Singleton case – all pairs

(a) All-pairs, singleton case ($|R_i| = 1\ \forall i = 1, \dots, k$).

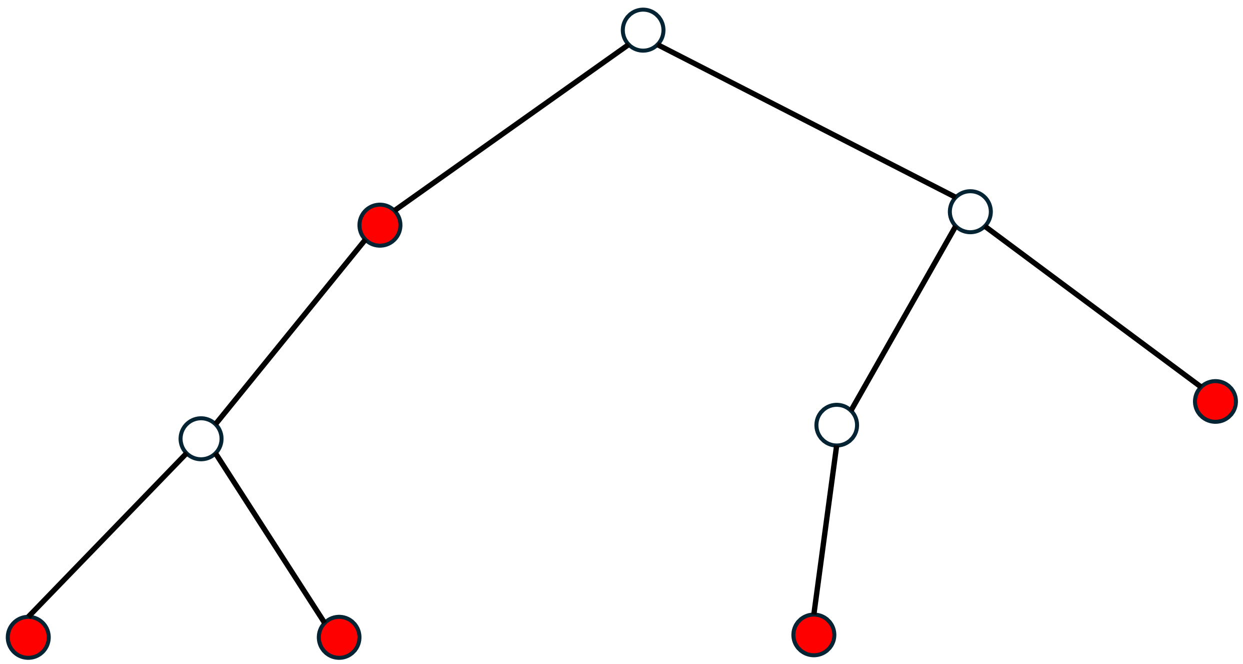
	Stretch	Size	Building time	Reference	D.O. query time
☆	1	$O(kn)$	polynomial	Theorem 7	$O(2^k \cdot k^3)$
	$1+\varepsilon$	$O(n/\varepsilon^2)$	FPT	Theorem 8	$O(1/\varepsilon^2)$
	$\gamma+\varepsilon$	$O(n/\varepsilon^2)$	polynomial	Theorem 9	$O(1/\varepsilon^2)$
☆	2	$n-1$	polynomial	Theorem 11	$O(1)$

$n-1$ edge are sufficient to achieve a stretch 2

Tight bound: stretch $< 2 \Rightarrow n$ edges are required

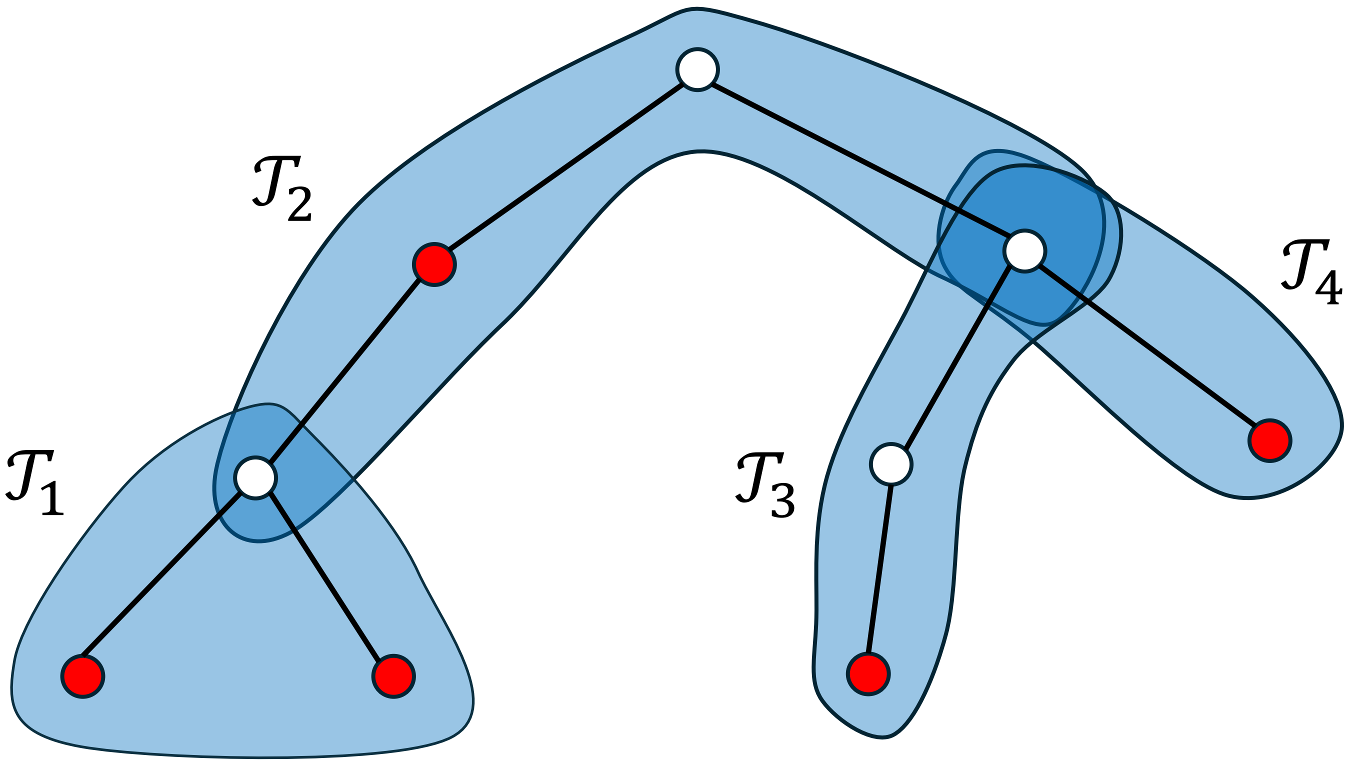
Tight bound: stretch 1 $\Rightarrow \Omega(kn)$ edges are required

$(1+\varepsilon)$-GS spanner – singleton



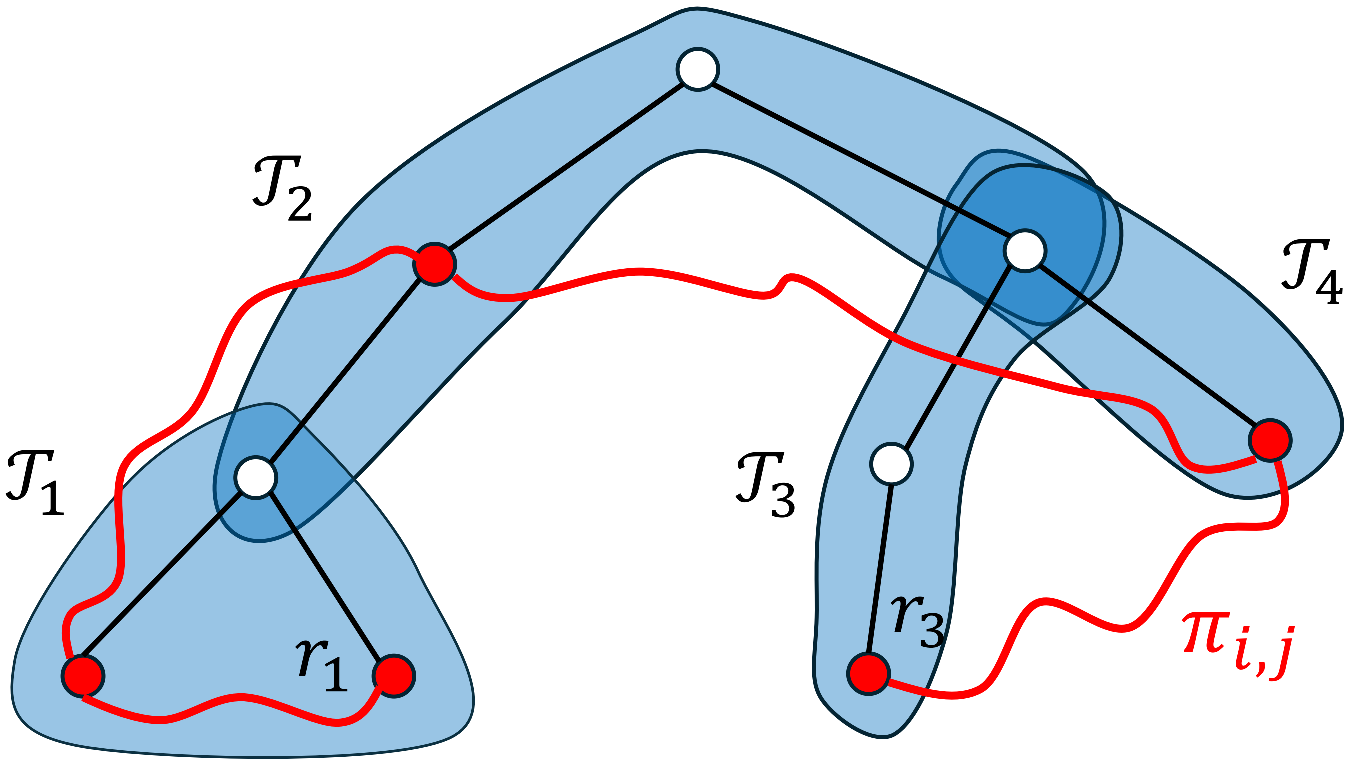
Compute a 2-apx steiner tree T (in *poly-time*)

$(1+\varepsilon)$-GS spanner – singleton



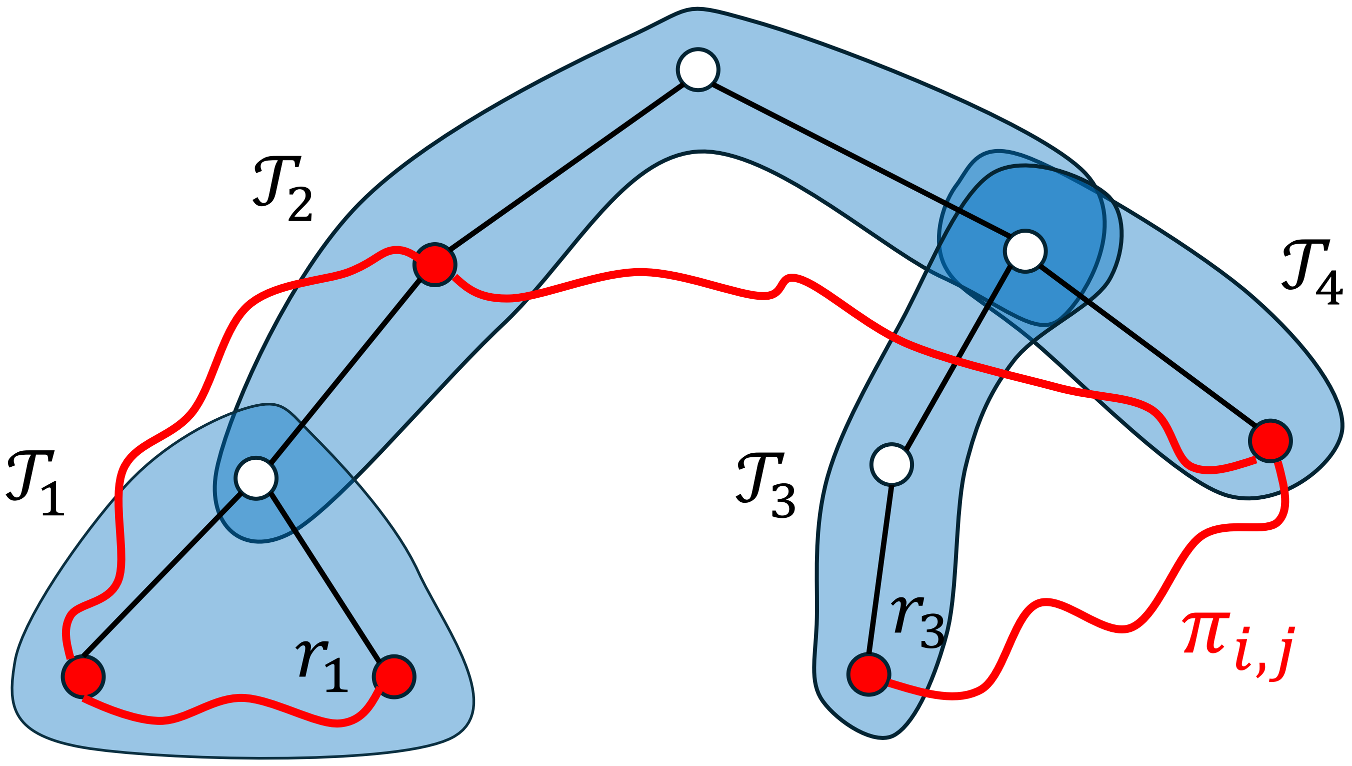
Partition T into $O(1/\varepsilon)$ **micro-trees** $\mathcal{T}_1, \dots, \mathcal{T}_h$,
all of them with weight **at most** $W = \frac{\varepsilon}{4} w(T)$ (in *poly-time*).

$(1+\varepsilon)$-GS spanner – singleton



$\pi_{i,j}$: the **best** shortest GS-path between two required $r_i \in \mathcal{J}_i$ and $r_j \in \mathcal{J}_j$, one for each pair $\mathcal{J}_i, \mathcal{J}_j$ (in *FPT-time*).

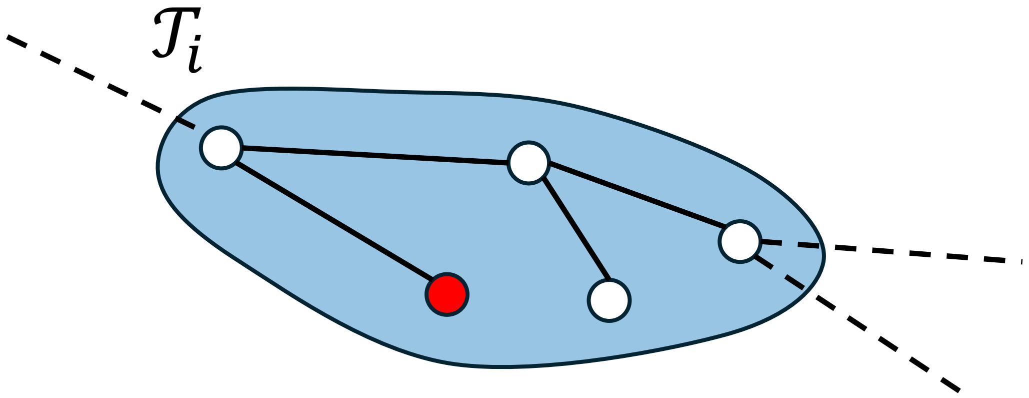
$(1 + \varepsilon)$-GS spanner – singleton



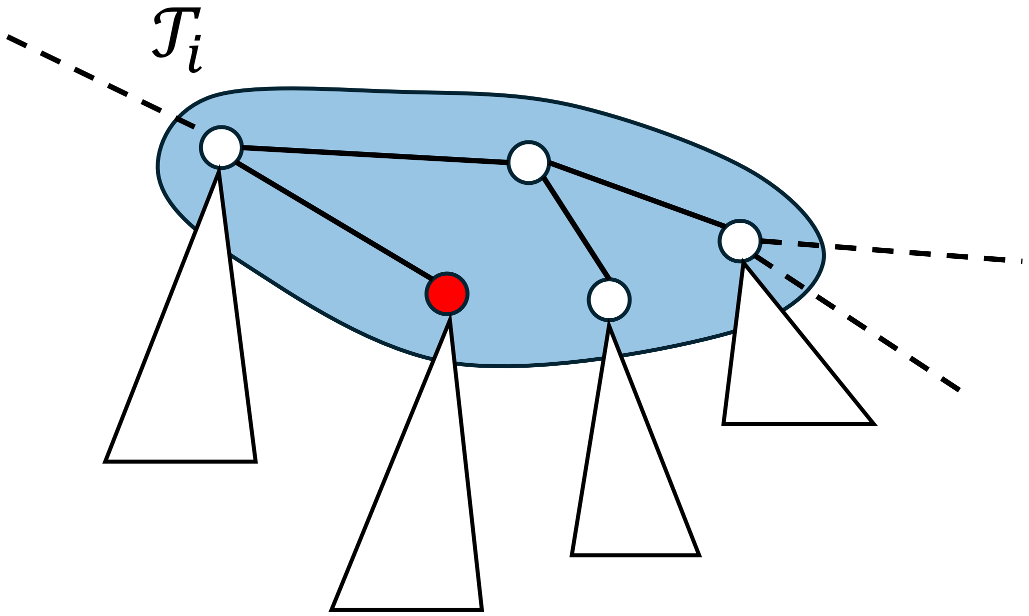
$\pi_{i,j}$: the **best** shortest GS-path between two required $r_i \in \mathcal{T}_i$ and $r_j \in \mathcal{T}_j$, one for each pair $\mathcal{T}_i, \mathcal{T}_j$ (in *FPT-time*).

$O\left(\frac{1}{\varepsilon^2}\right)$ of these GS-pahts $\pi_{i,j}$.

$(1 + \varepsilon)$-GS spanner – singleton



$(1+\varepsilon)$-GS spanner – singleton



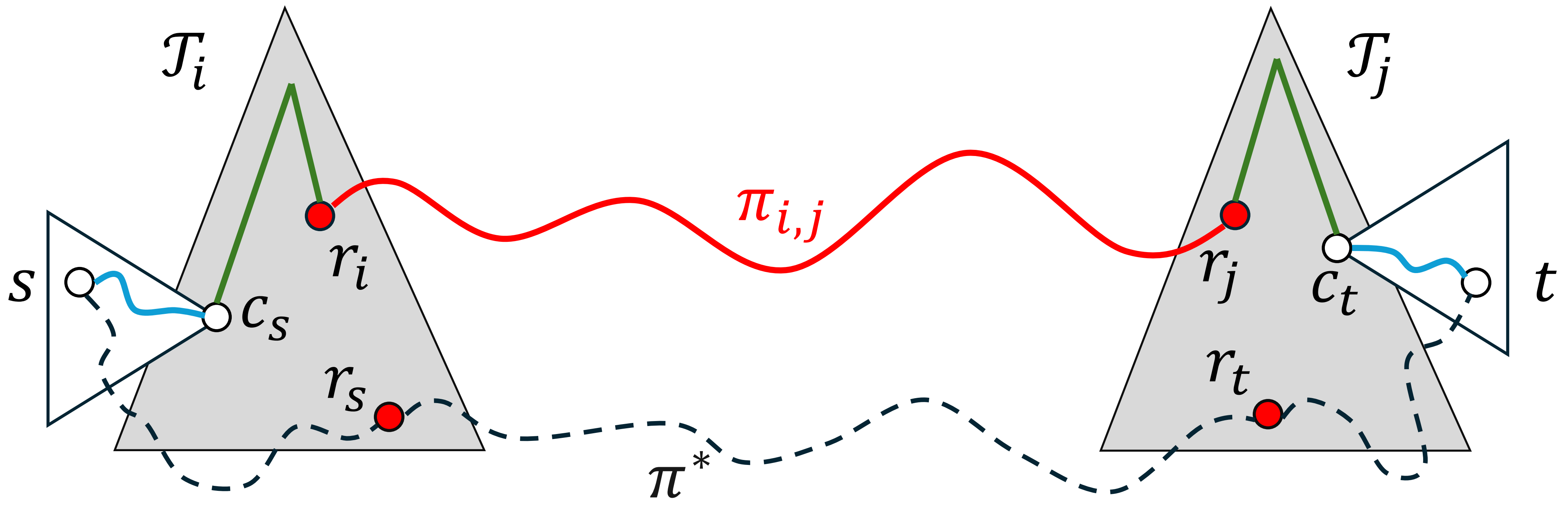
We compute a **spanning forest** F_i consisting of trees rooted at each vertex in $\mathcal{T}_i$, where every path from a node to its root is the shortest.

$(1+\varepsilon)$-GS spanner – singleton

GS-spanner = $\cup_i \mathcal{T}_i + \cup_i F_i + \cup_{i,j} \pi_{i,j}$

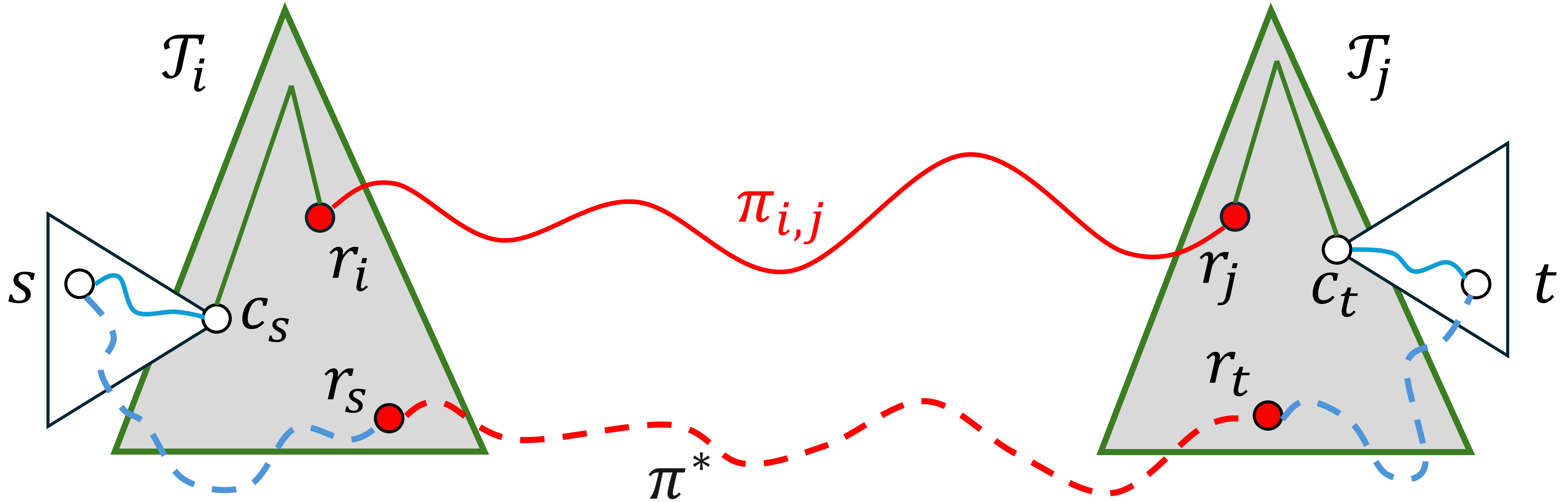
$O\left(\frac{n}{\varepsilon}\right)$, $O\left(\frac{n}{\varepsilon}\right)$, $O\left(\frac{n}{\varepsilon^2}\right)$

$(1+\varepsilon)$-GS spanner – singleton



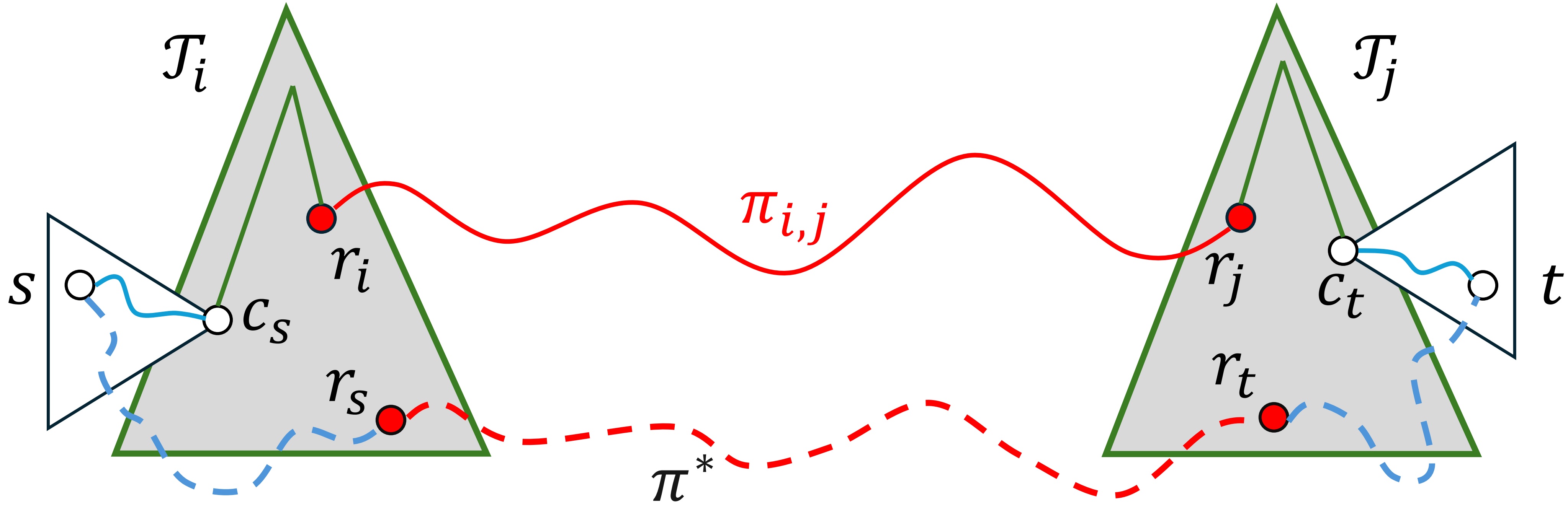
$$\sigma_H(s,t) \leq d(s,c_s) + d_{\mathcal{T}_i}(c_s,r_i) + w(\pi_{i,j}) + d_{\mathcal{T}_j}(r_j,c_t) + d(c_t,t)$$

$(1+\varepsilon)$-GS spanner – singleton



$$\leq w(\pi^*[s : r_s]) + W + w(\pi^*[r_i : r_s]) + W + w(\pi^*[r_t : t])$$

$(1+\varepsilon)$-GS spanner – singleton



$$\leq \sigma_G(s,t) + 2W = \sigma_G(s,t) + \frac{\varepsilon}{2}w(T) = (1+\varepsilon)\sigma_G(s,t)$$

$(\gamma + \varepsilon)$-GS spanner – singleton

Stretch	Building time
$(1 + \varepsilon)$	FPT

⇓

γ = apx ratio of a poly-time algorithm for minimum-cost metric Hamiltonian path

Stretch	Building time
$(\gamma + \varepsilon)$	Polynomial

Open problems

1. Ad-hoc contructions for general group size

Open problems

1. Ad-hoc contructions for general group size

2. Better bounds for single-source preserver

Open problems

1. Ad-hoc contructions for general group size

2. Better bounds for single-source preserver *(eg. we give a preserver of size $O(2^k n)$)*

Open problems

1. Ad-hoc contructions for general group size

2. Better bounds for single-source preserver *(eg. we give a preserver of size $O(2^k n)$)*

3. Ordered version

Open problems

1. Ad-hoc contructions for general group size

2. Better bounds for single-source preserver *(eg. we give a preserver of size $O(2^k n)$)*

3. Ordered version

Thanks !