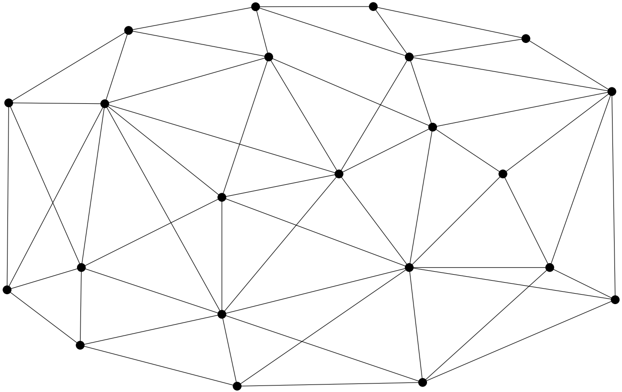
Distributed Construction of Light Networks

Michael Elkin[1] **Arnold Filtser**[2] Ofer Neiman[1]

[1] Ben-Gurion University of the Negev
[2] Columbia University

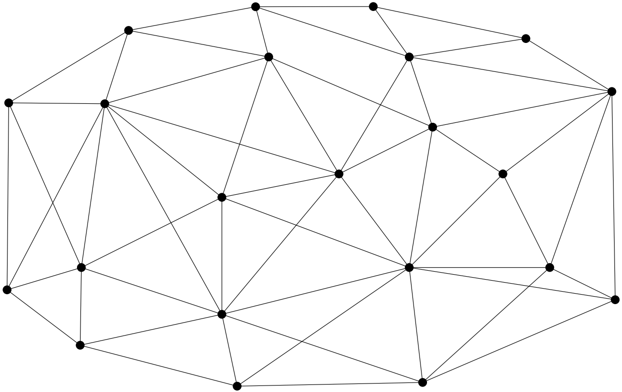
PODC 2020 online presentation

CONGEST model of distributed computation



- The network is represented by a **graph**
- **Synchronized** rounds
- $O(\log n)$ bits messages

CONGEST model of distributed computation

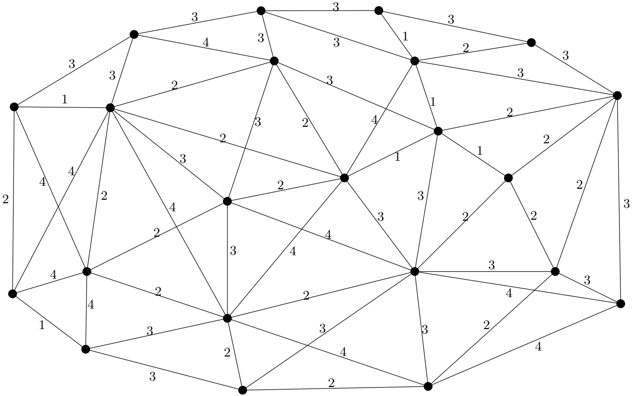


- The network is represented by a **graph**
- **Synchronized** rounds
- $O(\log n)$ bits messages

$\boldsymbol{D}$: **Hop** diameter

$\boldsymbol{n}$: number of vertices

CONGEST model of distributed computation

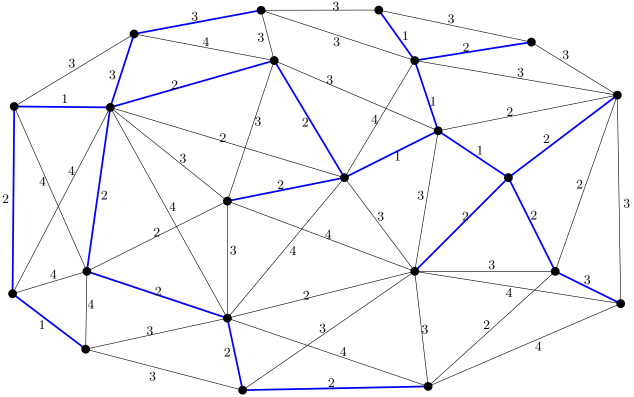


- The network is represented by a **graph**
- **Synchronized** rounds
- $O(\log n)$ bits messages

D: **Hop** diameter

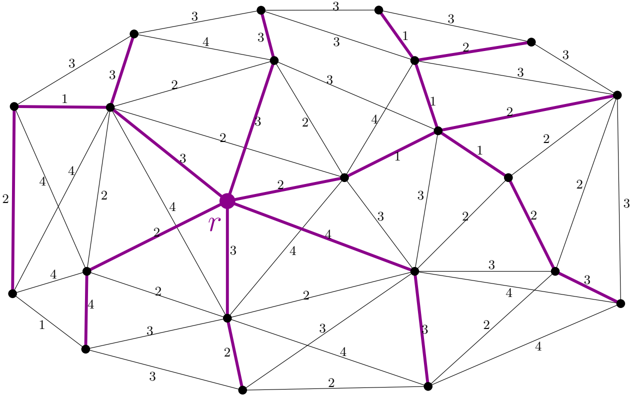
n: number of vertices

CONGEST model of distributed computation



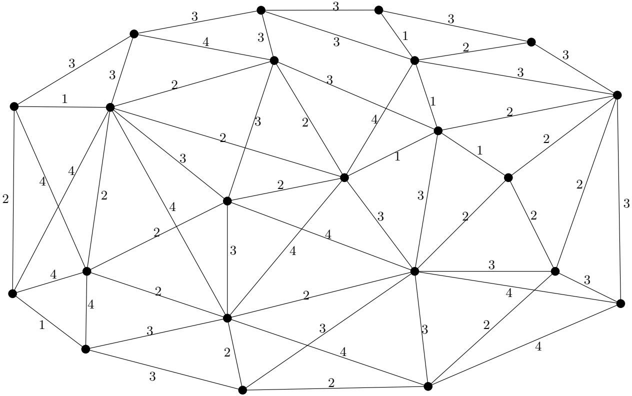
MST: minimum spanning tree

CONGEST model of distributed computation



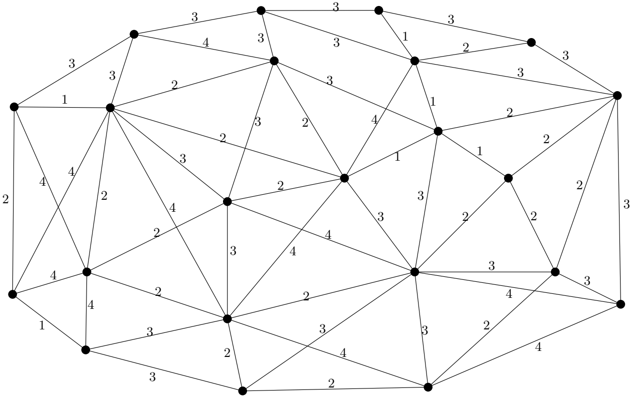
SPT: shortest path tree

CONGEST model of distributed computation



Tension between the **shortest path metric** and the **hop-metric**

CONGEST model of distributed computation

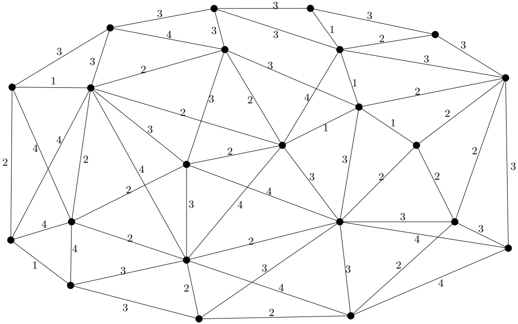


Theorem ([Sarma, Holzer, Kor, Korman, Nanongkai, Pandurangan, Peleg, Wattenhofer 2012])

For every parameter $\alpha \geq 1$,

every α*-approximation algorithm for* **MST** *takes* $\tilde{\Omega}(\sqrt{n} + D)$ **rounds**.

CONGEST model of distributed computation

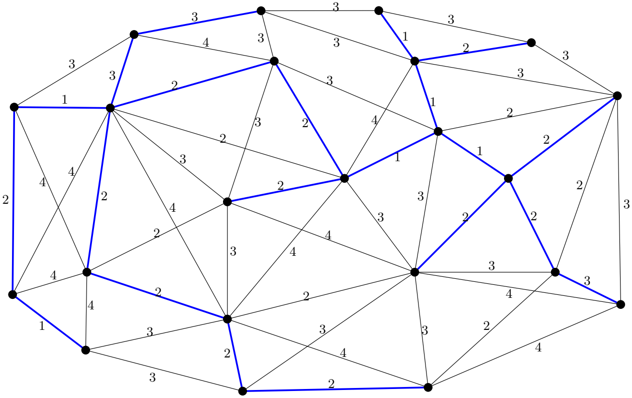


Theorem ([SHKKNPPW12])

For every parameter $\alpha \geq 1$,
every α-approximation algorithm for **MST** *takes $\tilde{\Omega}(\sqrt{n} + D)$* **rounds**.

Also holds for approximating shortest s-t path, shortest path tree, and minimum cut

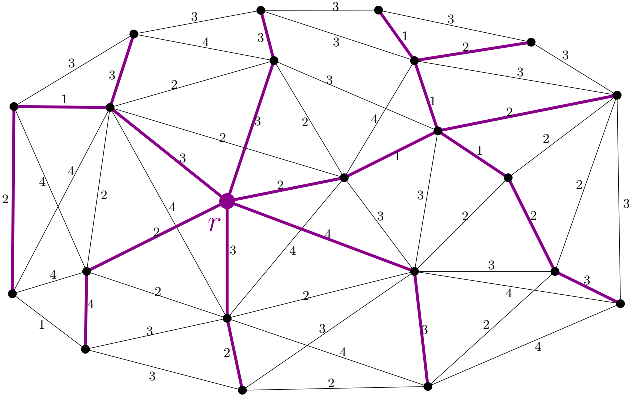
CONGEST model of distributed computation



Theorem (Elkin 2017])

MST construction in $\tilde{O}(\sqrt{n} + D)$ rounds.

CONGEST model of distributed computation



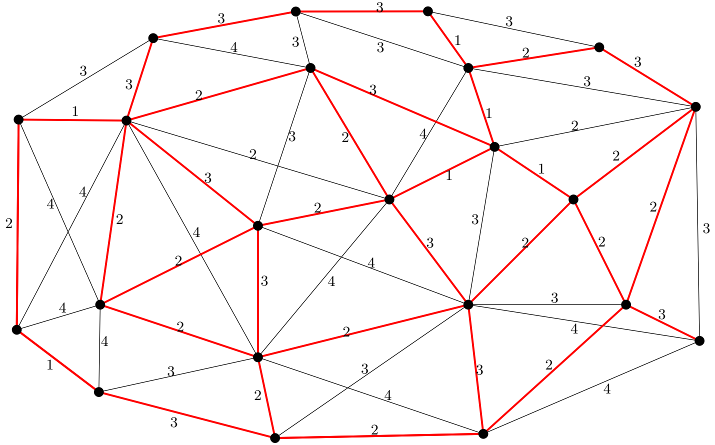
Theorem (Becker, Karrenbauer, Krinninger, Lenzen 2017])

Approximate SPT *in* $\tilde{O}_\epsilon(\sqrt{n} + D)$ *rounds.*
root rt, $\epsilon > 0$, *a spanning tree* T *s.t. for all* $v \in V$, $d_T(\mathrm{rt}, \mathrm{v}) \leq (1 + \epsilon)\mathrm{d_G}(\mathrm{rt}, \mathrm{v})$.

Graph Spanners

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

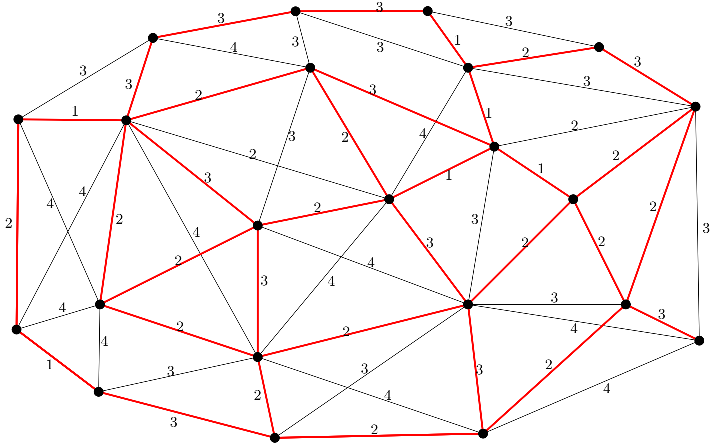
$$\forall u, v \in V, \quad d_H(u, v) \leq t \cdot d_G(u, v)$$



Graph Spanners

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

$$\forall u, v \in V, \quad d_H(u,v) \le t \cdot d_G(u,v)$$



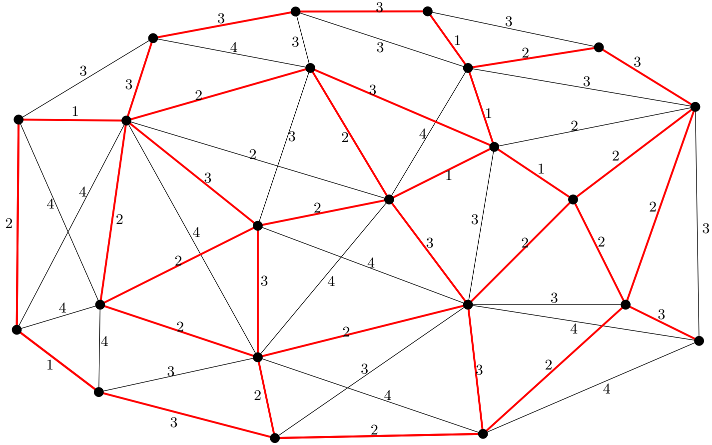
Stretch t

Sparsity $|H|$

Graph Spanners

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

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



Stretch t

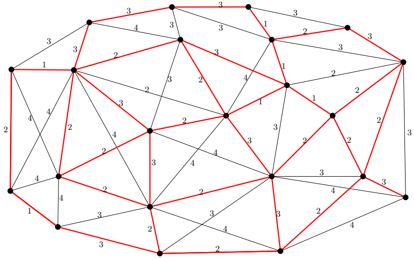
Sparsity $|H|$

Lightness $\dfrac{\sum_{e \in E_H} w(e)}{w(\mathrm{MST})}$

Graph Spanners

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

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



Stretch	t
Sparsity	$\lvert H \rvert$
Lightness	$\frac{\sum_{e \in E_H} w(e)}{w(\mathrm{MST})}$

For $k \geq 1$

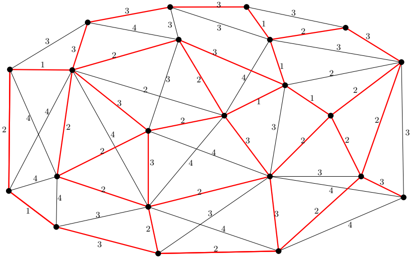
- [Althofer, Das, Dobkin, Joseph, Soares 1993]:

  $2k - 1$ spanner with $O(n^{1+\frac{1}{k}})$ edges.

Graph Spanners

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

$$\forall u, v \in V, \quad d_H(u, v) \le t \cdot d_G(u, v)$$



Stretch t

Sparsity $|H|$

Lightness $\frac{\sum_{e \in E_H} w(e)}{w(\mathrm{MST})}$

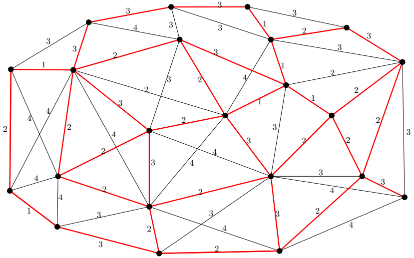
For $k \ge 1$,

- [ADDJS93]: $2k - 1$ spanner with $O(n^{1+\frac{1}{k}})$ edges.
- [Chechik, Wulff-Nilsen 2018]:
  $(2k-1)(1+\epsilon)$ spanner with $O(n^{1+\frac{1}{k}})$ edges and **lightness** $O_\epsilon(n^{1/k})$.

Graph Spanners

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

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



Stretch	t
Sparsity	$\lvert H \rvert$
Lightness	$\frac{\sum_{e \in E_H} w(e)}{w(\mathrm{MST})}$

For $k \geq 1$ and $\epsilon \in (0, 1)$,

- [ADDJS93]: $2k - 1$ spanner with $O(n^{1+\frac{1}{k}})$ edges.
- [CW18]: $(2k - 1)(1 + \epsilon)$ spanner of $O_\epsilon(n^{1/k})$ **lightness** and $O(n^{1+\frac{1}{k}})$ edges.
- [Baswana, Sen 2007]:

  $2k - 1$ spanner with $O(k \cdot n^{1+\frac{1}{k}})$ edges in $O(k)$ CONGEST **rounds**.

Graph Spanners

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

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

For $k \geq 1$ and $\epsilon \in (0, 1)$,

- [ADDJS93]: $2k - 1$ spanner with $O(n^{1+\frac{1}{k}})$ edges.
- [CW18]: $(2k - 1)(1 + \epsilon)$ spanner of $O_\epsilon(n^{1/k})$ **lightness** and $O(n^{1+\frac{1}{k}})$ edges.
- [BS07]: $2k - 1$ spanner with $O(k \cdot n^{1+\frac{1}{k}})$ edges in $O(k)$ CONGEST rounds.

Theorem ([Elkin, Filtser, Neiman 2020] (This paper))

$(2k - 1)(1 + \epsilon)$-**spanner** *of* $O_\epsilon(k \cdot n^{1/k})$ **lightness** *and* $O_\epsilon(k \cdot n^{1+\frac{1}{k}})$ *edges in* $\tilde{O}_\epsilon(n^{\frac{1}{2}+\frac{1}{4k+2}} + D)$ *CONGEST* **rounds**.

Graph Spanners

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

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

For $k \geq 1$ and $\epsilon \in (0, 1)$,

- [ADDJS93]: $2k - 1$ spanner with $O(n^{1+\frac{1}{k}})$ edges.
- [CW18]: $(2k - 1)(1 + \epsilon)$ spanner of $O_\epsilon(n^{1/k})$ **lightness** and $O(n^{1+\frac{1}{k}})$ edges.
- [BS07]: $2k - 1$ spanner with $O(k \cdot n^{1+\frac{1}{k}})$ edges in $O(k)$ CONGEST rounds.

Theorem ([EFN20] (This paper))

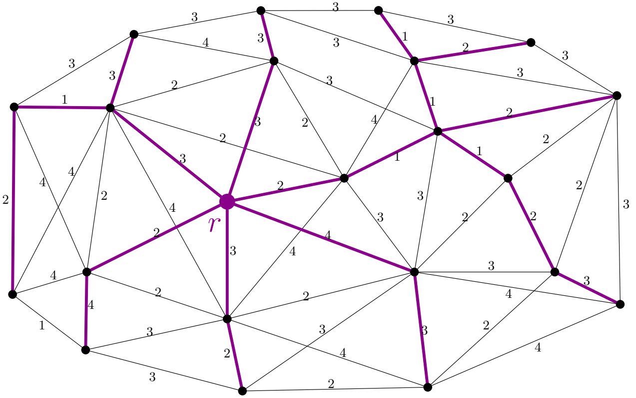
$(2k - 1)(1 + \epsilon)$-**spanner** *of* $O_\epsilon(k \cdot n^{1/k})$ **lightness** *and* $O_\epsilon(k \cdot n^{1+\frac{1}{k}})$ *edges in* $\tilde{O}_\epsilon(n^{\frac{1}{2}+\frac{1}{4k+2}} + D)$ *CONGEST* **rounds**.

[SHKKNPPW12]: $\tilde{\Omega}(\sqrt{n} + D)$ lower bound.

Shallow Light Tree

Fix root $r \in V$.

SPT: tree T s.t. $\forall v \in V$, $d_T(r, v) = d_G(r, v)$.

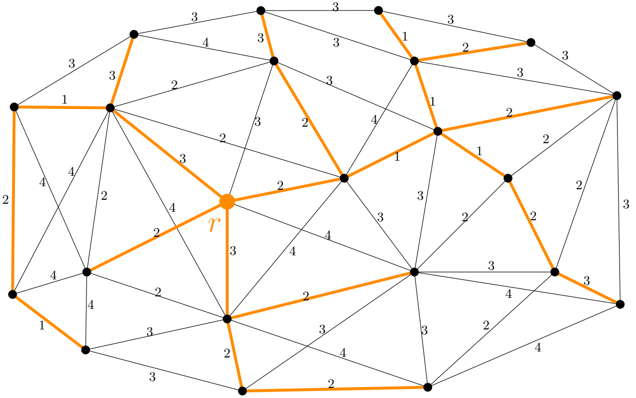


Shallow Light Tree

Fix root $r \in V$.

SPT: tree T s.t. $\forall v \in V$, $d_T(r, v) = d_G(r, v)$.

(α, β)-**SLT**: tree T of lightness β s.t. $\forall v \in V$, $d_T(r, v) \leq \alpha \cdot d_G(r, v)$.

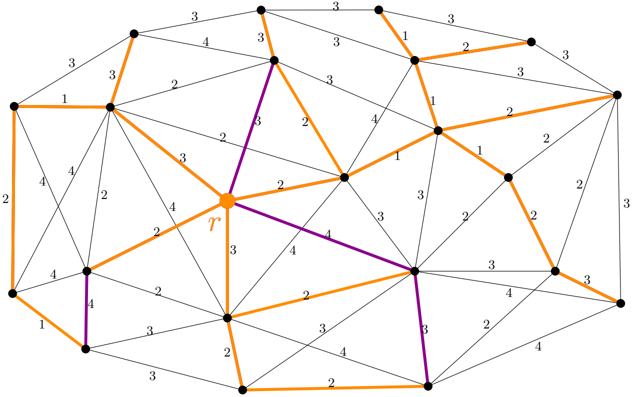


Shallow Light Tree

Fix root $r \in V$.

SPT: tree T s.t. $\forall v \in V$, $d_T(r,v) = d_G(r,v)$.

(α,β)-**SLT**: tree T of lightness β s.t. $\forall v \in V$, $d_T(r,v) \leq \alpha \cdot d_G(r,v)$.

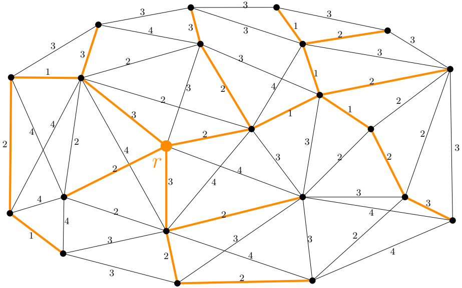


Shallow Light Tree

Fix root $r \in V$.

SPT: tree T s.t. $\forall v \in V$, $d_T(r, v) = d_G(r, v)$.

(α, β)-**SLT**: tree T of lightness β s.t. $\forall v \in V$, $d_T(r, v) \le \alpha \cdot d_G(r, v)$.



Theorem ([Khuller, Raghavachari, Young 1995])

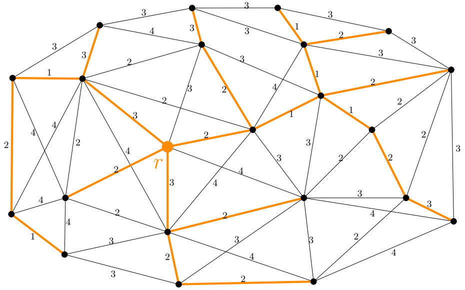
For every $\epsilon > 0$ *exist* $(1 + \epsilon, 1 + \frac{2}{\epsilon})$-**SLT**.

Shallow Light Tree

Fix root $r \in V$.

SPT: tree T s.t. $\forall v \in V$, $d_T(r, v) = d_G(r, v)$.

(α, β)-**SLT**: tree T of lightness β s.t. $\forall v \in V$, $d_T(r, v) \leq \alpha \cdot d_G(r, v)$.



Theorem ([KRY95])

*For every $\epsilon > 0$ exist $(1 + \epsilon, 1 + \frac{2}{\epsilon})$-***SLT***.* *(tight)*

Shallow Light Tree

Fix root $r \in V$.

SPT: tree T s.t. $\forall v \in V$, $d_T(r,v) = d_G(r,v)$.

(α,β)-**SLT**: tree T of lightness β s.t. $\forall v \in V$, $d_T(r,v) \leq \alpha \cdot d_G(r,v)$.

Theorem ([KRY95])

*For every $\epsilon > 0$ exist $(1+\epsilon, 1+\frac{2}{\epsilon})$-**SLT**.* *(tight)*

Theorem ([EFN20] (This paper))

*For every $\epsilon > 0$ algorithm that constructs $(1+\epsilon, 1+\frac{O(1)}{\epsilon})$-**SLT** in $\tilde{O}_\epsilon(\sqrt{n}+D)$ rounds*

Shallow Light Tree

Fix root $r \in V$.

SPT: tree T s.t. $\forall v \in V$, $d_T(r,v) = d_G(r,v)$.

(α, β)-**SLT**: tree T of lightness β s.t. $\forall v \in V$, $d_T(r,v) \leq \alpha \cdot d_G(r,v)$.

Theorem ([KRY95])

*For every $\epsilon > 0$ exist $(1+\epsilon, 1+\frac{2}{\epsilon})$-**SLT**.* *(tight)*

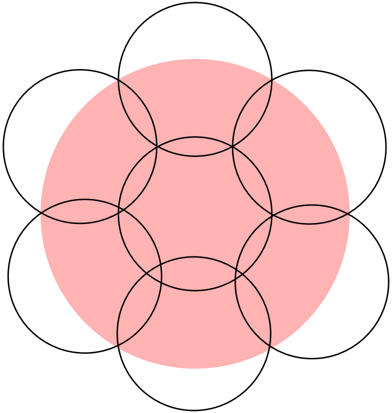
Theorem ([EFN20] (This paper))

*For every $\epsilon > 0$ algorithm that constructs $(1+\epsilon, 1+\frac{O(1)}{\epsilon})$-**SLT** in $\tilde{O}_\epsilon(\sqrt{n}+D)$ rounds*

[SHKKNPPW12]: $\tilde{\Omega}(\sqrt{n}+D)$ lower bound.

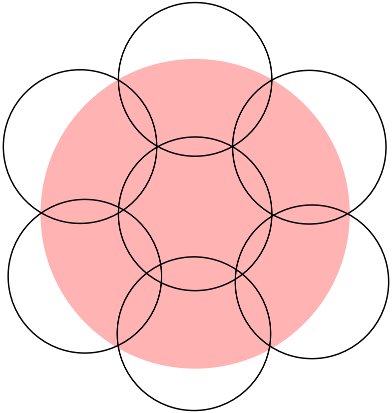
Doubling Metrics

Metric space has **doubling dimension *d*** if
every radius r ball can be **covered** by $2^{\boldsymbol{d}}$ balls of radius $\frac{r}{2}$.



Doubling Metrics

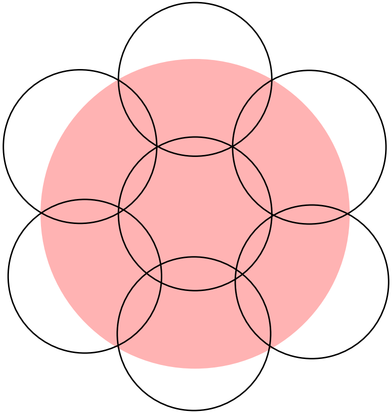
Metric space has **doubling dimension $\boldsymbol{d}$** if
every radius r ball can be **covered** by $2^{\boldsymbol{d}}$ balls of radius $\frac{r}{2}$.



Example: Every d-dimensional Euclidean space has doubling dimension $\boldsymbol{O(d)}$.

Doubling Metrics

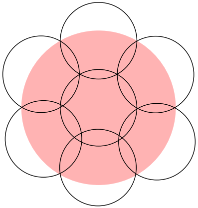
Metric space has **doubling dimension $\boldsymbol{d}$** if
every radius r ball can be **covered** by $2^{\boldsymbol{d}}$ balls of radius $\frac{r}{2}$.



$G = (V, E, w)$ has doubling dim. $\boldsymbol{d} \iff (V, d_G)$ has doubling dim. $\boldsymbol{d}$.

Doubling Metrics

Metric space has **doubling dimension** $\boldsymbol{d}$ if
every radius r ball can be **covered** by $2^{\boldsymbol{d}}$ balls of radius $\frac{r}{2}$.

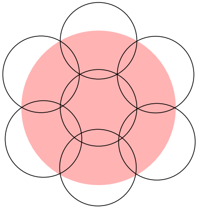


Theorem ([Borradaile, Le, Wulff-Nilsen 2019] improving ([Got15,FS20]))

$1+\epsilon$-**spanner** *of* $\epsilon^{-O(\boldsymbol{d})}$ **lightness** *and* $\epsilon^{-O(\boldsymbol{d})} \cdot n$ *edges.*

Doubling Metrics

Metric space has **doubling dimension $\boldsymbol{d}$** if
every radius r ball can be **covered** by $2^{\boldsymbol{d}}$ balls of radius $\frac{r}{2}$.



Theorem ([BLW19])

$1+\epsilon$-**spanner** *of* $\epsilon^{-O(\boldsymbol{d})}$ **lightness** *and* $\epsilon^{-O(\boldsymbol{d})} \cdot n$ *edges.*

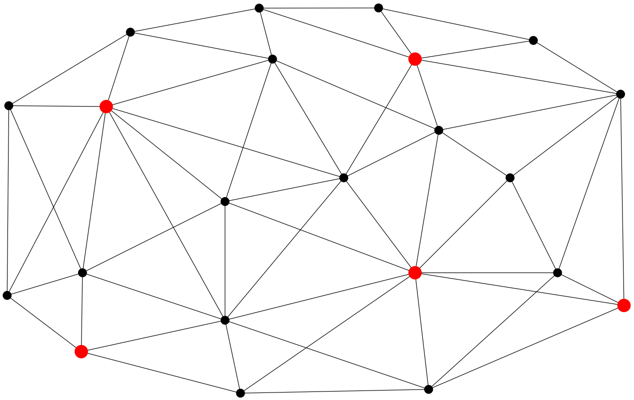
Theorem ([EFN20] (This paper))

$1+\epsilon$-**spanner** *of* $\epsilon^{-O(\boldsymbol{d})} \cdot \log n$ **lightness** *and* $\epsilon^{-O(\boldsymbol{d})} \cdot n \log n$ *edges*
in $\tilde{O}_{\epsilon}(\sqrt{n}+D) \cdot n^{o(1)}$ *CONGEST* **rounds**.

Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

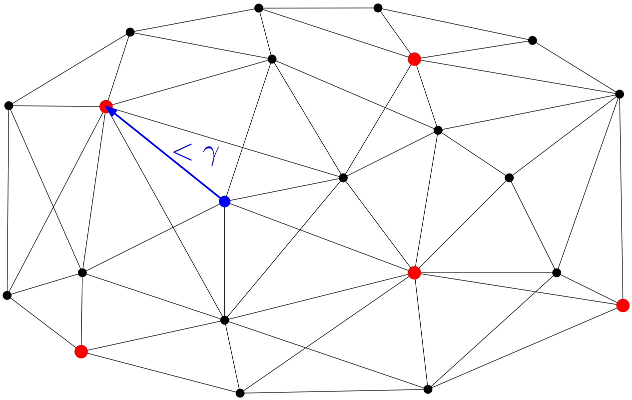
- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.



Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

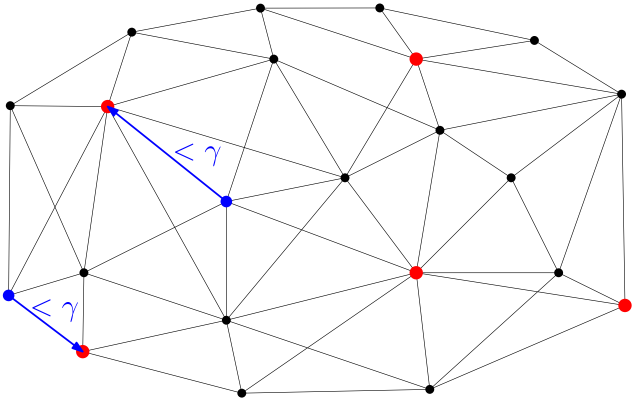
- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.



Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

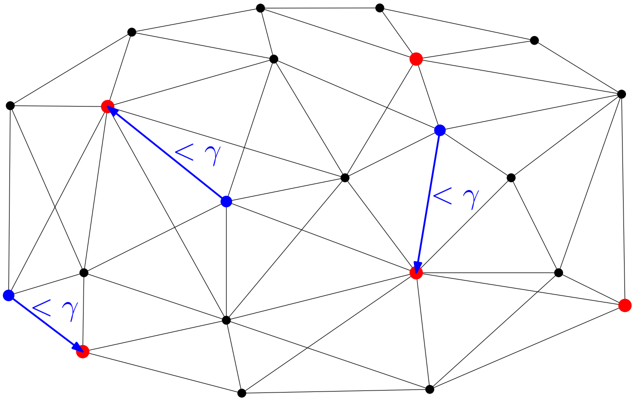
- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.



Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

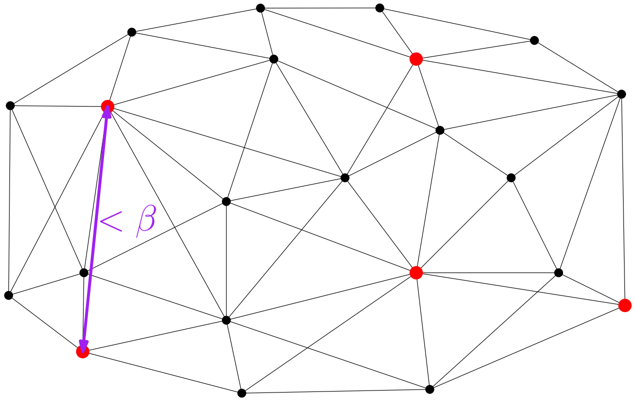
- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.



Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

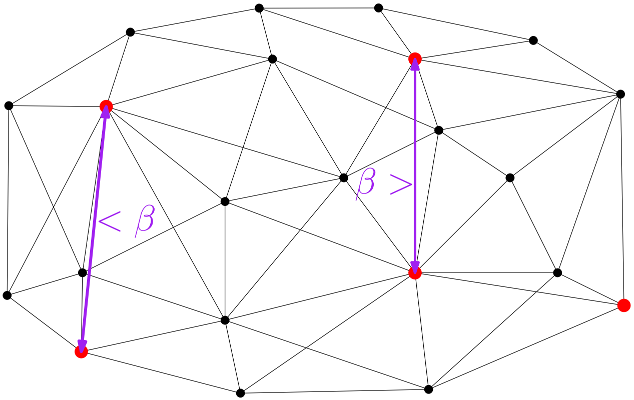
- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.



Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

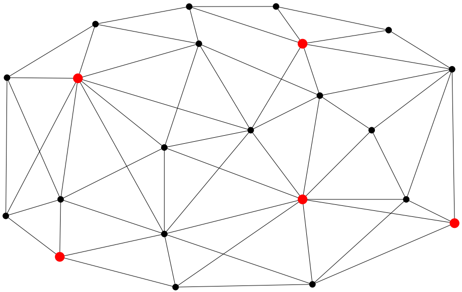
- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.



Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.

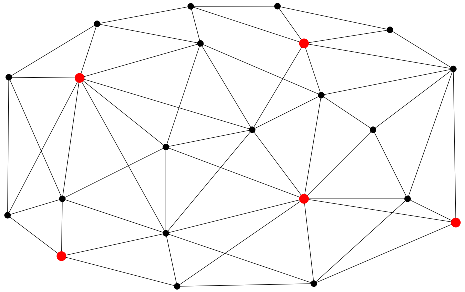


***k*-ruling set**: when G is unweighted and $\gamma = \beta = k$

Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.



***k*-ruling set**: when G is unweighted and $\gamma = \beta = k$

1-ruling set: **MIS** (maximal independent set)

Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.

***k*-ruling set**: when G is unweighted and $\gamma = \beta = k$

1-ruling set: **MIS** (maximal independent set)

Theorem ([Luby 1986])

(k, k)-ruling set in $O(k \log n)$ CONGEST rounds.

Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.

***k*-ruling set**: when G is unweighted and $\gamma = \beta = k$

1-ruling set: **MIS** (maximal independent set)

Theorem ([Lub86])

(k, k)-ruling set in $O(k \log n)$ CONGEST rounds.

Theorem ([EFN20] (This paper))

For any fixed $0 < \beta < \gamma < 2\beta$ algorithm for (γ, β)-net in $\tilde{O}_\epsilon(\sqrt{n} + D) \cdot n^{o(1)}$ CONGEST **rounds**.

Definition ((γ, β)-net)

Set $N \subseteq V$ is an (γ, β)-net if:

- **γ-covering:** $\forall v \in V$ there is a net point $u \in N$ s.t. $d_G(u, v) \leq \gamma$.
- **β-separating:** $\forall u, v \in N$, $d_G(u, v) > \beta$.

Theorem ([Lub86])

(k, k)-ruling set in $O(k \log n)$ CONGEST rounds.

Theorem ([EFN20] (This paper))

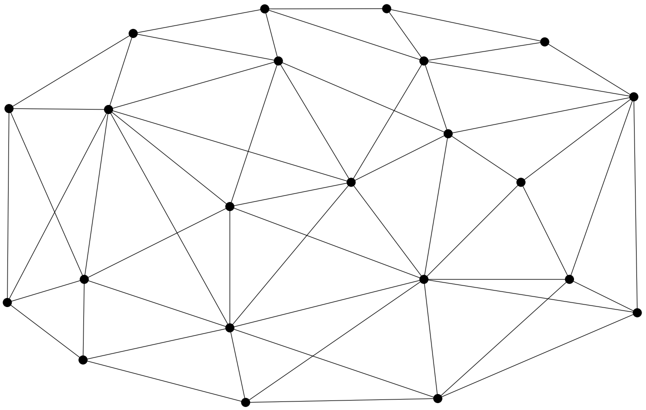
For any fixed $0 < \beta < \gamma < 2\beta$ algorithm for (γ, β)-net in $\tilde{O}_\epsilon(\sqrt{n} + D) \cdot n^{o(1)}$ CONGEST **rounds**.

Theorem ([EFN20] (This paper))

For $\alpha \leq \mathrm{poly(n)}$, every algorithm for $(\alpha \cdot \Delta, \Delta)$-net takes $\tilde{\Omega}(\sqrt{n} + D)$ CONGEST **rounds**.

Algorithm

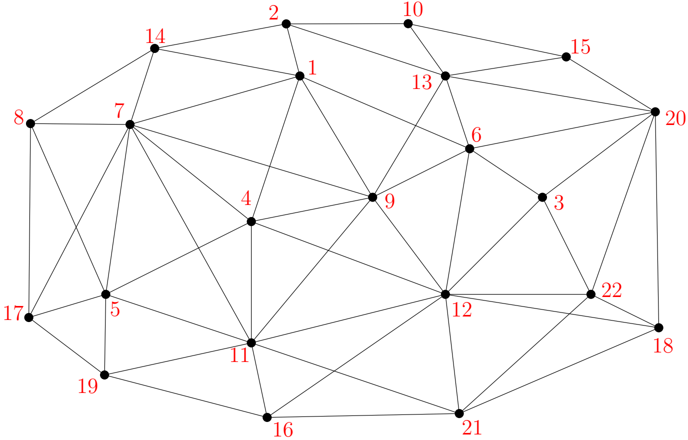
$A \leftarrow V,\ N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
    sample permutation π over A;
    for $v \in A$ **do**
        if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
            $N \leftarrow N \cup \{v\}$
    $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



Inspired by MIS algorithms of and [Métivier, Robson, Saheb-Djahromi, Zemmari 2011] and [Luby 1986].

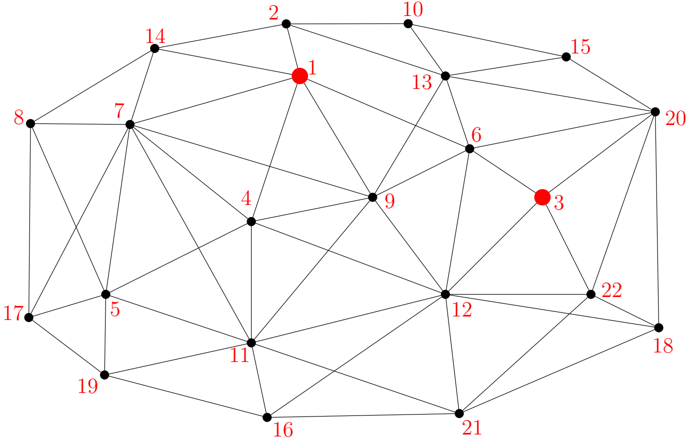
Algorithm

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



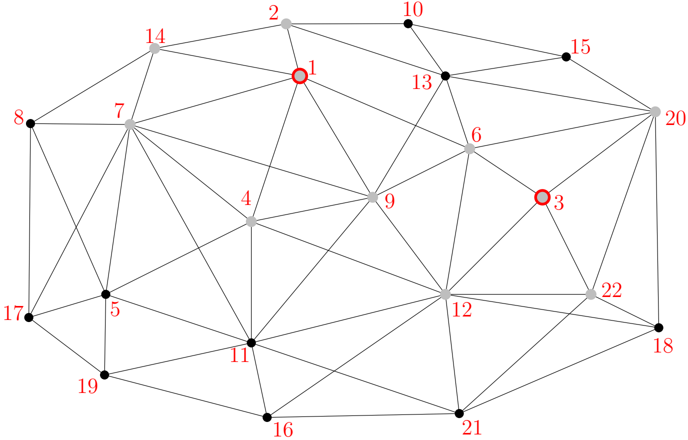
Algorithm

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



Algorithm

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



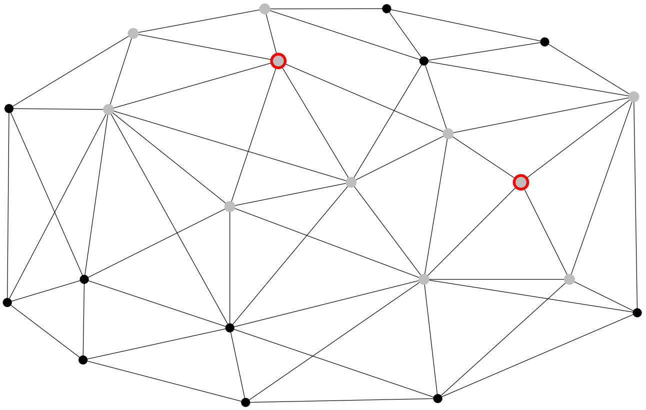
Algorithm

$A \leftarrow V$, $N \leftarrow \emptyset$;

while $A \neq \emptyset$ **do**

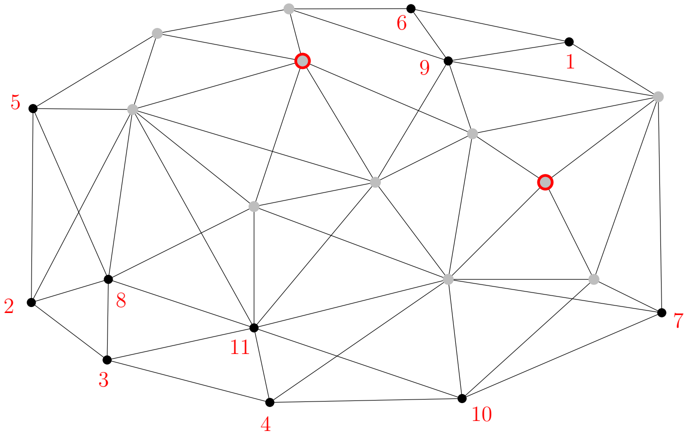
- sample permutation π over A;
- **for** $v \in A$ **do**
  - **if** $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
    - $N \leftarrow N \cup \{v\}$
- $A \leftarrow A \setminus B_G(N, \gamma)$;

return N



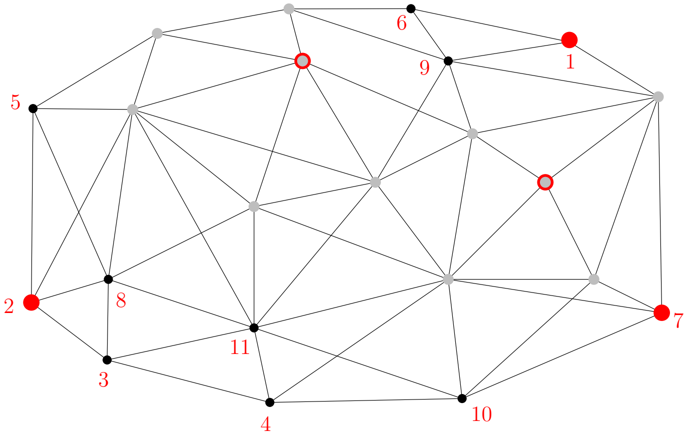
Algorithm

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



Algorithm

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



Algorithm

$A \leftarrow V$, $N \leftarrow \emptyset$;

while $A \neq \emptyset$ **do**

  sample permutation π over A;

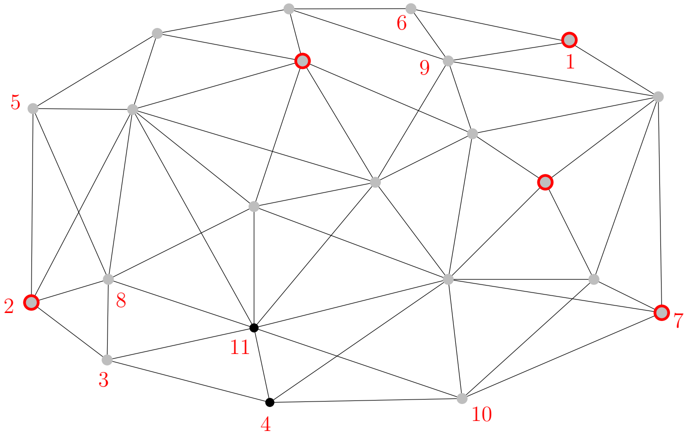
  for $v \in A$ **do**

    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**

      $N \leftarrow N \cup \{v\}$

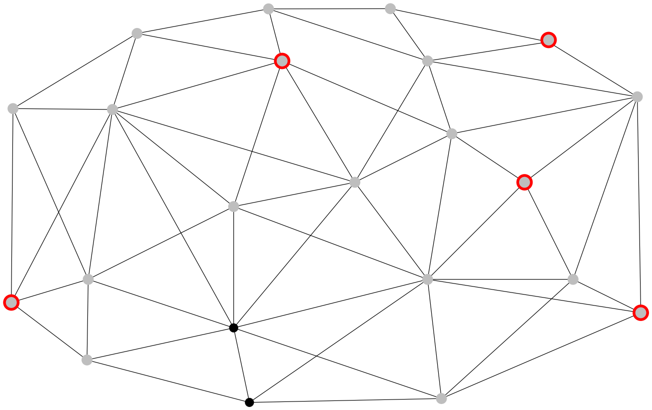
  $A \leftarrow A \setminus B_G(N, \gamma)$;

return N



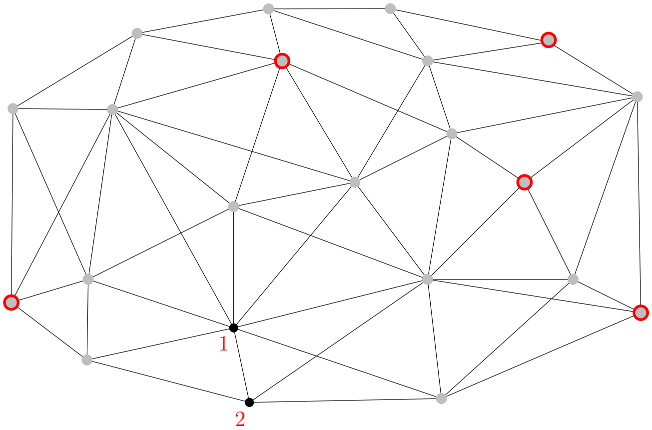
Algorithm

$A \leftarrow V,\ N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
    sample permutation π over A;
    for $v \in A$ **do**
        if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
            $N \leftarrow N \cup \{v\}$
    $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



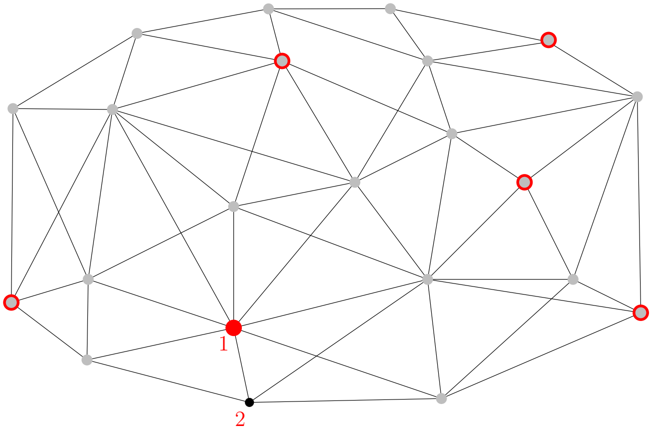
Algorithm

$A \leftarrow V,\ N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



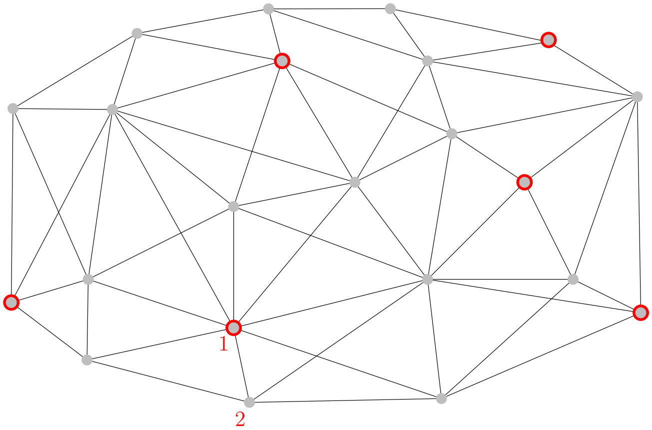
Algorithm

$A \leftarrow V,\ N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
    sample permutation π over A;
    for $v \in A$ **do**
        if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
            $N \leftarrow N \cup \{v\}$
    $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



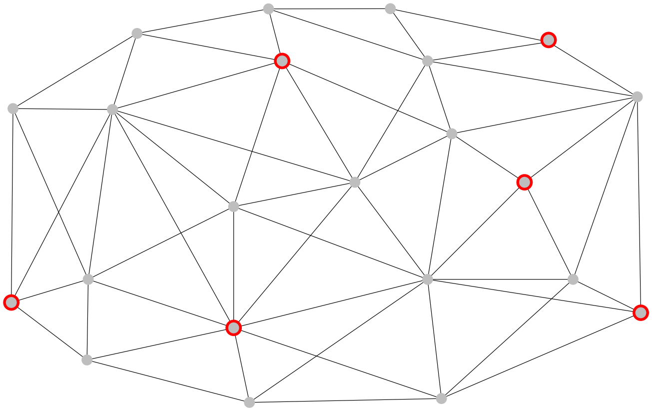
Algorithm

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



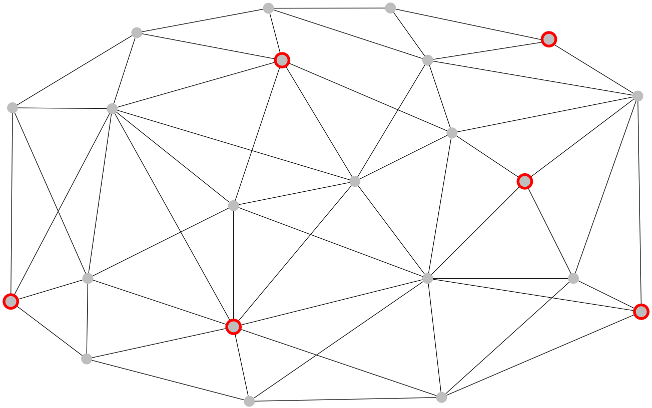
Algorithm

$A \leftarrow V,\ N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
    sample permutation π over A;
    for $v \in A$ **do**
        if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
            $N \leftarrow N \cup \{v\}$
    $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



Algorithm

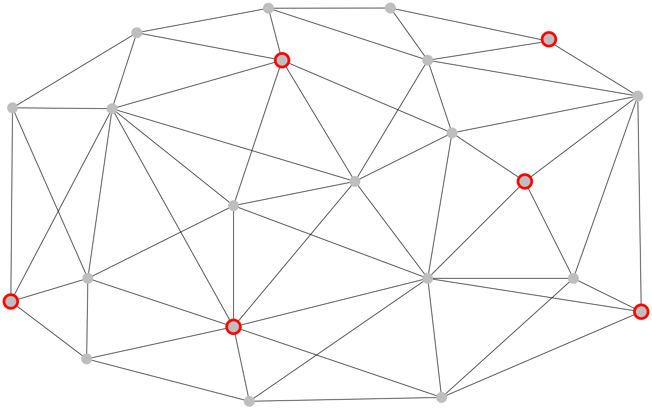
$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



Clearly N is (γ, β)-net.

Algorithm

$A \leftarrow V,\ N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



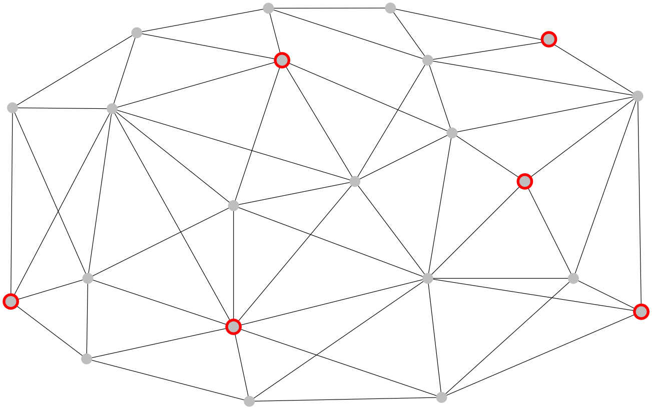
Clearly N is (γ, β)-net.

Lemma

With high probability, the process **will terminate** *in $O(\log n)$ rounds.*

Algorithm

```
A ← V, N ← ∅;
while A ≠ ∅ do
    sample permutation π over A;
    for v ∈ A do
        if π(v) = min_{u∈B_G(v,β)} π(u) then
            N ← N ∪ {v}
    A ← A \ B_G(N, γ);
return N
```



Clearly N is (γ, β)-net.

Lemma

With high probability, the process **will terminate** *in* $O(\log n)$ *rounds.*

But how to implement in the CONGEST model?

Least Elements (LE) lists

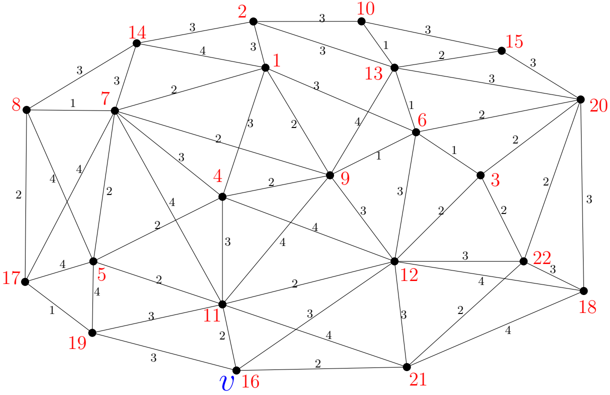
For permutation π,

$$\mathrm{LE}_{G,\pi}(v) = \Big\{(u, d_G(u,v)) \;\mid\; u,\; \nexists w \text{ s.t. } d_G(v,w) \leq d_G(v,u) \text{ and } \pi(w) < \pi(u)\Big\}$$

Least Elements (LE) lists

For permutation π,

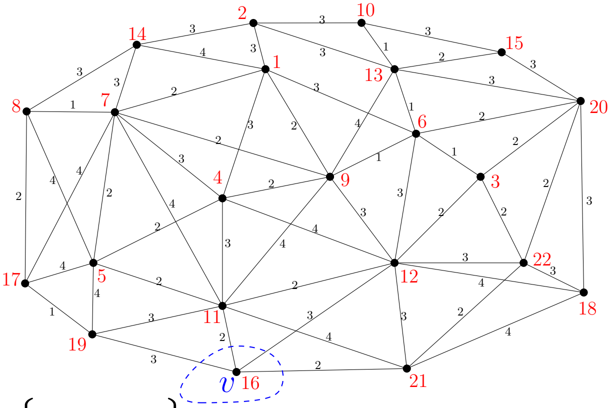
$$\mathsf{LE}_{G,\pi}(v) = \Big\{ (u, d_G(u,v)) \;\Big|\; u,\ \nexists w \text{ s.t. } d_G(v,w) \le d_G(v,u) \text{ and } \pi(w) < \pi(u) \Big\}$$



Least Elements (LE) lists

For permutation π,

$$\mathsf{LE}_{G,\pi}(v) = \Big\{ (u, d_G(u,v)) \;\Big|\; u,\ \nexists w \text{ s.t. } d_G(v,w) \le d_G(v,u) \text{ and } \pi(w) < \pi(u) \Big\}$$

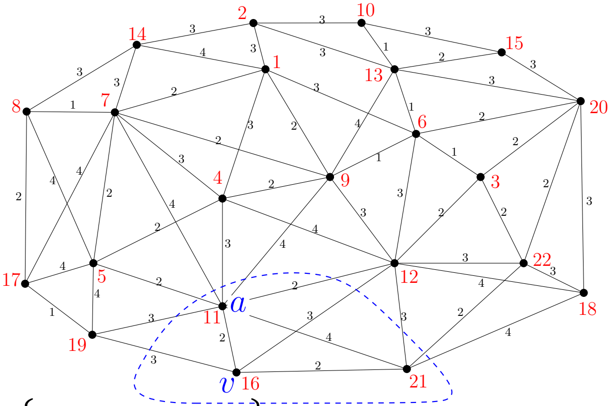


$$\mathsf{LE}_{G,\pi}(v) = \Big\{ (v, 0), \dots \Big\}$$

Least Elements (LE) lists

For permutation π,

$$\mathrm{LE}_{G,\pi}(v) = \Big\{ (u, d_G(u,v)) \;\Big|\; u,\ \nexists w \text{ s.t. } d_G(v,w) \leq d_G(v,u) \text{ and } \pi(w) < \pi(u) \Big\}$$

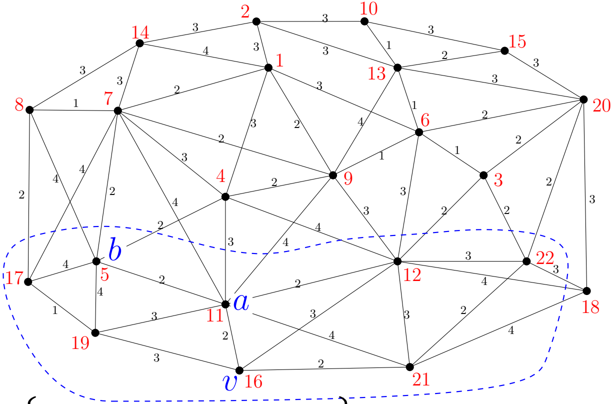


$$\mathrm{LE}_{G,\pi}(v) = \Big\{ (v, 0), (a, 2), \ldots \Big\}$$

Least Elements (LE) lists

For permutation π,

$$\mathrm{LE}_{G,\pi}(v) = \Big\{(u, d_G(u,v)) \;\Big|\; u,\ \nexists w \text{ s.t. } d_G(v,w) \le d_G(v,u) \text{ and } \pi(w) < \pi(u)\Big\}$$

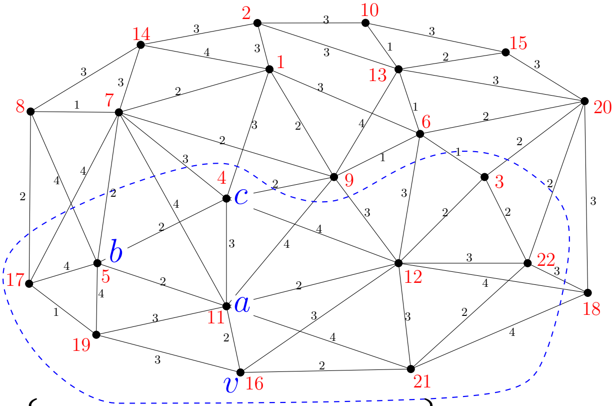


$$\mathrm{LE}_{G,\pi}(v) = \Big\{(v,0),(a,2),(b,4),\ldots\Big\}$$

Least Elements (LE) lists

For permutation π,

$$\mathrm{LE}_{G,\pi}(v) = \left\{ (u, d_G(u,v)) \;\middle|\; u,\ \nexists w \text{ s.t. } d_G(v,w) \leq d_G(v,u) \text{ and } \pi(w) < \pi(u) \right\}$$

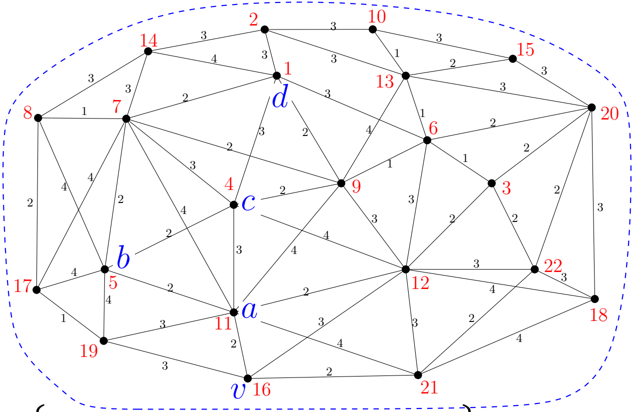


$$\mathrm{LE}_{G,\pi}(v) = \left\{ (v,0), (a,2), (b,4), (c,5), \ldots \right\}$$

Least Elements (LE) lists

For permutation π,

$$\mathsf{LE}_{G,\pi}(v) = \left\{ (u, d_G(u,v)) \;\middle|\; u,\ \nexists w \text{ s.t. } d_G(v,w) \le d_G(v,u) \text{ and } \pi(w) < \pi(u) \right\}$$



$$\mathsf{LE}_{G,\pi}(v) = \left\{ (v,0), (a,2), (b,4), (c,5), (d,8) \right\}$$

Least Elements (LE) lists

For permutation π,

$$\mathsf{LE}_{G,\pi}(v) = \Big\{(u, d_G(u,v)) \;\mid\; u,\ \nexists w \text{ s.t. } d_G(v,w) \leq d_G(v,u) \text{ and } \pi(w) < \pi(u)\Big\}$$

[Khan, Kuhn, Malkhi, Pandurangan, Talwar 2012]:

W.H.P. $\forall v \in V$, $|\mathsf{LE}_{G,\pi}(v)| = O(\log n)$.

Least Elements (LE) lists

For permutation π,

$$\mathrm{LE}_{G,\pi}(v) = \Big\{(u, d_G(u, v)) \;\mid\; u,\ \nexists w \text{ s.t. } d_G(v, w) \le d_G(v, u) \text{ and } \pi(w) < \pi(u)\Big\}$$

[KKMPT12]: W.H.P. $\forall v \in V$, $|\mathrm{LE}_{G,\pi}(v)| = O(\log n)$.

[Friedrichs and Lenzen 2016]: For $\delta \in (0, 1)$, $(\sqrt{n} + D) \cdot 2^{\tilde{O}(\sqrt{\log n} \cdot \log(1/\delta))}$ round algorithm that samples π, and computes $\{\mathrm{LE}_{H,\pi}(v)\}_{v \in V}$.

Least Elements (LE) lists

For permutation π,

$$\mathsf{LE}_{G,\pi}(v) = \Big\{(u, d_G(u,v)) \;\mid\; u,\ \nexists w \text{ s.t. } d_G(v,w) \le d_G(v,u) \text{ and } \pi(w) < \pi(u)\Big\}$$

[KKMPT12]: W.H.P. $\forall v \in V$, $|\mathsf{LE}_{G,\pi}(v)| = O(\log n)$.

[FL16]: For $\delta \in (0,1)$, $(\sqrt{n} + D) \cdot 2^{\tilde{O}(\sqrt{\log n} \cdot \log(1/\delta))}$ round algorithm that samples π, and computes $\{\mathsf{LE}_{H,\pi}(v)\}_{v \in V}$.

Here H is a graph s.t.: $\quad \forall u, v \in V, \quad d_G(u,v) \le d_H(u,v) \le (1+\delta) \cdot d_G(u,v)$.

CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;

while $A \neq \emptyset$ **do**

  sample permutation π over A;

  for $v \in A$ **do**

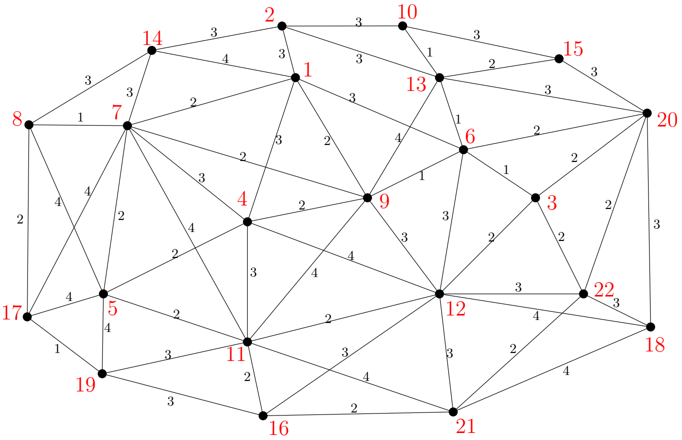
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**

      $N \leftarrow N \cup \{v\}$

  $A \leftarrow A \setminus B_G(N, \gamma)$;

return N

Fix $\gamma = \beta = 3$.



CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;

while $A \neq \emptyset$ **do**

  sample permutation π over A;

  for $v \in A$ **do**

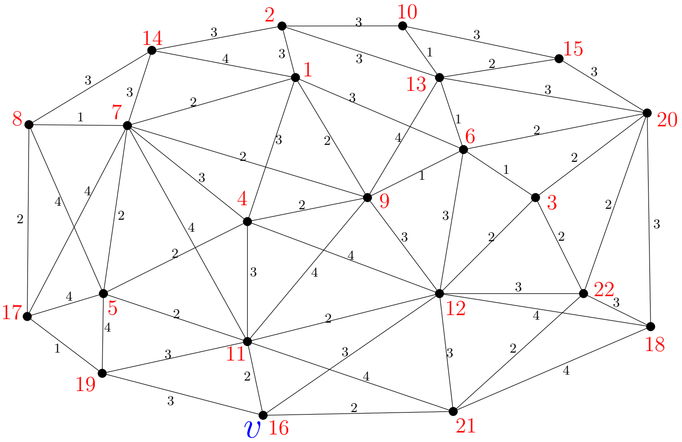
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**

      $N \leftarrow N \cup \{v\}$

  $A \leftarrow A \setminus B_G(N, \gamma)$;

return N

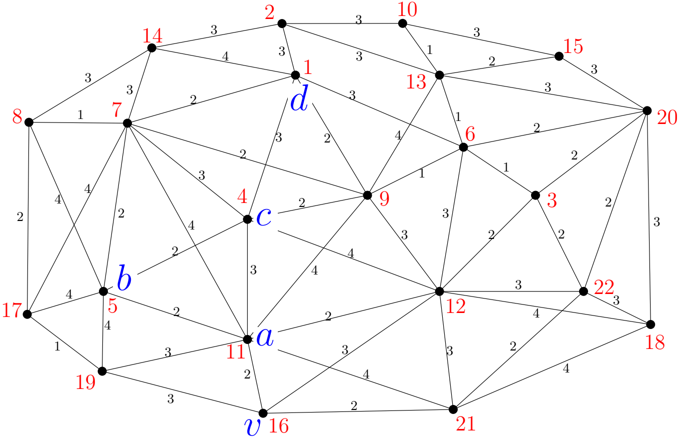
Fix $\gamma = \beta = 3$.



CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N

Fix $\gamma = \beta = 3$.



$$\mathrm{LE}_{G,\pi}(v) = \Big\{(v,0),(a,2),(b,4),(c,5),(d,8)\Big\}$$

CONGEST implementation

$A \leftarrow V,\ N \leftarrow \emptyset$;

while $A \neq \emptyset$ **do**

  sample permutation π over A;

  for $v \in A$ **do**

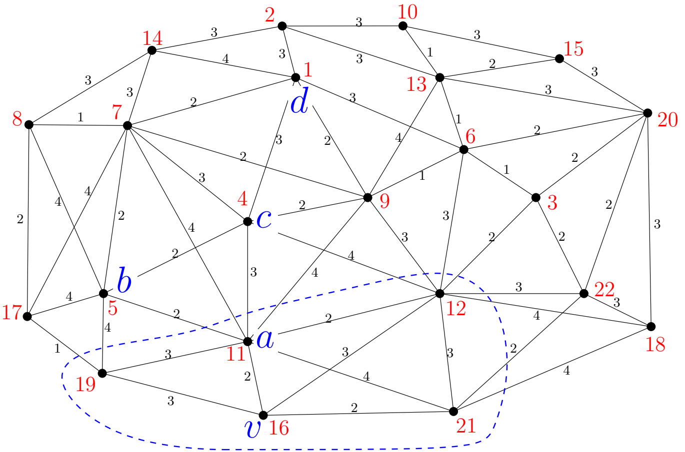
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**

      $N \leftarrow N \cup \{v\}$

  $A \leftarrow A \setminus B_G(N, \gamma)$;

return N

Fix $\gamma = \beta = 3$.



$$\mathrm{LE}_{G,\pi}(v) = \Big\{(v,0),(a,2),(b,4),(c,5),(d,8)\Big\}$$

CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;

while $A \neq \emptyset$ **do**

  sample permutation π over A;

  for $v \in A$ **do**

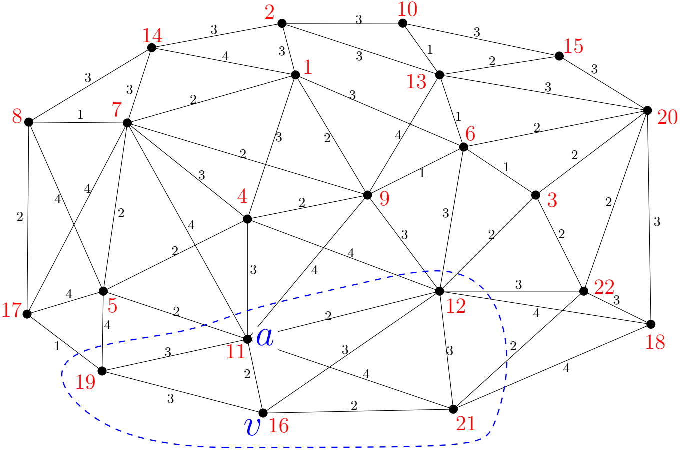
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**

      $N \leftarrow N \cup \{v\}$

  $A \leftarrow A \setminus B_G(N, \gamma)$;

return N

Fix $\gamma = \beta = 3$.



$$\mathrm{LE}_{G,\pi}(v) = \Big\{(v,0),(a,2),\ldots\Big\}$$

CONGEST implementation

$A \leftarrow V,\ N \leftarrow \emptyset$;

while $A \neq \emptyset$ **do**

  sample permutation π over A;

  for $v \in A$ **do**

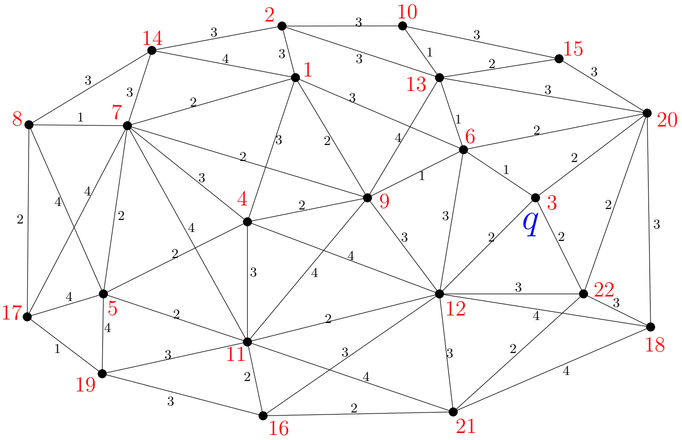
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**

      $N \leftarrow N \cup \{v\}$

  $A \leftarrow A \setminus B_G(N, \gamma)$;

return N

Fix $\gamma = \beta = 3$.



CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;

while $A \neq \emptyset$ **do**

  sample permutation π over A;

  for $v \in A$ **do**

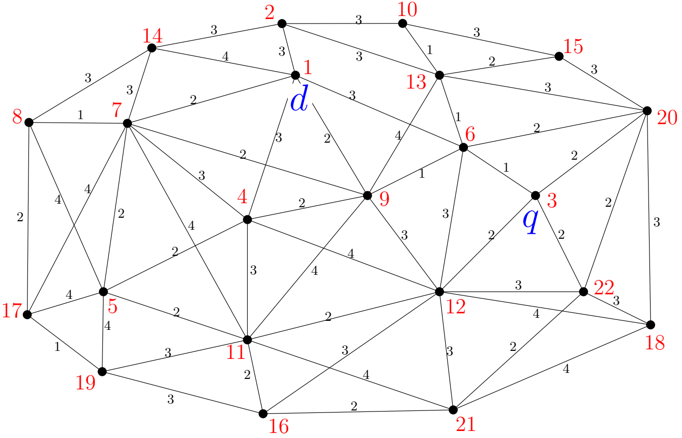
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**

      $N \leftarrow N \cup \{v\}$

  $A \leftarrow A \setminus B_G(N, \gamma)$;

return N

Fix $\gamma = \beta = 3$.

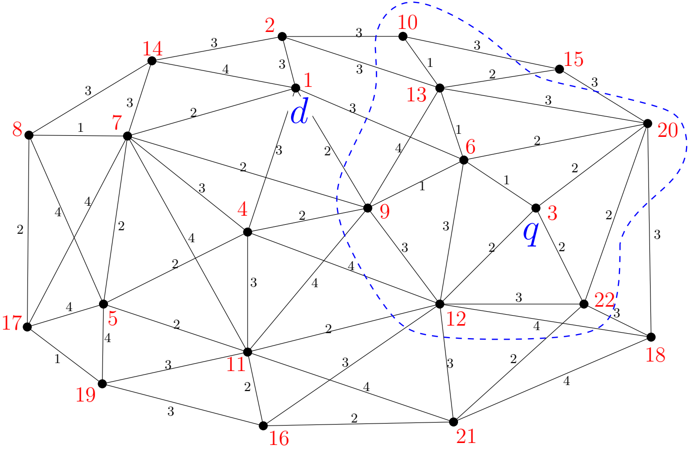


$$\mathrm{LE}_{G,\pi}(q) = \Big\{(q,0), (d,4)\Big\}$$

CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N

Fix $\gamma = \beta = 3$.

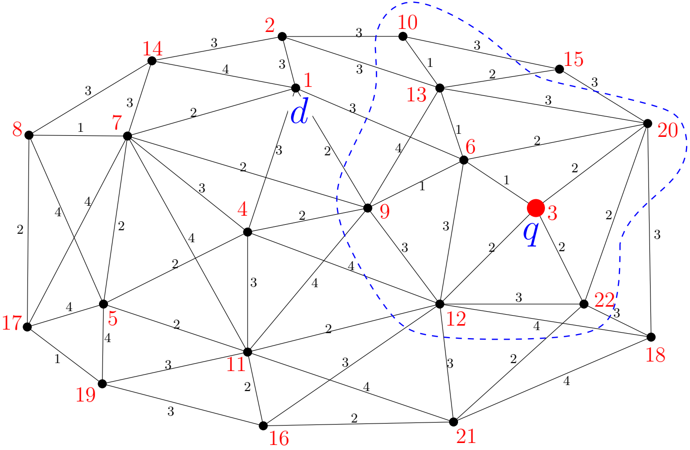


$$\mathrm{LE}_{G,\pi}(q) = \left\{ (q, 0), \dots \right\}$$

CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N

Fix $\gamma = \beta = 3$.



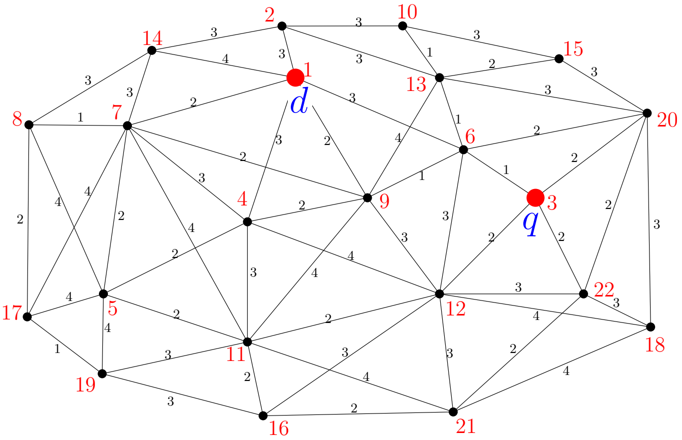
$$\mathrm{LE}_{G,\pi}(q) = \Big\{(q, 0), \ldots \Big\}$$

CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N

Fix $\gamma = \beta = 3$.

$N = \{q, d, \ldots\}$

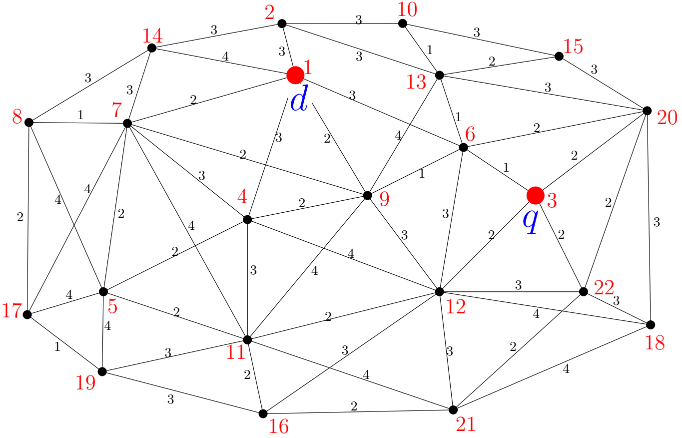


CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N

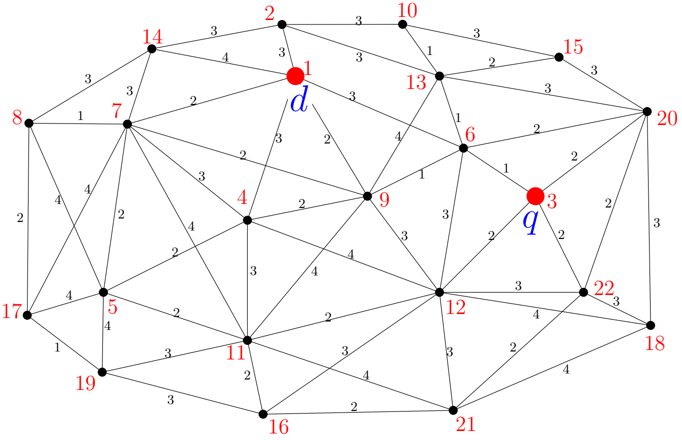
Fix $\gamma = \beta = 3$.

$N = \{q, d, \dots\}$



CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
    sample permutation π over A;
    for $v \in A$ **do**
        if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
            $N \leftarrow N \cup \{v\}$
    $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



Fix $\gamma = \beta = 3$.
$N = \{q, d, \dots\}$

[BKKL17]: Approximate SPT in $\tilde{O}_\epsilon(\sqrt{n} + D)$ rounds.

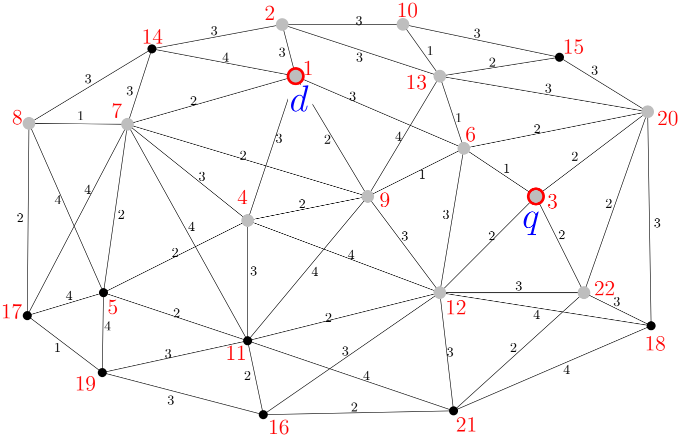
CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N

Fix $\gamma = \beta = 3$.

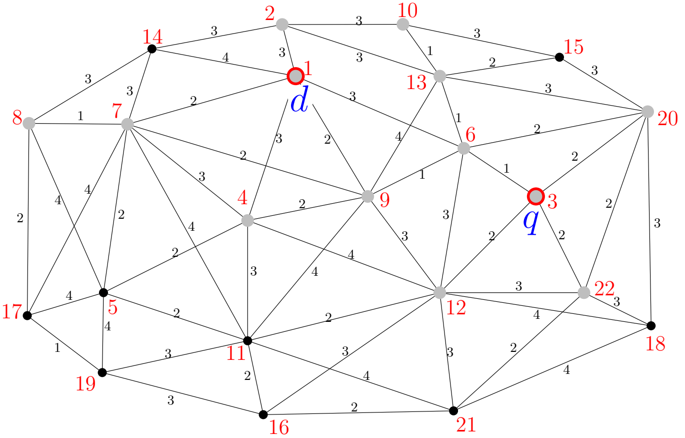
$N = \{q, d, \dots\}$

[BKKL17]: Approximate SPT in $\tilde{O}_\epsilon(\sqrt{n} + D)$ rounds.



CONGEST implementation

$A \leftarrow V$, $N \leftarrow \emptyset$;
while $A \neq \emptyset$ **do**
  sample permutation π over A;
  for $v \in A$ **do**
    if $\pi(v) = \min_{u \in B_G(v,\beta)} \pi(u)$ **then**
      $N \leftarrow N \cup \{v\}$
  $A \leftarrow A \setminus B_G(N, \gamma)$;
return N



Fix $\gamma = \beta = 3$.
$N = \{q, d, \ldots\}$

[BKKL17]: Approximate SPT in $\tilde{O}_\epsilon(\sqrt{n} + D)$ rounds.

Theorem ([EFN20] (This paper))

Algorithm for (γ, β)-net in $\tilde{O}_\epsilon(\sqrt{n} + D) \cdot n^{o(1)}$ CONGEST **rounds**.

Theorem ([EFN20])

$(2k-1)(1+\epsilon)$-**spanner** *of* $O_\epsilon(k \cdot n^{1/k})$ **lightness** *and* $O_\epsilon(k \cdot n^{1+\frac{1}{k}})$ *edges* *in* $\tilde{O}_\epsilon(n^{\frac{1}{2}+\frac{1}{4k+2}} + D)$ *CONGEST* **rounds**.

Open questions:

Theorem ([EFN20])

*$(2k-1)(1+\epsilon)$-**spanner** of $O_\epsilon(k \cdot n^{1/k})$ **lightness** and $O_\epsilon(k \cdot n^{1+\frac{1}{k}})$ edges in $\tilde{O}_\epsilon(n^{\frac{1}{2}+\frac{1}{4k+2}} + D)$ CONGEST **rounds**.*

Open questions:

- **Remove** $\frac{1}{4k+2}$ factor in rounds, remove k factor in lightness/sparsity.

Theorem ([EFN20])

*$(2k-1)(1+\epsilon)$-**spanner** of $O_\epsilon(k \cdot n^{1/k})$ **lightness** and $O_\epsilon(k \cdot n^{1+\frac{1}{k}})$ edges in $\tilde{O}_\epsilon(n^{\frac{1}{2}+\frac{1}{4k+2}} + D)$ CONGEST **rounds**.*

Open questions:

- **Remove** $\frac{1}{4k+2}$ factor in rounds, remove k factor in lightness/sparsity.
- Construct $O(\log n)$ spanner with $O(1)$ lightness in $\tilde{O}(\sqrt{n}+D)$ CONGEST **rounds**.

Theorem ([EFN20])

For every $\epsilon > 0$ *algorithm that constructs* $(1+\epsilon, 1+\frac{O(1)}{\epsilon})$-**SLT** *in* $\tilde{O}_\epsilon(\sqrt{n}+D)$ *rounds*

Open questions:

- **Remove** $\frac{1}{4k+2}$ factor in rounds, remove k factor in lightness/sparsity.
- Construct $O(\log n)$ spanner with $O(1)$ lightness in $\tilde{O}(\sqrt{n}+D)$ CONGEST **rounds**.

Theorem ([EFN20])

For every $\epsilon > 0$ *algorithm that constructs* $(1+\epsilon, 1+\frac{O(1)}{\epsilon})$-**SLT** *in* $\tilde{O}_{\epsilon}(\sqrt{n}+D)$ *rounds*

Open questions:

- **Remove** $\frac{1}{4k+2}$ factor in rounds, remove k factor in lightness/sparsity.
- Construct $O(\log n)$ spanner with $O(1)$ lightness in $\tilde{O}(\sqrt{n}+D)$ CONGEST **rounds**.
- Reduce $O(1)$ to 2.

Theorem ([EFN20])

*$1+\epsilon$-**spanner** of $\epsilon^{-O(\boldsymbol{d})}\cdot\log n$ **lightness** and $\epsilon^{-O(\boldsymbol{d})}\cdot n\log n$ edges in $\tilde{O}_\epsilon(\sqrt{n}+D)\cdot n^{o(1)}$ CONGEST **rounds**.*

Open questions:

- **Remove** $\frac{1}{4k+2}$ factor in rounds, remove k factor in lightness/sparsity.
- Construct $O(\log n)$ spanner with $O(1)$ lightness in $\tilde{O}(\sqrt{n}+D)$ CONGEST **rounds**.
- Reduce $O(1)$ to 2.

Theorem ([EFN20])

*$1+\epsilon$-**spanner** of $\epsilon^{-O(\boldsymbol{d})} \cdot \log n$ **lightness** and $\epsilon^{-O(\boldsymbol{d})} \cdot n \log n$ edges in $\tilde{O}_\epsilon(\sqrt{n}+D) \cdot n^{o(1)}$ CONGEST **rounds**.*

Open questions:

- **Remove** $\frac{1}{4k+2}$ factor in rounds, remove k factor in lightness/sparsity.
- Construct $O(\log n)$ spanner with $O(1)$ lightness in $\tilde{O}(\sqrt{n}+D)$ CONGEST **rounds**.
- Reduce $O(1)$ to 2.
- **Remove** $\log n$ factors.

Theorem ([EFN20])

*$1+\epsilon$-**spanner** of $\epsilon^{-O(\boldsymbol{d})} \cdot \log n$ **lightness** and $\epsilon^{-O(\boldsymbol{d})} \cdot n \log n$ edges in $\tilde{O}_\epsilon(\sqrt{n}+D) \cdot n^{o(1)}$ CONGEST **rounds**.*

Open questions:

- **Remove** $\frac{1}{4k+2}$ factor in rounds, remove k factor in lightness/sparsity.
- Construct $O(\log n)$ spanner with $O(1)$ lightness in $\tilde{O}(\sqrt{n}+D)$ CONGEST **rounds**.
- Reduce $O(1)$ to 2.
- **Remove** $\log n$ factors.
- Is it possible to **construct** light spanner for bounded doubling dimension graph in $o(\sqrt{n})$ CONGEST **rounds**?

Theorem ([EFN20])

$1+\epsilon$-**spanner** *of* $\epsilon^{-O(\boldsymbol{d})}\cdot\log n$ **lightness** *and* $\epsilon^{-O(\boldsymbol{d})}\cdot n\log n$ *edges in* $\tilde{O}_\epsilon(\sqrt{n}+D)\cdot n^{o(1)}$ *CONGEST* **rounds**.

Open questions:

- **Remove** $\frac{1}{4k+2}$ factor in rounds, remove k factor in lightness/sparsity.
- Construct $O(\log n)$ spanner with $O(1)$ lightness in $\tilde{O}(\sqrt{n}+D)$ CONGEST **rounds**.
- Reduce $O(1)$ to 2.
- **Remove** $\log n$ factors.
- Is it possible to **construct** light spanner for bounded doubling dimension graph in $o(\sqrt{n})$ CONGEST **rounds**?
- **Metric TSP**: find approximation better than 2 in $\tilde{O}(\sqrt{n}+D)$ rounds.

Theorem ([EFN20])

$1+\epsilon$-**spanner** *of* $\epsilon^{-O(\mathbf{d})}\cdot\log n$ **lightness** *and* $\epsilon^{-O(\mathbf{d})}\cdot n\log n$ *edges in* $\tilde{O}_\epsilon(\sqrt{n}+D)\cdot n^{o(1)}$ *CONGEST* **rounds**.

Open questions:

- **Remove** $\frac{1}{4k+2}$ factor in rounds, remove k factor in lightness/sparsity.
- Construct $O(\log n)$ spanner with $O(1)$ lightness in $\tilde{O}(\sqrt{n}+D)$ CONGEST **rounds**.
- Reduce $O(1)$ to 2.
- **Remove** $\log n$ factors.
- Is it possible to **construct** light spanner for bounded doubling dimension graph in $o(\sqrt{n})$ CONGEST **rounds**?
- **Metric TSP**: find approximation better than 2 in $\tilde{O}(\sqrt{n}+D)$ rounds.

Thank you for listening!