

Fast Deterministic Distributed Algorithms for Sparse Spanners

Bilel Derbel

(LaBRI, Bordeaux 1)

joint work with

Cyril Gavoille

(LaBRI, Bordeaux1)

03 June 2006

SIROCCO06



Problem

Definition: H is a k -spanner of G if $H \subseteq G$ and $d_H(u, v) \leq k \cdot d_G(u, v) \quad \forall u, v \in V(G)$

Problem

Definition: H is a k -spanner of G if $H \subseteq G$ and $d_H(u, v) \leq k \cdot d_G(u, v) \quad \forall u, v \in V(G)$

- G is a 1-spanner of G
- A BFS spanning tree of G is a D -spanner of G .

Problem

Definition: H is a k -spanner of G if $H \subseteq G$ and
 $d_H(u, v) \leq k \cdot d_G(u, v) \quad \forall u, v \in V(G)$

We want to **efficiently** compute a **high quality**
spanner in a **distributed way**.

Problem

Definition: H is a k -spanner of G if $H \subseteq G$ and $d_H(u, v) \leq k \cdot d_G(u, v) \quad \forall u, v \in V(G)$

We want to **efficiently** compute a **high quality** spanner in a **distributed way**.

- Model of computation = *LOCAL* model
 - *locality nature* of a distributed problem
 - in 1 time unit, a node can send messages of unlimited size to its neighbours.
 - negligible time for local computations.

Problem

Definition: H is a k -spanner of G if $H \subseteq G$ and $d_H(u, v) \leq k \cdot d_G(u, v) \quad \forall u, v \in V(G)$

We want to **efficiently** compute a **high quality** spanner in a **distributed way**.

- Model of computation = *LOCAL* model
- *high quality* = good stretch-size trade-offs

Problem

Definition: H is a k -spanner of G if $H \subseteq G$ and $d_H(u, v) \leq k \cdot d_G(u, v) \quad \forall u, v \in V(G)$

We want to **efficiently** compute a **high quality** spanner in a **distributed way**.

- Model of computation = *LOCAL* model
- *high quality* = good stretch-size trade-offs
 - **[Folklore]** Every graph G has a $(2k - 1)$ -spanner with $O(n^{1+1/k})$ edges
 - this stretch-size trade-off is believed to be tight according to a girth conjecture by Erdos (proved for $k = 1, 2, 3, 5$).

Problem

Deterministic Constructions (**A. Awerbuch, D. Peleg ...**)

• $O(n^{\epsilon+1/k})$ time \iff Sparse covers and decompositions

Problem

Deterministic Constructions (A. Awerbuch, D. Peleg ...)

- $O(n^{\epsilon+1/k})$ time \iff Sparse covers and decompositions
 - $\implies O(k)$ -spanner with $O(n^{1+1/k})$ edges.

Problem

Deterministic Constructions (A. Awerbuch, D. Peleg ...)

- $O(n^{\epsilon+1/k})$ time \iff Sparse covers and decompositions
- $\implies O(k)$ -spanner with $O(n^{1+1/k})$ edges.

Can we construct a constant stretch spanner with $o(n^2)$ edges in $O(n^\epsilon)$ **(deterministic) time** ?

Problem

Deterministic Constructions (A. Awerbuch, D. Peleg ...)

- $O(n^{\epsilon+1/k})$ time \iff Sparse covers and decompositions
- $\implies O(k)$ -spanner with $O(n^{1+1/k})$ edges.

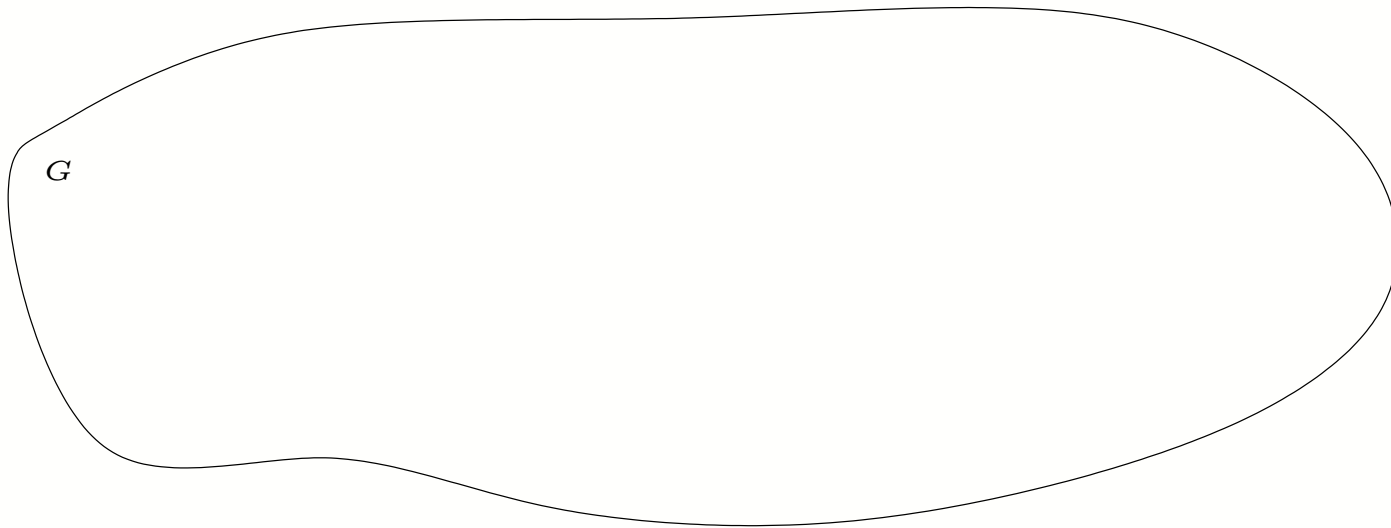
Can we construct a constant stretch spanner with $o(n^2)$ edges in $O(n^\epsilon)$ **(deterministic) time** ?

YES

Main idea of our algorithm

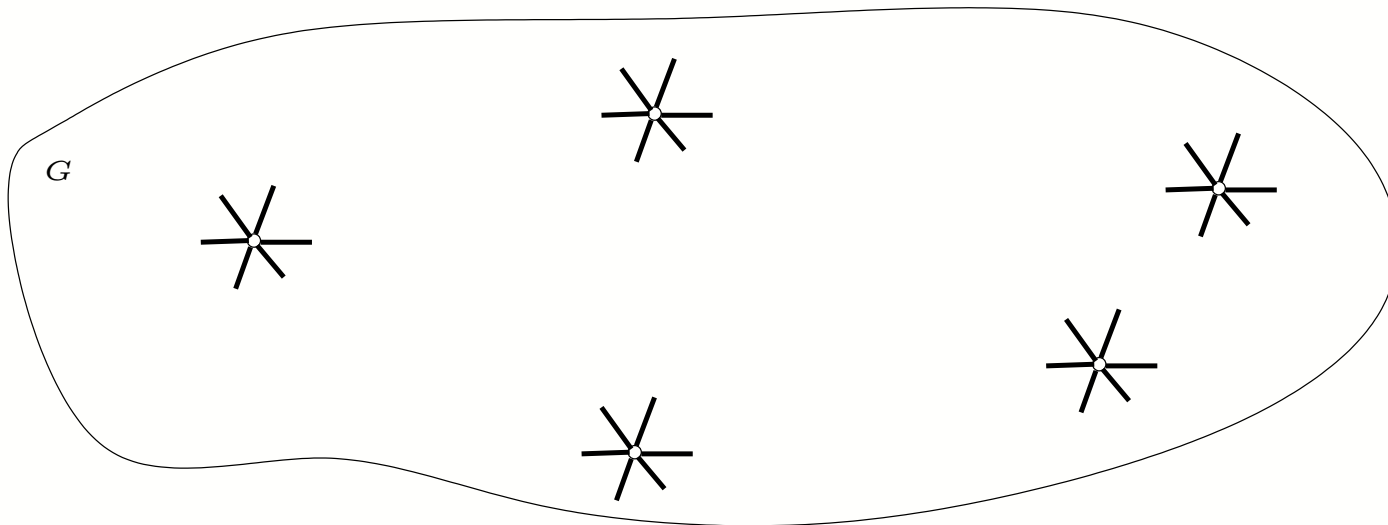
- add edges to the spanner in *parallel* at different regions of the graph
- break the symmetry
- we use a maximal independent ρ -dominating set $IDS(G)$
 - $\forall u \in V, \exists v \in IDS(G)$ such that $d(u, v) \leq \rho$.
 - $\forall u, v \in IDS(G), d(u, v) \geq 2$, i.e., $(u, v) \notin E$.
 - maximal for inclusion.
- a MIS is a maximal independent 1-dominating set

Main idea of our algorithm : $k = 2$



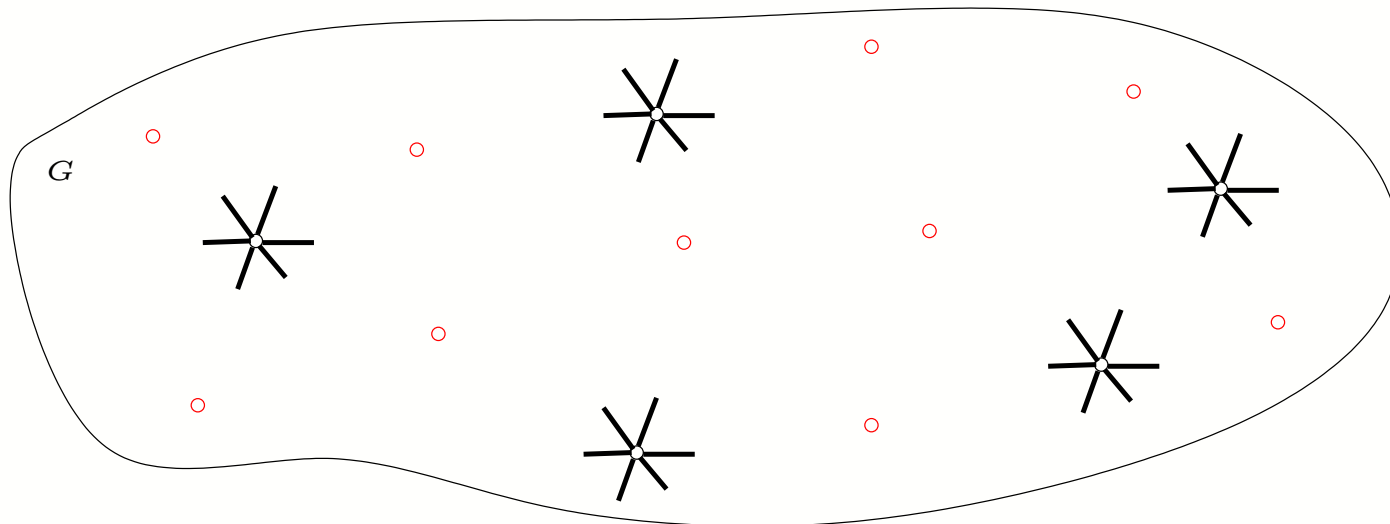
Main idea of our algorithm : $k = 2$

- if the degree d_v of a node v verifies $d_v \leq \sqrt{n}$, then
 - add the star $\mathcal{N}(v)$ to the spanner.
 - delete v from G



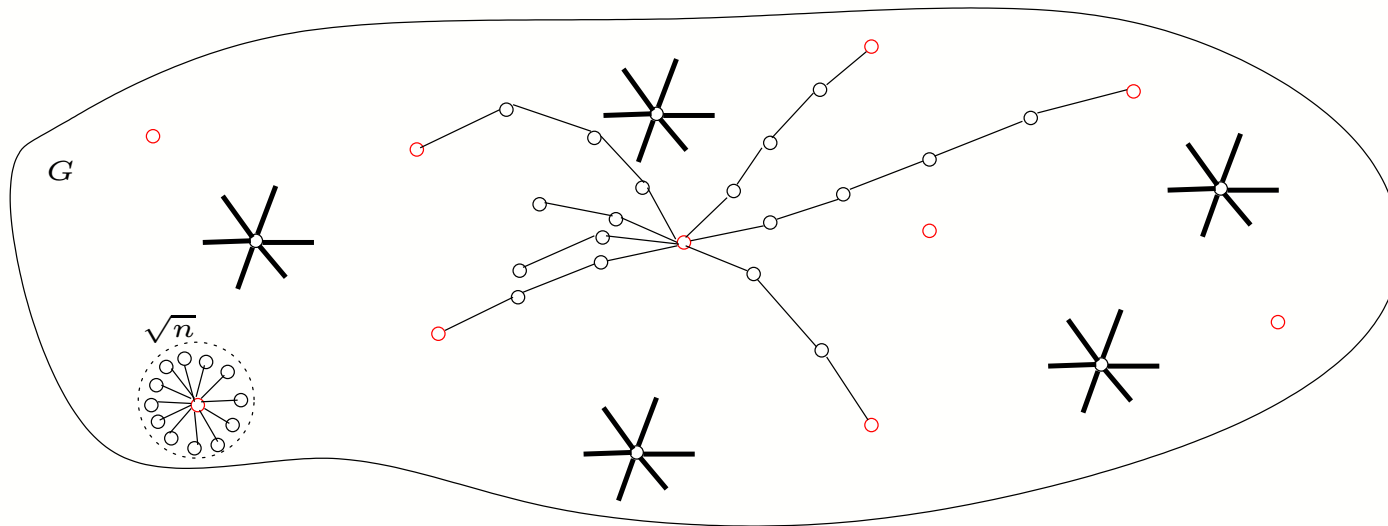
Main idea of our algorithm : $k = 2$

- if the degree d_v of a node v verifies $d_v \leq \sqrt{n}$, then
 - add the star $\mathcal{N}(v)$ to the spanner.
 - delete v from G
- find a maximal independent set I of G^2 .



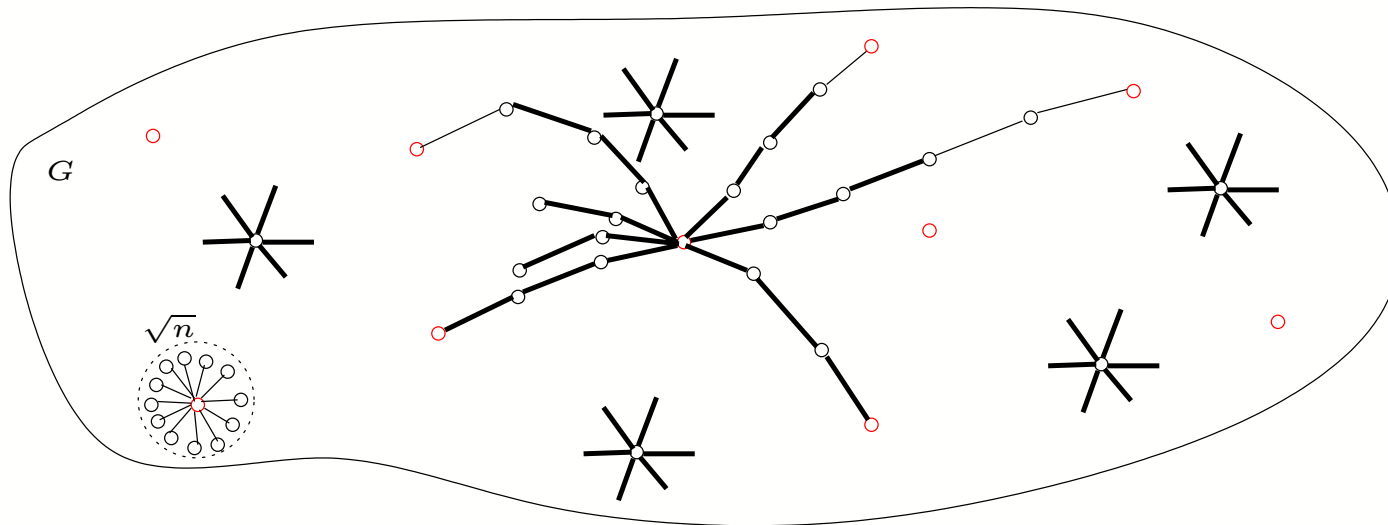
Main idea of our algorithm : $k = 2$

- if the degree d_v of a node v verifies $d_v \leq \sqrt{n}$, then
 - add the star $\mathcal{N}(v)$ to the spanner.
 - delete v from G
- find a maximal independent set I of G^2 .
 - $\implies \forall u \in V, d(u, I) \leq 2$
 - $\implies |I| \leq n/\sqrt{n} = \sqrt{n}$



Main idea of our algorithm : $k = 2$

- if the degree d_v of a node v verifies $d_v \leq \sqrt{n}$, then
 - add the star $\mathcal{N}(v)$ to the spanner.
 - delete v from G
- find a maximal independent set I of G^2 .
 - $\implies \forall u \in V, d(u, I) \leq 2$
 - $\implies |I| \leq n/\sqrt{n} = \sqrt{n}$
- $\forall v \in I$, add a BFS tree of $\mathcal{N}_3(v)$ to the spanner



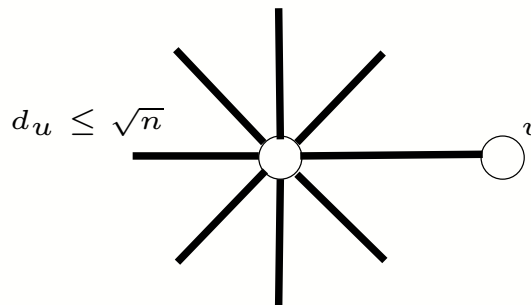
Main idea of our algorithm : $k = 2$

- if the degree d_v of a node v verifies $d_v \leq \sqrt{n}$, then
 - add the star $\mathcal{N}(v)$ to the spanner.
 - delete v from G
- find a maximal independent set I of G^2 .
 - $\implies \forall u \in V, d(u, I) \leq 2$
 - $\implies |I| \leq n/\sqrt{n} = \sqrt{n}$
- $\forall v \in I$, add a BFS tree of $\mathcal{N}_3(v)$ to the spanner
- Analysis : Let us consider an edge u, v



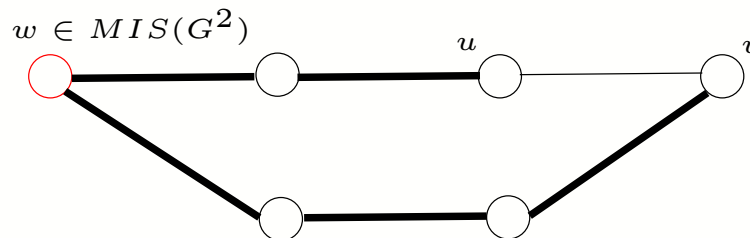
Main idea of our algorithm : $k = 2$

- if the degree d_v of a node v verifies $d_v \leq \sqrt{n}$, then
 - add the star $\mathcal{N}(v)$ to the spanner.
 - delete v from G
- find a maximal independent set I of G^2 .
 - $\implies \forall u \in V, d(u, I) \leq 2$
 - $\implies |I| \leq n/\sqrt{n} = \sqrt{n}$
- $\forall v \in I$, add a BFS tree of $\mathcal{N}_3(v)$ to the spanner
- Analysis : Let us consider an edge u, v
 - case 1 : $d_u \leq \sqrt{n} \implies \text{stretch} = 1$



Main idea of our algorithm : $k = 2$

- if the degree d_v of a node v verifies $d_v \leq \sqrt{n}$, then
 - add the star $\mathcal{N}(v)$ to the spanner.
 - delete v from G
- find a maximal independent set I of G^2 .
 - $\implies \forall u \in V, d(u, I) \leq 2$
 - $\implies |I| \leq n/\sqrt{n} = \sqrt{n}$
- $\forall v \in I$, add a BFS tree of $\mathcal{N}_3(v)$ to the spanner
- Analysis : Let us consider an edge u, v
 - case 1 : $d_u \leq \sqrt{n} \implies \text{stretch} = 1$
 - case 2 : $d_u > \sqrt{n}$ and $d_v > \sqrt{n} \implies \text{stretch} = 5$



Main idea of our algorithm : $k = 2$

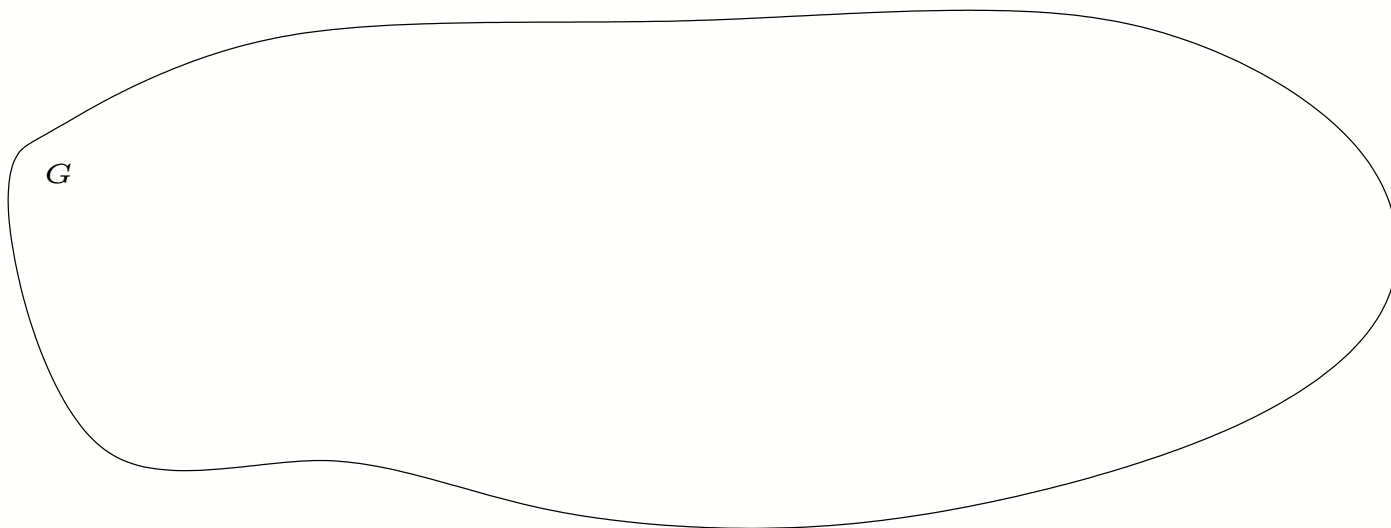
- if the degree d_v of a node v verifies $d_v \leq \sqrt{n}$, then
 - add the star $\mathcal{N}(v)$ to the spanner.
 - delete v from G
- find a maximal independent set I of G^2 .
 - $\implies \forall u \in V, d(u, I) \leq 2$
 - $\implies |I| \leq n/\sqrt{n} = \sqrt{n}$
- $\forall v \in I$, add a BFS tree of $\mathcal{N}_3(v)$ to the spanner
- Analysis
 - stretch = 5
 - size = $n \cdot \sqrt{n} + \sqrt{n} \cdot n = O(n^{3/2})$

Main idea of our algorithm : $k = 2$

- if the degree d_v of a node v verifies $d_v \leq \sqrt{n}$, then
 - add the star $\mathcal{N}(v)$ to the spanner.
 - delete v from G
- find a maximal independent set I of G^2 .
 - $\implies \forall u \in V, d(u, I) \leq 2$
 - $\implies |I| \leq n/\sqrt{n} = \sqrt{n}$
- $\forall v \in I$, add a BFS tree of $\mathcal{N}_3(v)$ to the spanner
- Analysis
 - stretch = 5
 - size = $O(n^{3/2})$

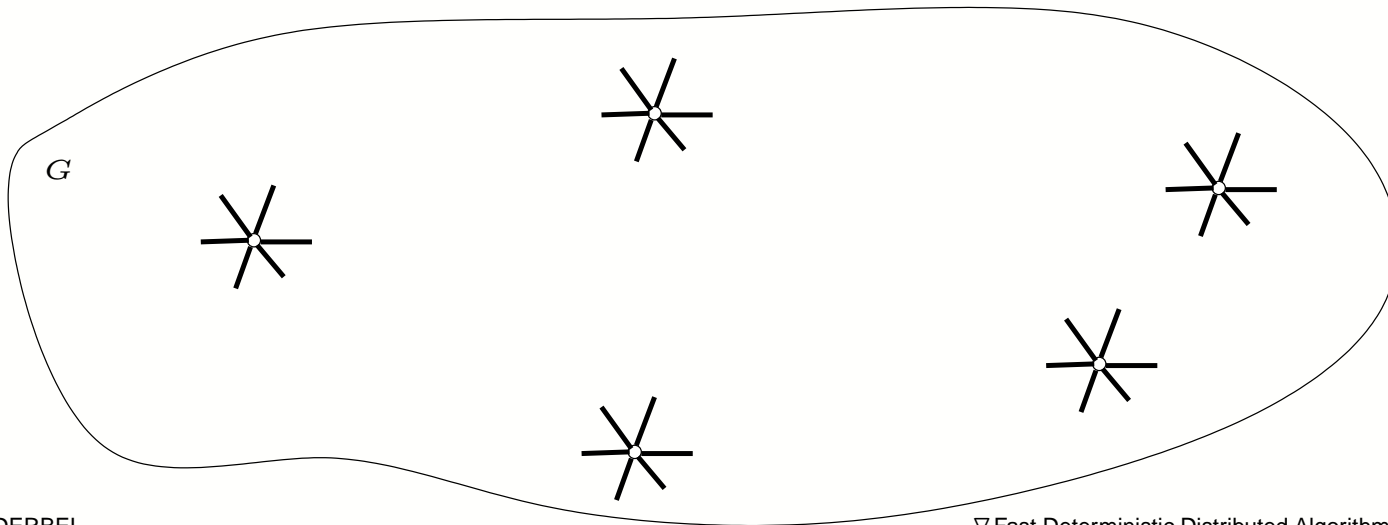
5-spanner with $O(n^{3/2})$ edges.

Main idea of our algorithm : $k = 3$



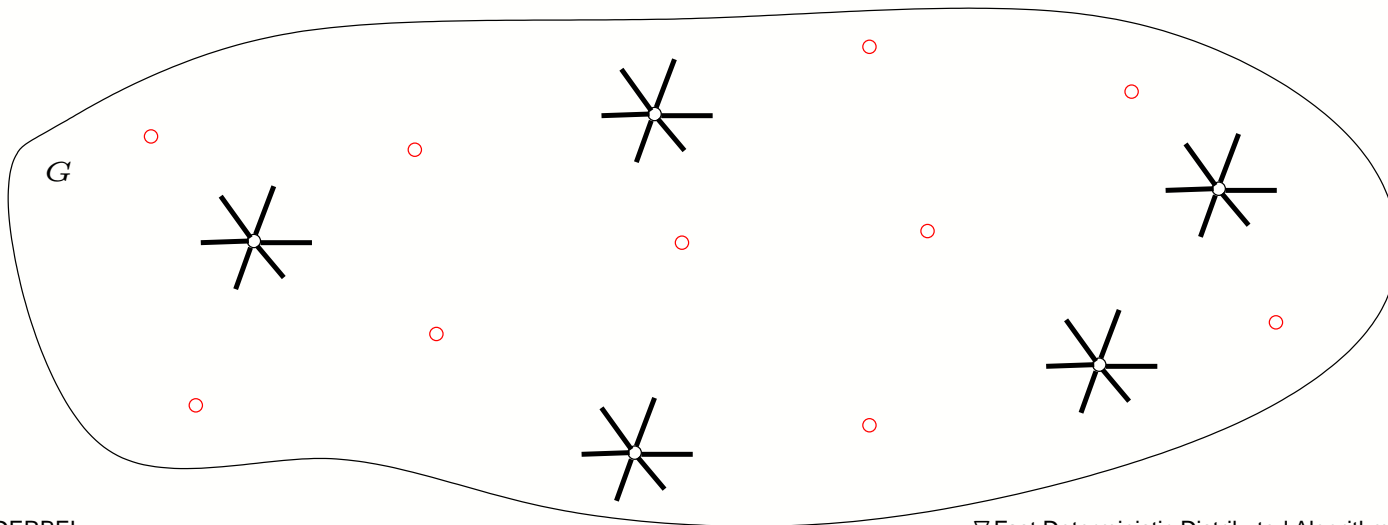
Main idea of our algorithm : $k = 3$

- $d_v \leq n^{1/3} \implies$ add the star $\mathcal{N}(v)$ to the spanner.



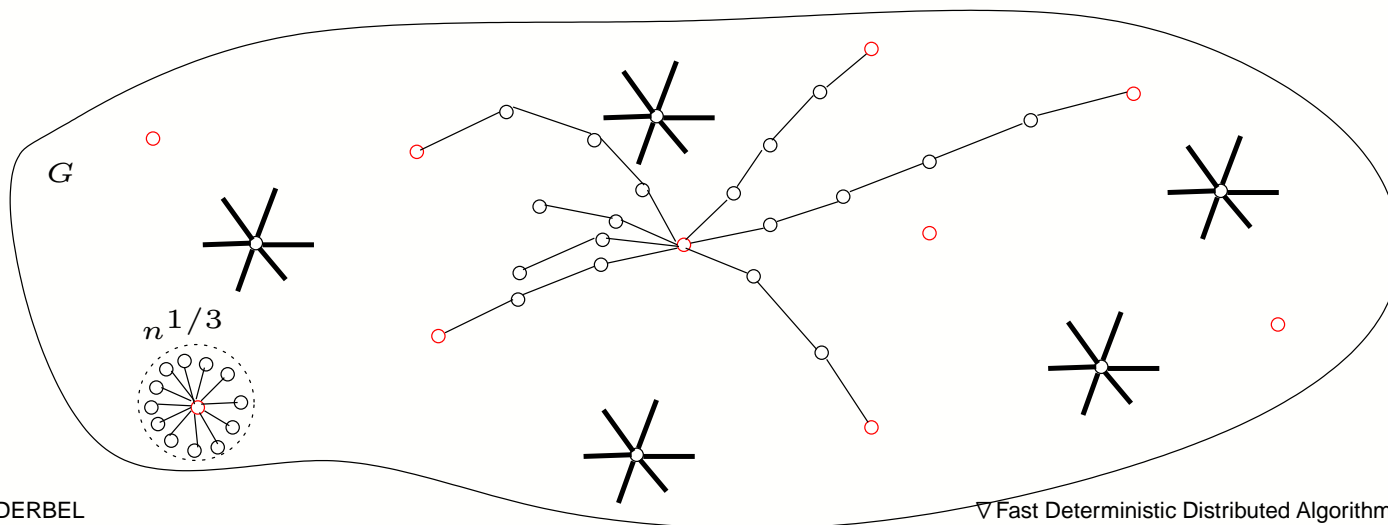
Main idea of our algorithm : $k = 3$

- $d_v \leq n^{1/3} \implies$ add the star $\mathcal{N}(v)$ to the spanner.
- find a MIS I of G^2 .



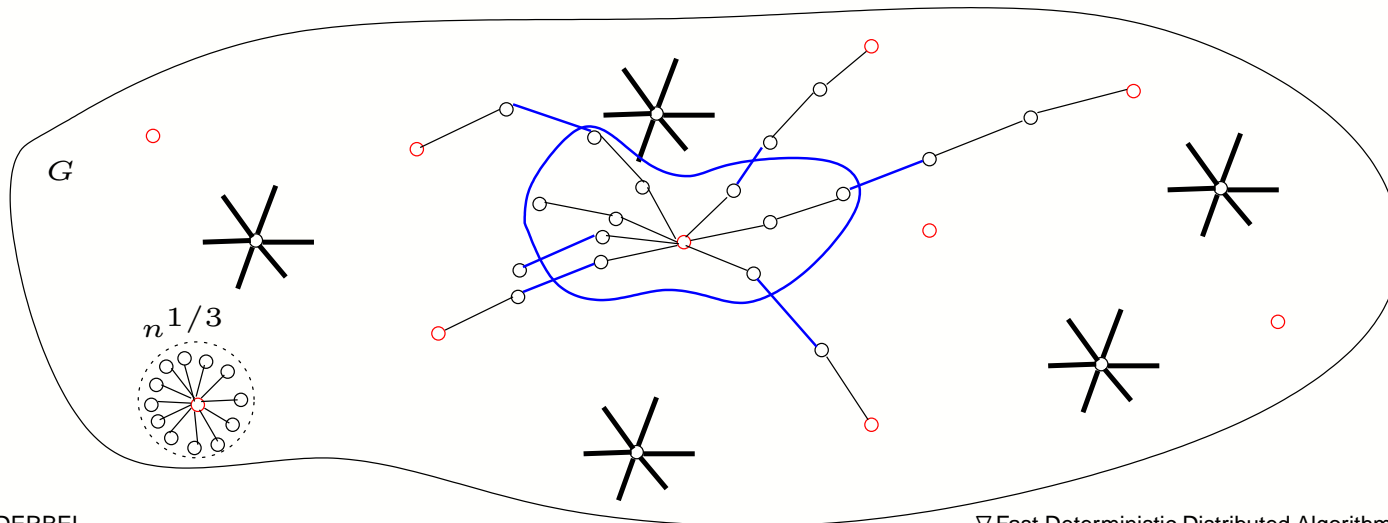
Main idea of our algorithm : $k = 3$

- $d_v \leq n^{1/3} \implies$ add the star $\mathcal{N}(v)$ to the spanner.
- find a MIS I of G^2 .
 - $\implies |I| \leq n/n^{1/3} = n^{2/3}$
 - $n \cdot n^{2/3} > n^{1+1/3} !$



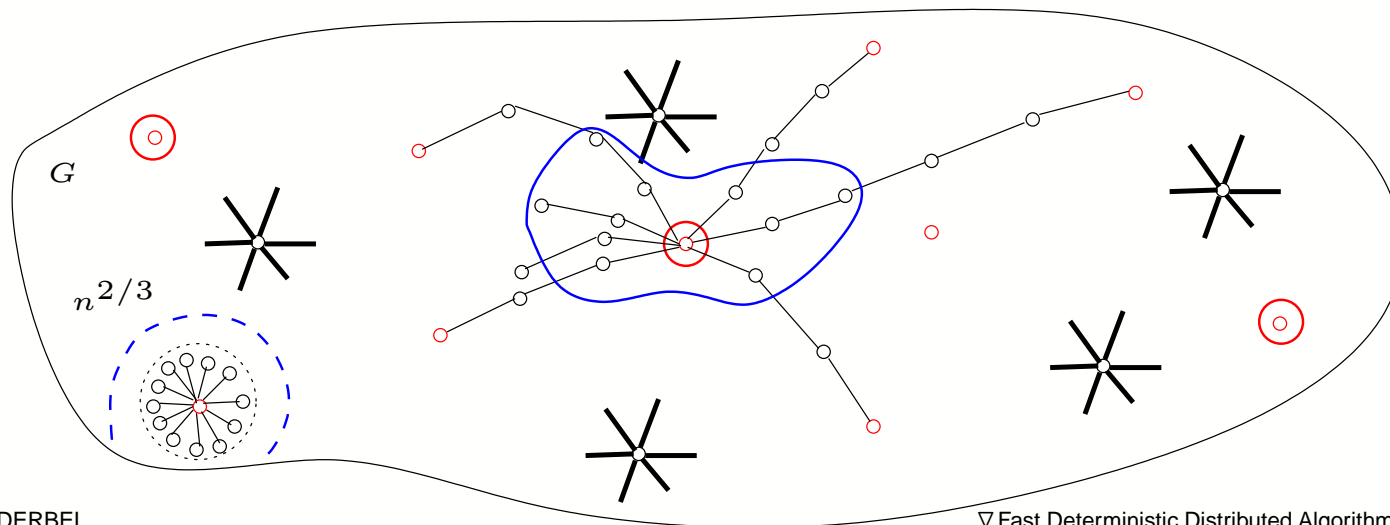
Main idea of our algorithm : $k = 3$

- $d_v \leq n^{1/3} \implies$ add the star $\mathcal{N}(v)$ to the spanner.
- find a MIS I of G^2 .
 - $\implies |I| \leq n/n^{1/3} = n^{2/3}$
 - $n \cdot n^{2/3} > n^{1+1/3} !$
- $\forall v \in I$, construct the region $R(v)$ of v
 - if $|\Gamma(R(v))| \leq n^{1/3}|R(v)|$, then span $\Gamma(R(v))$
 - $\implies \sum |\Gamma(R(v))| \leq n^{1/3} \sum |R(v)| = n^{1/3} \cdot n$



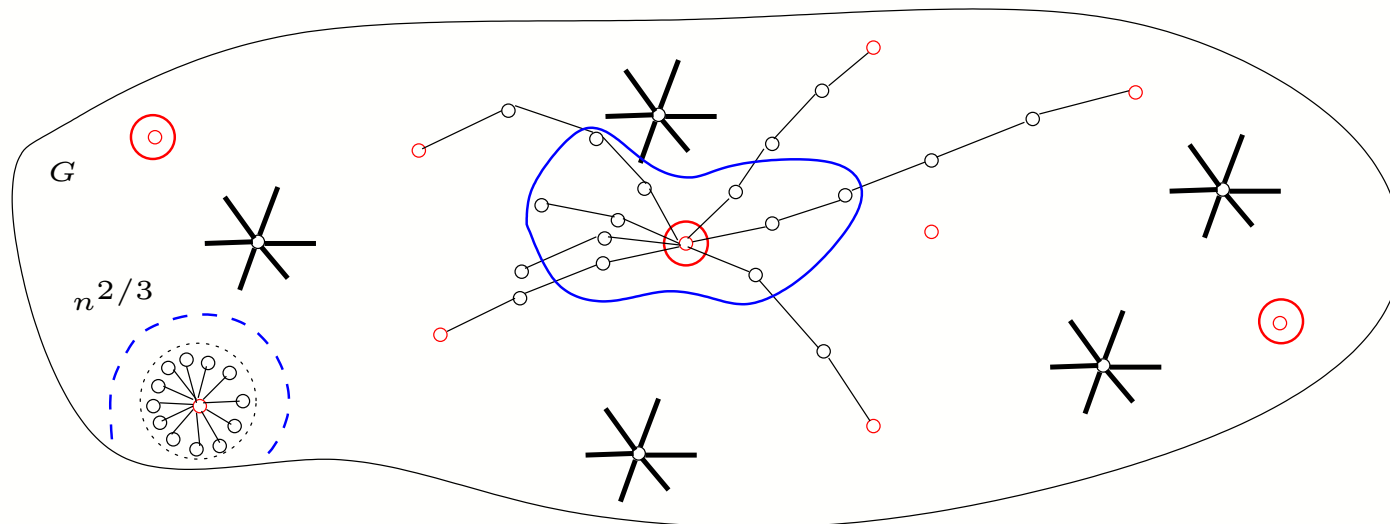
Main idea of our algorithm : $k = 3$

- $d_v \leq n^{1/3} \implies$ add the star $\mathcal{N}(v)$ to the spanner.
- find a MIS I of G^2 .
 - $\implies |I| \leq n/n^{1/3} = n^{2/3}$
 - $n \cdot n^{2/3} > n^{1+1/3} !$
- $\forall v \in I$, construct the region $R(v)$ of v
- find a MIS I' of $G^6[I]$
 - $\implies \forall u \in I, d(u, I') \leq 6 \implies \forall v \in V, d(v, I') \leq 8$
 - $\implies |I'| \leq n/n^{2/3} = n^{1/3}$



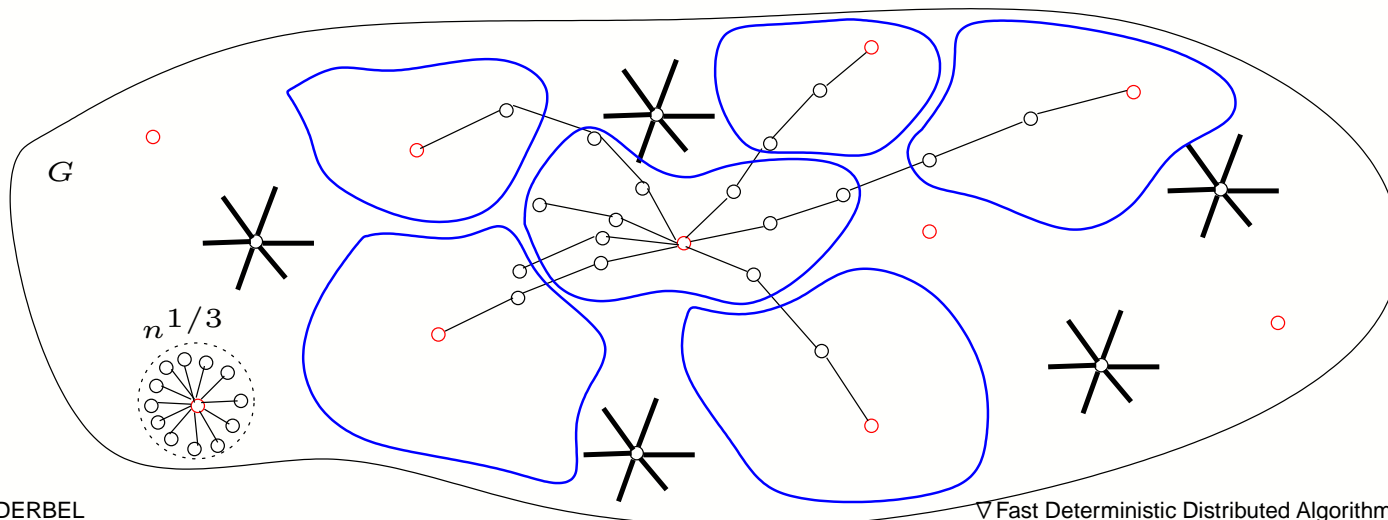
Main idea of our algorithm : $k = 3$

- $d_v \leq n^{1/3} \implies$ add the star $\mathcal{N}(v)$ to the spanner.
- find a MIS I of G^2 .
 - $\implies |I| \leq n/n^{1/3} = n^{2/3}$
 - $n \cdot n^{2/3} > n^{1+1/3} !$
- $\forall v \in I$, construct the region $R(v)$ of v
- find a MIS I' of $G^6(I)$
- $\forall v \in I'$, add a BFS tree of $\mathcal{N}_9(v)$ to the spanner
 - stretch = $2 \cdot 8 + 1 = 17$



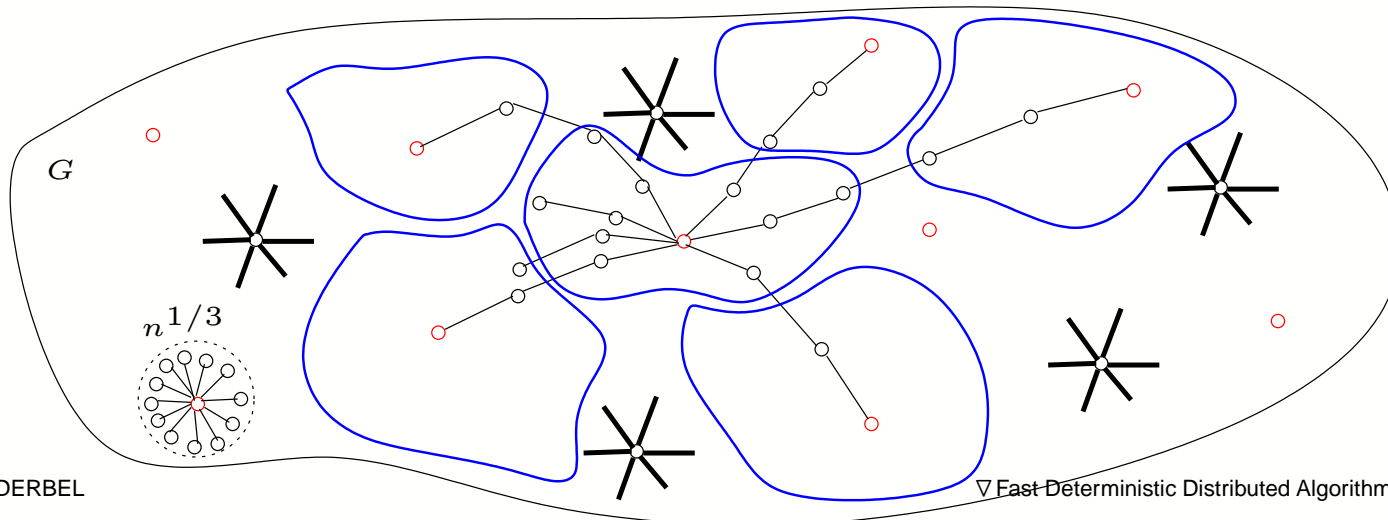
Main idea of our algorithm : $k = 3$

- $d_v \leq n^{1/3} \implies$ add the star $\mathcal{N}(v)$ to the spanner.
- find a MIS I of G^2 .
 - $\implies |I| \leq n/n^{1/3} = n^{2/3}$
 - $n \cdot n^{2/3} > n^{1+1/3} !$
- $\forall v \in I$, construct the region $R(v)$ of v
 - there are at most $n^{2/3}$ regions!



Main idea of our algorithm : $k = 3$

- $d_v \leq n^{1/3} \implies$ add the star $\mathcal{N}(v)$ to the spanner.
- find a MIS I of G^2 .
 - $\implies |I| \leq n/n^{1/3} = n^{2/3}$
 - $n \cdot n^{2/3} > n^{1+1/3} !$
- $\forall v \in I$, construct the region $R(v)$ of v
 - there are at most $n^{2/3}$ regions!
- connect each two neighbouring regions
 - size = $n + n^{2/3}n^{2/3} = O(n^{1+1/3})$
 - stretch = $2 + 2 + 1 + 2 + 2 = 9$



Main idea of our algorithm : $k = 3$

- $d_v \leq n^{1/3} \implies$ add the star $\mathcal{N}(v)$ to the spanner.
- find a MIS I of G^2 .
 - $\implies |I| \leq n/n^{1/3} = n^{2/3}$
 - $n \cdot n^{2/3} > n^{1+1/3} !$
- $\forall v \in I$, construct the region $R(v)$ of v
 - there are at most $n^{2/3}$ regions !
- connect each two neighbouring regions
 - size = $n + n^{2/3}n^{2/3} = O(n^{1+1/3})$
 - stretch = $2 + 2 + 1 + 2 + 2 = 9$
- sparsity condition using the number of neighbouring regions

Main idea : general framework

- associate a region $R(v)$ to each node v
- select some regions to enlarge in parallel
 - independent dominating set
- enlarge the dense regions
- merge the regions together in order to span all nodes in non selected regions

How to merge the regions ?

- if $R(v)$ is sparse, then span the edges of $R(v)$

How to decide if a region is sparse ?



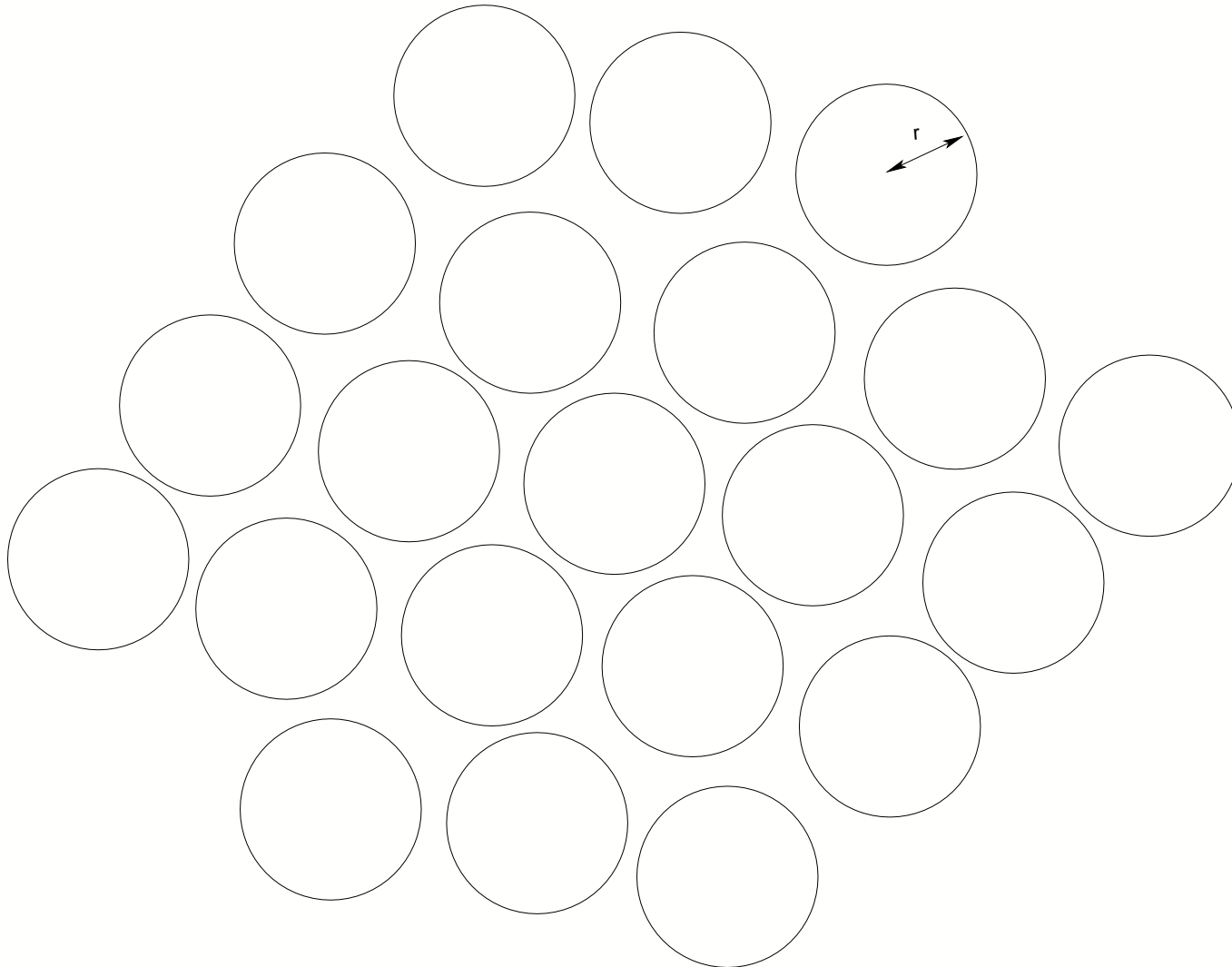
First strategy

- sparsity = the neighbourhood of a region in G



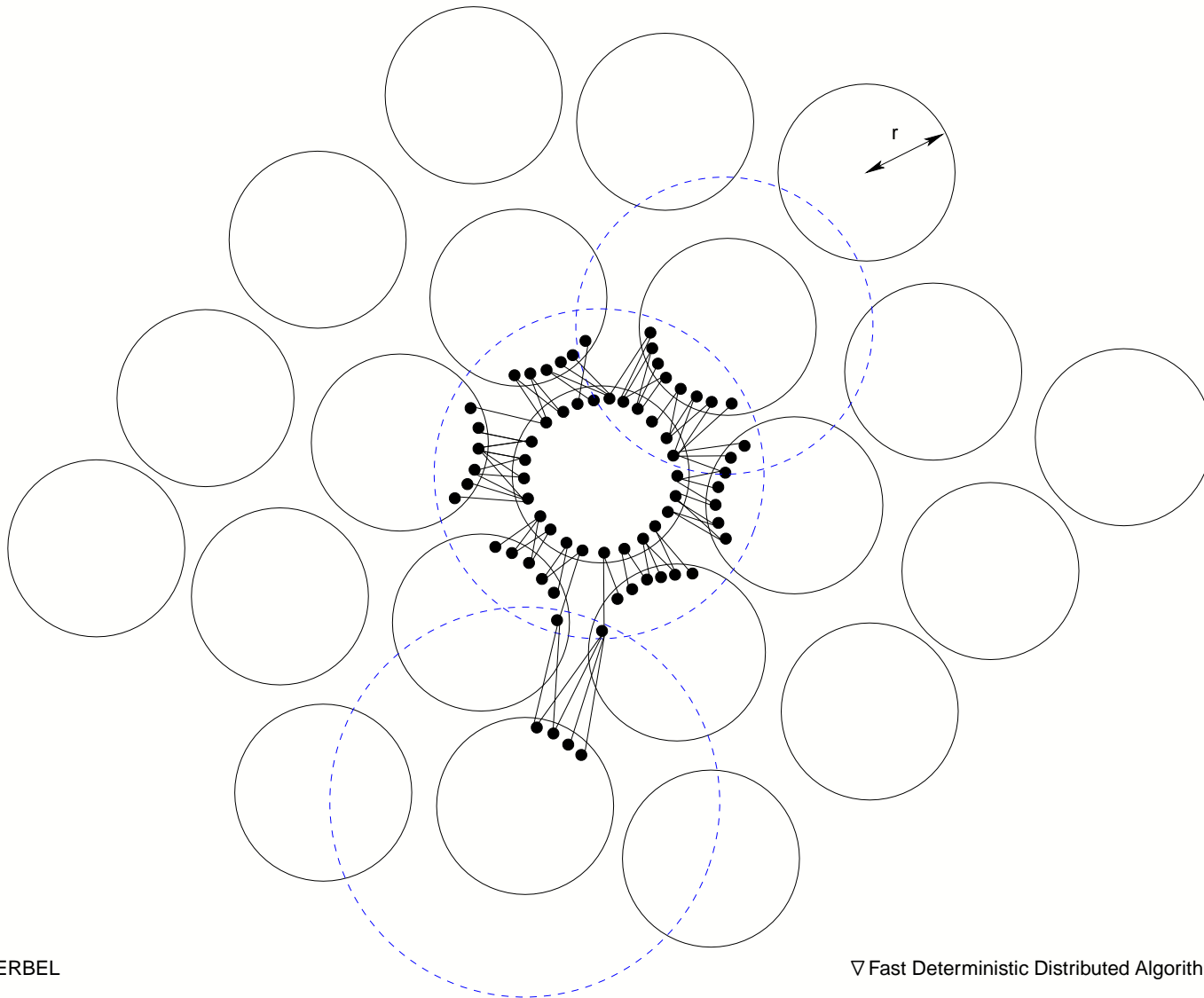
First strategy

- sparsity = the neighbourhood of a region in G



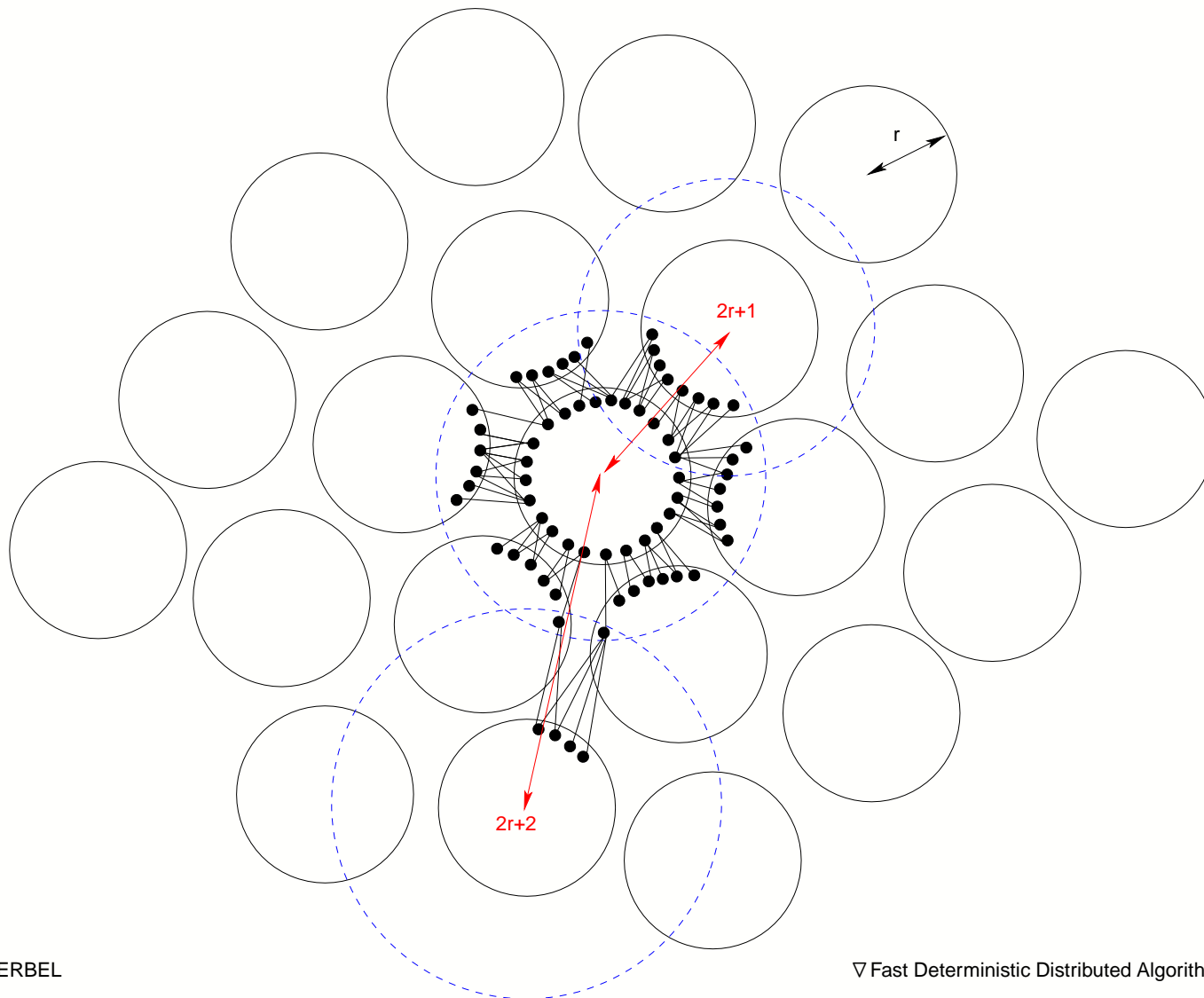
First strategy

- sparsity = the neighbourhood of a region in G



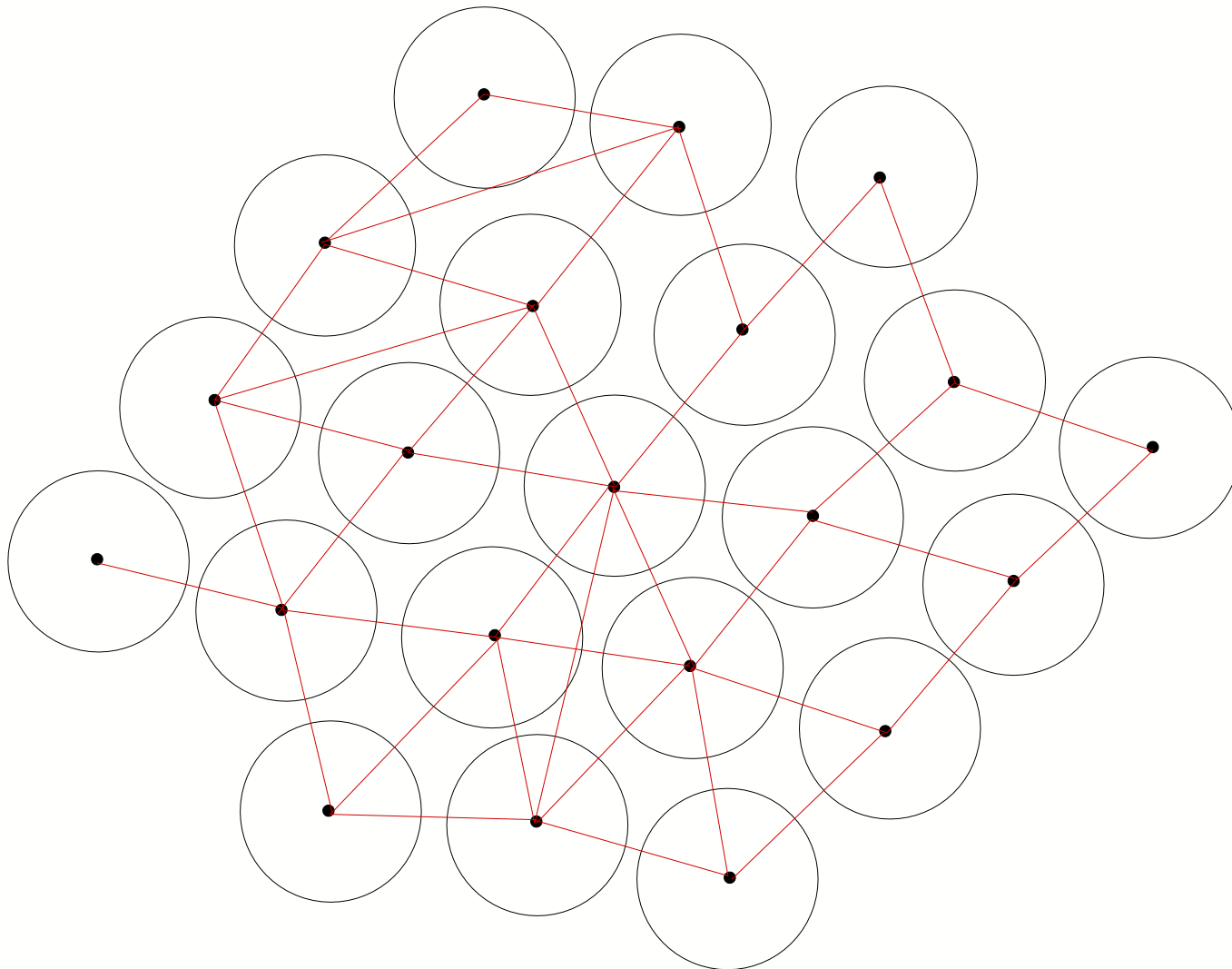
First strategy

- sparsity = the neighbourhood of a region in G



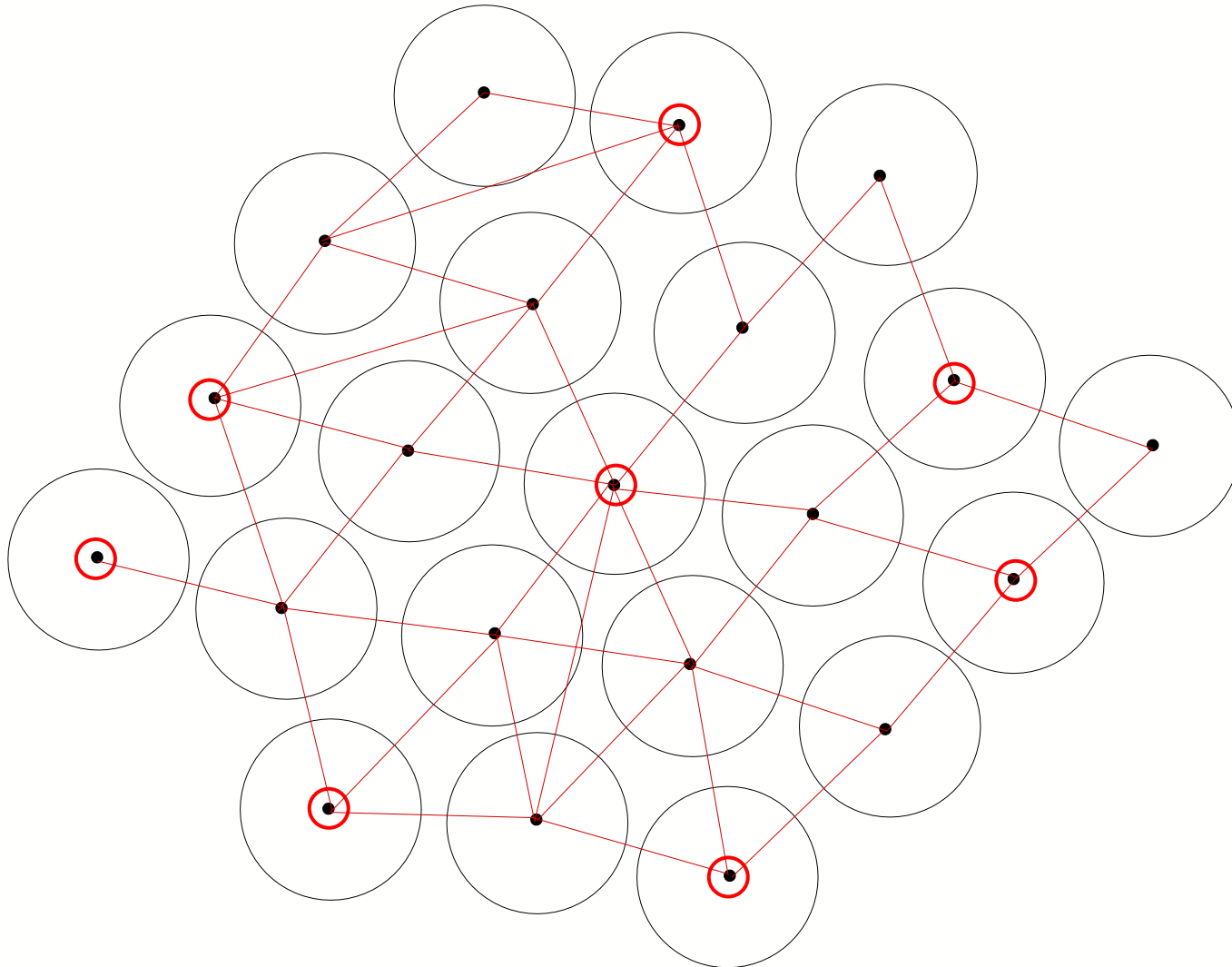
First strategy

- sparsity = the neighbourhood of a region in G



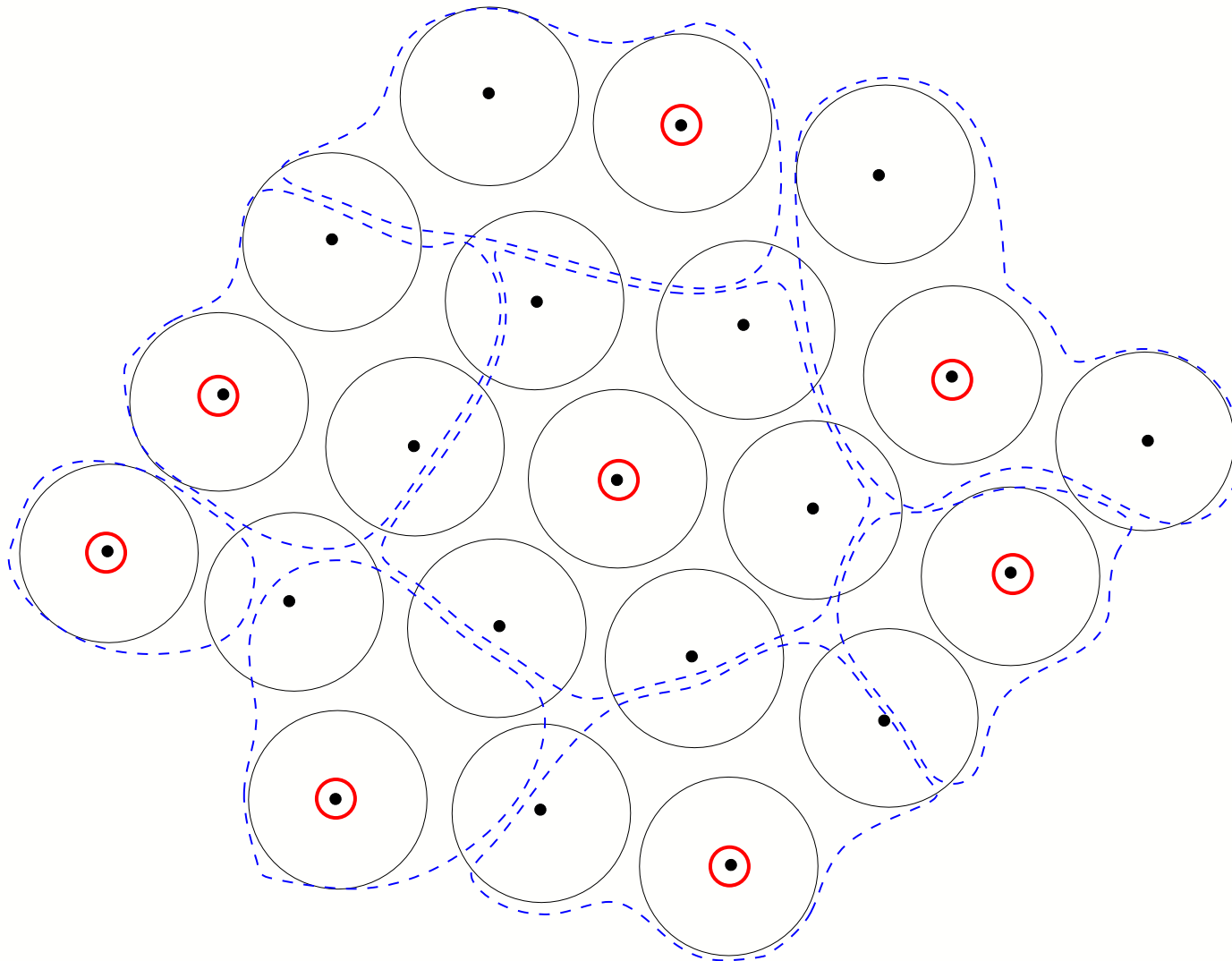
First strategy

- sparsity = the neighbourhood of a region in G



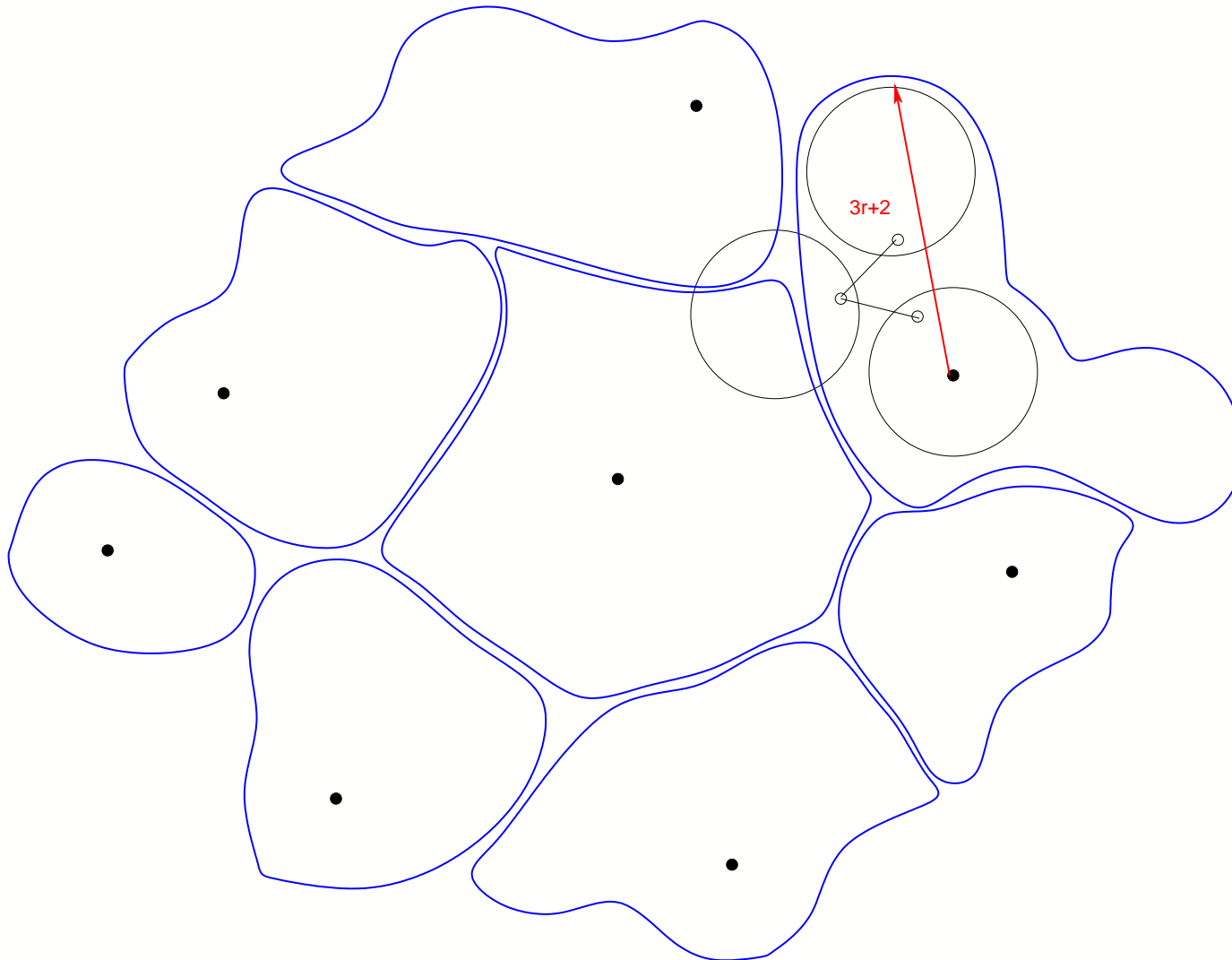
First strategy

- sparsity = the neighbourhood of a region in G



First strategy

- sparsity = the neighbourhood of a region in G



First strategy

- sparsity = the neighbourhood of a region in G

1. $L := \{v \in C, |R^+(v)| \leq n^{1/k} \cdot |R(v)|\}$ and $H := C \setminus L$;
2. $\forall (u, v) \in L \times V$ such that \exists edge e between $R(u)$ and v , $S := S \cup \{e\}$
3. $X := \text{IDS}(G^{2(r+1)}[H], \rho)$
4. $\forall z \in V$, if $d_G(z, X) \leq (2\rho + 1)r + 2\rho$, then set $c(z)$ to be its closest node of X , breaking ties with identities.
5. $\forall v \in X, R(v) := \{z \in V \mid c(z) = v\}$
6. $\forall v \in X, S := S \cup \text{BFS}(v, R(v))$
7. $C := X$ and $r := (2\rho + 1)r + 2\rho$

- $r := 3r + 2$
- $|R(v)| > n^{1/k} |R(v)|$
 - k steps
 - exponential stretch



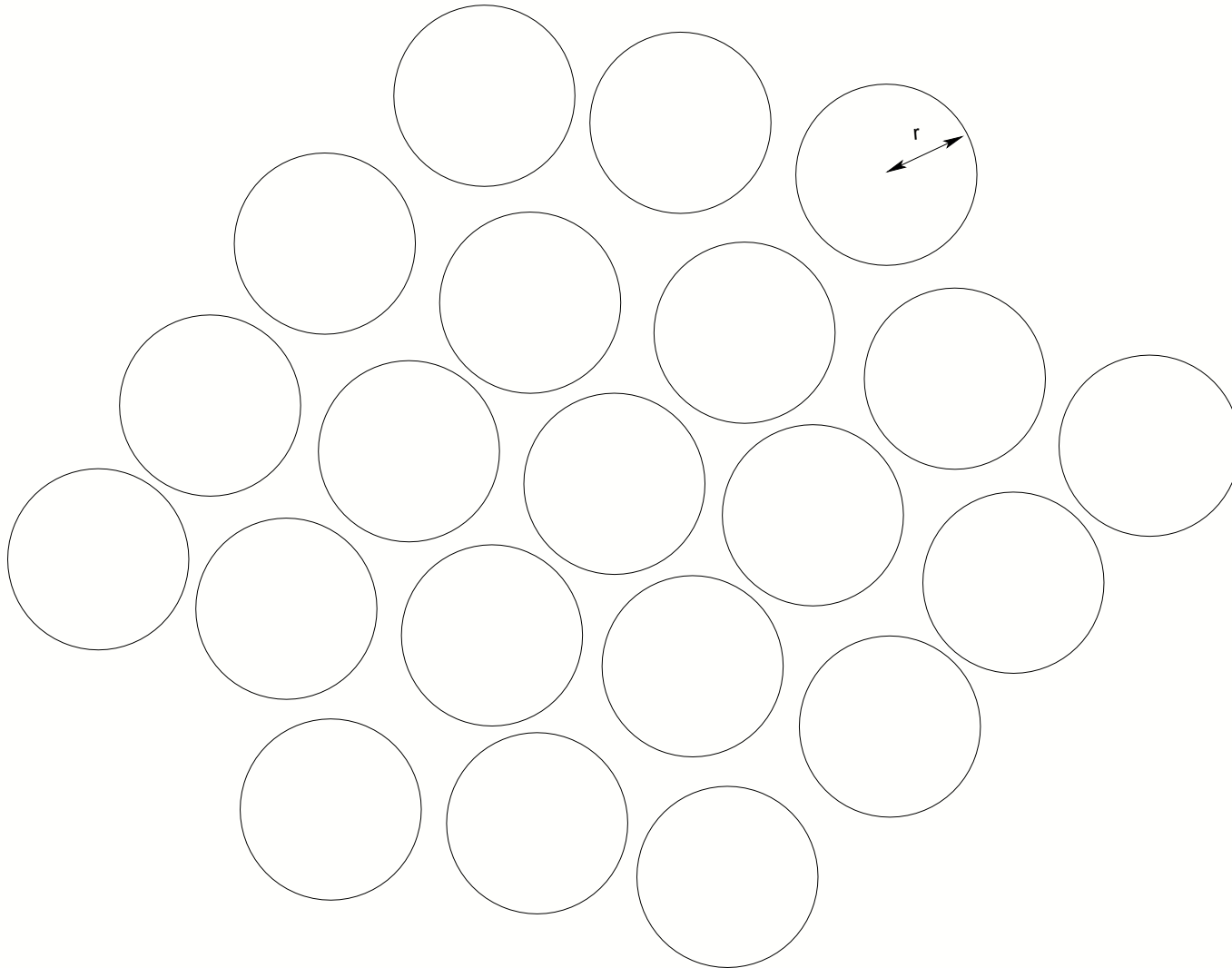
Second strategy

- sparsity = the number of neighbouring regions



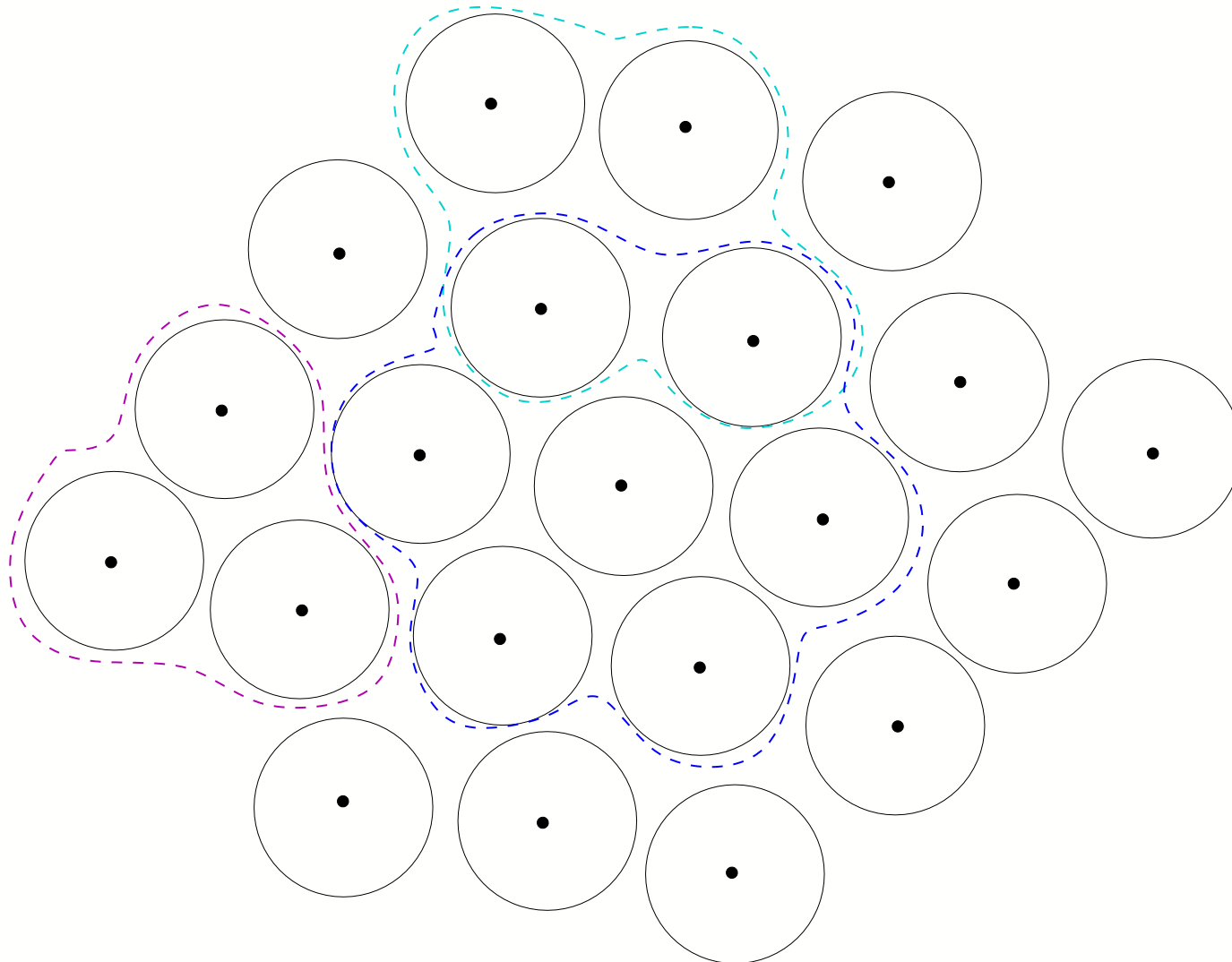
Second strategy

- sparsity = the number of neighbouring regions



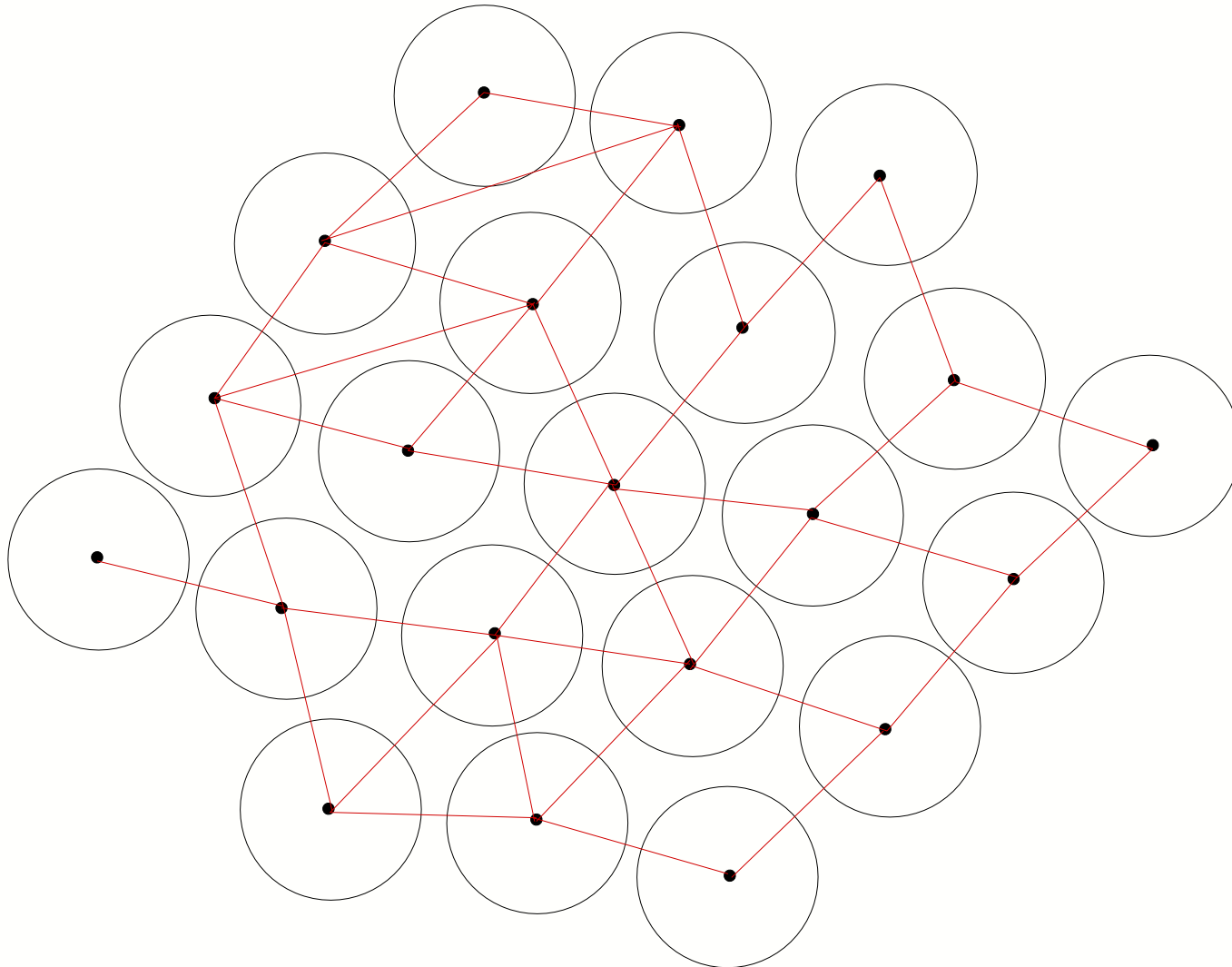
Second strategy

- sparsity = the number of neighbouring regions



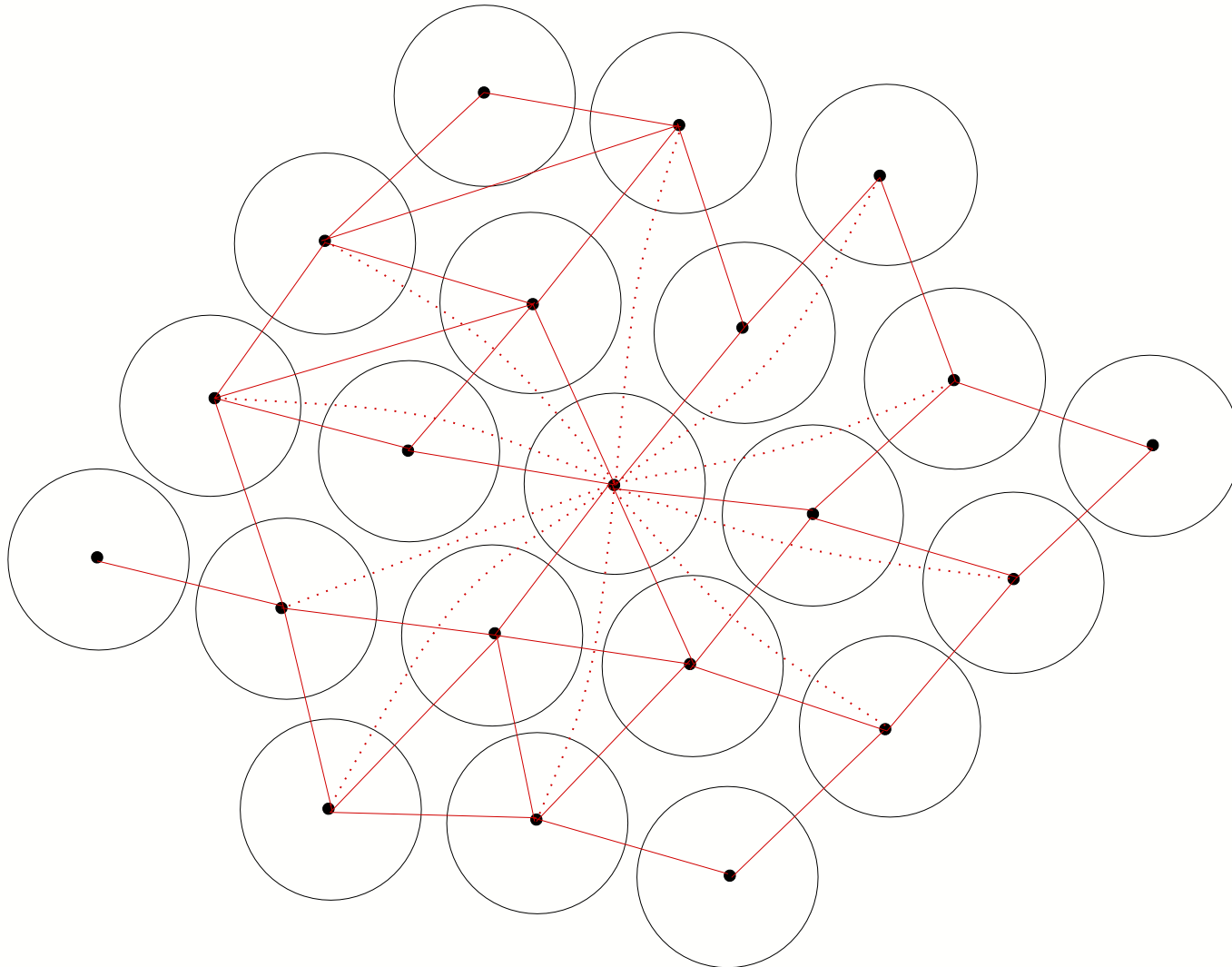
Second strategy

- sparsity = the number of neighbouring regions



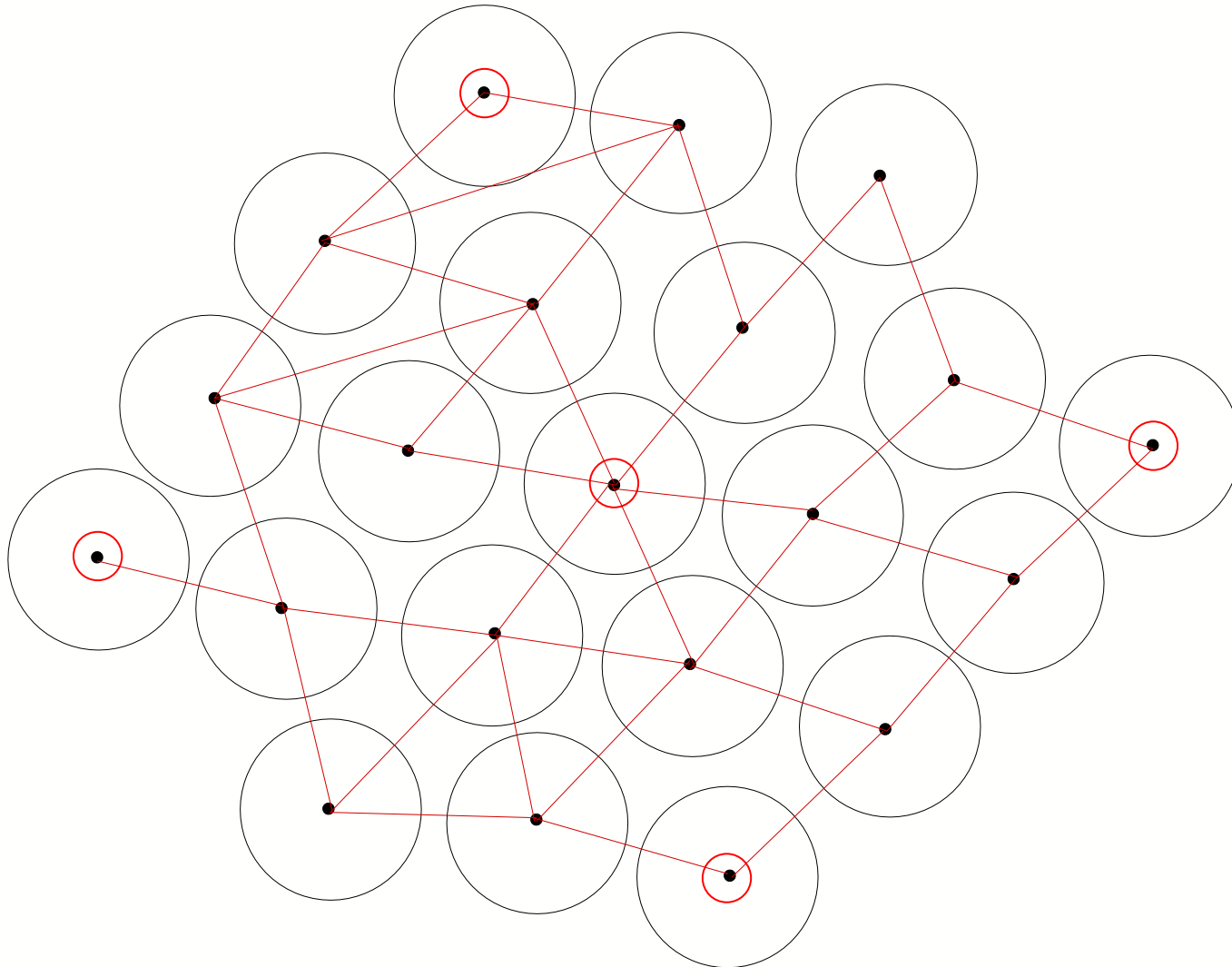
Second strategy

- sparsity = the number of neighbouring regions



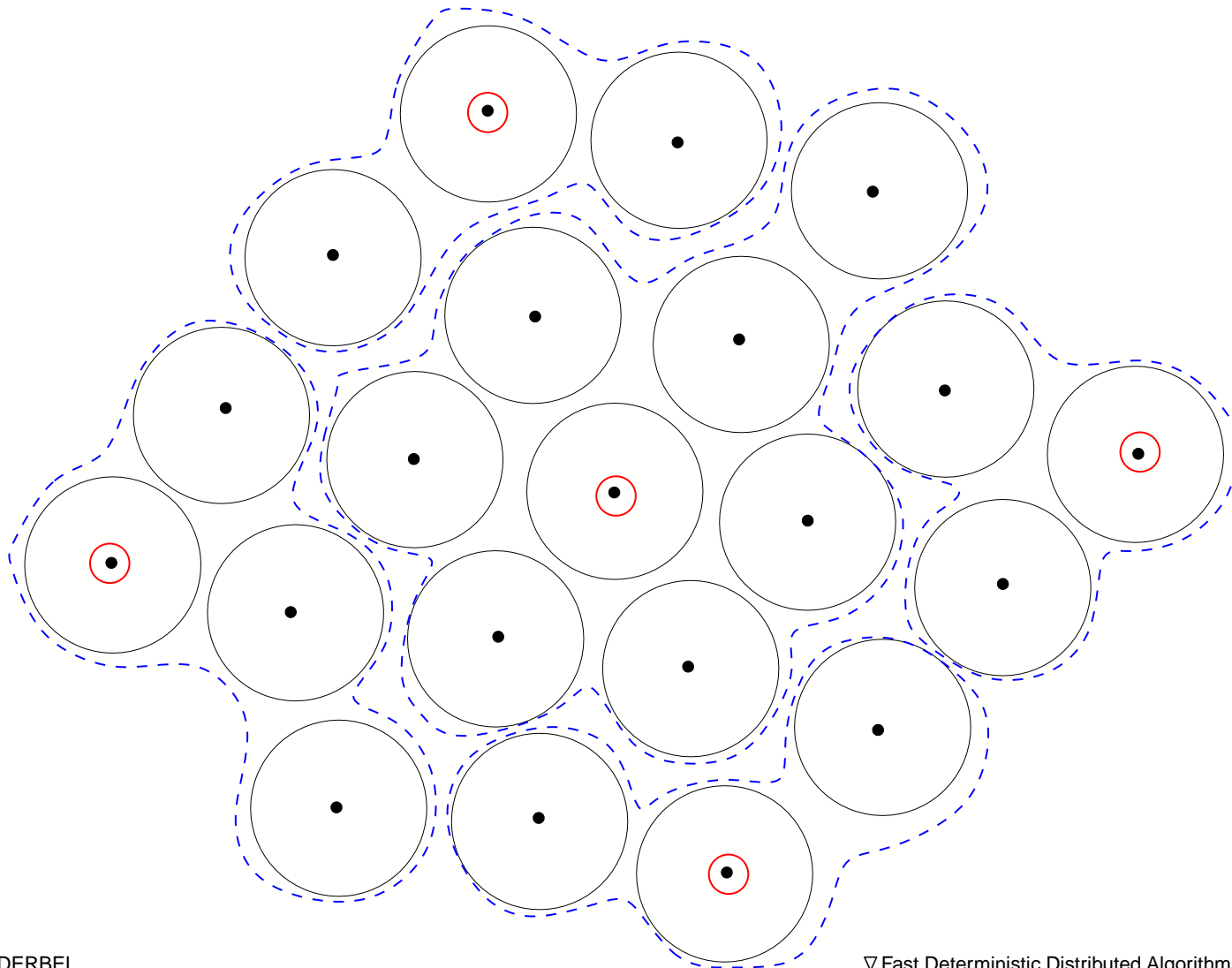
Second strategy

- sparsity = the number of neighbouring regions



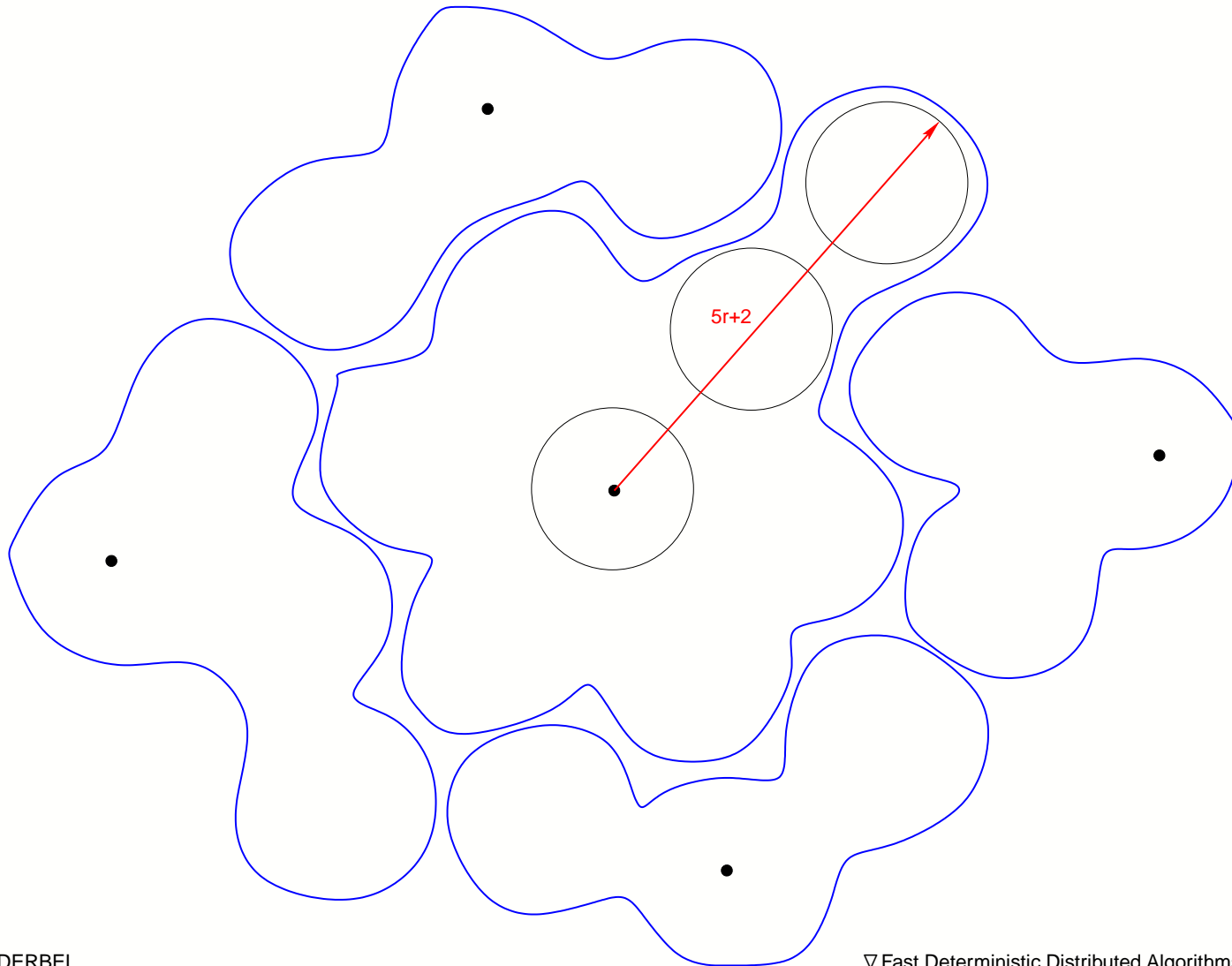
Second strategy

- sparsity = the number of neighbouring regions



Second strategy

- sparsity = the number of neighbouring regions



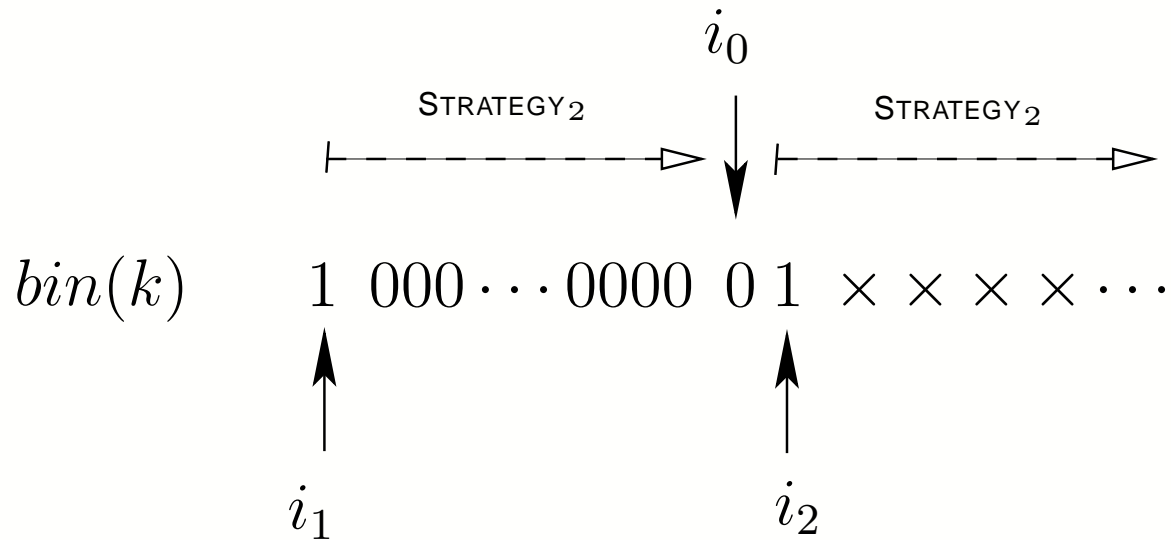
Second strategy

- sparsity = the number of neighbouring regions

1. $L := \{v \in C, |R_C^+(v)| \leq n^{1/k} \cdot |R(v)|\}$ and $H := C \setminus L$
2. $\forall (u, v) \in L \times C$ such that \exists edge e between $R(u)$ and $R(v)$, $S := S \cup \{e\}$
3. $X := \text{IDS}((G_C)^2[H], \rho)$
4. $\forall u \in C$, if $d_{G_C}(u, X) \leq 2\rho$, then set $c(u)$ to be its closest node of X in G_C , breaking ties with identities.
5. $\forall v \in X$, $R(v) := \{R(u) \mid u \in C \text{ and } c(u) = v\}$
6. $\forall v \in X$, $S := S \cup \text{BFS}(v, R(v))$
7. $C := X$ and $r := (4\rho + 1)r + 2\rho$

- $r := 5r + 2$
- $|R(v)| > n^{1/k} |R_{\min}|^2$
 - $\log(k)$ steps
 - polynomial stretch

General algorithm



- the radius increases essentially in a similar way in both strategies
- the size of a region increases very quickly with STRATEGY₂

$$\text{sparsity condition} = n^{2^{i_1} + 2^{i_1 - i_0 + 1} - 1}$$

Some Results

- if $k = 2^p \implies i_0 = \text{last iteration}$
 - stretch = $k^{\log 5} = k^{2.321\dots}$
 - size = $O(n^{1+1/k})$

Some Results

- if $k = 2^p \implies i_0 = \text{last iteration}$
 - stretch = $k^{\log 5} = k^{2.321\dots}$
 - size = $O(n^{1+1/k})$

- if $k = 2^p + 2^q - 1 \implies i_0 = p - q + 1$
 - stretch = $6 \cdot 5^{p-1} + 4 \cdot 5^{q-1} - 1$
 - size = $O(n^{1+1/k})$

Some Results

- if $k = 2^p \implies i_0 = \text{last iteration}$
 - stretch = $k^{\log 5} = k^{2.321\dots}$
 - size = $O(n^{1+1/k})$
- if $k = 2^p + 2^q - 1 \implies i_0 = p - q + 1$
 - stretch = $6 \cdot 5^{p-1} + 4 \cdot 5^{q-1} - 1$
 - size = $O(n^{1+1/k})$

k	1	2	3	4	5
$s[k]$	1	5	9	25	33

Distributed implementation

- *LOCAL* model
 - MIS in the logical graphs $G^{O(r)}$ and $(G_C)^2[H]$
 - $O(r \cdot \tau)$ time consuming with τ the time needed to construct a MIS in G
 - forming the new regions = $O(r)$ time consuming
 - constructing the BFS trees = $O(r)$ time consuming
 - connecting the regions = $O(r)$ time consuming
- Deterministic Implementation
 - $n^{O(1/\sqrt{\log n})}$ time consuming
- Randomised Implementation
 - $O(\log n)$ time consuming (*Las Vegas*)

Summary

	<i>Time</i>	<i>k</i>	2	3	4	5
our method	$O(n^\epsilon)$	$s[k]$	5	9	25	33
sparse decomposition	$O(n^{\epsilon+1/k})$	$4k - 2$	6	10	14	18
best possible	?	$2k - 1$	3	5	7	9

Graphs with high minimum degree δ

- Suppose we are given a ρ -dominating set X of G
 - we can construct a $(4\rho + 1)$ -spanner with $n + |X|^2/2$ edges in ρ time

Graphs with high minimum degree δ

- Suppose we are given a ρ -dominating set X of G
 - we can construct a $(4\rho + 1)$ -spanner with $n + |X|^2/2$ edges in ρ time
- If $\delta \geq \sqrt{n}$
 - we can construct a 9-spanner with $3n/2$ edges
 - in $O(MIS)$ deterministic time
 - in $O(\log n)$ expected time

Graphs with high minimum degree δ

- Suppose we are given a ρ -dominating set X of G
 - we can construct a $(4\rho + 1)$ -spanner with $n + |X|^2/2$ edges in ρ time
- If $\delta \geq \sqrt{n}$
 - we can construct a 9-spanner with $3n/2$ edges
 - in $O(MIS)$ deterministic time
 - in $O(\log n)$ expected time
 - we can construct a 5-spanner with $O(n \log^2 n)$ edges w.h.p in $O(\log n)$ time (*Monte Carlo*)
 - G has a dominating set with size $O(\sqrt{n \log n})$ (greedy technique)
 - we use the constant approximation algorithm of Khun et al **[KMW](SODA 06)**

Concluding remarks

- The challenge is to efficiently construct an independent ρ -dominating set.

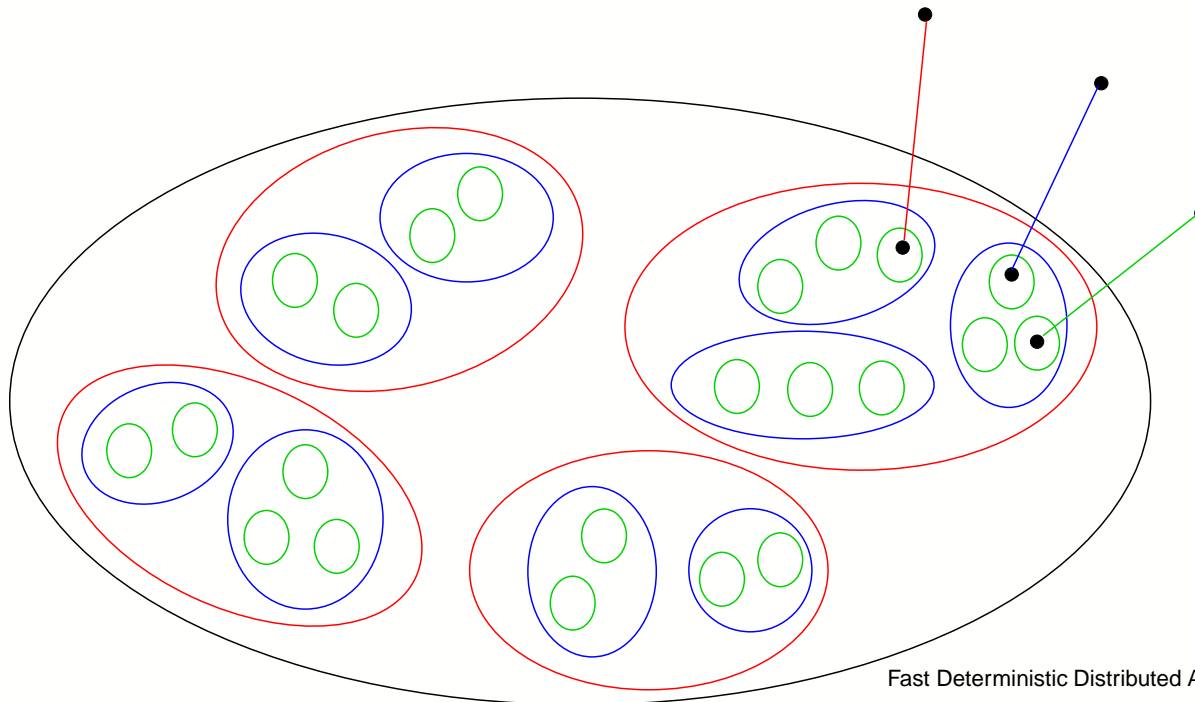
Concluding remarks

- The challenge is to efficiently construct an independent ρ -dominating set.
- Can we construct a 3-spanner using a MIS ?
 - a more careful analysis shows that :
 - if $d_G(u, v)$ is even $\implies d_H(u, v) \leq 3 \cdot d_G(u, v)$
 - if $d_G(u, v)$ is odd $\implies d_H(u, v) \leq 3 \cdot d_G(u, v) + 2$

Concluding remarks

- The challenge is to efficiently construct an independent ρ -dominating set.
- Can we construct a 3-spanner using a MIS ?
 - a more careful analysis shows that :
 - if $d_G(u, v)$ is even $\implies d_H(u, v) \leq 3 \cdot d_G(u, v)$
 - if $d_G(u, v)$ is odd $\implies d_H(u, v) \leq 3 \cdot d_G(u, v) + 2$

Idea



THE END