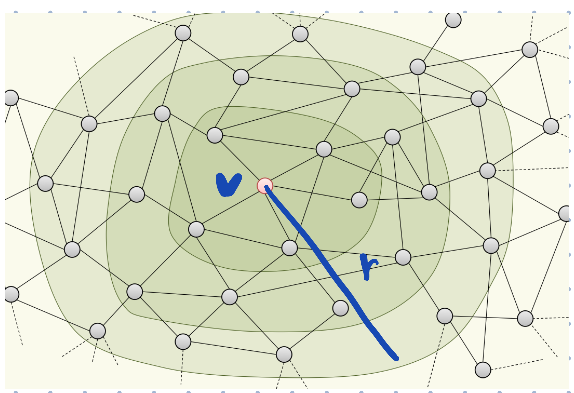
Distributed Graph Algorithms, Class 6

Recall two main challanges in LOCAL:

(1 **Locality**: determine output of a node based on local view

r-round: Function of r-ball.

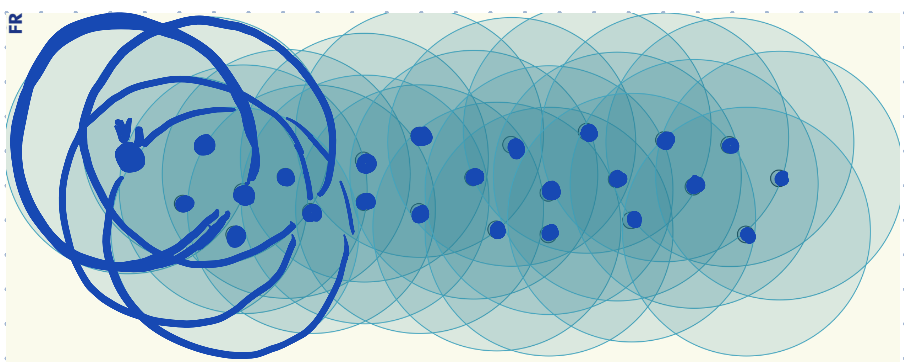


2) **Coordinaton / Symmetry breaking**
- Nodes make decisions in Parallel

Today: Focus only on 1st challange

SLOCAL: Sequential local Algorithms

- locality parameter $r = r(n)$
- Alg. go over all nodes in a seq. manner (arbitrary ordering): $v_1, \dots, v_n$
- output of node v_i depends on the output of $\underbrace{\{v_1, \dots, v_{i-1}\} \cap B_r(v_i)}$



$(\Delta+1)$-
<u>Example:</u> Coloring

SLOCAL generalizes the simple greed alg.

classical LOCAL problems

$(\Delta+1)$-coloring
$(2\Delta-1)$ edge coloring
MIS
Matching

SLOCAL $r = O(1)$

* Seq. alg. in which we only look at the nearby prior nodes.

Complexity Classes

PLOCAL:

Problems that can be solved deterministically in $\boxed{\text{poly-log}n}$ LOCAL rounds

P-RLOCAL: prob. that can be solved w.h.p by an $\boxed{\text{Poly-log}n}$ round randomized LOCAL alg.

P-SLOCAL:

problems that can be solved deterministically by an SLOCAL alg with locality of poly-logn

FACT:

① PLOCAL $\subseteq$ PSLOCAL (easy)

② PRLOCAL $\subseteq$ PSLOCAL [GHK'18]

$\boxed{\text{randomized}}$ r round LOCAL alg

⬇

$\boxed{\text{det.}}$ sequential alg. with loc. 2r

Main Theorem: [GKM '17 + RG '20]

$$P\text{-}SLOCAL \subseteq P\text{-}LOCAL$$

That is:

any sequential alg. with locality r

can be translated into $\boxed{\text{LOCAL}}$

alg. with locality $O(r \log^2 n)$

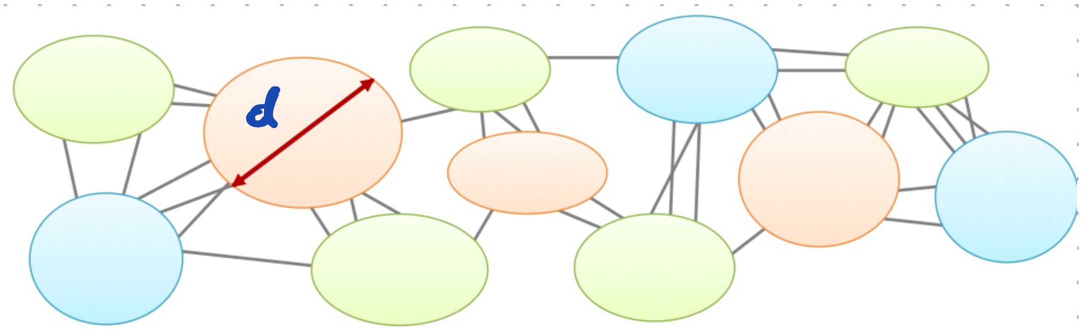
Putting it all together:

PRLOCAL $\subseteq$ PSLOCAL $\subseteq$ PLOCAL

Key Tool: Network Decomposition

(c,d) decomposition: $V_1, \ldots, V_c$

Each V_i made of non-neighboring clusters of weak/strong diam d



$(\Delta+1)$ coloring: $O(c \cdot d)$ rounds.

→ Seq. alg of locality 1 into $O(c \cdot d)$ LOCAL alg.

Goal: generalize to sequential alg. with locality r

$O(c \cdot d \cdot r)$

SLOCAL: Detailed Definition

* Each node v has an unbounded local memory S_v (local state)

* Initially S_v contains private input x_v (if exists)

* Alg processes nodes seq. in some order $\boxed{\pi = v_1, v_2, \ldots, v_n}$ provided to it

* The algorithm must work for $\boxed{\text{any}}$ ordering π

* Processing v_i:

① alg. views r-ball of v_i & local states of all $\{v_1 \dots v_{i-1}\} \cap B_r(v_i)$

② compute state s_{v_i}

unbounded local comp. & space

Useful observations:

Lemma 1:

Any SLOCAL alg $\mathcal{A}$ with locality $\boxed{R}$ in which each node v can write into local memory S_u of $\boxed{u \in B_R(v)}$

for $\boxed{r \leq R}$ can be transferred into SLOCAL(β) with locality $\boxed{r+R}$ in which v writes only in its own memory S_v.

Lemma 2: Multi-Phase SLOCAL

Any $\boxed{k}$-Phase SLOCAL alg. $\mathcal{A}$

with locality r_i in phase $\boxed{i=1,\ldots,k}$

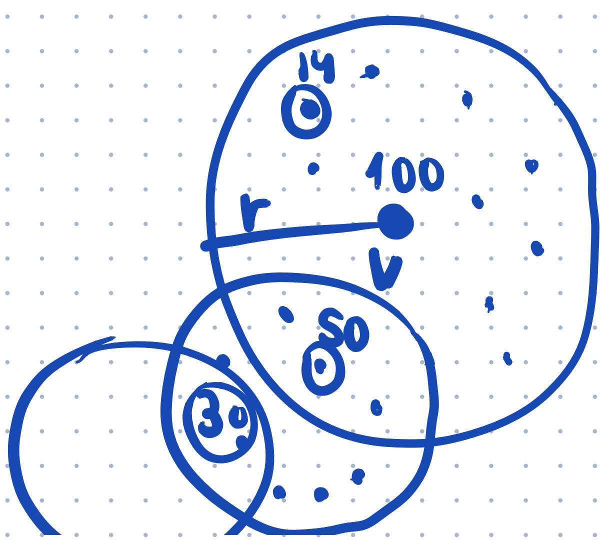
can be transformed into single-Phase

SLOCAL with locality $\boxed{O(\sum r_i)}$

Simulating SLOCAL alg w. locality r in the local model:

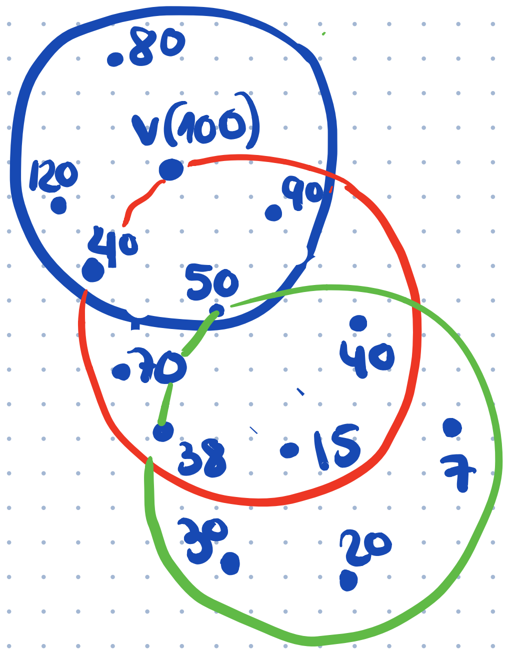
* Ordering $\boxed{\pi = v_1 \ldots v_n}$ given to SLOCAL alg.

* v_i can compute output as long as all nodes $\boxed{\{v_1, \ldots, v_{i-1}\} \cap B_r(v_i)}$ computed their output.



Dependency Chains

SLOCAL alg.
with locality r



max length of dep. chain T

→ LOCAL alg. with $O(T r)$ rounds.

Complete graphs: $\boxed{T = \Omega(n)!}$

Note:

In LOCAL model sufficient to ensure that any dep. chain is contained in a small ball!

Notation [Power graph]:

G^r: $V(G^r) = V(G)$

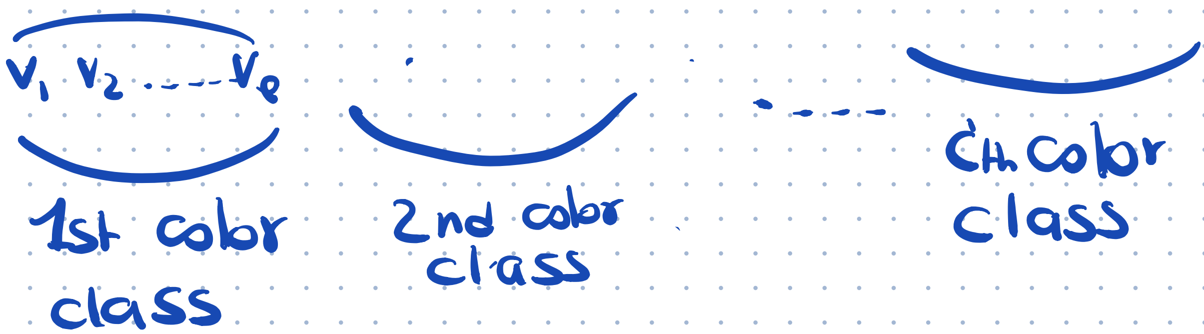
$E(G^r) = \{(u,v) \mid d_G(u,v) \leq r\}$

From SLOCAL alg $\mathcal{A}$ with loc. r

$\Rightarrow$ LOCAL alg $\mathcal{B}$ with $r \cdot \text{poly-log} n$

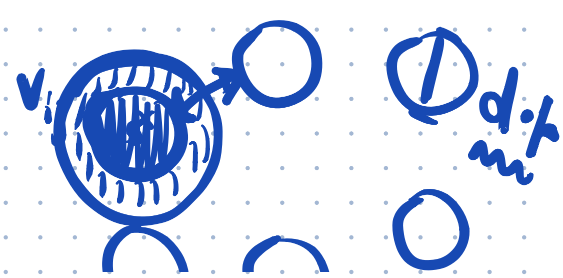
Alg B:

- Compute (c, d) ND of G^{2r}
- Let π be defined by



- Iterate over color classes

$V_1 \ldots \ldots V_c$



$\cup\ \cup$

- $\boxed{\text{Processing } V_i\text{:}}$

1. Collect $\boxed{B_r(V_{ij})}$ for each <u>cluster</u> $V_{ij} \subseteq V_i$

2. Simulate $\boxed{\mathcal{A}_\Pi}$ on V_{ij}

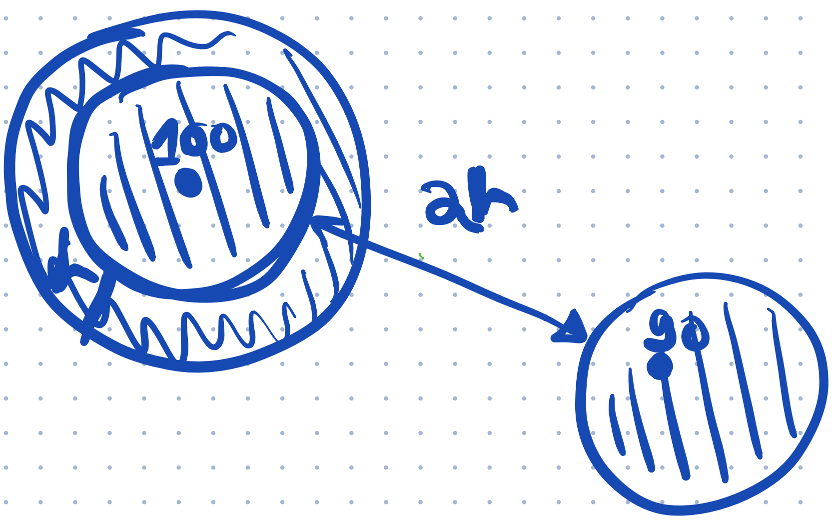
3. Send states V_{ij} to neighbors in G^{2r}

<u>Round complexity</u>: ?

$$\tilde{O}(c \cdot d \cdot r)$$

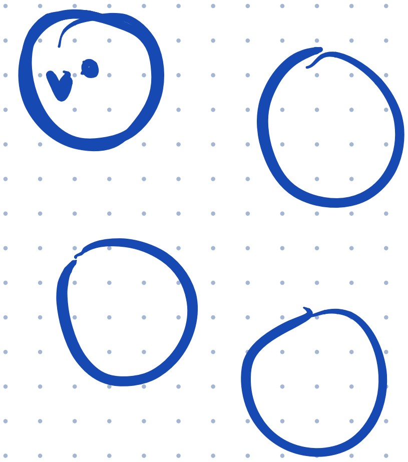
Analysis

1 color class



v_{ij}

Color class V_i :



* node $v \in V_{ij}$

* nodes u in $B_r(v)$ s.t

$u \leq_\pi v$:

① nodes $u < v$ in class $j < i$

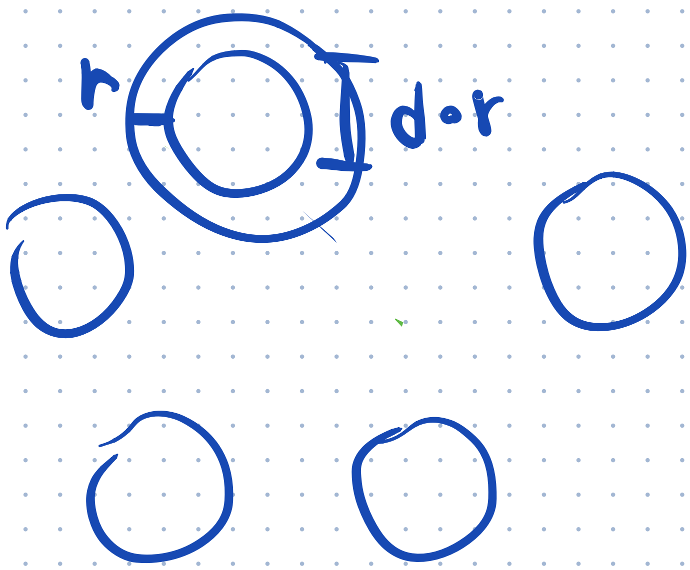
② nodes $u < v$ in class i

$\Rightarrow$ only care for nodes in V_{ij}

Lemma: Given a $\boxed{(c,d)\text{ ND}}$ of G^{2r} can simulate any SLOCAL alg. with locality r within

$O(r \cdot c \cdot d)$ rounds.

c color classes



Approximating Covering & Packing Integer Linear Prog.

* $(1+\varepsilon)$ approx of MaxIS

* $(1+\varepsilon)$ approx min Dominating Sets

Show:

SLOCAL alg. with locality $\boxed{O\left(\frac{\log n}{\varepsilon}\right)}$

$\Rightarrow$ LOCAL alg. with $\mathrm{Poly}\left(\frac{\log n}{\varepsilon}\right)$ rounds.

$(1+\varepsilon)$ approx. max-IS:

SLOCAL alg. with locality $\boxed{R = O\left(\frac{\log n}{\varepsilon}\right)}$

$$\pi = (v_1, \dots\dots v_n)$$

Define: $V' \subseteq V$

$\alpha(V')$: indep. number [redacted] of $G(V')$.

IDEA

Ball growing argument

SLOCAL alg

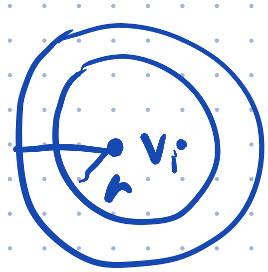
1) $S \leftarrow \phi$

2) For $v_1 \ldots\ldots v_n$:

* Let r be s.t

$$\alpha(B_{r+1}(v_i)) \leq (1+\varepsilon)\,\alpha(B_r(v_i))$$

* Add max-IS in $B_r(v_i)$ to S
* Omit $B_{r+1}(v_i)$ from G



Locality arg: $R = O\left(\frac{\log n}{\varepsilon}\right)$

Approx. argument

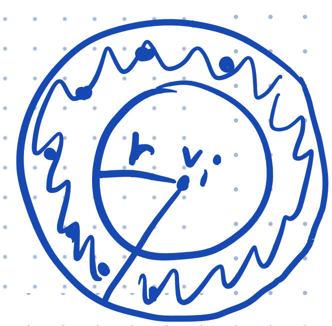
OPTIMAL IS I^*

$$\boxed{I_1 \mathbin{\dot\cup} I_2 \mathbin{\dot\cup} \dots \mathbin{\dot\cup} I_n \subseteq I^*}$$

I_i = set of IS nodes removed when processing v_i

S_i = set of IS nodes taken

$$|S_i| \geq \frac{I_i}{(1+\varepsilon)}$$

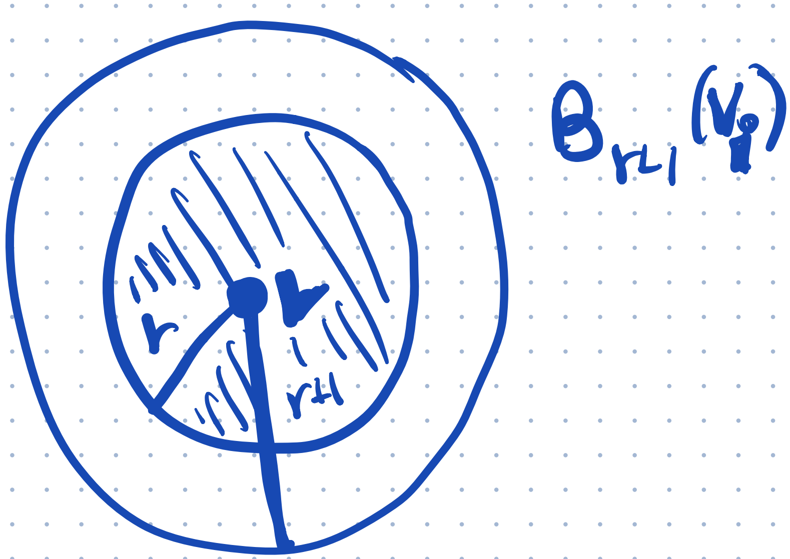


$(1+\varepsilon)$ Approximation of Min DS

SLOCAL alg with $r = O\left(\frac{\log n}{\varepsilon}\right)$

Define:

$g(v,r) :=$ size of smallest set in $B_{r+1}(v)$ that dominates $B_r(v)$.



For node v, define r s.t

$$g(v, r+2) \leq (1+\varepsilon) g(v, r)$$

- compute minimal set of vertices in $\boxed{B_{r+3}(v)}$ that dominates $\boxed{B_{r+2}(v)}$, add to D

- Omit all vertices in $\boxed{B_{r+2}(v)}$

Central ball of v: $B_r(v)$

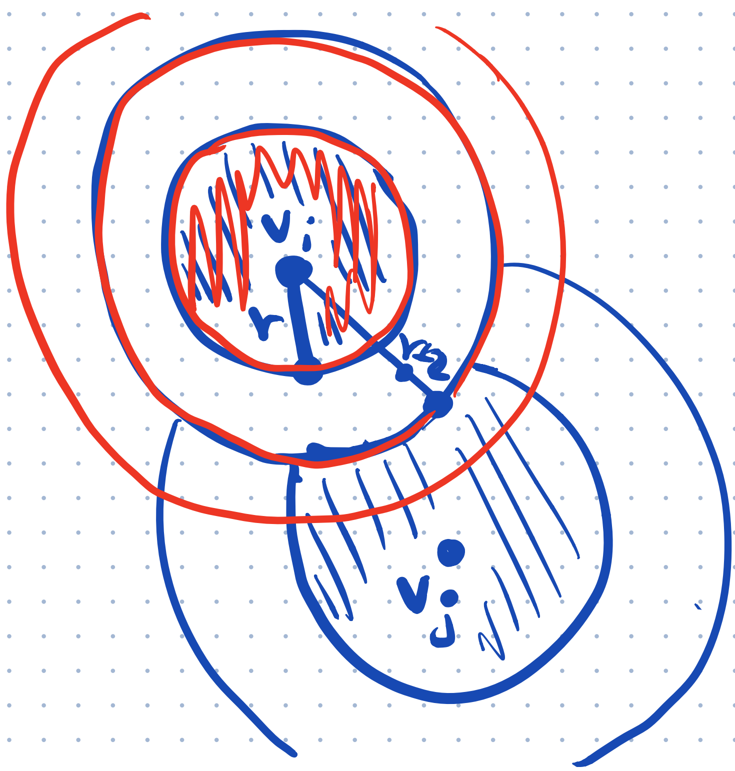
Approximation argument:

Define:

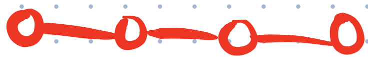
$v_1, \ldots\ldots, v_n$

central ball

$V_i :=$ set of nodes in $B_r(v_i)$ when processing v_i



$d_G(u,v) \geq 3$, $\forall\, u \in V_i$
$v \in V_j$



→ No node can dominate nodes from ≥ 2 V_i sets

→ OPTIMAL D^*

$$\boxed{D^* \supseteq D_1^* \cup D_2^* \cup \ldots \cup D_n^*}$$

D_i^* = set of nodes in D^* dominating V_i

D_i = set of nodes added in step i (processing V_i)

$\Rightarrow \quad |D_i^o| \leq (1+\varepsilon)\, |D_i^{*}|$

Recall:

For node v, define r s.t

$$\boxed{g(v, r+2) \leq (1+\varepsilon)\, g(v, r)}$$