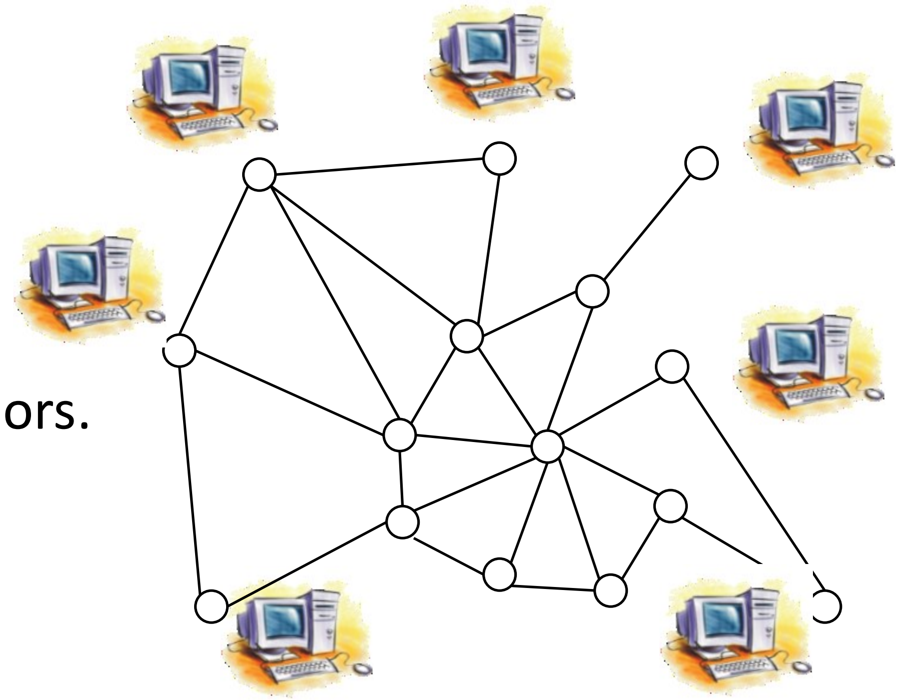
LOCAL Algorithms:
The Chasm Between Deterministic & Randomized

Mohsen Ghaffari
ETH Zurich

The LOCAL Model of Distributed Graph Algorithms

LOCAL model [Linial FOCS'87**]**

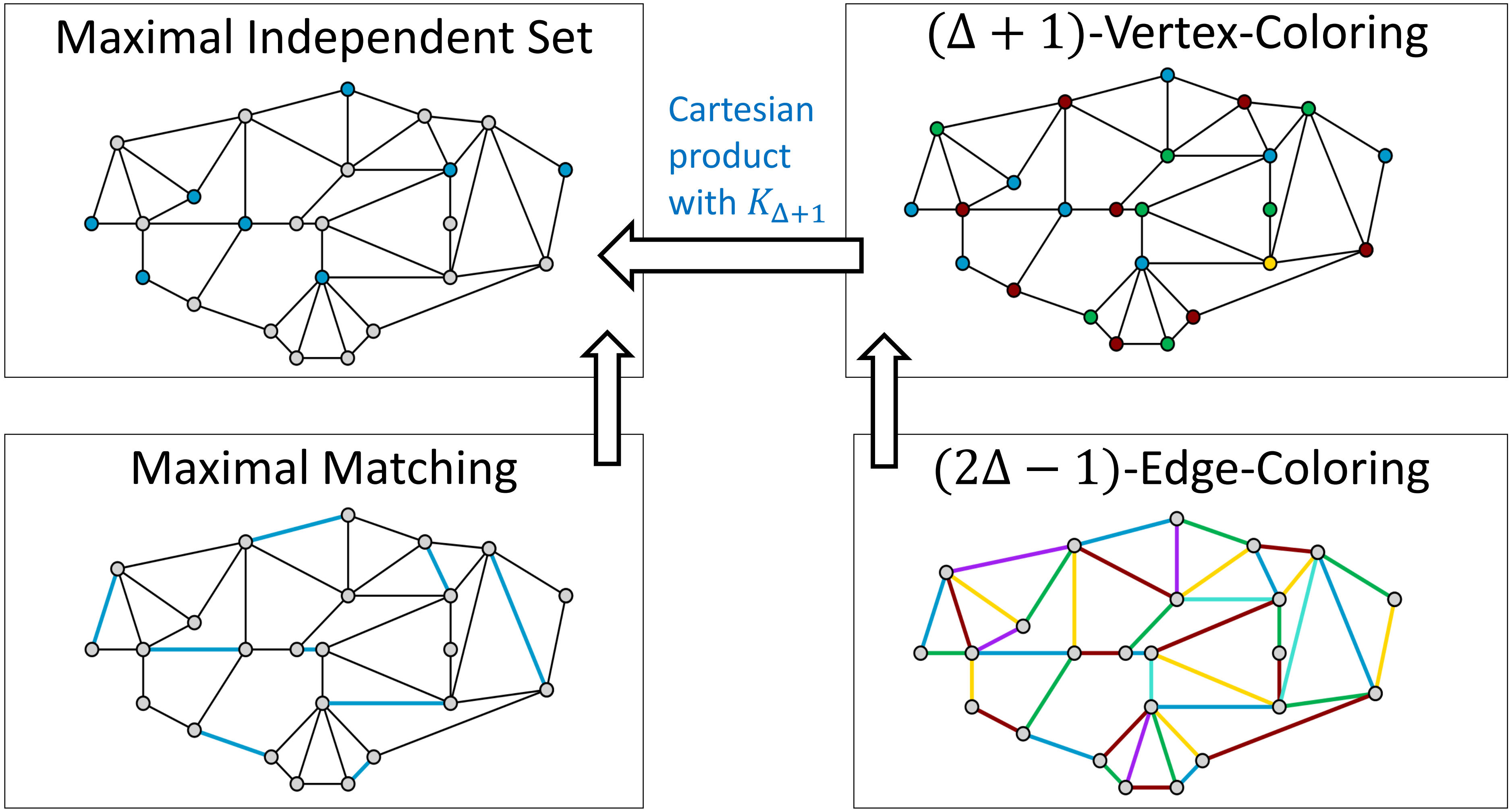
- undirected graph $G = (V, E)$, n nodes, max degree Δ
- one computer on each graph node,
- Synchronous message-passing rounds 1, 2, 3, …
  per round, each node sends one message to its neighbors.
- Unbounded message size & computation.



- Initially nodes do not know the topology
- Each node should learn its own part of the output, e.g., its color.

Time-Complexity: number of rounds until all nodes are done.

Four Classic Problems (since 1980's)

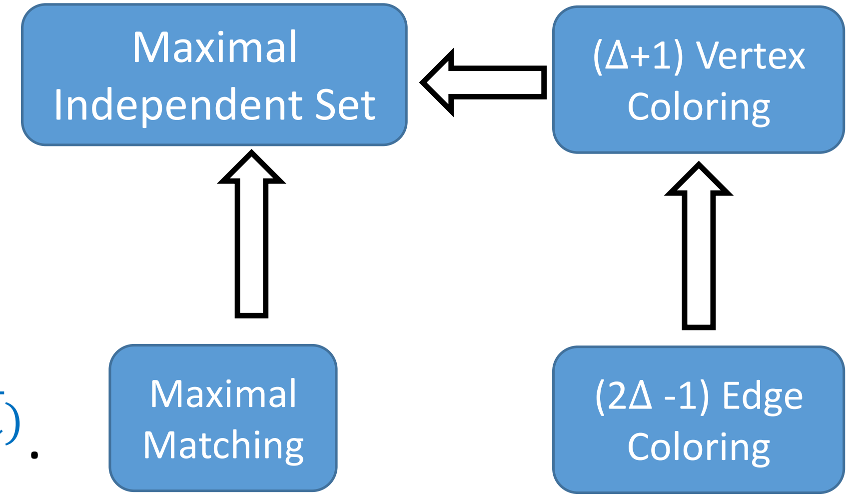


State of the Art

Randomized vs. Deterministic LOCAL Algorithms

First-Order Summary: Significant gap between Randomized & Deterministic

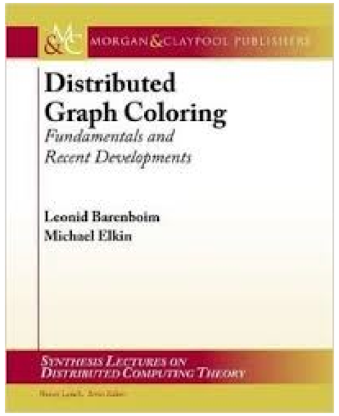
- **Randomized**:
  Very simple O(log n)-round algorithms,
  and even some o(log n)-round algorithms
- **Deterministic**: polylog n is the dream, for most problems, the the best known is $2^{O(\sqrt{\log n})}$.



Linial's Open Question [FOCS'87, SICOMP'92]:

"Can it [MIS] always be found [deterministically] in polylogarithmic time?"

Also the first 5 open problems of the Distributed Graph Coloring book [Barenboim & Elkin]



Maximal Independent Set (MIS): Current State

Lower Bound

- $\Omega\left(\min\{\frac{\log \Delta}{\log\log \Delta}, \sqrt{\frac{\log n}{\log \log n}}\}\right)$ rounds needed [Kuhn, Moscibroda, Wattenhofer PODC'04]

Efficient Randomized Algorithms:

- An $O(\log^4 n)$-time algorithm [Karp, Wigderson STOC'84]
- A truly-simple $O(\log n)$-time algorithm [Luby STOC'85; Alon, Isreali, Itai JALG'86]
- Best upper bound: $O(\log \Delta) + 2^{O\left(\sqrt{\log \log n}\right)}$ [G., SODA'16]

Best Known Deterministic Algorithm

- Based on network decomposition: $2^{O\left(\sqrt{\log n}\right)}$ [Panconesi, Srinivasan STOC'92]

$(\boldsymbol{\Delta} + \mathbf{1})$-Vertex-Coloring: Current State

Lower Bound

- $\Omega(\log^* n)$ rounds needed even on the ring [Linial FOCS'87]

Efficient Randomized Algorithms

- Simple randomized $O(\log n)$-time algorithms [Luby STOC'86; Alon, Isreali, Itai JALG'86]
- Best current upper bound: $O\left(\sqrt{\log \Delta}\right) + 2^{O\left(\sqrt{\log \log n}\right)}$ [Harris, Schneider, Su STOC'16]

Best Known Deterministic Algorithm

- Based on network decomposition: $2^{O\left(\sqrt{\log n}\right)}$ [Panconesi, Srinivasan STOC'92]

State of the Art: Deterministic vs. Randomized

$2^{O(\sqrt{\log n})}$
[Panconesi, Srinivasan STOC'92]

$2^{O(\sqrt{\log n})}$
[Panconesi, Srinivasan STOC'92]

$O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$
[G., SODA'16]

$O(\sqrt{\log \Delta}) + 2^{O(\sqrt{\log \log n})}$
Harris, Schneider, Su [STOC'16]

In

- In RANDOMIZED, the n-dependencies come from deterministically solving polylog n-size graphs.
- All are necessary, unless we improve the deterministic algorithms.
  Thm [Chang, Kopelowitz, & Pettie FOCS'16]:

$$\text{RandomizedComplexity}(n) \geq \text{DeterministicComplexity}(\sqrt{\log n})$$

$O(\log^4 n)$
[Hanckowiak, Karonski, Panconesi SODA'98 & PODC'99]

$2^{O(\sqrt{\log n})}$
[Panconesi, Srinivasan STOC'92]

$O(\log \Delta) + O(\log^4 \log n)$
Barenboim, Elkin, Pettie, & Schneider [FOCS'11]

$O(\sqrt{\log \Delta}) + 2^{O(\sqrt{\log \log n})}$
Harris, Schneider, Su [STOC'16]

DETERMINISTIC

RANDOMIZED

Some Other Related Work

Exponential Separations

[Chang, Kopelowitz, Pettie FOCS‘16], [Brandt et al. STOC’16], [G., Su SODA’17]

If we do not ignore log-factors, there is an exponential separation between randomized and deterministic LOCAL algorithm.

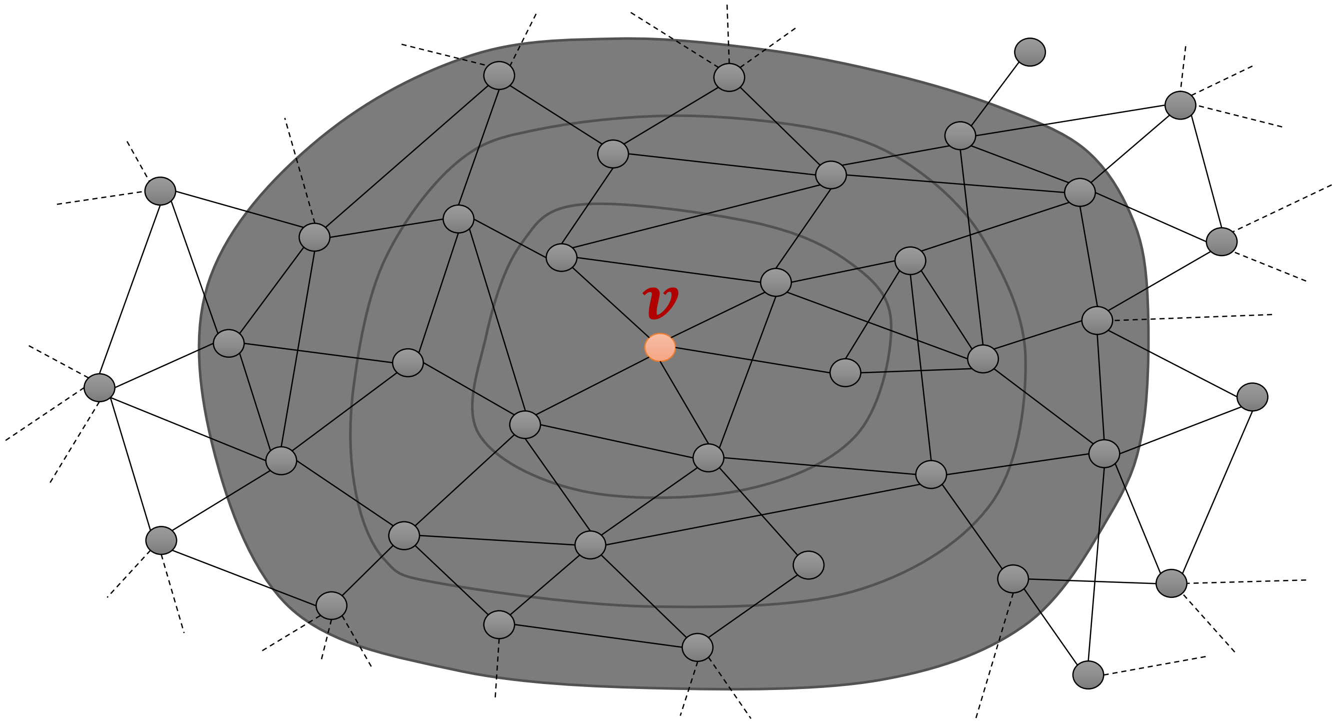
– *Δ-coloring trees* has randomized round complexity $\Theta(\log\log n)$
and deterministic round complexity $\Theta(\log n)$

– *Sinkless-orientation* has randomized round complexity $\Theta(\log\log n)$
and deterministic round complexity $\Theta(\log n)$

Challenges in the LOCAL Model

Challenges in the LOCAL Model

(1) Locality:



In any r-round Algorithm, each node computes its output as a function of the initial state of its r-neighborhood.

Challenges in the LOCAL Model

(2) Local Coordination / Symmetry Breaking

- Nearby (symmetric) nodes need to output different values
  - E.g., different colors
- Nodes decide in parallel based on their r-neighborhoods
- Need local coordination among nearby nodes

✓ Randomization naturally helps

- E.g., choose random color, keep if no conflict with neighbors

SLOCAL: A sequential variant of the LOCAL Model

Sequential LOCAL Model

SLOCAL Model

- locality $r(n)$
- sequentially go over all nodes $v_1, v_2, \dots, v_n$ (an arbitrary given order)
- compute output of each node based on the current state of its $r(n)$-neighborhood

SLOCAL Model

The SLOCAL model is much more powerful than LOCAL model

- $(\Delta + 1)$-coloring and MIS can easily be solved with locality 1
  - The sequential greedy algorithm is an SLOCAL-algorithm.
  - The output a node v only depends on the outputs of neighbors that were processed before v.
- SLOCAL is a generalization of sequential greedy algorithms
  - if for each node, one only looks at previous nearby nodes

Complexity Classes

LOCAL$(\boldsymbol{t(n)})$: graph problems that can be solved deterministically in $t(n)$ rounds in the LOCAL model

SLOCAL$(\boldsymbol{t(n)})$**:** graph problems that can be solved deterministically with locality $t(n)$ in the SLOCAL model,

- e.g., MIS, $(\Delta + 1)$-coloring $\in$ SLOCAL(1)

P-LOCAL $:=$ **LOCAL**$(\mathbf{poly}\log \boldsymbol{n})$

P-SLOCAL $:=$ **SLOCAL**$(\mathbf{poly}\log \boldsymbol{n})$

Randomized classes:
RLOCAL, RSLOCAL, P-RLOCAL, P-RSLOCAL

Relations between the Complexity Classes

Basic: $\text{LOCAL}(t(n)) \subseteq \text{SLOCAL}(t(n))$, P-LOCAL $\subseteq$ P-SLOCAL

Fact 1: P-SLOCAL $\subseteq$ P-RLOCAL

– randomized $\text{poly}\log n$-round distributed alg. for all problems in P-SLOCAL

Fact 2: P-SLOCAL $\subseteq$ $\text{LOCAL}\left(2^{O\left(\sqrt{\log n}\right)}\right)$

– deterministic $2^{O\left(\sqrt{\log n}\right)}$-round distributed alg. for all problems in P-SLOCAL

Proofs via
Network Decompositions

Open Problem: P-LOCAL $\stackrel{?}{=}$ P-SLOCAL

P-SLOCAL Completeness

Problems in P-SLOCAL that if proven to be in P-LOCAL, imply **P-SLOCAL = P-LOCAL**.

P-SLOCAL Completeness

Local Reduction: We say that a distr. graph problem P_1 is polylog n-reducible to P_2 if a deterministic $\text{poly}\log n$-round distr. algorithm for P_2 implies a deterministic $\text{poly}\log n$-round distr. algorithm for P_1.

P-SLOCAL Completeness: A problem P in P-SLOCAL is called P-SLOCAL-complete if every problem P' in P-SLOCAL is polylog n-reducible to P

Example: $\boldsymbol{(O(\log n), O(\log n))}$-decomposition is P-SLOCAL-complete

- $(O(\log n), O(\log n))$-decomp is in SLOCAL$\left(O(\log^2 n)\right)$
- polylog n round decomposition alg. $\Longrightarrow$ polylog n round P-SLOCAL alg.

Local Splitting: A Simple Yet Complete Problem

λ-Local Splitting for $\lambda \in (0, \frac{1}{2})$:

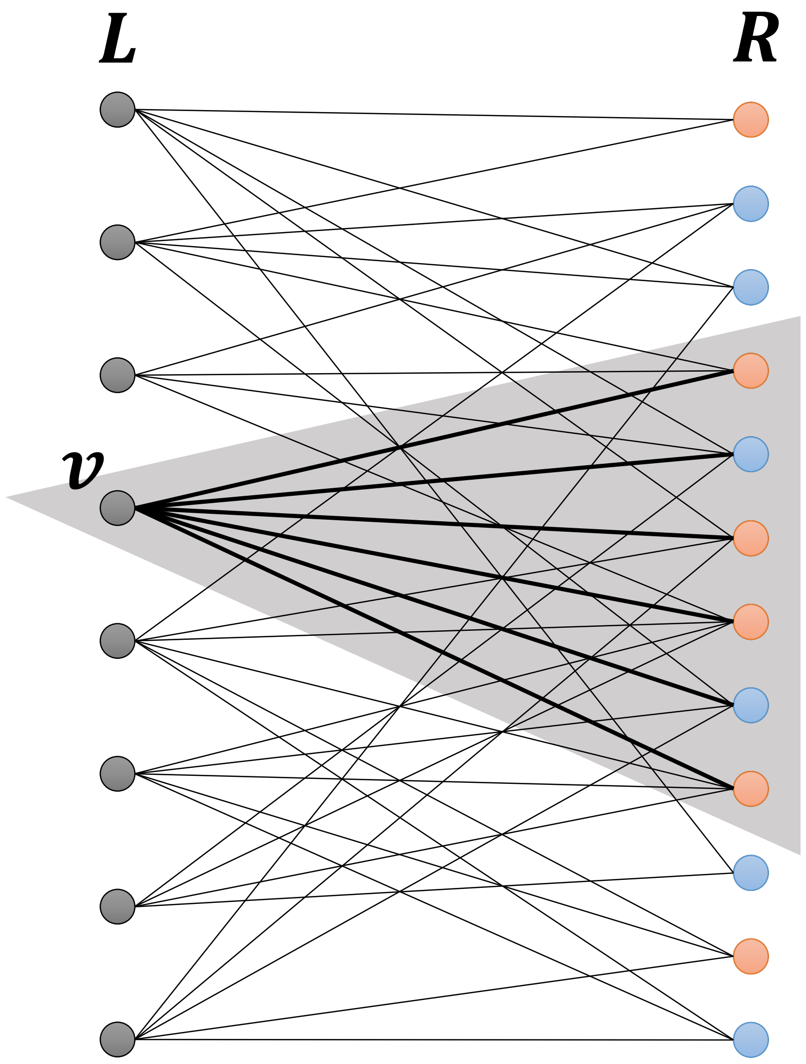
– Color R red/blue s.t. each $v \in L$ with $\deg(v)=\Omega(\log n)$ has at least $\lfloor \lambda \deg(v) \rfloor$ neighbors in each color.

Weak Local Splitting:

– Every $v \in L$ with $\deg(v)=\Omega(\log n)$ has at least one neighbor in each color.

Trivial Randomized Solution:

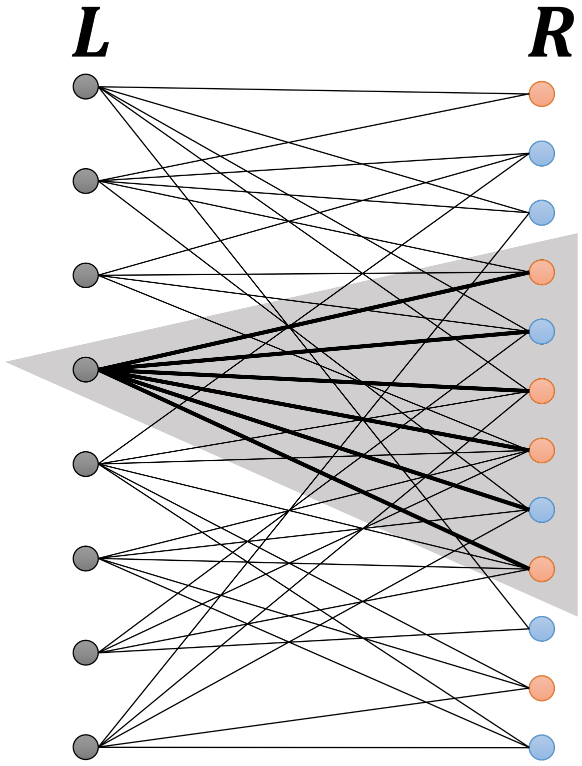
- Independently color each node red/blue with probability $^1/_2$.



(Weak) Local Splitting is PSLOCAL-Complete

Theorem: Weak local splitting for bipartite graph where all nodes in L have a large polylogarithmic degree --- say $\Theta(\log^{10} n)$ --- is P-SLOCAL-complete.

- It can be seen as a rounding fractional (1/2) values to integer values (0 or 1), while preserving some linear constraints.



Take-Home Message: Rounding fractional values to integer values, while coarsely preserving some linear constraints, is the only obstacle to obtaining efficient (polylog n-time) deterministic LOCAL algorithms.

(Weak) Local Splitting is PSLOCAL-Complete

Theorem: Weak local splitting for bipartite graph where all nodes in L have a large polylogarithmic degree --- say $\Theta(\log^{10} n)$ --- is P-SLOCAL-complete.

Proof.

network decomposition

↘ polylog-reducible

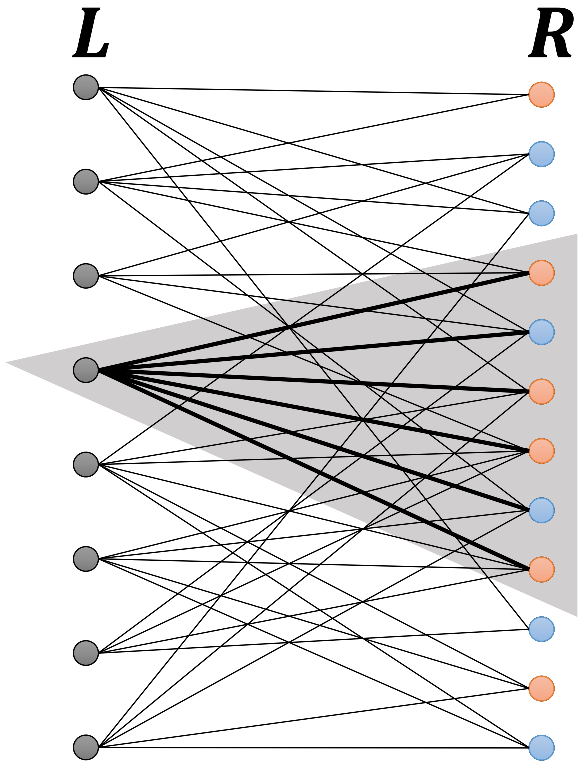
conflict-free coloring

↘ polylog-reducible

λ-local splitting

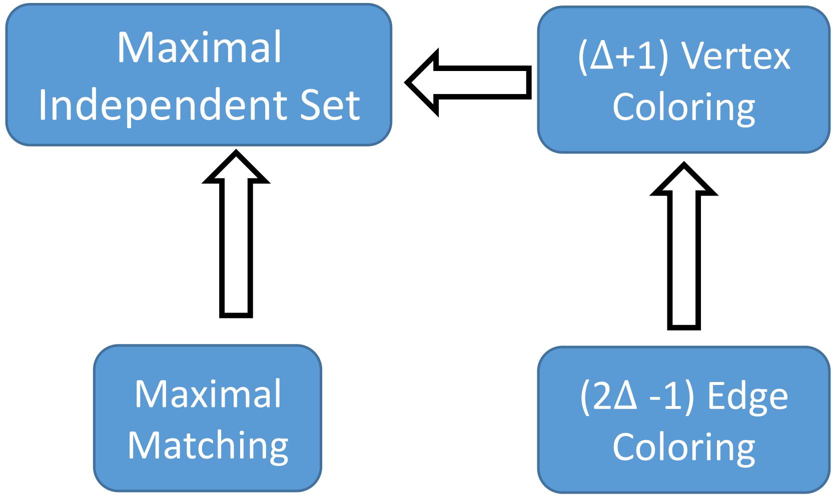
↘ polylog-reducible

weak local splitting



DET LOCAL Algorithms via Rounding

1. Maximal Matching
2. $(2\Delta - 1)$ edge coloring



Maximal Matching via Rounding

THEOREM [Fischer'17]
There is a $O\left(\log^2\Delta \cdot \log n\right)$-round deterministic algorithm for maximal matching.

$O(\log^4 n)$ Hańćkowiak, Karoński, Panconesi [SODA'98, PODC'99]

Algorithm Outline (Core Part):

O(1) - Approximate Bipartite Matching $O\left(\log^2 \Delta\right)$ rounds

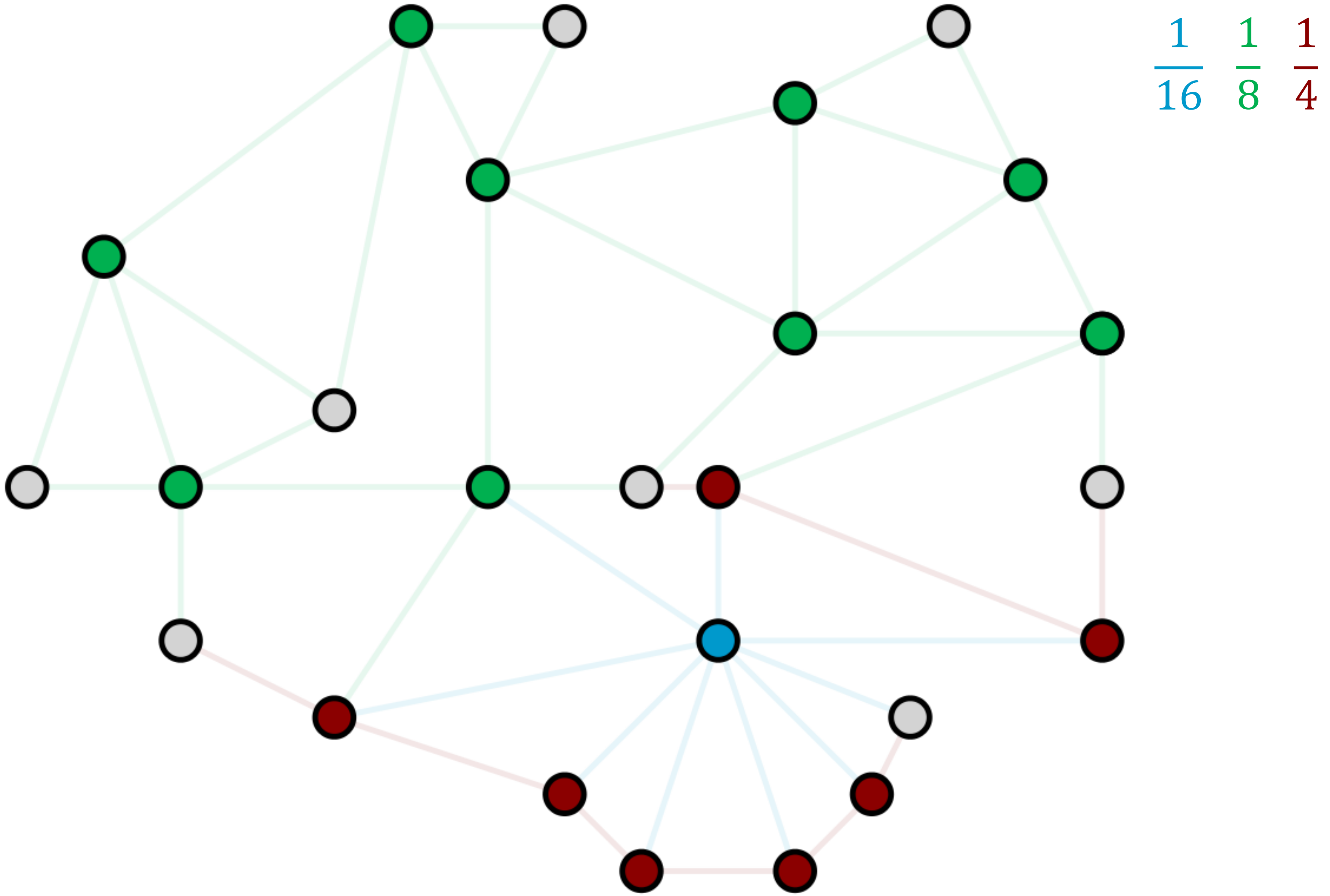
- **4 - Approximate Fractional Matching**

  $O(\log \Delta)$ rounds

- **Rounding Fractional Bipartite Matching**

  $O\left(\log^2 \Delta\right)$ rounds, $O(1)$ los
s

I) 4- Approximate Fractional Matching $\quad O(\log \Delta)$ rounds



Fractional Maximum Matching

$$\max \sum_{e \in E} x_e$$

value of v

$$\text{s.t.} \sum_{e \in E(v)} x_e \leq 1 \quad \text{for all } v \in V$$

$$x_e \in [0,1] \quad \text{for all } e \in E$$

v is half-tight if its value is $\geq \frac{1}{2}$

LOCAL GREEDY ALGORITHM

$x_e = 2^{-\lceil \log \Delta \rceil}$ for all $e \in E$

repeat until all edges are blocked

mark half-tight nodes

block its edges

double value of unblocked edges

Constant - Approximate Bipartite Matching $O\left(\log^2 \Delta\right)$ rounds

I) 4-Approximate Fractional Matching

$O(\log \Delta)$ rounds

II) Rounding Fractional Bipartite Matching

$O\left(\log^2 \Delta\right)$ rounds, $O(1)$ loss

II) Rounding Fractional Bipartite Matching

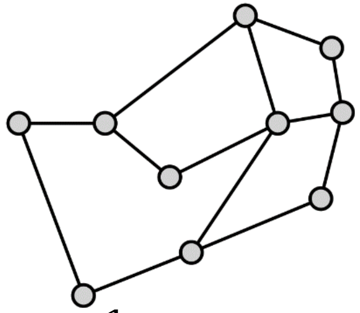
$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Fractionality of a Matching

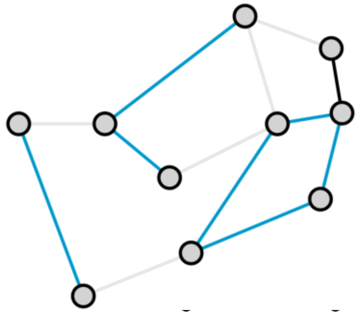
Matching is $\frac{1}{d}$ - fractional if all non-zero values are $\geq \frac{1}{d}$

1-fractional means integral

Factor-2-Rounding

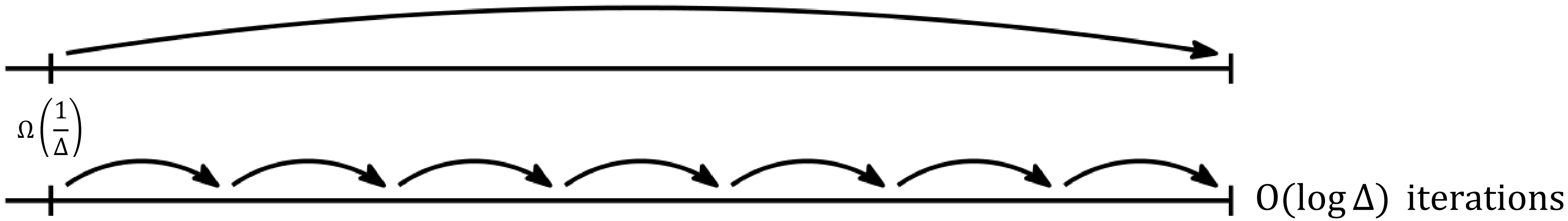


$\supseteq$



$\frac{1}{\mathrm{d}}$ - fractional $\longrightarrow$ $\frac{2}{\mathrm{d}}$ - fractional

DIRECT ROUNDING



GRADUAL ROUNDING

II) Rounding Fractional Bipartite Matching

$O(\log^2 \Delta)$ rounds, $O(1)$ loss

Iterated Factor-2-Rounding using Locally Balanced Splitting

<u>**Locally Balanced Splitting:**</u>
2-edge-coloring so that
every node roughly balanced

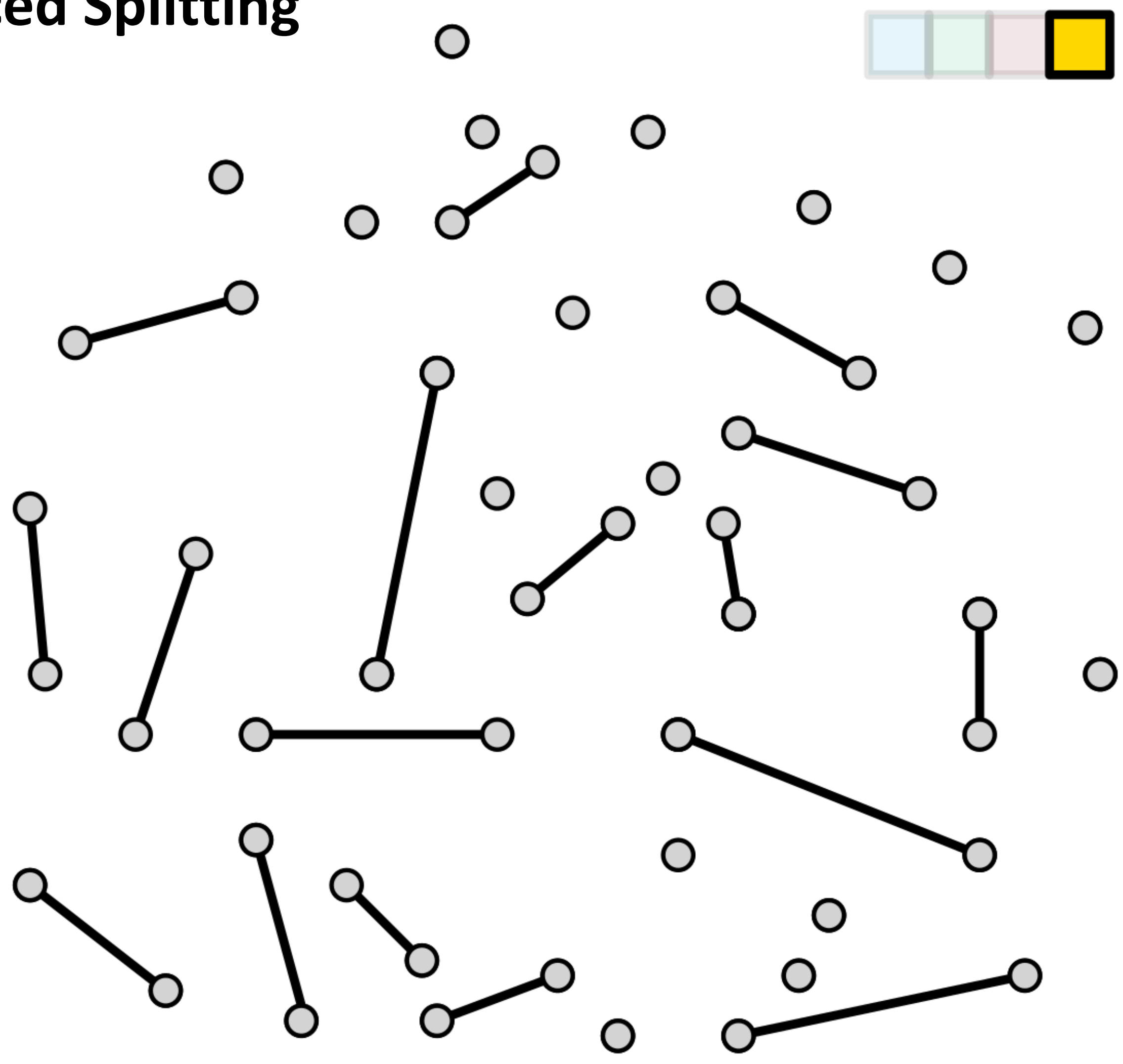
for i $= \lceil \log \Delta \rceil, \dots, 1$

$E_i = \{e \in E : x_e = 2^{-i}\}$

splitting of E_i into

increase to 2^{-i+1}

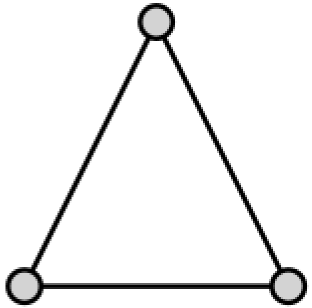
decrease to 0



In case of <u>perfect</u> locally balanced splitting:
no constraint violated & no loss in total value

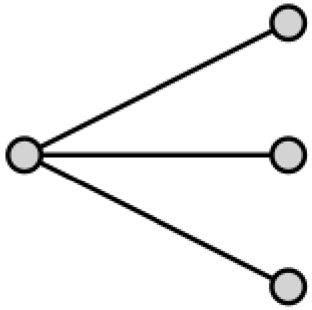
II) Rounding Fractional Bipartite Matching $O(\log^2 \Delta)$ rounds, $O(1)$ loss

perfect splitting not possible in case of…



… odd cycles

bipartite graph!



… odd-degree vertices

small technicality.

Suppose that bipartite even-degree graph!

II) Rounding Fractional Bipartite Matching — $O(\log^2 \Delta)$ rounds, $O(1)$ loss

LOCAL Almost-Perfect Splitting

Decompose into cycle

In parallel, for all cycles

A) **Short cycles** of length $O(\log \Delta)$

alternate ■□

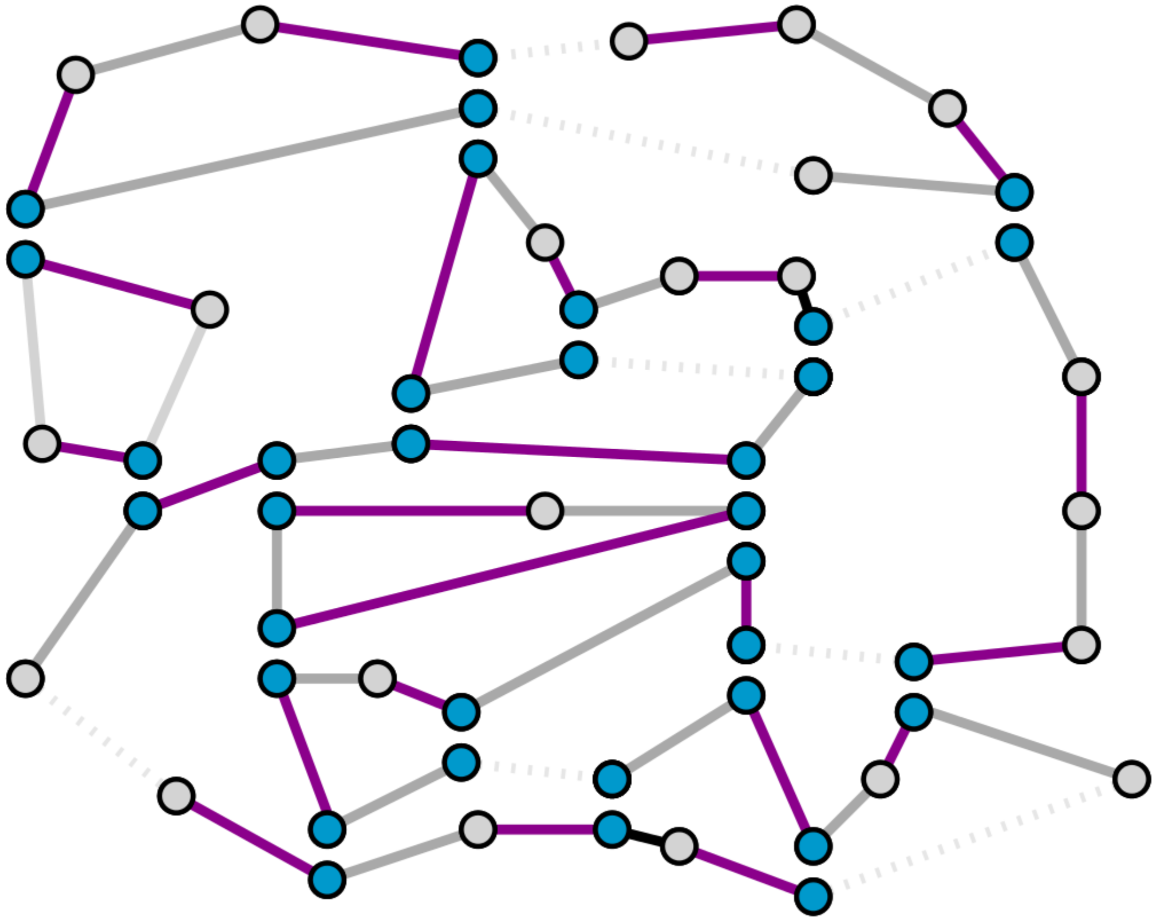
B) **Long cycles**

chop at length $\Theta(\log \Delta)$

set boundary to 0

alternate ■□ in between

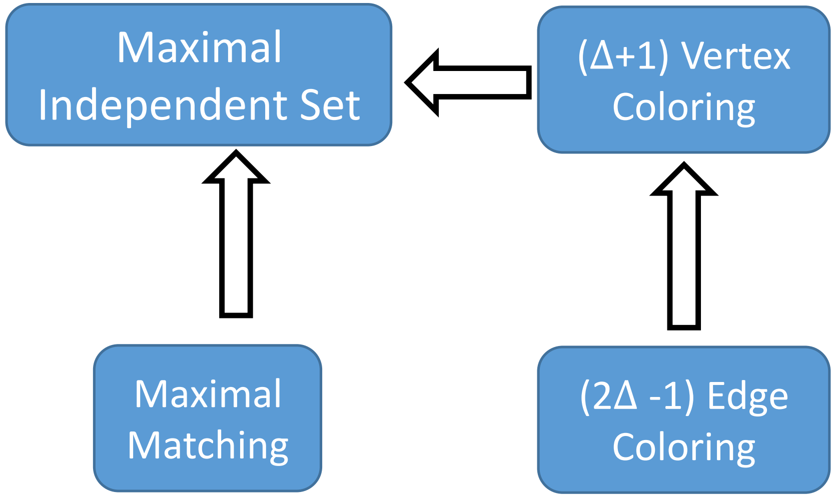
$\Theta\left(\frac{1}{\log \Delta}\right)$ **loss**



Over all O($\log \Delta$) rounding iterations, the overall loss still a constant!

DET LOCAL Algorithms via Rounding

1. Maximal Matching
2. $(2\Delta - 1)$ edge coloring



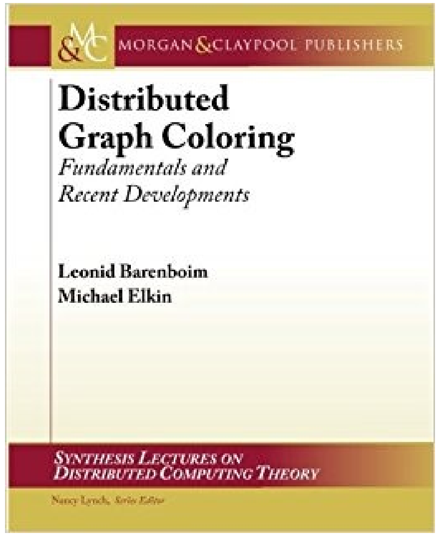
$(2\Delta - 1)$ edge coloring

"While maximal matchings can be computed in polylogarithmic time […], it is a decade old open problem whether the same running time is achievable for the remaining three structures."

Panconesi, Rizzi ‘01

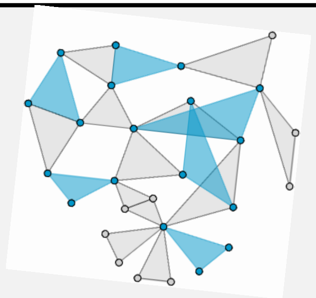
Open Problem 11.4:
Devise or rule out a deterministic $(2\Delta - 1)$-edge-coloring algorithm that runs in polylogarithmic time.

Barenboim, Elkin ‘13



- We resolve this problem and give a polylog n round algorithm for it.
- The solution goes via hypergraph maximal matching.

Hypergraph Maximal Matching & Implications

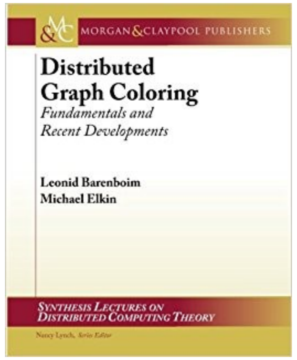


Rank-r-Hypergraph Maximal Matching
in poly $r \cdot \log^{O(\log r)} \Delta \cdot \log n$ rounds

$(2\Delta - 1)$-Edge-Coloring
in $O(\log^8 n)$ rounds

Randomized
$(2\Delta - 1)$-Edge-Coloring
in $O(\log^8 \log n)$rounds

Maximal Independent Set and $(\Delta + 1)$-Vertex-Coloring
for graphs with bounded neighborhood independence



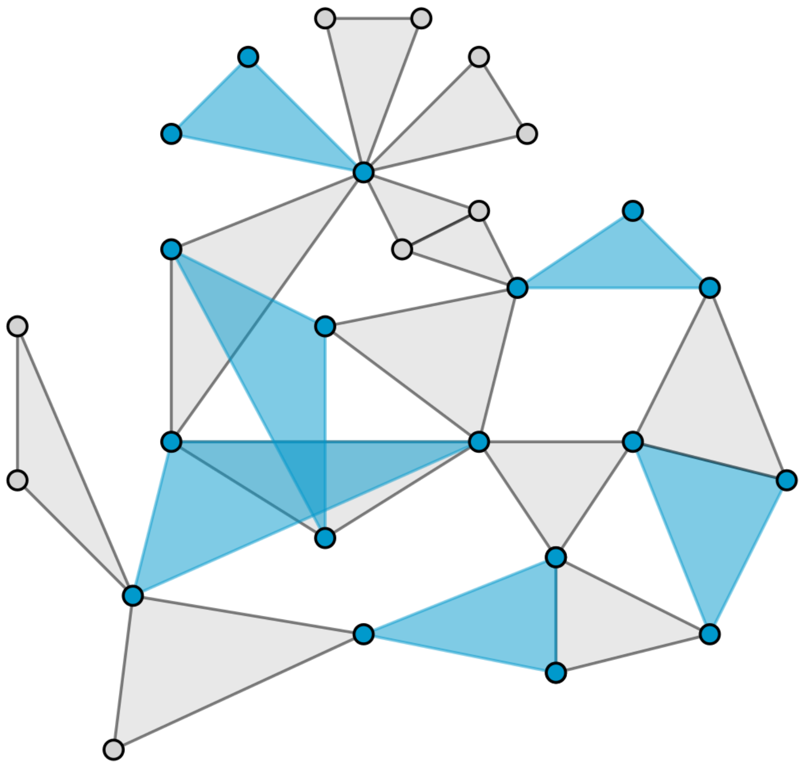
Open Problem 11.4

Open Problem 11.5

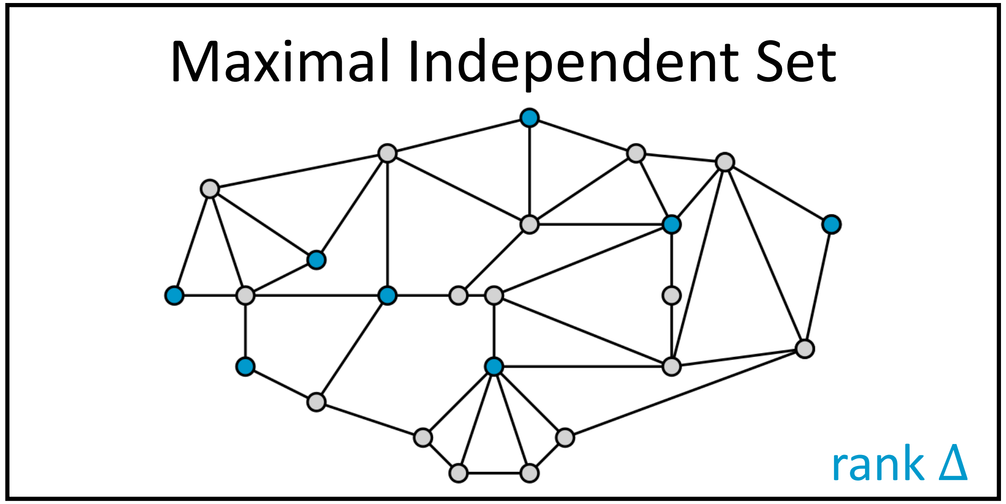
I) Formulation as Hypergraph Maximal Matching

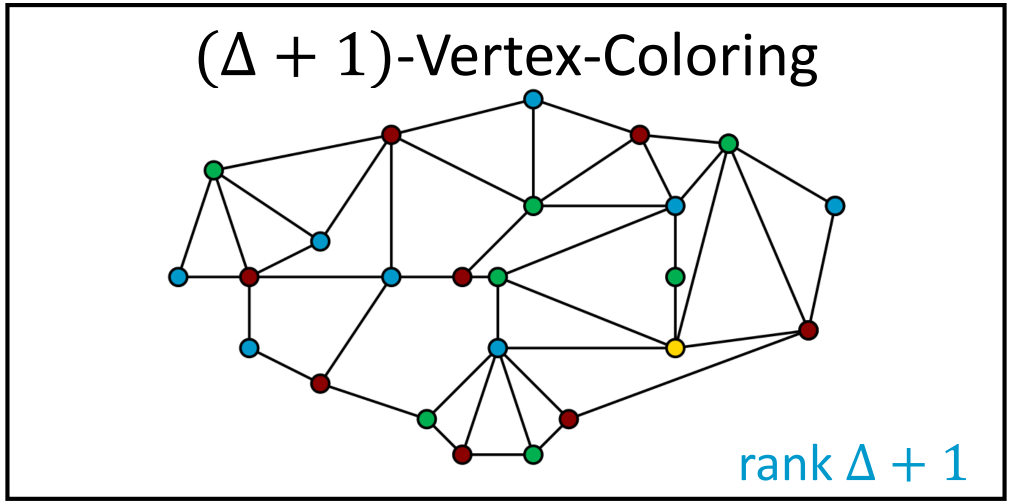
II) Hypergraph Maximal Matching Algorithm

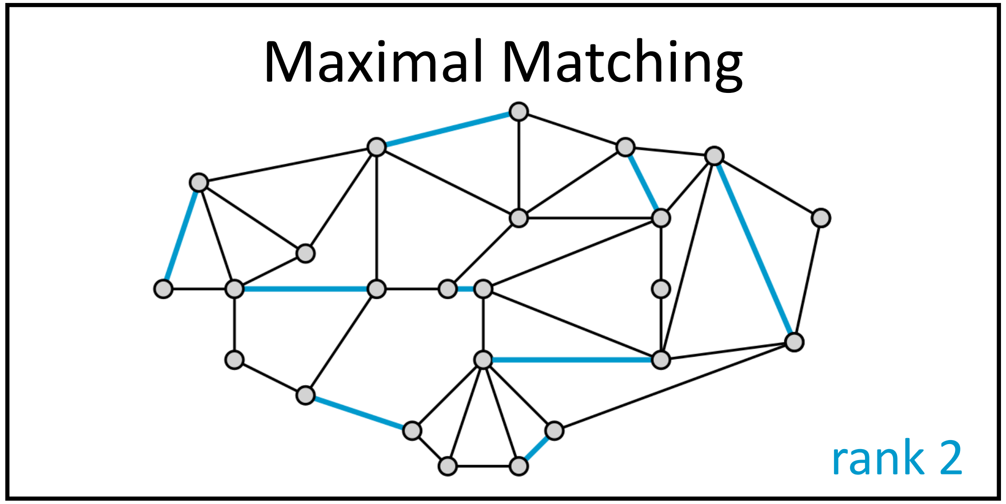
I) Formulation as Hypergraph Maximal Matching

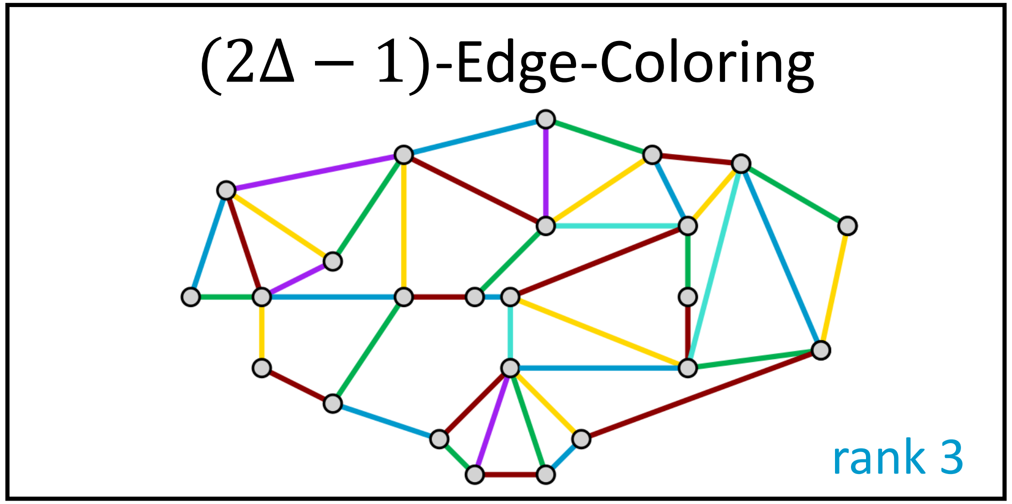


Unified Formulation as Hypergraph Maximal Matching Problem





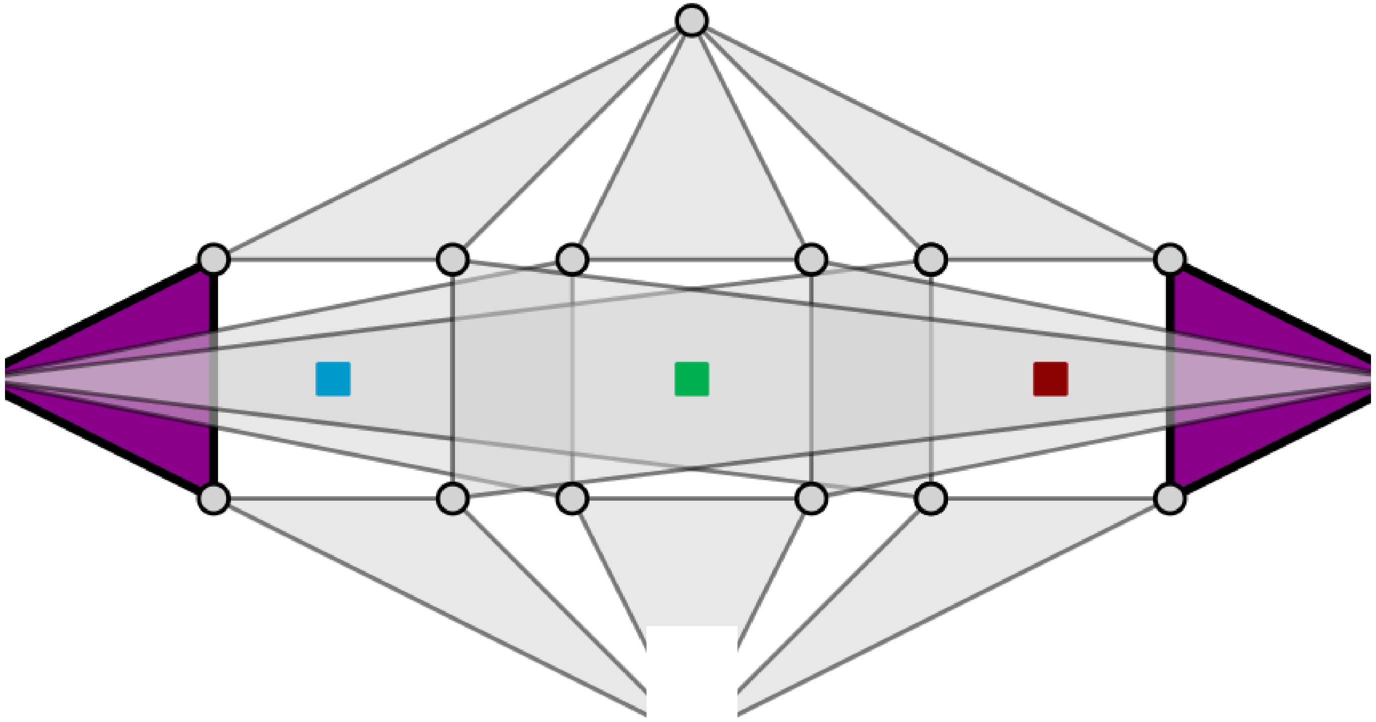




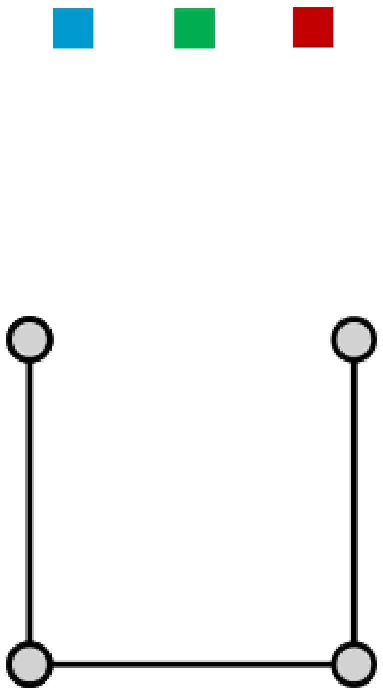
cast classic LOCAL graph problems as hypergraph maximal matching problems (LOCAL reductions)

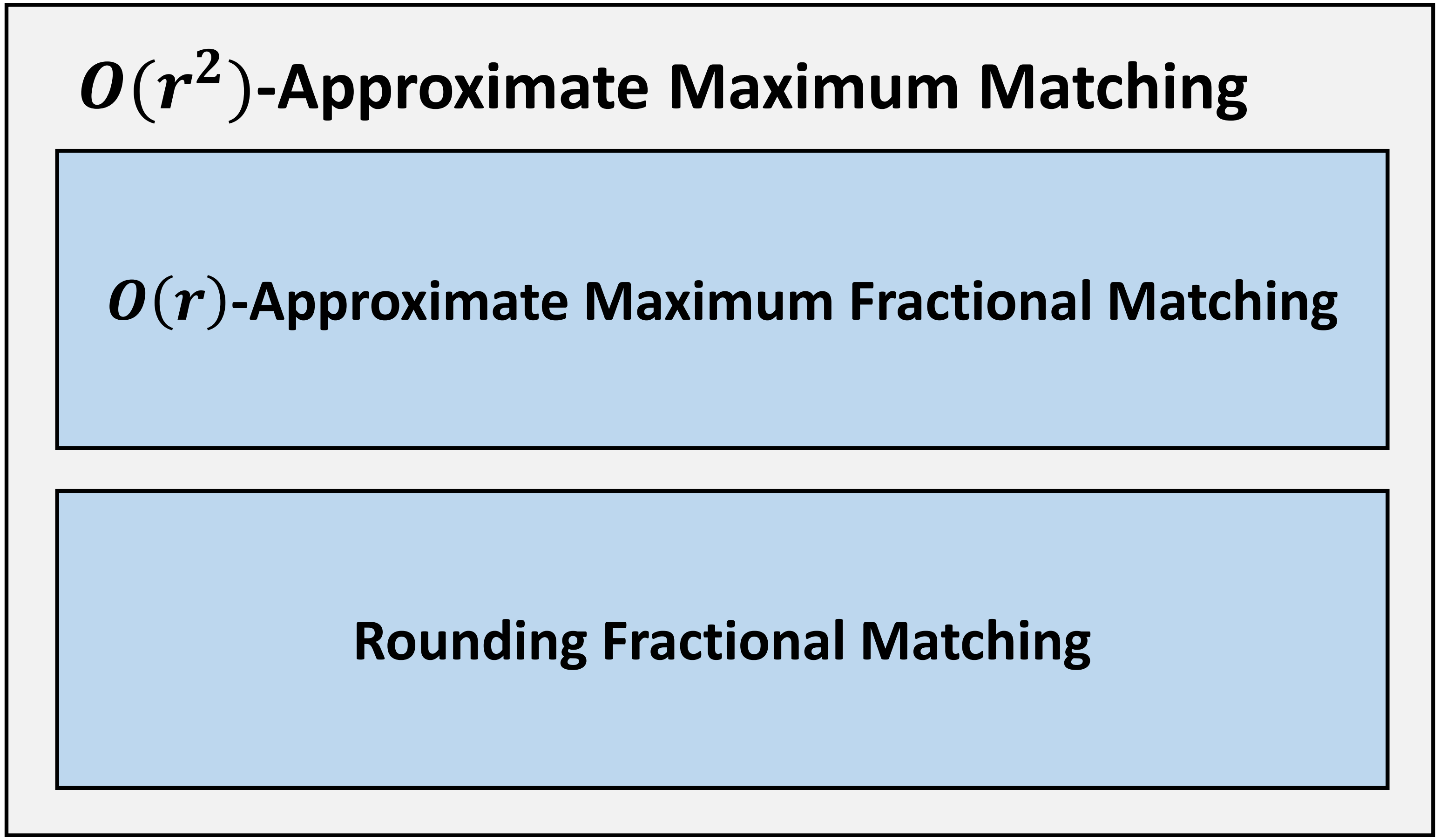
$(2\Delta - 1)$-Edge-Coloring as Rank-3-Hypergraph Maximal Matching

Rank-3-Hypergraph Maximal Matching



Edge-Coloring of Graph




$O(r^2)$-Approximate Maximum Matching
$O(r)$-Approximate Maximum Fractional Matching
Rounding Fractional Matching

Basic Rounding

Sequential Greedy Factor-L-rounding from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

for all unblocked edges with value $< \frac{L}{d}$

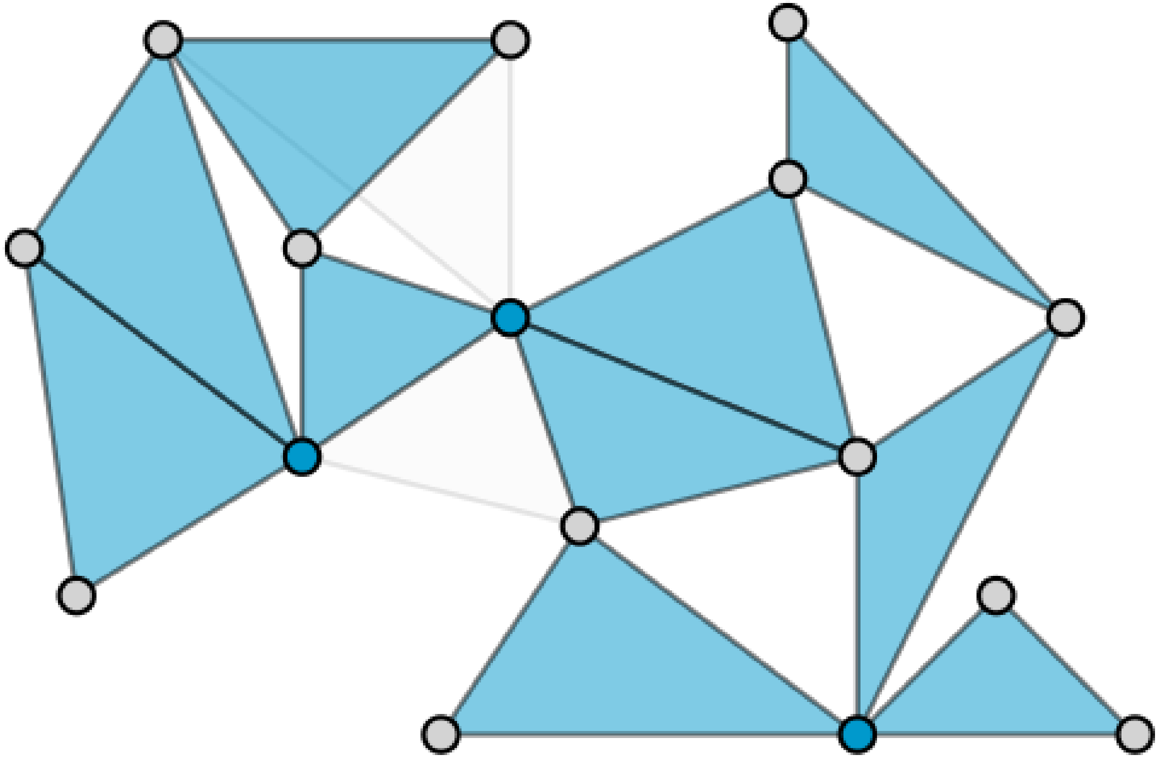
set value to $\frac{L}{d}$

mark tight nodes

block their edges

for all blocked edges with value $< \frac{L}{d}$

set value to 0



$O(r)$ loss

Basic Rounding

Factor-L-Rounding in $O\left(\log \Delta + (L \cdot r)^2\right)$ rounds with $O(r)$ loss

LOCAL Greedy Factor-L-Rounding

from $\geq \frac{1}{d}$ to $\geq \frac{L}{d}$

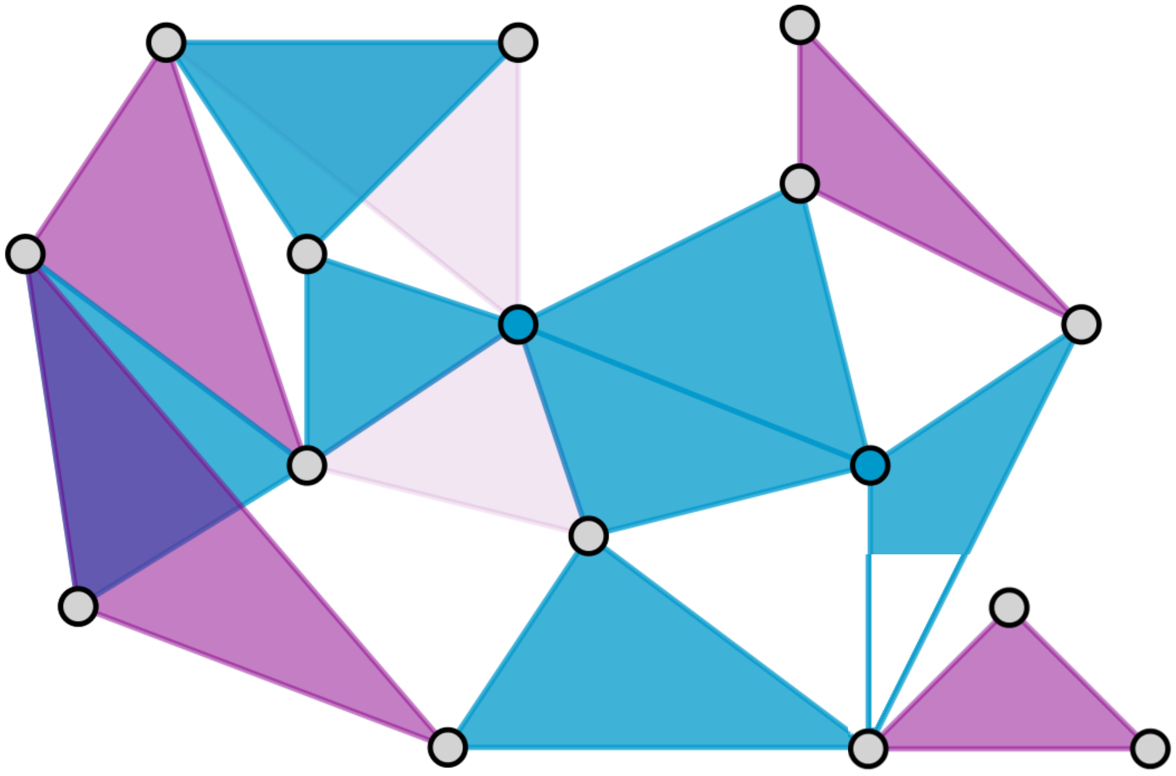
$\frac{d}{2L}$ - Defective $O\left(L^2 r^2\right)$-Edge-Coloring

for each color class

- mark half-tight nodes
- block their edges
- set value of edges in color class to $\frac{L}{d}$

for all blocked edges with value $< \frac{L}{d}$

- set value to 0



In each step, value of a node increased by at most $+\frac{d}{2L} \cdot \frac{L}{d} = \frac{1}{2}$

$O\left(\log \Delta + (L \cdot r)^2\right)$ rounds using Defective-Coloring Algorithm by Kuhn [SPAA'09]

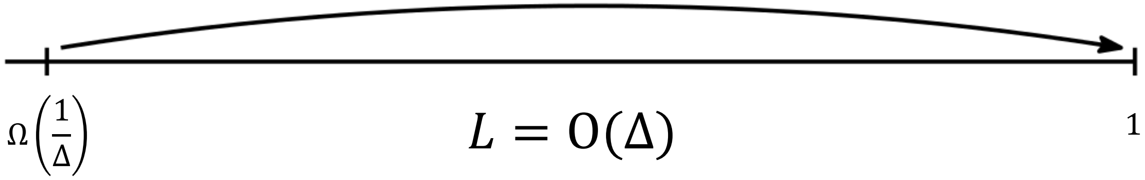
Rounding

Basic Rounding:
Factor-L-Rounding in $O\left(\log \Delta + (L \cdot r)^2\right)$ rounds with $O(r)$ loss

Direct Basic Rounding

$O(r)$ loss

$O\left(\log \Delta + (\Delta \cdot r)^2\right)$ rounds

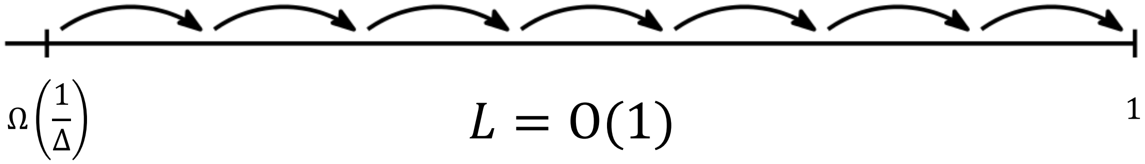


too slow!

Iterative Basic Rounding

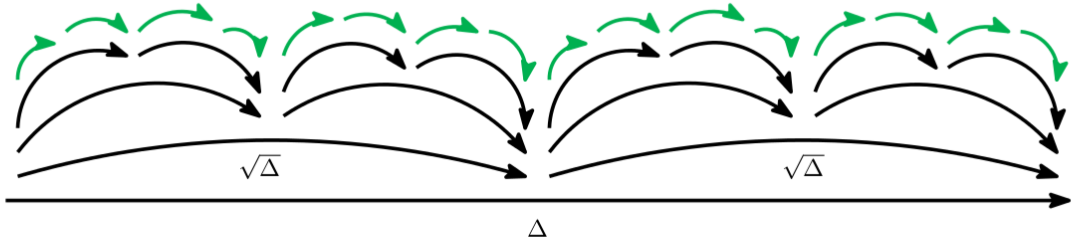
$r^{\Theta(\log \Delta)}$ loss

$O\left(\log \Delta \cdot \left(\log \Delta + r^2\right)\right)$ rounds



too lossy!

Recursive Rounding
&
Iterative Refilling



Basic Rounding

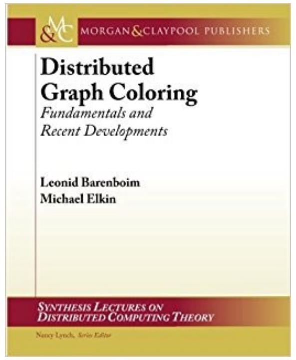
Invariant: $O(r)$ loss, maintained via refilling iterations

Further Improvements & Open Problems

Further Improvements (a general derandomization recipe)

- **Rank-r-Hypergraph Maximal Matching** in $\mathrm{poly}\,(r \cdot \log n)$ rounds
  - ✓ $(\mathbf{a}(\mathbf{1}+\boldsymbol{\varepsilon}))$ **Out-Degree Orientation** in $\mathrm{poly}\,(\log n/\boldsymbol{\varepsilon})$
  - ✓ $(\mathbf{1}+\boldsymbol{\varepsilon})$**-Approximation of Matching** in $\mathrm{poly}\,(\log n/\boldsymbol{\varepsilon})$

Open Problem 11.10

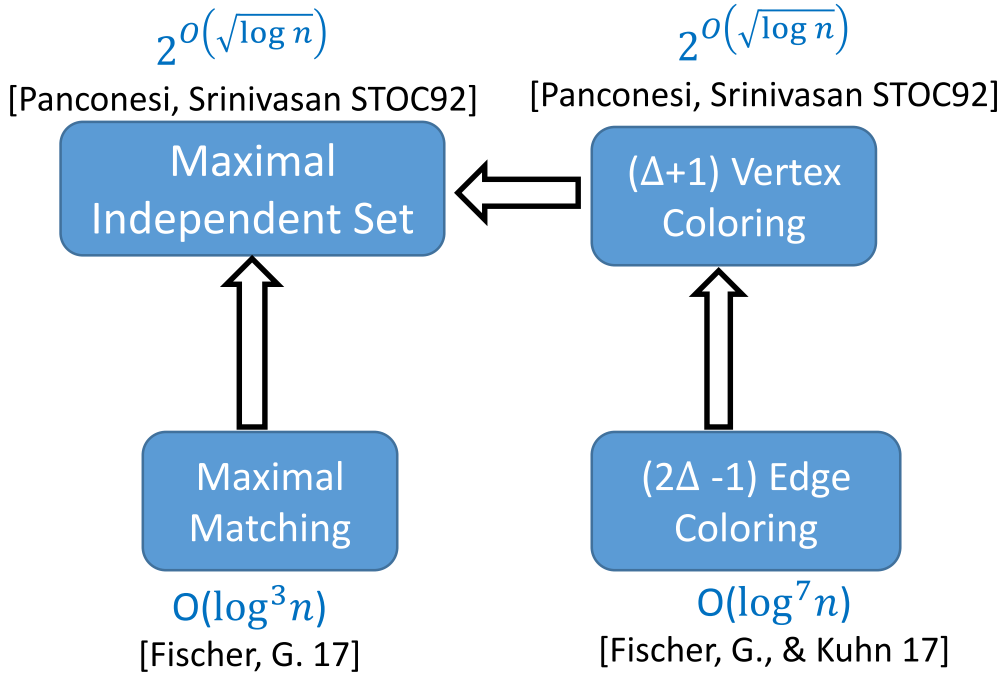


- $((\mathbf{1}+\boldsymbol{\varepsilon})\boldsymbol{\Delta})$**-Edge Coloring** in $\mathrm{poly}\,(\log n/\boldsymbol{\varepsilon})$ rounds, assuming $\Delta = \Omega_\epsilon(\log n)$
- Faster algorithms for th Lovasz Local Lemma
- For LCL problems, **P-SLOCAL = P-RSLOCAL**
- …

Open Problems

The SLOCAL model & Rounding as keys towards efficient DET LOCAL Algo.

- Linial's Q.: Is either of MIS or $(\Delta+1)$-vertex-coloring in P-LOCAL?
- Are they P-SLOCAL-complete?
- Solve splitting/rounding for $r = \log^{\omega(1)} n$



Thanks!