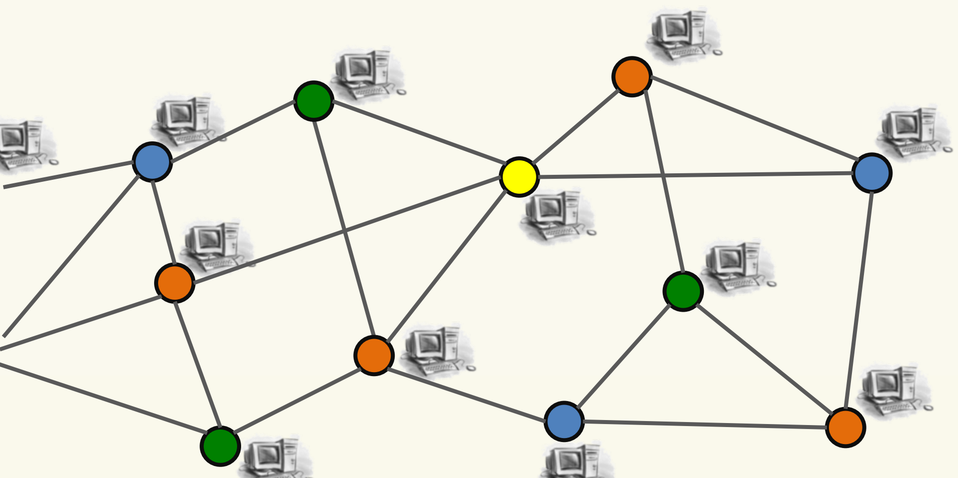
Local Distributed Rounding:

A Tool for Efficient Deterministic Distributed Symmetry Breaking

Fabian Kuhn, University of Freiburg, Germany

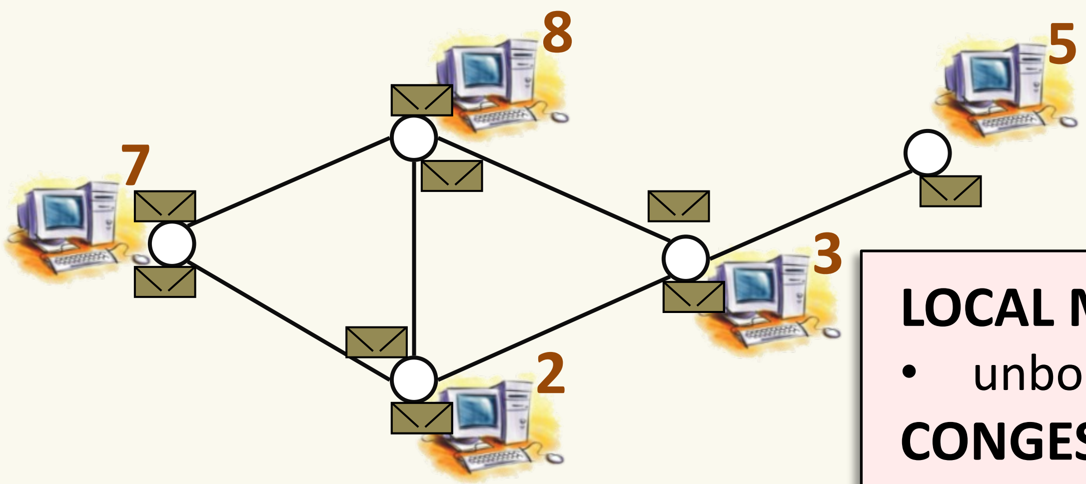
joint work with

Salwa Faour (U. Freiburg)
Mohsen Ghaffari (MIT)
Christoph Grunau (ETH Zurich)
Václav Rozhoň (ETH Zurich)



Distributed Graph Algorithms

Network is modeled as a graph



General assumption

- n nodes
- $O(\log n)$ bit IDs
  (IDs $\in \{1, \dots, N\}$)

LOCAL Model

- unbounded message size

CONGEST Model

- $O(\log n)$-bit messages

Synchronous rounds

1. Each node/computer does some (unrestricted) internal computation
2. Send a message to each neighbor
3. Receive message from each neighbor

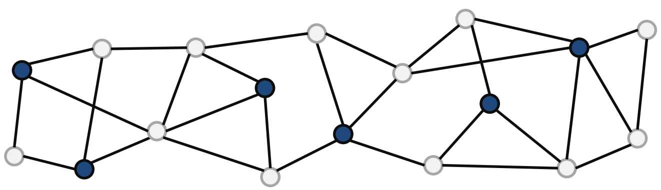
time complexity $=$ number of rounds

Four Classic Problems (since 1980s)

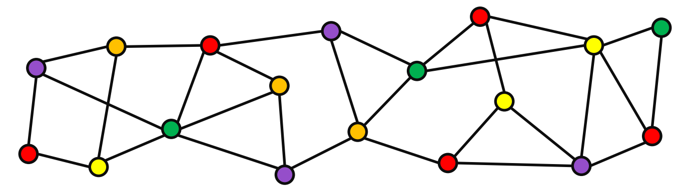
Simple reduction

[Luby; STOC ‘85], [Linial; FOCS ’87]

MIS



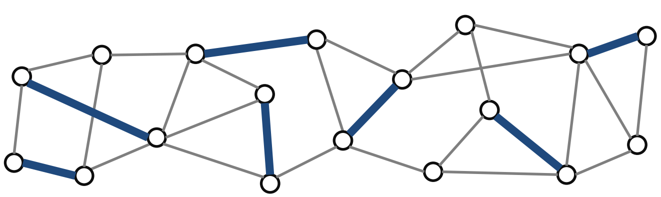
$(\Delta + 1)$-Vertex Coloring



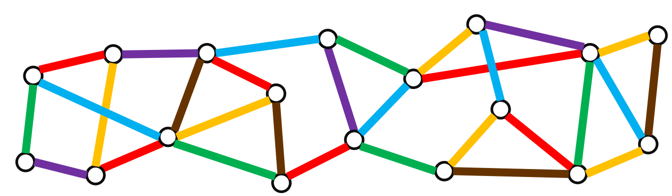
MIS on line graph

$(\Delta + 1)$-coloring on line graph

Maximal Matching



$(2\Delta - 1)$-Edge Coloring



Four Problems, State of the Art, Early 2019

MIS

Rand.: $O(\log\Delta) + 2^{O(\sqrt{\log\log n})}$

[Ghaffari; SODA '16]

Det.: $2^{O(\sqrt{\log n})}$

[Panconesi,Srinivasan; STOC '92]

Maximal Matching

Rand.: $O(\log\Delta) + O(\log^3 \log n)$

[Barenboim,Elkin,Pettie,Schneider; FOCS '12]

Det.: $O(\log^2\Delta \cdot \log n)$

[Fischer; DISC '17]

Det.: $\Omega(\log n / \log\log n)$

[Balliu,Brandt,Hirvonen,Olivetti,Rabie,Suomela; FOCS '19]

Rand.: $\Omega\left(\sqrt{\log n / \log\log n}\right)$

[K.,Moscibroda,Wattenhofer; PODC '04]

$(\Delta + 1)$-Vertex Coloring

Rand.: $2^{O(\sqrt{\log\log n})}$

[Chang,Li,Pettie; STOC '18]

Det.: $2^{O(\sqrt{\log n})}$

[Panconesi,Srinivasan; STOC '92]

$(2\Delta - 1)$-Edge Coloring

Rand.: $\tilde{O}(\log^3 \log n)$

[Elkin,Pettie,Su; SODA '15]

Det.: $\tilde{O}(\log^2\Delta \cdot \log n)$

[Harris; FOCS '19]

Rand.: $\Omega(\log^* n)$

[Linial; FOCS '87], [Naor; 1990]

Four Problems, State of the Art, Early 2019

MIS

Rand.: $O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$

[Ghaffari; SODA '16]

Det.: $2^{O(\sqrt{\log n})}$

[Panconesi,Srinivasan; STOC '92]

Rand.: $O(\log \Delta) + O(\log^3 \log n)$

[Barenboim,Elkin,Pettie,Schneider; FOCS '12]

Det.: $O(\log^2 \Delta \cdot \log n)$

[Fischer; DISC '17]

Det.: $\Omega(\log n / \log \log n)$

[Balliu,Brandt,Hirvonen,Olivetti,Rabie,Suomela; FOCS '19]

Rand.: $\Omega\left(\sqrt{\log n / \log \log n}\right)$

[K.,Moscibroda,Wattenhofer; PODC '04]

$(\Delta + 1)$-Vertex Coloring

Rand.: $2^{O(\sqrt{\log \log n})}$

[Chang,Li,Pettie; STOC '18]

Det.: $2^{O(\sqrt{\log n})}$

[Panconesi,Srinivasan; STOC '92]

Rand.: $\tilde{O}(\log^3 \log n)$

[Elkin,Pettie,Su; SODA '15]

Det.: $\tilde{O}(\log^2 \Delta \cdot \log n)$

[Harris; FOCS '19]

Rand.: $\Omega(\log^* n)$

[Linial; FOCS '87], [Naor; 1990]

$$\textbf{RandomizedTime}(n) \geq \textbf{DeterministicTime}\left(\sqrt{\log n}\right)$$

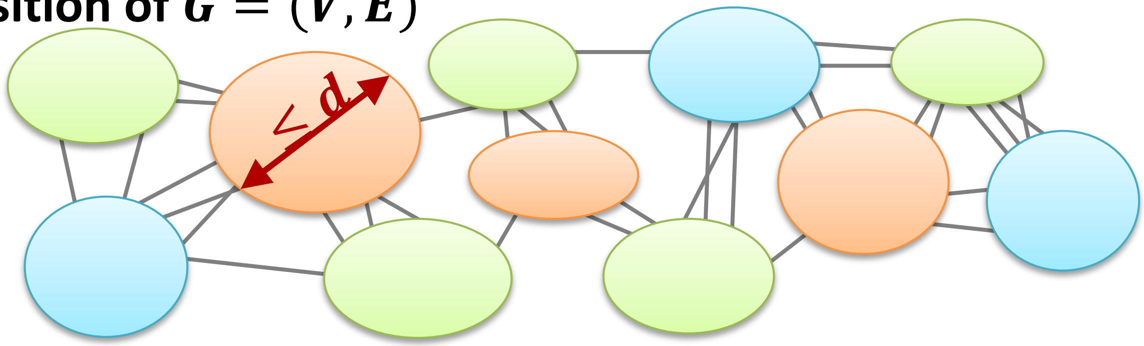
[Chang,Kopelowitz,Pettie; FOCS '16]

The Breakthrough of Rozhoň & Ghaffari

[Awerbuch,Goldberg,Luby,Plotkin; FOCS '89]

Definition: (d, c)-decomposition of $G = (V, E)$

- Partition of V into clusters of diameter $\leq d$
- Coloring of the cluster graph with c colors



Fast Deterministic Decomposition: [Rozhoň, Ghaffari; STOC '20]

- There is an $O(\log^7 n)$**-round deterministic distributed alg.** to compute an

$$(O(\log n), O(\log n))\text{-decomposition}.$$

- Implies $O(\log^7 n)$-round deterministic distributed algorithms for MIS and $(\Delta + 1)$-coloring.
- Implies poly $\log n$-round deterministic distributed algorithms for all locally checkable problems with poly $\log n$-round randomized algorithms.
- The time was improved to $O(\log^5 n)$ in [Ghaffari,Grunau,Rozhoň; SODA '21]

Four Classic Problems

MIS

Rand.: $O(\log\Delta) + O(\log^5 \log n)$

[Ghaffari; SODA '16]

Det.: $O(\log^5 n)$

[Rozhoň,Ghaffari; STOC '20]
[Ghaffari,Grunau,Rozhoň; SODA '21]

$(\Delta+1)$-Vertex Coloring

Rand.: $O(\log^5 \log n)$

[Chang,Li,Pettie; STOC '18]

Det.: $O(\log^5 n)$

[Rozhoň,Ghaffari; STOC '20]
[Ghaffari,Grunau,Rozhoň; SODA '21]

Maximal Matching

Rand.: $O(\log\Delta) + O(\log^3 \log n)$

[Barenboim,Elkin,Pettie,Schneider; FOCS '12]

Det.: $O(\log^2 \Delta \cdot \log n)$

[Fischer; DISC '17]

Det.: $\Omega(\log n / \log\log n)$

[Balliu,Brandt,Hirvonen,Olivetti,Rabie,Suomela; FOCS '19]

Rand.: $\Omega\left(\sqrt{\log n / \log\log n}\right)$

[K.,Moscibroda,Wattenhofer; PODC '04]

$(2\Delta-1)$-Edge Coloring

Rand.: $\tilde{O}(\log^3 \log n)$

[Elkin,Pettie,Su; SODA '15]

Det.: $\tilde{O}(\log^2 \Delta \cdot \log n)$

[Harris; FOCS '19]

Rand.: $\Omega(\log^* n)$

[Linial; FOCS '87], [Naor; 1990]

New More Direct Deterministic Algorithms

- Algorithms based on network decomposition are quite brute-force
  - The algorithms really exploit the LOCAL model
    - even for the four classic problems, and especially when derandomizing by using the method of conditional expectations

- **Can we get similar results more directly?**

Theorem: The MIS and $(\Delta + 1)$-coloring problems can be solved **deterministically** in $\boldsymbol{O\left(\log^2 \Delta \cdot \log n\right)}$ **rounds** in the LOCAL model.

- Based on a ***generic technique*** for ***rounding fractional solutions***
  - Algorithm for $(\Delta + 1)$-coloring appeared in **[Ghaffari, K.; FOCS '21]**
  - MIS alg. / generic rounding in **[Faour, Ghaffari, Grunau, K., Rozhoň; SODA '23]**

- Almost the same bounds hold in the CONGEST model (msg. of $O(\log n)$ bits)
  - MIS at the cost of an $O(\log \log \Delta)$ factor

Complexity of Four Classic Problems

MIS

Rand.: $O(\log \Delta) + O(\log^3 \log n)$
[Ghaffari; SODA '16]

Det.: $O(\log^2 \Delta \cdot \log n)$
[Faour, Ghaffari, Grunau, K., Rozhoň; SODA '23]

Maximal Matching

Rand.: $O(\log \Delta) + O(\log^3 \log n)$
[Barenboim,Elkin,Pettie,Schneider; FOCS '12]

Det.: $O(\log^2 \Delta \cdot \log n)$
[Fischer; DISC '17]

Det.: $\Omega(\log n / \log \log n)$
[Balliu,Brandt,Hirvonen,Olivetti,Rabie,Suomela; FOCS '19]

Rand.: $\Omega\left(\sqrt{\log n / \log \log n}\right)$
[K.,Moscibroda,Wattenhofer; PODC '04]

$(\Delta + 1)$-Vertex Coloring

Rand.: $O(\log^3 \log n)$
[Chang,Li,Pettie; STOC '18]

Det.: $O(\log^2 \Delta \cdot \log n)$
[Ghaffari, K.; FOCS '21]

$(2\Delta - 1)$-Edge Coloring

Rand.: $O(\log^3 \log n)$
[Elkin,Pettie,Su; SODA '15]

Det.: $O(\log^2 \Delta \cdot \log n)$
[Ghaffari, K.; FOCS '21]

Rand.: $\Omega(\log^* n)$
[Linial; FOCS '87], [Naor; 1990]

Highlevel Idea: Deterministic Rounding

Observations

- Solving a fractional variant of a problem is often easier
- Typically does not require to break symmetries
- Often simple deterministic solutions or easy to derandomize
- Round gradually $\Longrightarrow$ break symmetries gradually (and more efficiently)

Deterministic Distributed Rounding of Fractional Solutions

- Has successfully been used for computing maximal matchings in graphs and bounded-rank hypergraphs (with applications to distr. edge coloring)

  [Fischer; DISC '17], [Fischer, Ghaffari, K.; FOCS '17]

  – also more implicitly in [Hańćkowiak, Karonski, Panconesi; SODA '98 / PODC '99]
  – and for minimum dominating set in [Deurer, K., Maus; PODC '19]

For this talk, we first consider a simpler problem

compute a large independent set

Simple Randomized Independent Set Algorithm

Setting: Graph $G = (V, E)$ with an edge orientation

- Orientation: nodes give priority to join indep. set to their out-neighbors

Input: Parameter $\lambda \in (0,1)$ and $\forall v \in V$, a value $x_v \in [0,1]$ s.t.

$$\forall v \in V : \sum_{u \in N_{out(v)}} x_u \leq \lambda$$

Algorithm to compute an independent set S:

1. $\forall v \in V$: mark v with probability x_v
2. $\forall v \in V$: v joins S if v is marked and no $u \in N_{out}(v)$ is marked

Analysis:

$$|S| \geq \#(\textit{marked nodes}) - \#(\textit{edges with 2 marked nodes})$$

$$\mathbb{E}[|\boldsymbol{S}|] \geq \sum_{v\in V} x_v - \sum_{v\in V} \sum_{u\in N_{out}(v)} x_v \cdot x_u \;=\; \sum_{v\in V} x_v - \sum_{v\in V} x_v \cdot \underbrace{\sum_{u\in N_{out}(v)} x_u}_{\leq \lambda}$$

$$\geq \sum_{v\in V} x_v - \sum_{v\in V} x_v \cdot \lambda \;=\; (\mathbf{1}-\boldsymbol{\lambda}) \cdot \sum_{v\in V} \boldsymbol{x_v}$$

Example:

$$\forall v \in V : x_v = \frac{\lambda}{\deg(v)} \;\Longrightarrow\; \mathbb{E}[|S|] \geq \lambda \cdot (1-\lambda) \cdot \sum_{v\in V} \frac{1}{\deg(v)}$$

Observation:

- If the node values are integers (i.e., $x_v \in \{0,1\}$), we have

$$|S| = \mathbb{E}[|S|] \geq \sum_{v\in V} x_v - \sum_{v\in V} \sum_{u\in N_{out}(v)} x_v \cdot x_u$$

Highlevel Idea: Fractional Problem and Rounding

Fractional Independent Set

- Each node v has a fractional value $x_v \in [0,1]$
- "Size" of fractional indep. set: expected indep. size of randomized alg.

$$\mathbb{E}[|S|] \geq \sum_{v \in V} x_v - \sum_{v \in V} \sum_{u \in N_{out}(v)} x_v \cdot x_u$$

Gradual Rounding

- Fractionality of a fractional solution $=$ smallest non-zero x_v value
  - E.g., if $x_v \in \{0, 1/2, 1\}$, fractionality is $1/2$
- Start with fractional solution and gradually round to integer value x_v
- Gradual rounding $=$ gradually increase the fractionality
- **Goal:** approximately preserve expected independent set size while rounding the solution

Rounding Overview

Fractional Solution

utility, cost

potential $\Phi(\vec{x}) = \sum_{v \in V} x_v - \sum_{\{u,v\} \in E} x_u \cdot x_v$

Gradual Rounding:

- At all times, for all v : $x_v = 0$ or $x_v = 2^{-k}$ for some integer $k \geq 0$
- Initially, for all v: $x_v = 2^{-k_0}$, where $k_0 = \lceil \log_2 \Delta \rceil$
- **After i $\geq$ 0 rounding steps:**

$$x_v = 0 \text{ or } x_v = 2^{-k_i}, \text{where } k_i = k_0 - i$$

- After $k_0 = O(\log \Delta)$ steps, $x_v = 0$ or $x_v = 1$
  - *And we thus have an independent set of size* $\Phi(\vec{x})$

Goal:

- Implement rounding step s.t. $\Phi(\vec{x})$ drops by factor $\leq 1 - O(1/\log \Delta)$

Rounding Overview

Fractional Solution

$$\Phi(\vec{x}) = \sum_{v\in V} x_v - \sum_{\{u,v\}\in E} x_u \cdot x_v$$

$$= \sum_{v\in V} \text{utility}(v) - \sum_{e\in E} \text{cost}(e)$$

Rounding of a node:

- v can round x_v such that $\text{utility}(v) - \sum_{e \text{ of } v} \text{cost}(e)$ does not increase
  - Gives a simple sequential rounding algorithm
- As long as no two neighbors are processed at the same time, this allows to round such that $\Phi(\vec{x})$ does not increase
  - This is however way too slow to be interesting…
- **Idea:** Try to use a defective coloring (some monochromatic edges) with a small number of colors and show that $\Phi(\vec{x})$ does not increase too much.

Using a Defective Coloring

Weighted Average Defective Coloring:

- For a graph $G = (V, E)$ with edge weights $w_e \geq 0$ and $\varepsilon > 0$, one can compute a coloring with $O(1/\varepsilon)$ colors such that at most a ε-factor of the total edge weight is on monochromatic edges.
- Such a coloring can be computed in (essentially) $O(1/\varepsilon)$ rounds.
  - Follows from work in [K.; SPAA '09], [Barenboim, Elkin, Goldenberg; PODC '18], [Kawarabayashi, Schwartzman; DISC '18]

In $O(1/\varepsilon + \log^* \Delta)$ rounds if an initial $O(\Delta^2)$-coloring is given.

Recall: $\Phi(\vec{x}) = \sum_{v \in V} \text{utility}(v) - \sum_{e \in E} \text{cost}(e)$

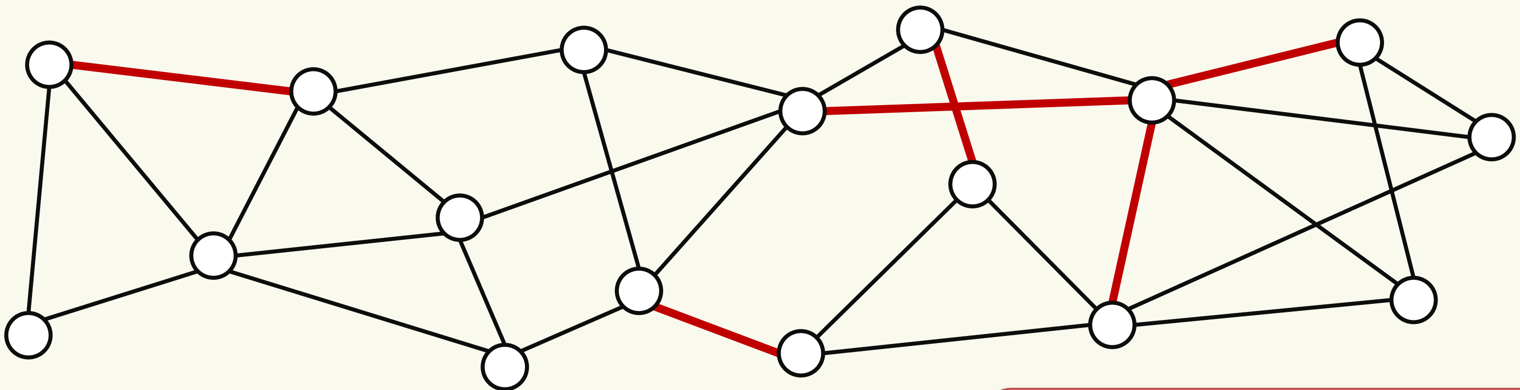
Algorithm:

1. Defective coloring for edge weights $\text{cost}(e)$ and $\varepsilon = 1/\log \Delta$
2. Rounding on the subgraph induced by bichromatic edges

Using a Defective Coloring

Algorithm:

1. Defective coloring for edge weights $\mathrm{cost}(e)$ and $\varepsilon = 1/\log\Delta$
2. Rounding on the subgraph induced by the bichromatic edges



bichromatic edges

monochromatic edges, at most an ε-factor of total cost

$$\Phi(\vec{x}) = \sum_{v\in V} \mathrm{utility}(v) - \sum_{e\in E_b} \mathrm{cost}(e) - \sum_{e\in E_m} \mathrm{cost}(e)$$

does not decrease

grows by factor ≤ 4
(because fractional values at most double)

Using a Defective Coloring

bichromatic edges

monochromatic edges, at most an ε-factor of total cost

$$\Phi(\vec{x}) = \sum_{v \in V} \text{utility}(v) - \sum_{e \in E_b} \text{cost}(e) - \sum_{e \in E_m} \text{cost}(e)$$

does not decrease

grows by factor ≤ 4
(because fractional values at most double)

$$\Phi(\vec{x}') \geq \Phi(\vec{x}) - 4 \cdot \sum_{e \in E_m} \text{cost}(e) \geq \Phi(\vec{x}) - 4\varepsilon \cdot \sum_{e \in E} \text{cost}(e)$$

new potential

old potential

We have $\Phi(\vec{x}') \geq \big(1 - O(\varepsilon)\big) \cdot \Phi(\vec{x})$ only if $\mathbf{cost} = \boldsymbol{O}(\mathbf{utility} - \mathbf{cost})$

- This is true at the beginning, but not necessarily after a few rounding steps
- We can however enforce something sufficient, by slightly using a potential function that slightly changes between rounding steps.

Adaptive Potential Function

Potential Function of Rounding Step $\boldsymbol{i} \in \{\mathbf{1}, \dots, \mathbf{\log\Delta}\}$:

$$\Phi_{\mathrm{i}}(\vec{x}) = \sum_{v \in V} \mathrm{utility}(v) - \left(\frac{3}{2} - \frac{i}{2\log\Delta}\right) \cdot \sum_{e \in E} \mathrm{cost}(e)$$

Intuition:

- Initially utility $\geq 2 \cdot$ cost, and hence, $\Phi_1(\vec{x}) = \Omega(\text{utility})$
  - We can implement the first rounding step while maintaining the value of $\Phi_1(\vec{x})$ up to a $1 + O(\varepsilon)$ factor.

- For the next rounding step, the gap between positive and negative term grows by $\Theta(\text{cost}/\log\Delta)$
  - If the gap was already $\Theta(\text{utility})$, the next rounding step works anyways
  - Otherwise, the gap grows by $\Theta(\text{utility}/\log\Delta)$ and becomes sufficiently large to make the rounding step work (still with $\varepsilon = 1/\log\Delta$)

Using a Defective Coloring

Algorithm:

1. Defective coloring for edge weights $\mathrm{cost}(e)$ and $\varepsilon = 1/\log\Delta$
2. Rounding on the subgraph induced by the bichromatic edges

Round complexity of one rounding step:

- Both steps require $O(1/\varepsilon) = O(\log\Delta)$ rounds.

We need to compute an initial $O(\Delta^2)$-coloring for the defective coloring to be as fast as claimed.

Total round complexity:

- $O(\log\Delta)$ rounding steps $\Longrightarrow$ $\boldsymbol{O(\log^2\Delta + \log^* n)}$ **rounds**

Time to deterministically compute and independent set of size $\Omega\left(\sum_{v\in V}\frac{1}{\deg(v)}\right)$.

Generic Rounding Algorithm

General Setting, graph $G = (V, E)$:

- Every node $v \in V$ picks a label ℓ_v from a finite alphabet Σ

Potential function

- Quality of labeling is measured by a potential function Φ
- Potential $\Phi = U - C$ for utility $U \geq 0$ and cost $C \geq 0$
- Cost and utility are defined as sums over node and edge utilities/costs
  - Utility/cost functions can differ arbitrarily between different nodes/edges

Fractional label assignment

- Each node gets assigned a probability distribution over the labels in Σ
- Utility and cost of fractional label assignment are defined as the expected cost if all nodes pick their labels independently according to the given probability distributions (fractional solution)

Generic Rounding Algorithm

Fractionality of fractional label assignment

- For node v and label $\ell \in \Sigma$, let $x_{v,\ell} \in [0,1]$ be the fractional value (i.e., probability) for node v and label ℓ
- **Fractionality of label assignment:** minimum non-zero $x_{v,\ell}$-value
  - E.g., if all $x_{v,\ell}$ are $x_{v,\ell} = 0$ or $x_{v,\ell} \geq 1/F$, the fractionality is $1/F$

Assume we are given a **fractional label assignment** with potential

$$\mathbf{\Phi_0 = U_0 - C_0},$$

Where U_0 is the utility and C_0 is the cost.

If $\mathbf{\Phi_0 = \Omega(U_0)}$, we are given an **initial q-coloring**, and the fractionality of the fractional label assignment is **$1/F$-fractional**, then one can compute an **integral label assignment** with potential

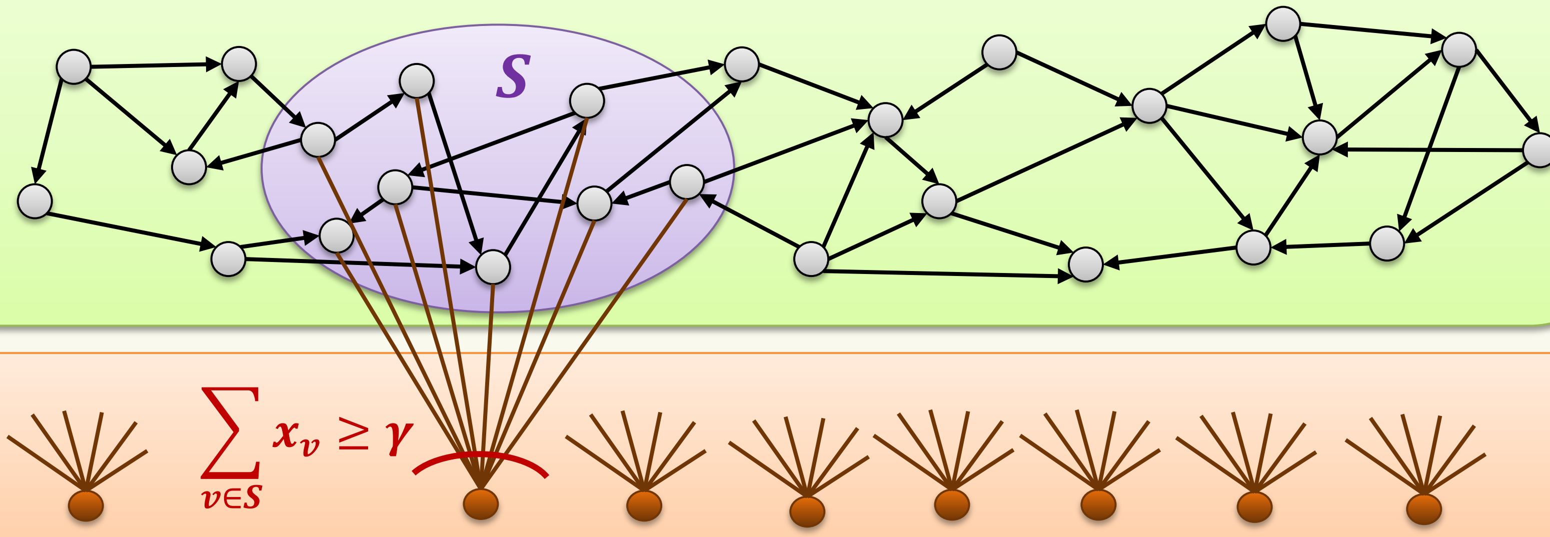
$$\mathbf{\Phi = \Omega(\Phi_0)}$$

in $\mathbf{O\left(\log^2 F + \log^* q\right)}$ **rounds.**

Highlevel Idea for MIS Algorithm

Oriented graph G:

each node v has a fractional value x_v s.t. $\sum_{u \in N_{out(v)}} x_u \leq \lambda$



Set U of subsets of V: $\forall S: \sum_{v \in S} x_v \geq \gamma$

Goal: Compute an indep. set of G that hits a constant fraction of the sets

- In our randomized alg., each v joins ind. set with prob. $\geq (1 - \lambda) \cdot x_v$
- Each of the sets S in U is hit with constant probability
- Let's try to set up edge potentials that allow to derandomize this alg.

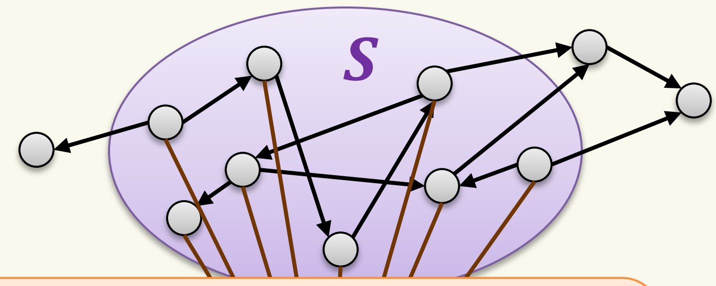
Building a Potential Function : First Try

Define $\boldsymbol{\sigma_S \coloneqq \sum_{v\in S} x_v}$ (note that $\sigma_S \geq \gamma$)

$$\forall v \sum_{u\in N_{out}(v)} x_u \leq \lambda$$

Break down potential as $\boldsymbol{\Phi = \sum_{S\in U} \Phi_S}$**:**

$$\Phi_S \coloneqq \frac{1}{\sigma_S}\cdot\left[\sum_{v\in S} x_v \cdot\left(1 - \sum_{u\in N_{out}(v)} x_u\right)\right]$$



Initial Potential:

$$\Phi_S \leq 1, \qquad \Phi_S \geq 1-\lambda$$

Total initial potential Φ is proportional to number of the sets in U.

After Rounding if #nodes $\boldsymbol{v \in S}$ **with** $\boldsymbol{x_v = 1}$ **is** $\boldsymbol{k}$**:**

$$\Phi_S \leq \frac{1}{\sigma_S}\cdot k$$

After rounding, $\Phi_S \gg 1$ possible:
Number of sets hit can be much smaller than Φ

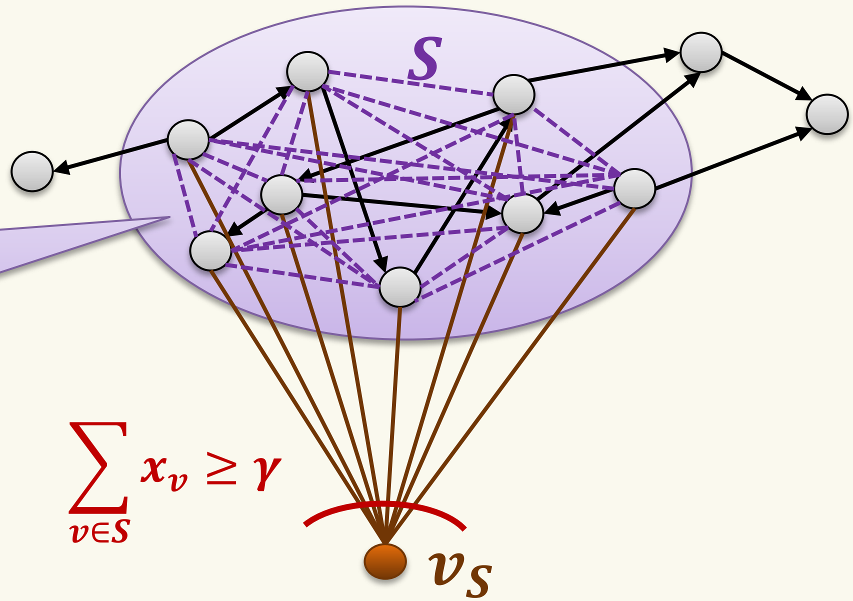
Building a Potential Function : Second Try

Define $\boldsymbol{\sigma_S} \coloneqq \sum_{v \in S} \boldsymbol{x_v}$ (note that $\sigma_S \geq \gamma$)

Break down potential as $\boldsymbol{\Phi} = \sum_{S \in U} \boldsymbol{\Phi_S}$:

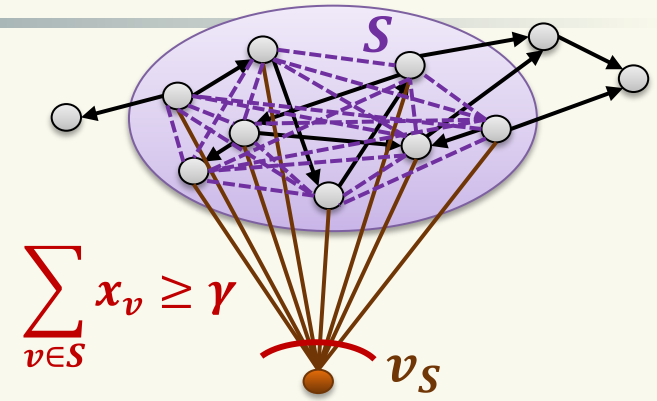
$$\Phi_S \coloneqq \frac{1}{\sigma_S} \cdot \left[\sum_{v \in S} x_v \cdot \left(1 - \sum_{u \in N_{out}(v)} x_u \right) \qquad \right]$$

The rounding graph also needs to have virtual edges between all nodes in S



Building a Potential Function : Second Try

Define $\boldsymbol{\sigma_S} \coloneqq \sum_{v\in S} \boldsymbol{x_v}$ (note that $\sigma_S \geq \gamma$)



Break down potential as $\boldsymbol{\Phi} = \sum_{S\in U} \boldsymbol{\Phi_S}$:

$$\Phi_S \coloneqq \frac{1}{\sigma_S} \cdot \left[\underbrace{\sum_{v\in S} x_v}_{=\sigma_S} \cdot \left(1 - \underbrace{\sum_{u\in N_{out}(v)} x_u}_{\leq \lambda}\right) - \frac{\mu}{\sigma_S} \underbrace{\sum_{\{u,v\}\in\binom{S}{2}} x_u \cdot x_v} \right]$$

if

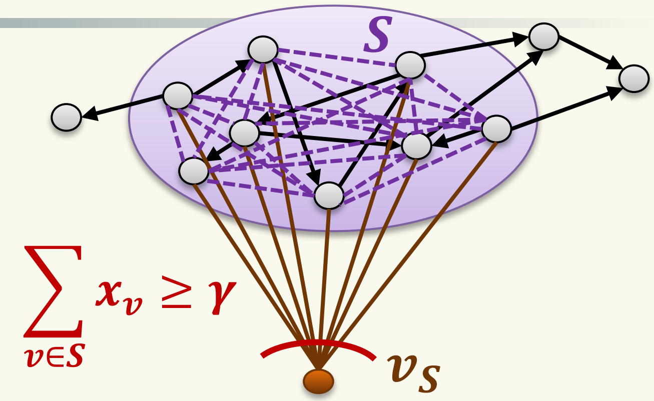
Total initial potential Φ is proportional to number of the sets in U.

Initial Potential:

$1 \geq \Phi_S$

Building a Potential Function : Second Try

Define $\boldsymbol{\sigma_S} \coloneqq \sum_{v\in S} \boldsymbol{x_v}$ (note that $\sigma_S \geq \gamma$)



Break down potential as $\boldsymbol{\Phi} = \sum_{S\in U} \boldsymbol{\Phi_S}$:

$$\Phi_S \coloneqq \frac{1}{\sigma_S} \cdot \left[\sum_{v\in S} x_v \cdot \left(1 - \sum_{u\in N_{out}(v)} x_u\right) - \frac{\mu}{\sigma_S} \sum_{\{u,v\}\in\binom{S}{2}} x_u \cdot x_v \right]$$

Potential after rounding if $\boldsymbol{k}$ nodes in $\boldsymbol{S}$ have $\boldsymbol{x_v = 1}$:

- If $\Phi_S > 0$, then $k \geq 1$ and thus, the set S is hit
- Φ_S can be upper bounded as

$$\Phi_S \leq \frac{1}{\sigma_S} \cdot \left[k - \frac{\mu}{\sigma_S} \cdot \binom{k}{2}\right] = \left(\frac{1}{\sigma_S} + \frac{\mu}{2\sigma_S^2}\right) \cdot k - \frac{\mu}{2\sigma_S^2} \cdot k^2$$

We have $\boldsymbol{\sigma_S \coloneqq \sum_{v\in S} x_v \geq \gamma = \Theta(1)}$

Potential after rounding if $\boldsymbol{k}$ nodes in $\boldsymbol{S}$ have $\boldsymbol{x_v = 1}$:

- If $\Phi_S > 0$, then $k \geq 1$ and thus, the set S is hit
- Φ_S can be upper bounded as

$$\Phi_S \leq \frac{1}{\sigma_S} \cdot \left[k - \frac{\mu}{\sigma_S} \cdot \binom{k}{2}\right] = \left(\frac{1}{\sigma_S} + \frac{\mu}{2\sigma_S^2}\right) \cdot k - \frac{\mu}{2\sigma_S^2} \cdot k^2$$

- Φ_S is maximized for $k = \frac{\sigma_S}{\mu} + \frac{1}{2}$
- We then have

$$\Phi_S \leq \left(\frac{1}{\sigma_S} + \frac{\mu}{2\sigma_S^2}\right) \cdot k = \frac{1}{\mu} + \frac{1}{\sigma_S} + \frac{\mu}{4\sigma_S^2} \leq \frac{1}{\mu} + \frac{1}{\gamma} + \frac{\mu}{4\gamma^2} = O(1)$$

Building a Potential Function : Summary

Potential contribution of set S

$$\Phi_S := \frac{1}{\sigma_S} \cdot \left[\sum_{v \in S} x_v \cdot \left(1 - \sum_{u \in N_{out}(v)} x_u \right) \; - \; \frac{\mu}{\sigma_S} \sum_{\{u,v\} \in \binom{S}{2}} x_u \cdot x_v \right]$$

Initial potential of fractional solution: $\boldsymbol{\Phi_S = \Theta(1)}$

Potential after rounding :

- If $\Phi_S > 0$, then the set S is hit and we have $\boldsymbol{\Phi_S = O(1)}$

- Initially, we have $\Phi = \Theta(|U|)$
- If we also have $\Phi = \Theta(|U|)$ after rounding, a constant fraction of the sets S in U are hit

Relation to MIS

Luby's MIS Algorithm

- Each node v tries to join the MIS with prob. $x_v = \dfrac{1}{2\deg(v)}$

Application of the generic rounding algorithm:

Given an initial $\Delta^{O(1)}$-coloring of the rounding graph, we can find an independent set that covers a constant fraction of all edges in $O(\log^2 \Delta)$ rounds and thus an MIS in $O(\log^2 \Delta \cdot \log n)$ rounds.

- Edge $\{u, v\}$ is called good if u or v is good
  - A constant fraction of the edges is good
  - Hitting set problem: define a set S for each good edge e, where S consists of all nodes of edges adjacent to e (all nodes that remove e)
  - For each such set S, we have $\sum_{v \in S} x_v \geq \frac{1}{2} =: \gamma$

Application of Rounding to Other Problems

Remark: $O(\log^2 \Delta \cdot \log n)$-round MIS algorithm implies $O(\log^2 \Delta \cdot \log n)$-round algorithms for $(\Delta + 1)$-coloring, maximal matching, edge coloring.

We can also get those results more directly:

- **Maximal matching:** sufficient to apply the *„large independent set" algo.* on the line graph with an appropriate initial fractional solution.
- **$(\boldsymbol{\Delta + 1})$-coloring:** *Fractional solution* of each node s given by a *probability distribution over colors* for this node.

Beyond algorithms for the four standard problems

- An $O(\log^2 \Delta + \log^* n)$-round algorithm to *compute large ind. sets*
  - $(1 - \varepsilon)/\beta$-approximation with neighborhood independence β
- $O(\log s \cdot \log^2 t + \log^* n)$-round algorithm for $O(\log s)$-approximation of *minimum set cover*
  - s: maximum set size, t: maximum number of sets containing each element

Application of Rounding to Other Problems

New algorithm for computing a network decomposition

[Ghaffari, Grunau, Haeupler, Ilchi, Rozhoň; SODA '23]

$\big(\boldsymbol{O}(\log n), \boldsymbol{O}(\log n)\big)$**-network decomposition**
in $\boldsymbol{O}\big(\log^3 n \cdot \text{poly}\log\log n\big)$ **rounds**.

- improves over the $O(\log^5 n)$-round alg. of [Ghaffari,Grunau, Rozhoň; SODA '21]
- almost the same result also in the CONGEST model

Even faster algorithm for MIS, etc. [Ghaffari, Grunau; STOC '23]

MIS in $\boldsymbol{O}\big(\log^2 n \cdot \text{poly}\log\log n\big)$ **rounds**.

- implies the same bound for all four classic problems
- Requires the LOCAL model
- $O(1)$-approximation for maximum matching in $O\big(\log^{4/3} n \cdot \text{poly}\log\log n\big)$ rounds.

Thanks!