Chapter 8
Distributed Coloring & MIS II: Randomization

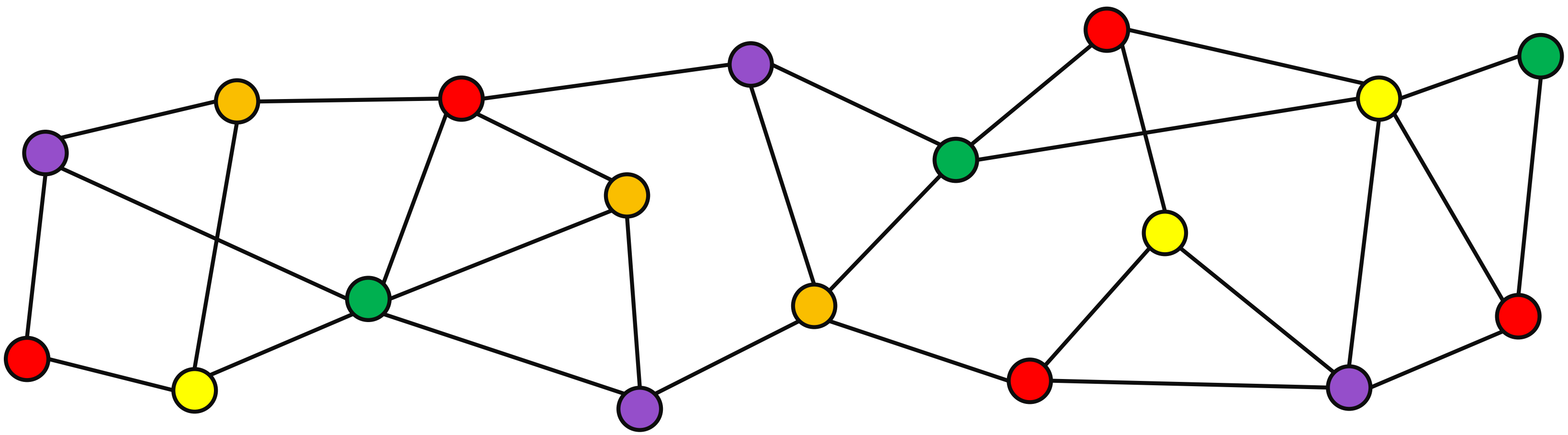
Distributed Systems

Summer Term 2020

Fabian Kuhn

Distributed Coloring Problem

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

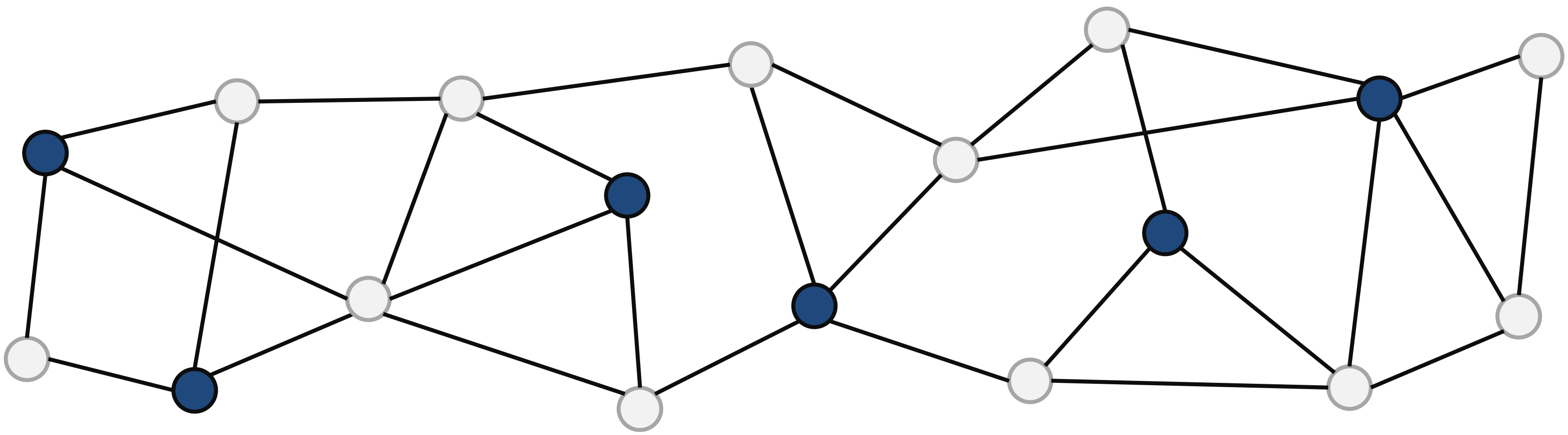


Objective: properly color the nodes with $\leq \Delta + 1$ colors

- Δ : maximum degree
- $\Delta + 1$ colors: what a simple sequential greedy algorithm achieves

Maximal Independent Set

Maximal Independent Set (MIS)



Objective: compute a maximal independent set (MIS)

- **Independent Set:** set of pairwise non-adjacent nodes
- **Maximal:** adding any additional node destroys independence (non-extendible set of pair-wise non-adjacent nodes)

Summary from last time

Coloring Trees

- Trees can be colored with 2 colors, this however requires time $\Omega(D)$.
- Rooted trees can be 3-colored in time $O(\log^* n)$.
- Unrooted trees can be 3-colored in time $O(\log n)$.

Coloring General Graphs

- 3^Δ-coloring of graphs with max. degree Δ in time $O(\log^* n)$
- $(\Delta + 1)$-coloring of graphs with max. degree Δ in time $O\left(3^\Delta + \log^* n\right)$
  - If $\Delta = O(1)$, this is $O(\log^* n)$.
  - This algorithm can be improved significantly.

Today

- So far, we looked at deterministic algorithms, today we will look at randomized algorithms for $(\Delta + 1)$-coloring and MIS in general graphs.

Randomized Coloring : Ideas

How can we use randomization to color a graph?

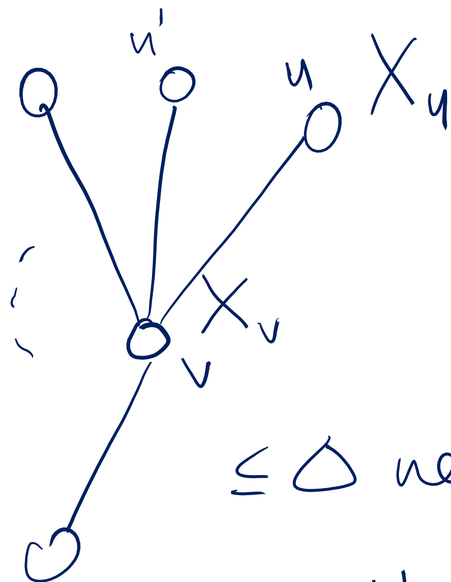
- Assume that each node v should get a color in $\{1, \dots, \Delta + 1\}$
  - The sequential greedy algorithm guarantees this…

Simple Idea:

- Each node could pick a random color
- If no neighbor picks the same color, the node can keep the color
- Repeat until all nodes are colored

Random Colors

Lemma: If each node $v \in V$ of a graph $G = (V, E)$ independently picks a uniformly random color X_v from $\{1, \dots, \Delta + 1\}$, for each node $v \in V$, the probability that $X_v \neq X_u$ for all neighbors u of v is at least ${}^1/_e$.



$\mathbb{P}(X_u = X_v) = \frac{1}{\Delta + 1}$

$\leq \Delta$ neighbors

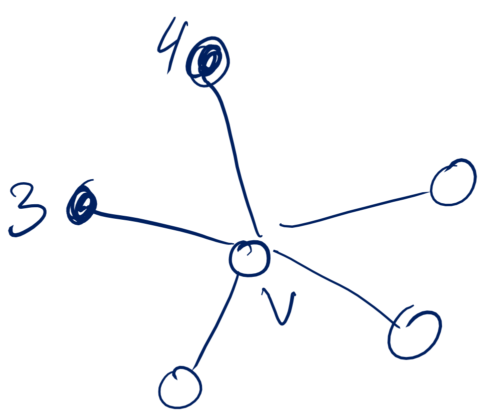
$\mathbb{P}(\text{all neighbors choose diff colors}) \geq \left(1 - \frac{1}{\Delta+1}\right)^{\Delta} > \frac{1}{e}$

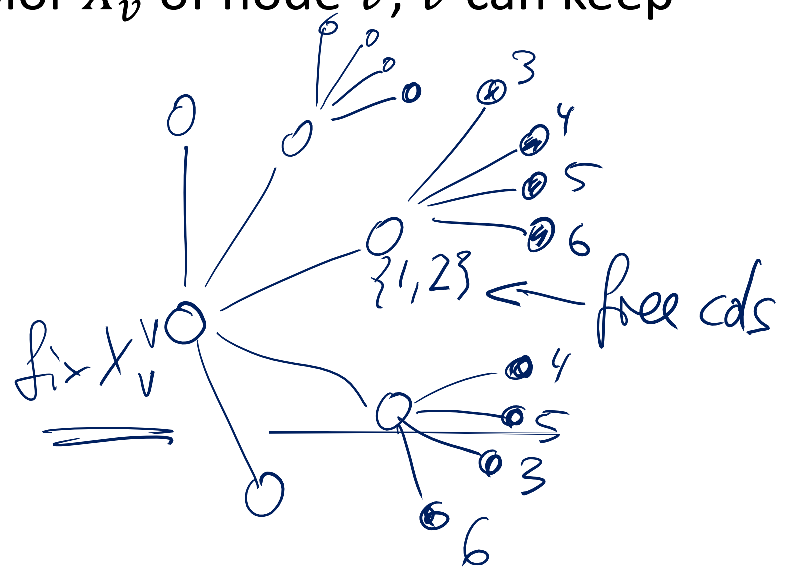
Extending an Existing Coloring

- Assume that some nodes of a graph already have a color
  – So that no two neighbors have the same color

- Each uncolored node now picks a random color
  – If we want the already colored nodes to keep their colors, it makes sense to only pick the random color among the colors that are still free for the node.
  – What is now probability that a node can keep its color?

- Is it still true that for every possible color X_v of node v, v can keep color X_v with constant probability?





Extending an Existing Coloring

Let's directly make the problem a bit harder (and also more precise)

- Each node $v \in V$ in $G = (V, E)$ should get a color in $\{1, \dots, \deg(v) + 1\}$
- Subset $V_C \subset V$ of the nodes: nodes in $v \in V_C$ already have a color x_v such that the induced subgraph $G[V_C]$ is properly colored
- $V_U := V \setminus V_C$ is the set of uncolored nodes, for all $u \in V_U$, we define

$$F_u := \{1, \dots, \deg(u) + 1\} \setminus \bigcup_{v \in V_C \cap N(u)} x_v$$

free colors of u

as the set of free colors for node u.

- Each node $u \in V_U$ picks a color X_u uniformly at random from F_u
- Node $u \in V_U$ keeps the color if no neighbor in V_U picks the same random color.
  - There clearly cannot be any conflicts with the already colored nodes in V_C.

Extending an Existing Coloring

Weight $\boldsymbol{w_x}$ of a color $\boldsymbol{x \in F_u}$ for $\boldsymbol{u \in V_U}$:

- $N_x(u)$: uncolored neighbors u' of u for which $x \in F_{u'}$. Then,

nodes in $V_U \cap N(u)$, s.t. $x \in F_u$

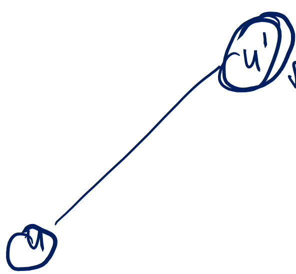
$$w_x := \sum_{u' \in N_x(u)} \frac{1}{|F_{u'}|}$$

prob. that neighbor u' picks color x

Intuition: Weight w_x corresponds to probability that some neighbor of u picks color x as its random color.

Lemma: For every $u \in U$, we have $\boxed{\sum_{x \in F_u} w_x \leq |N(u) \cap V_U|} \leq |F_u| - 1 \; (\leq |F_u|)$

set of uncolored neighbors.



contributes $\frac{1}{|F_{u'}|}$ to $\sum_x w_x$ for each $x \in F_u \cap F_{u'}$

total contribution of u' to $\sum_{x \in F_u} w_x$ is ≤ 1

Extending an Existing Coloring

Lemma: If node $u \in V_U$ picks the random color $x \in F_u$, the probability that u can keep its color is at least 4^{-w_x}.

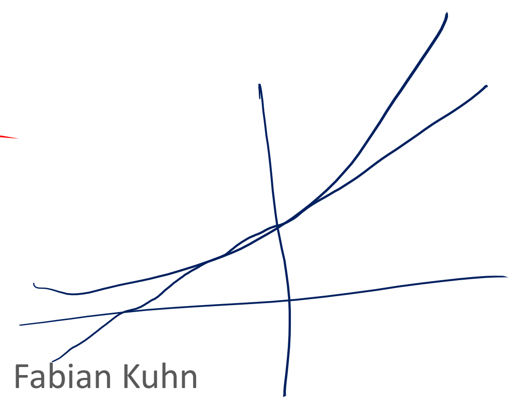
Assume u picks color x as random color

$$\mathbb{P}(u \text{ can keep color}) = \prod_{u' \in N_x(u)} \left(1 - \frac{1}{|F_{u'}|}\right) \geq \prod_{u' \in N_x(u)} \cancel{e}\,4^{-1/|F_{u'}|}$$

$$= \cancel{e}\,4^{-\sum_{u' \in N(u)} 1/|F_{u'}|} = \cancel{e}\,4^{-w_x}$$

$$\boxed{\forall x \in [0, 1/2] : 1 - x \geq 4^{-x}}$$

Ineq: $\forall x \in \mathbb{R} : 1 + x \leq e^x$



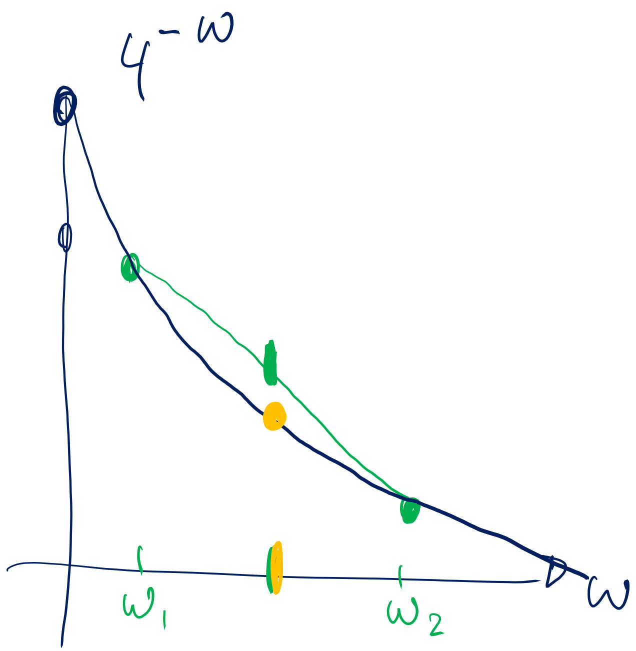
Extending an Existing Coloring

Theorem: The probability that a node $u \in V_U$ can keep its random color is at least $1/_4$.

$$\mathbb{P}(u \text{ keeps color } X_u) = \sum_{x \in F_u} \frac{1}{|F_u|} \cdot \underbrace{\mathbb{P}(u \text{ keeps color} \mid X_u = x)}_{\text{prev. lemma: } \geq 4^{-w_x}}$$

$$= \sum_{x \in F_u} \frac{1}{|F_u|} \cdot 4^{-w_x} \geq 4^{-\frac{1}{|F_u|}\sum_{x \in F_u} w_x} > \frac{1}{4}.$$

convexity of 4^{-w}



Randomized Coloring

Theorem: The discussed randomized coloring algorithm computes a valid coloring of G in $O(\log n)$ rounds in expectation and w.h.p. Every node $v \in V$ gets a color in $\{1, \ldots, \deg(v) + 1\}$.

$$\mathbb{P}(\text{node } v \text{ uncolored after } T \text{ phases}) \le \left(\frac{3}{4}\right)^T = \frac{1}{n^c}$$

$$T = c \cdot \log_{4/3}(n)$$

$$\mathbb{P}(\text{some node uncolored after } T \text{ phases}) \le n \cdot \left(\frac{3}{4}\right)^T = \frac{1}{n^{c-1}}$$

union bound

$$\mathbb{P}(A \cup B) \le \mathbb{P}(A) + \mathbb{P}(B)$$

Randomized MIS : Ideas

- After solving coloring in only $O(\log n)$ rounds, let's also try MIS...

Ideas?

- With the coloring algorithm from before and the reduction from MIS to coloring from last lecture, we get an algorithm that runs in time

$$O(\Delta + \log n).$$

- We could run the distributed greedy algorithm of the last lecture with random IDs in $O(\log n)$ rounds
  - This actually works, but the analysis is highly non-trivial...

- We are going to slightly adapt this. Let's just look at one round of the greedy algorithm with random IDs.

$R_v^0 < R_u$

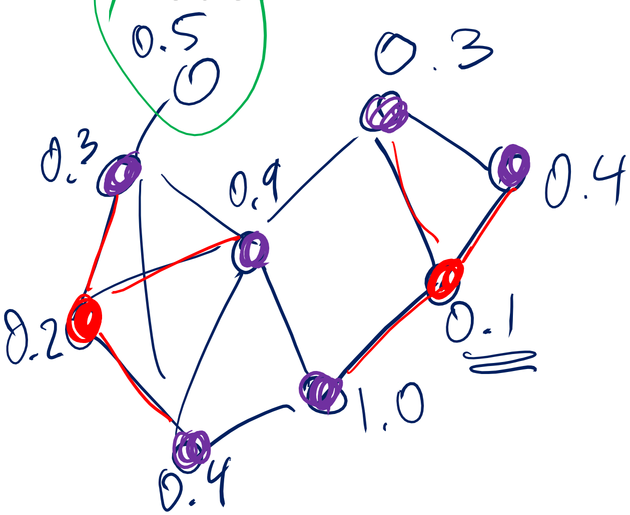
1. Each node $v \in V$ picks a random number $R_v \in [0,1]$
2. Node v joins the MIS if $R_v < R_u$ for all neighbors $u \in N(v)$
   - Afterwards, we can continue with new random IDs on the remaining graph ...

Randomized MIS : Luby's Algorithm

Repeat the following phase on all alive nodes:

1. Each node $v \in V$ picks a random number $R_v \in [0,1]$
2. Node v joins the MIS if $R_v < R_u$ for all neighbors $u \in N(v)$

- After each phase, the nodes that join the MIS and the nodes that have a neighbor that joins the MIS are removed from the graph of alive nodes.



$\mathbb{P}(v \text{ joins MIS}) = \frac{1}{d+1}$

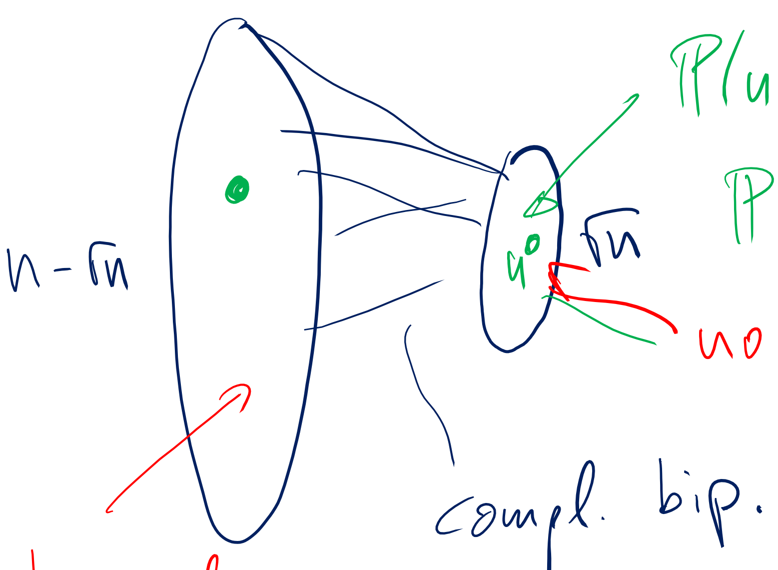
degree d in current graph

- Can we show that a large fraction of nodes is removed from the graph?

Randomized MIS : Luby's Algorithm

1. Each node $v \in V$ picks a random number $R_v \in [0,1]$
2. Node v joins the MIS if $R_v < R_u$ for all neighbors $u \in N(v)$

- The fraction of nodes deleted from the graph might be small.



$\mathbb{P}(u \text{ join MIS}) \approx \frac{1}{n}$

$\mathbb{P}(\text{some node from the right joins MIS}) \approx \frac{1}{\sqrt{n}}$

no node in MIS

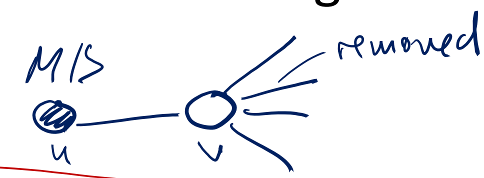
only nodes that join MIS are removed → every node with prob. $\approx \frac{1}{\sqrt{n}}$

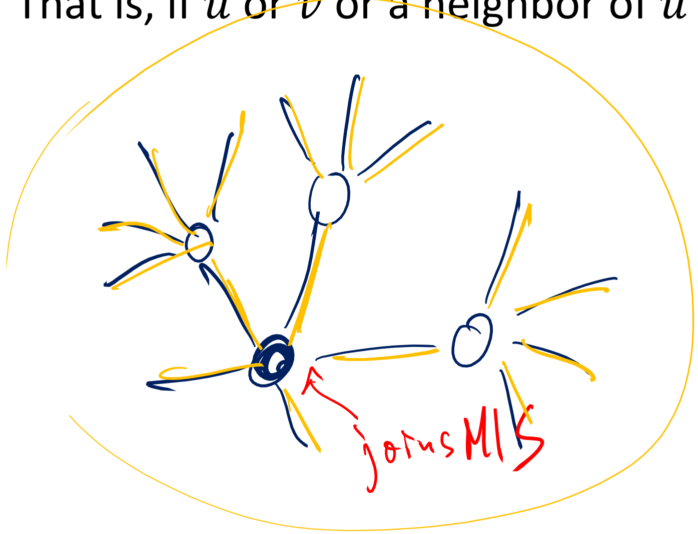
- How else can we show that we make fast progress?

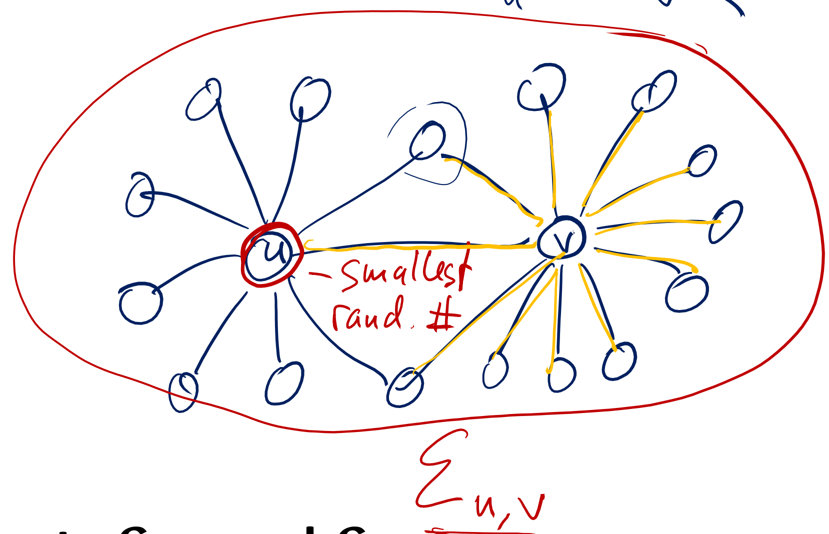
Randomized MIS : Luby’s Algorithm

Lemma: In expectation, at least half of the remaining edges are removed.

- Note that an edge $\{u, v\}$ gets removed if one of its nodes u or v gets removed.
  - That is, if u or v or a neighbor of u or v joins the MIS.







For each edge $\{u, v\}$, we define events $\mathcal{E}_{u,v}$ and $\mathcal{E}_{v,u}$:

$$\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$$

- If $\mathcal{E}_{u,v}$ is true, in particular all edges of v are deleted.

Randomized MIS : Luby's Algorithm

Lemma: In expectation, at least half of the remaining edges are removed.

For each edge $\{u, v\}$, we define events $\mathcal{E}_{u,v}$ and $\mathcal{E}_{v,u}$:

$$\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$$

- If $\mathcal{E}_{u,v}$ is true, in particular all edges of v are deleted.

We define random variables:

$$X_{u,v} := \begin{cases} \deg(v) & \text{if } \mathcal{E}_{u,v} \text{ holds} \\ 0 & \text{otherwise} \end{cases}, \qquad X := \sum_{\{u,v\} \in E} \left(X_{u,v} + X_{v,u}\right)$$

Randomized MIS : Luby's Algorithm

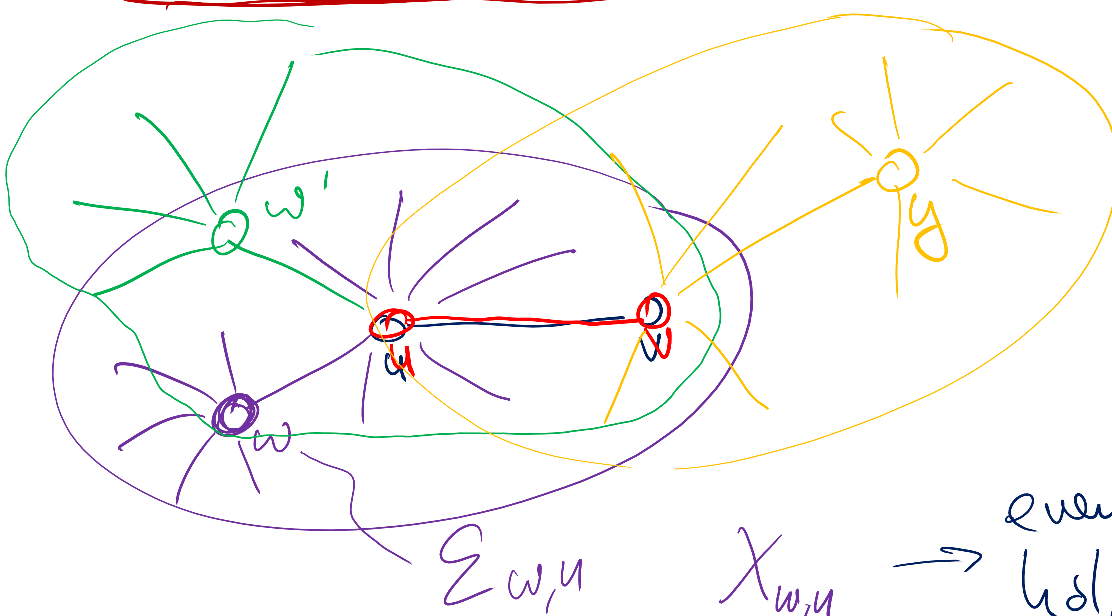
Lemma: In expectation, at least half of the remaining edges are removed.

Random Variables:

$$X_{u,v} := \begin{cases} \deg(v) & \text{if } \mathcal{E}_{u,v} \text{ holds} \\ 0 & \text{otherwise} \end{cases}, \qquad X := \sum_{\{u,v\} \in E} \left(X_{u,v} + X_{v,u} \right)$$

every deleted edge is counted at most twice

Claim: $X \leq 2 \cdot \#\text{deleted edges}.$



event $\mathcal{E}_{w,u}$ can only hold for one node w

Randomized MIS : Luby's Algorithm

Lemma: In expectation, at least half of the remaining edges are removed.

Random Variables:

$$X_{u,v} := \begin{cases} \deg(v) & \text{if } \mathcal{E}_{u,v} \text{ holds} \\ 0 & \text{otherwise} \end{cases}, \qquad X := \sum_{\{u,v\} \in E} \left(X_{u,v} + X_{v,u} \right)$$

Claim: $X \leq 2\cdot$ #deleted edges and $\mathbb{E}[X] \geq |E|/\cancel{2}$.

$$\mathbb{E}[X_{u,v}] = \deg(v) \cdot \mathbb{P}(\underline{\mathcal{E}_{u,v}}) \geq \frac{\deg(v)}{\deg(u)+\deg(v)}$$

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{\{u,v\} \in E} (X_{uv} + X_{vu}) \right]$$

Expectation of Sum of Random Variables

Linearity of Expectation:

For random variables X and Y, we have

$$\mathbb{E}[\boldsymbol{X}+\boldsymbol{Y}]=\mathbb{E}[\boldsymbol{X}]+\mathbb{E}[\boldsymbol{Y}]$$

Proof:

$$\mathbb{E}[X+Y]=\sum_{x,y}(x+y)\cdot\Pr(X=x\ \wedge\ Y=y)$$

$\Pr(X=x)$ $\qquad$ $\Pr(Y=y)$

$\mathbb{E}[X]$ $\qquad$ $\mathbb{E}[Y]$

Randomized MIS : Luby's Algorithm

Lemma: In expectation, at least half of the remaining edges are removed.

Random Variables:

$$X_{u,v} := \begin{cases} \mathbf{deg}(v) & \text{if } \mathcal{E}_{u,v} \text{ holds} \\ \mathbf{0} & \text{otherwise} \end{cases}, \qquad X := \sum_{\{u,v\} \in E} \left(X_{u,v} + X_{v,u}\right)$$

Claim: $X \leq 2 \cdot \#\text{deleted edges}$ and $\mathbb{E}[X] \geq |E|\cancel{/2}$.

$$\mathbb{E}[X_{u,v}] \geq \frac{\deg(v)}{\deg(u)+\deg(v)}$$

$$\mathbb{E}[X] = \sum_{\{u,v\}\in E} \left(\mathbb{E}[X_{u,v}] + \mathbb{E}[X_{v,u}]\right)$$

$$\geq \sum_{\{u,v\}\in E} \underbrace{\left(\frac{\deg(v)}{\deg(u)+\deg(v)} + \frac{\deg(u)}{\deg(u)+\deg(v)}\right)}_{=1} = |E|$$

Randomized MIS : Luby's Algorithm

Lemma: In expectation, at least half of the remaining edges are removed.

Theorem: Luby's randomized MIS algorithm computes an MIS in time $O(\log n)$ in expectation (and also w.h.p.).

with prob. $\geq \frac{1}{3}$, at least $\frac{|E|}{4}$ are deleted

Randomized MIS : Luby's Algorithm

Lemma: In expectation, at least half of the remaining edges are removed.

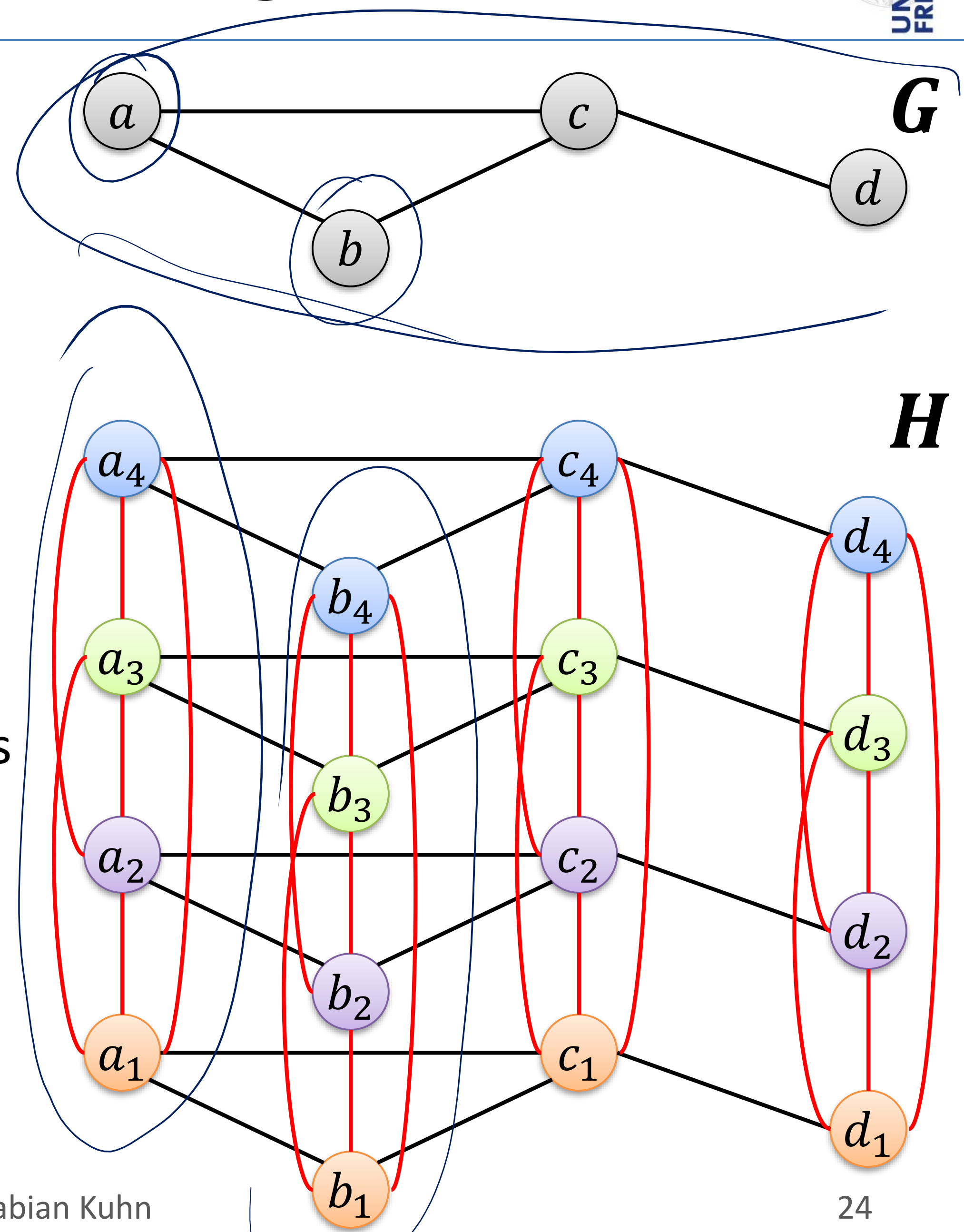
Theorem: Luby's randomized MIS algorithm computes an MIS in time $O(\log n)$ in expectation (and also w.h.p.).

From MIS to $(\Delta + 1)$-Coloring

Assume, we want to compute a coloring on graph G.

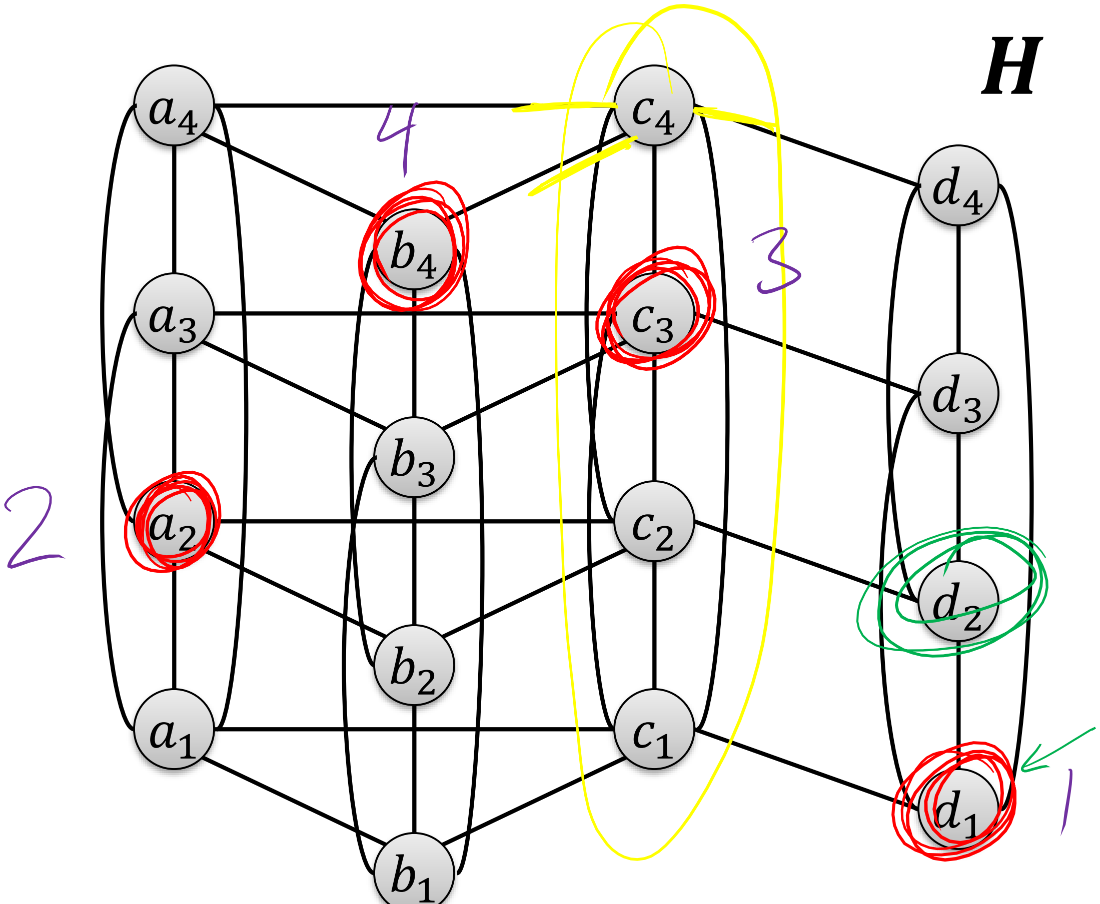
- We transform G into a new virtual graph H.
  - That can be simulated on G.

1. Create $\Delta + 1$ copies of G
2. Connect corresponding nodes in the copies to a clique.
3. Compute MIS on H.



From MIS to $(\Delta + 1)$-Coloring

Claim: MIS of H contains exactly one node from each column. If in column corresponding to some node v, node v_i is in the MIS, then in G node v can be colored with color $i \in \{1, \dots, \Delta + 1\}$.



≤1 node per column

=1 node " "

From MIS to $(\Delta + 1)$-Coloring

Theorem: Together with the randomized MIS algorithm, the reduction gives an alternative distributed algorithm to compute a $(\Delta + 1)$-coloring in $O(\log n)$ rounds.

Remark: The reduction can be adapted to assign a color from the set $\{1, \dots, \deg(v) + 1\}$ to each node v.

- It suffices to have $\deg(v) + 1$ copies of node v
- The additional copies can be removed from H.

Randomized Coloring & MIS : Summary

- We saw that there are randomized distributed $O(\log n)$-round algorithms to compute a $(\Delta+1)$-coloring or an MIS on a general graph.
  - The randomized MIS alg. and the coloring to MIS reduction is due to [Luby '86]
- Very recently (in July 2019), Rozhoň and Ghaffari showed that there are even deterministic distributed algorithms to solve these problems in time $O(\log^c n)$.
  - This was an open problem for 30+ years.
- The best randomized algorithms have the following time complexities
  - MIS: $O(\log\Delta + \log^c \log n)$ [Ghaffari '16]
  - $(\Delta+1)$-coloring: $O(\log^c \log n)$ [Chang, Li, Pettie '18]
- The best lower bounds are
  - $(\Delta+1)$-coloring: $\Omega(\log^* n)$ [Linial '87] 2
    (even for $\Delta = 3$, see next lecture)
  - MIS (randomized): $\Omega\left(\sqrt{\log n/\log\log n}\right)$ [Kuhn, Moscibroda, Wattenhofer '04]
  - MIS (deterministic): $\Omega(\log n/\log\log n)$ [Balliu, Brandt, Hirvonen, Olivetti, Rabie, Suomela '19]