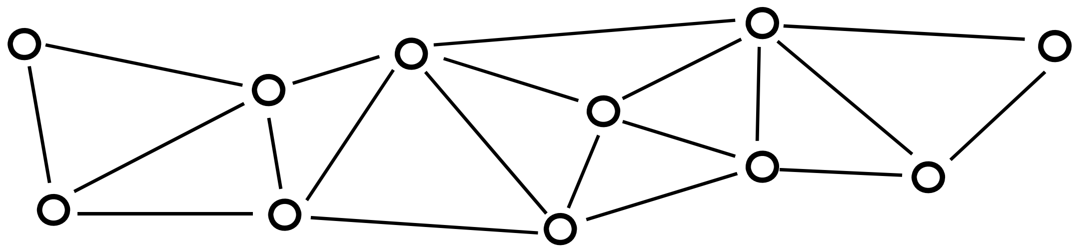
Algorithms and Lower Bounds for Distributed Coloring Problems

Fabian Kuhn

Parts are joint work with
Nicla Bernasconi, Dan Hefetz, Angelika Steger

Distributed Coloring

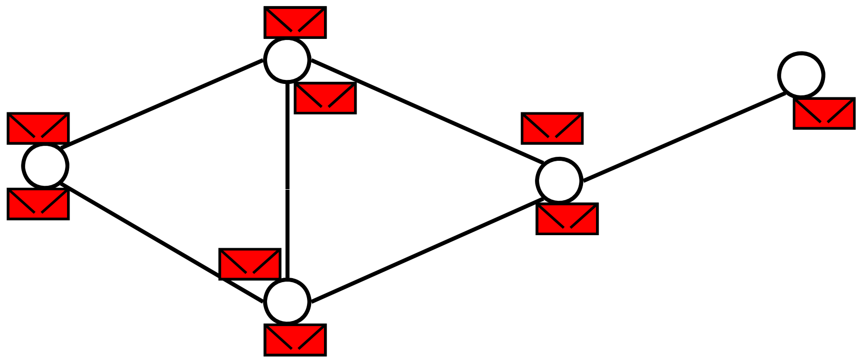
- Given: Network = Graph G



- Problem: Compute coloring of G by a distributed algorithm
- Goal: Minimize number of colors
- Applications:
  – Frequencies (FDMA), time slots (TDMA) in wireless network MAC prot.
  – Other distributed coordination tasks

Communication Model

- We use synchronous message passing model
- Network = graph (nodes: devices, edges: direct comm. links)
- Time is divided into rounds:



Each node sends message to each of its neighbors

time complexity = number of rounds

Symmetry Breaking

- Main challenge: How to break symmetries?

- Two ways to break symmetries:
  1. Randomization
  2. Deterministic Symmetry Breaking
     → Nodes have unique IDs or some other a priory labeling

- This talk: Mostly about deterministic symmetry breaking

Previous Work: Deterministic Algorithms

- 3-coloring of the ring in $O(\log^* n)$ rounds [Cole,Vishkin 86]
- Lower Bound: $\Omega(\log^* n)$ rounds needed for $O(1)$-coloring a ring [Linial 92]

- General Graphs:
  - $(\Delta+1)$-coloring in time $O(\Delta^2+\log^* n)$ [Goldberg,Plotkin,Shannon 88]
  - $O(\Delta^2)$-coloring in time $O(\log^* n)$ [Linial 92]
  - $(\Delta+1)$-coloring in time $O(\Delta\cdot\log n)$ [Awerbuch et al. 89]
    → Combining the last two: $(\Delta+1)$ colors in $O(\Delta\cdot\log\Delta+\log^* n)$ rounds
  - Using network decompositions: $(\Delta+1)$ colors in time $2^{O(\sqrt{\log n})}$ [Awerbuch,Goldberg,Luby,Plotkin 89], [Panconesi,Srinivasan 95]

- Algorithms efficient if degree small
- Else: Huge gap between upper and lower bounds

Previous Work: Randomized Algorithms

- Randomized O(Δ)-coloring in expected $O(\sqrt{\log n})$ rounds [Kothapalli,Scheideler,Onus,Schindelhauer 06]
- Randomized (Δ+1)-coloring in expected O(log n) rounds:
  - MIS in exp. O(log n) rounds [Alon,Babai,Itai 86], [Luby 86]
  - Reduction from (Δ+1)-coloring to MIS [Linial 92]

- Large gap between deterministic and randomized algorithms:

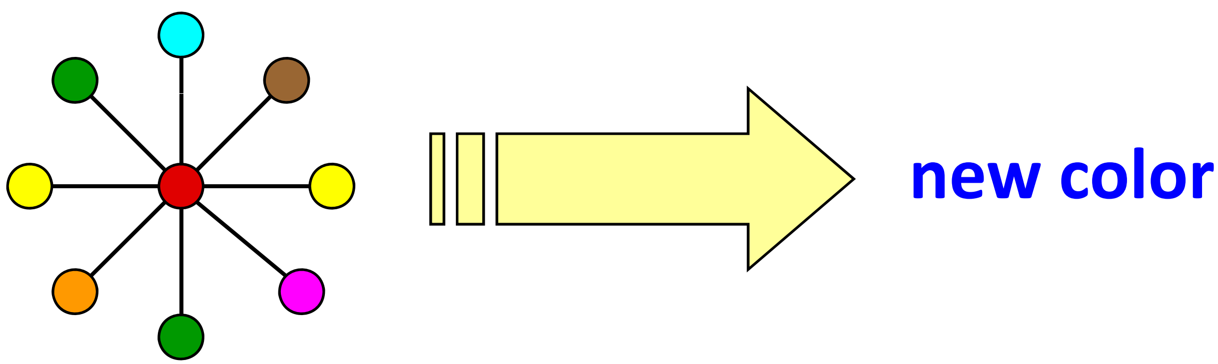
- O(Δ) colors, deterministic: $\min\left\{2^{O(\sqrt{\log n})}, O(\Delta \log \Delta + \log^* n)\right\}$

- O(Δ) colors, randomized: $O(\sqrt{\log n})$

One-Round Coloring Algorithms

- Easiest non-trivial case, most local algorithms

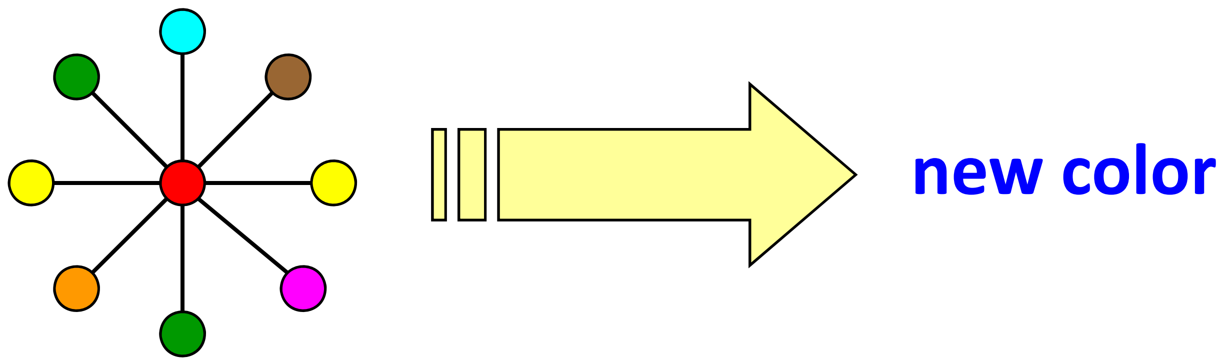
- Algorithm starts with unique IDs or initial coloring
- Each node collects IDs or initial colors of its neighbors
- Based on this information, a new color is determined



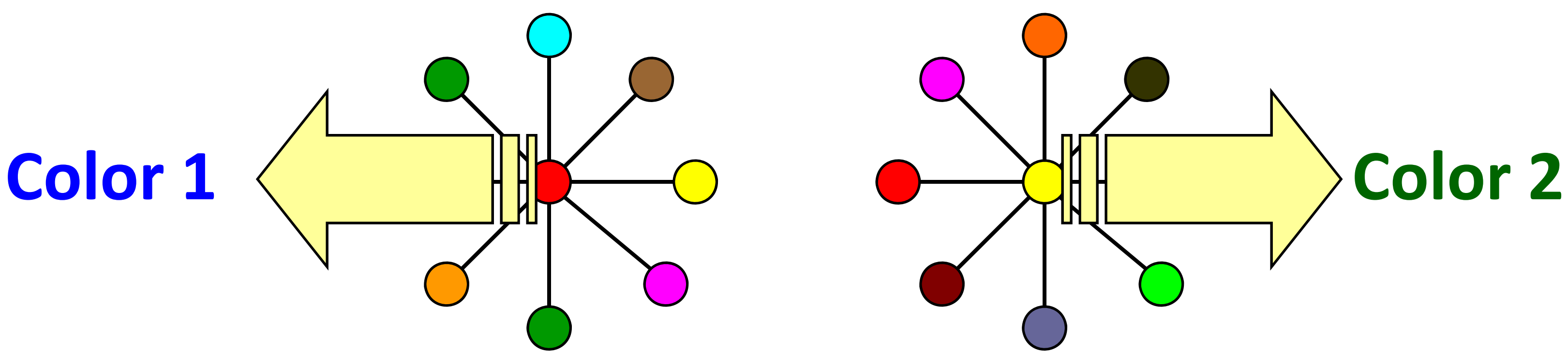
- Many existing coloring alg.: iterative applications of one-round alg.
- Results in algorithms with short messages

One-Round Coloring Algorithms

- Assign new color to every possible one-hop view



- Different colors for views of (possibly) neighboring nodes



Formally…

- Given graph with max degree Δ, initial m-coloring
- One-hop view of node u: (x_u, S_u)

  $x_u \in [m]$: color of u, $S_u \subset [m]$: colors of u's neighbors ($|S_u| \leq \Delta$, $x_u \notin S_u$)
- Views (x_u, S_u), (x_v, S_v) can be views of neighbors if $x_u \in S_v$ and $x_v \in S_u$
- q-coloring algorithm: function f: $(x,S) \to$ color $\in [q]$

  $f(x_u, S_u) \neq f(x_v, S_v)$ if $x_u \in S_v$ and $x_v \in S_u$
- q-coloring algorithm: q-coloring of neighborhood graph

  Nodes: all possible pairs (x,S), Edge if $x_u \in S_v$ and $x_v \in S_u$

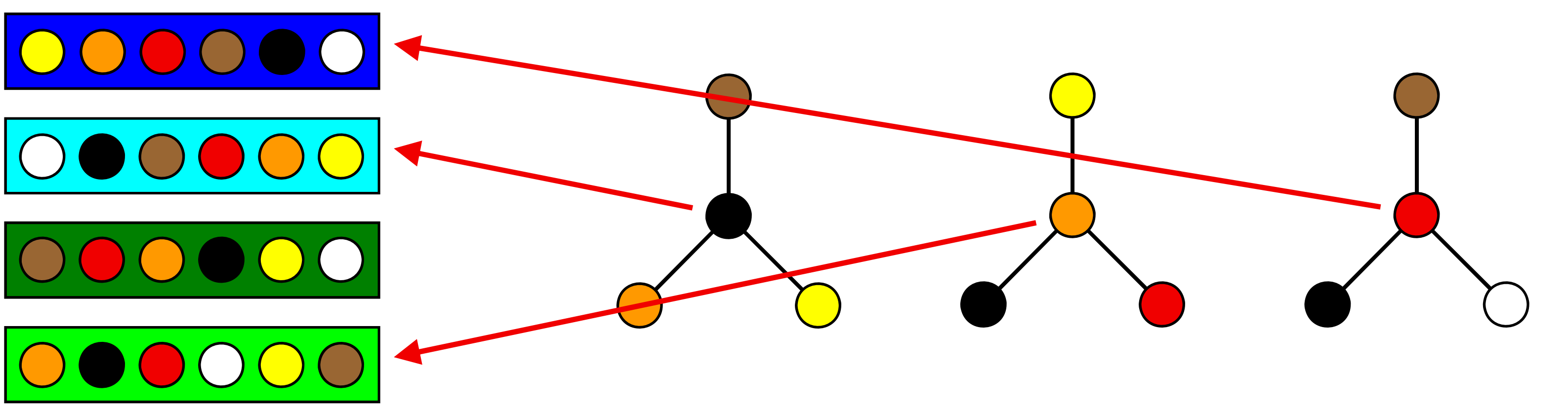
One-Round Coloring: Related Work

- Given, graph of max degree Δ, initial m-coloring
- Upper bound: $O(\Delta^2 \cdot \log m)$ [Linial 92]
- Lower bound: $\Omega(\log\log m)$ [Linial 92] (holds on ring)

Coloring Algorithm

- Algorithm Idea:
  - For each new color, determine a total order (permutation) on old colors
  - Node takes new color if its old color appears before colors of neighbors in corresponding order

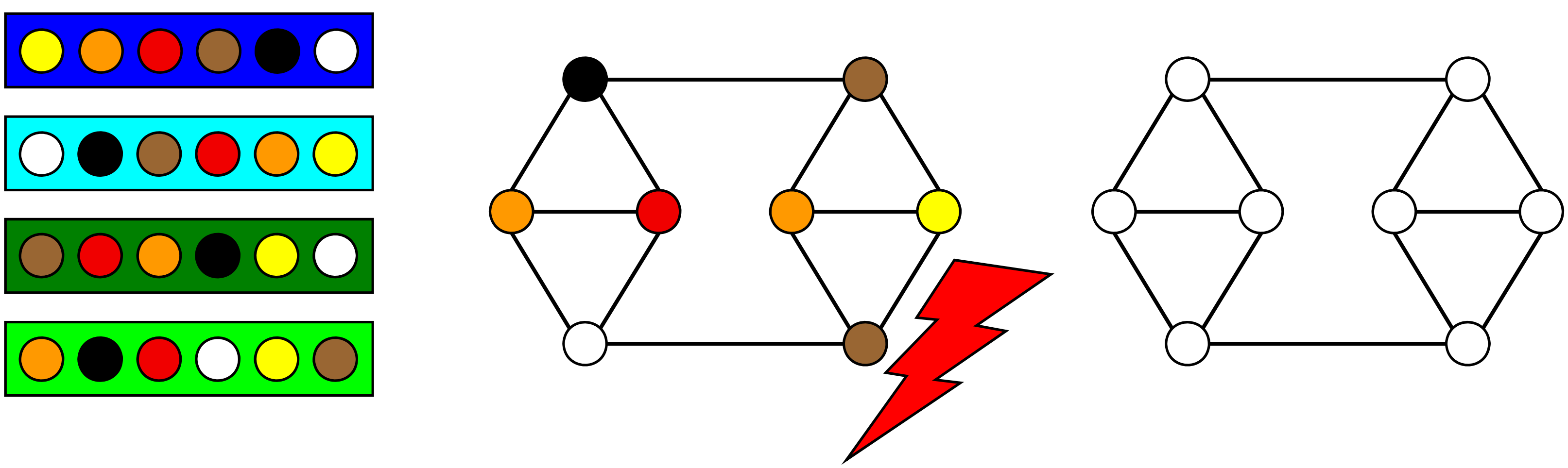
- Example: 6 old colors ( ), 4 new colors ( )



Coloring Algorithm

- Algorithm Idea:
  - For each new color, determine a total order (permutation) on old colors
  - Node takes new color if its old color appears before colors of neighbors in corresponding order

- Example: 6 old colors ( ), 4 new colors ( )



One-Round Coloring: Upper Bound

- **Theorem: $(\Delta+1)^2 \cdot \ln m$ different orders suffice**

- Proof Sketch: (simple probabilistic method proof)
  - Choose random orders
  - Any one-hop view (center color + Δ adjacent colors) covered with probability $1/(\Delta+1)$
  - Prob. that a given view is not covered by $(\Delta+1)^2 \cdot \ln m$ orders:

$$\left(1 - \frac{1}{\Delta + 1}\right)^{(\Delta+1)^2 \ln m} < \frac{1}{m^{\Delta+1}}$$

  - Number of one-hop views (center col, Δ adj. cols) is at most $m^{\Delta+1}$
  - Probability that there is a view that is not covered is < 1
  - Probability that all views are covered > 0
  - There is a set of $(\Delta+1)^2 \cdot \ln m$ orders which cover all views

Multicoloring

- Assign sets of colors such that neighboring sets are disjoint
  minimize total number of colors, maximize size of sets
- Taking $c \cdot \Delta^2 \cdot \ln m$ colors for sufficiently large constant c:
  Every possible one-hop view gets a color w.h.p.
- In fact, every node can choose many colors
- For $c = O(1/\delta^2)$, a node with degree d gets $(1-\delta)/d$-fraction of all colors (by a simple Chernoff argument)
- Might be useful for TDMA slot assignments
  (extremely local algorithm, better usage of channel by assigning multiple slots to each node)

Algebraic Constructions

- Uses a probabilistic argument → no explicit algorithm
- There are algebraic constructions that are almost as good:
  - $p_1, \ldots, p_k$: first k prime numbers
  - Node with ID x can take colors $(i, x \bmod p_i)$ for $i = 1,\ldots,k$
  - Choose a color that no neighbor can take
  - Chinese remainder theorem:
    upper bound on number of i such that for $x \neq y$: $x \equiv y \pmod{p_i}$
    (if $\prod_i p_i \geq m^{\Delta}$, every node finds a color)
  - Choosing $k = O(\Delta \cdot \log m / \log(\Delta \cdot \log m))$ suffices
  - Prime number theorem: there are sufficiently many small primes:
    $p_k = O(\Delta \cdot \log m)$ → #colors = $k \cdot p_k = O(\Delta^2 \cdot \log^2 m / \log \Delta)$
- Using similar algorithm based on polynomials over finite fields: $O(\Delta^2 \cdot \log^2 m / \log^2 \Delta)$ colors
- Iterating $O(\log^* m)$ times → $O(\Delta^2)$-coloring

Multicoloring II

- Explicit algorithms can be extended to also get bounds for multicoloring (parameter $\delta \in (0,1)$):
  - Total number of colors: $O(\Delta^2 \cdot \log^2 m / \log^2 \Delta)$
    node of degree d gets $O(\delta / (d^{1+\delta} + \log_\Delta N))$-fraction of colors
  - Total number of colors: $O(\Delta^{O(\log^* m)} \cdot \log m / \log \Delta)$
    node of degree d gets $O(\delta/d^{1+\delta})$-fraction of colors
  - Trade-off possible between the two extremes possible

- Open problem:
  Find better explicit algorithms for graph multicoloring

- Randomized: With $O(\Delta \cdot \log(n)/\delta^2)$ colors, possible to assign $(1-\delta)/d$-fraction to each node of deg. d w.h.p. (n = # of nodes)

Defective Coloring

- Generalization of the classical coloring problem
- The defect d of a coloring of the vertices of a graph is the maximum degree of a graph induced by one of the color classes. (classic coloring: d=0)
- Problem: Given d, minimize number of colors
- Using similar techniques as for standard coloring: deterministic d-defective $O((\Delta/d)^2)$-coloring in time $O(\log^* m)$
- Tricky part: iterative application

Deterministic (Δ+1)-Coloring Algorithm

- If number of colors m $\geq \Delta$+2
  We can reduce by a factor of (Δ+1)/(Δ+2) in one round
  [Kuhn,Wattenhofer 06]

- Algorithm (assume O(Δ^2)-coloring is given):
  – Compute (Δ/2)-defective c-coloring
    (O(1) colors in O(log* Δ) rounds)
  – Recursively compute (Δ/2 + 1)-coloring for each color class
    (can be done in parallel)
  – Combination of the two colorings gives c·(Δ/2 + 1) = O(Δ)-coloring
  – Reduce to Δ+1 colors in O(Δ) rounds

- Recursion for time: T(Δ) $\leq$ T(Δ/2) + O(Δ), T(2) = O(1)

$(\Delta+1)$-Coloring Algorithm: Analysis

- Recursion for time: $T(\Delta) \leq T(\Delta/2) + \alpha\cdot\Delta$, $T(2) \leq 4\alpha$ (for some constant α)

- Theorem: $T(\Delta) \leq 2\alpha\cdot\Delta$
  - True for $\Delta = 2$
  - $\Delta > 2$: $T(\Delta) \leq 2\alpha\cdot\Delta/2 + \alpha\cdot\Delta = 2\alpha\cdot\Delta$
- Obtaining $O(\Delta^2)$-coloring to begin: O(log*m) time

- Total time for $(\Delta+1)$-coloring: $O(\Delta + \log^* m)$

- Same idea gives $\lambda\cdot(\Delta+1)$-coloring in $O(\Delta/\lambda + \log^* m)$ rounds for every $\lambda \geq 1$ (e.g. $\Delta^{3/2}$-coloring in $O(\Delta^{1/2} + \log^* m)$ time)

Weak Colorings

- One of the first papers on local algorithms by Naor and Stockmeyer considered the following weak coloring problem: Assign colors to nodes such that every node has at least one neighbor with a different color

- Generalization: Assign colors to nodes such that every node has at least k neighbors with different color

- $O(k^2)$ colors in $O(\log^* m)$ rounds

- k+1 colors in $O(k + \log^* m)$ time
  (same technique as for $(\Delta+1)$-coloring, trade-off also possible)

Lower Bound for One-Round Algorithms

- $\Omega(\Delta^2/\log^2\Delta)$-lower bound on number of colors for deterministic algorithms in [Kuhn,Wattenhofer 06]

- New: Improved $\Omega(\Delta^2)$-lower bound, much simpler proof

- Observation (IDs can be replaced by initial colors):
  - Nodes u, v with IDs x, y, set of neighbor IDs: S_x, S_y
  - If $y \in S_x$ and $x \in S_y$, u and v must choose different colors (otherwise, there is a graph on which the algorithm does not work)
  - The color sets of nodes with ID x and a neighbor with ID y are disjoint from the color sets of nodes with ID y and a neighbor with ID x

Edge Orientations

- A new color can be seen as an orientation on the edges of K_m
- A node with ID x, neighbor IDs S_x can choose a new color α if in corresponding orientations all edges (x,y) for $y \in S_x$ are oriented as $x \rightarrow y$
- Find orientations on the edges of K_m such that $\forall$ x, S ($x \notin S$, $|S| \leq \Delta$), $\exists$ an orientation such that $x \rightarrow y$ for all $y \in S$
- Lower bound for coloring: Show that a certain number of orientations does not suffice!
- Remark: Edge orientation problem for general graphs G: Condition must hold for all x, S as before where all $y \in S$ are neighbors of x

Sources

- $X \subseteq [m]$ ($[m] = \{1, \ldots, m\}$)
- $x \in X$ is source w.r.t. X for a given orientation
  if $x \rightarrow y, \forall y \neq x, y \in X$
- For every orientation and every $X \subseteq [m]$: at most one source

- Theorem: If $m \geq \Delta^2/4 + \Delta/2$, $\Delta^2/4$ orientations do not suffice
- Proof (Δ even):
  – There are at most $\Delta^2/4$ sources w.r.t. [m] and some orientation
    $\rightarrow$ there are at least $\Delta/2$ non-sources
  – Let X be set of $\Delta/2$ of non-sources
  – Show: there is one-hop view (x, S) with $|S| = \Delta - 1$ that gets no color
    ($x \in X$, $S = X \setminus \{x\} \cup \{\Delta/2$ other IDs$\}$)

Lower Bound Proof

- Need to find (x, S) such that
  for all orientations, there is $y \in S$ with $y \rightarrow x$

- Per orientation, at most one source w.r.t. X
  $\rightarrow$ on average, IDs in X source for
  $(\Delta^2/4)/(\Delta/2) = \Delta/2$ orientations w.r.t. X
  $\rightarrow$ $\exists x \in X$ that is source for at most $\Delta/2$ orientations

- $\exists y \in S = X \setminus \{x\}$ with $y \rightarrow x$ for all but these $\Delta/2$ orientations
- Because x is non-source, for every orientation, there is $y \in [m]$ for which $y \rightarrow x$
- Hence, we can add $\Delta/2$ additional IDs to the set S to "cover" all orientations

Summary: One-Round Lower Bounds

- Hence, for m $\geq \Delta^2/4 + \Delta/2$, $\Omega(\Delta^2)$ colors are best possible for deterministic one-round algorithms
- Combined with Linial's ring lower bound: $\Omega(\Delta^2 + \log\log m)$
- Randomized algorithms:
- For $\Delta = \Omega(\log n)$, $\Omega(\Delta \cdot \log n / \log\log n)$ colors needed
- Proof based on more complicated counting argument and Yao's principle

The Color Reduction Problem

- We want an algorithm that works for any graph G with max. degree Δ and initial m-coloring (assume Δ and m are known)
- Goal: Reduce the number of colors as quickly as possible (time complexity of algorithm should be function of m and Δ)
- Note: There is no bound on the size of the graph
- Because size of graph is not bounded: randomization does not help!

Color Reduction Lower Bound

- Problem has nice recursive structure that can be exploited
- Proof Sketch:
  - $\mathcal{N}_r(m,\Delta)$: neighborhood graph for r rounds,
    $\chi(\mathcal{N}_r(m,\Delta))$ = number of colors needed by r-round algorithm
  - $\eta_{r,\Delta}(G)$: number of edge orientations needed for graph G
  - We have seen: $\chi(\mathcal{N}_1(m,\Delta)) = \eta_{r,\Delta}(K_m)$ (note that $K_m = \mathcal{N}_0(m,\Delta)$)
  - It can be shown: $\chi(\mathcal{N}_r(m,\Delta)) = \eta_{r,\Delta}(\mathcal{N}_{r-1}(m,\Delta))$
  - Recursive structure allows to show the following lemma:
    If after removing s independent sets from $\mathcal{N}_r(m,\Delta)$, a t-clique remains,
    removing s independent sets from $\mathcal{N}_{r+1}(m,\Delta)$ leaves a t'-clique
    (for some specific value of t and t')
  - Proof of lemma uses same basic technique as lower bound on $\eta_{r,\Delta}(K_m)$
- Result: $\chi(\mathcal{N}_r(m,\Delta)) = \Omega(\Delta^2/r)$
- Hence, our algorithm is essentially tight

Distributed Coloring: Open Problems

- Lower bound for deterministic distributed coloring algorithms
  (or is there really a polylog algorithm?)

- Lower bound for randomized algorithms
  (Ω(log*n) best current lower bound)

- Explicit multicoloring, other coloring variants

- Dynamic case?
  (maybe more realistic communication models in general)