Distributed Coloring Depending on the Chromatic Number or the Neighborhood Growth

Johannes Schneider[1], Roger Wattenhofer[1]

Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland

Abstract[1]

We deterministically compute a $\Delta+1$ coloring in time $O(\Delta_{5c+2}\cdot(\Delta_5)^{2/c}/(\Delta_1)^{\epsilon}+(\Delta_1)^{\epsilon}+\log^* n)$ and $\mathrm{O}(\Delta_{5c+2}\cdot(\Delta_5)^{1/c}/\Delta^{\epsilon}+\Delta^{\epsilon}+(\Delta_5)^{d}\log\Delta_5\log n)$ for arbitrary constants d,ϵ and arbitrary constant integer c, where Δ_i is defined as the maximal number of nodes within distance i for a node and $\Delta:=\Delta_1$. Our greedy algorithm improves the state-of-the-art $\Delta+1$ coloring algorithms for a large class of graphs, e.g. graphs of moderate neighborhood growth. We also state and analyze a randomized coloring algorithm in terms of the chromatic number, the run time and the used colors. If $\Delta\in\Omega(\log^{1+1/\log^* n} n)$ and $\chi\in O(\Delta/\log^{1+1/\log^* n} n)$ then our algorithm executes in time $\mathrm{O}(\log\chi+\log^* n)$ with high probability. For graphs of polylogarithmic chromatic number the analysis reveals an exponential gap compared to the fastest $\Delta+1$ coloring algorithm running in time $O(\log\Delta+\sqrt{\log n})$. The algorithm works without knowledge of χ and uses less than Δ colors, i.e., $(1-1/O(\chi))\Delta$ with high probability. To the best of our knowledge this is the first distributed algorithm for (such) general graphs taking the chromatic number χ into account.

1 Introduction

Coloring is a fundamental problem with many applications. Unfortunately, even in a centralized setting, where the whole graph is known, approximating the chromatic number (the minimal number of needed colors), is currently computationally infeasible for general graphs and believed to take exponential running time. Thus, basically any reduction of the used colors below $\Delta+1$ – even just to Δ – is non-trivial in general. Looking at the problem in a distributed setting, i.e., without global knowledge of the graph, makes the problem harder, since coloring is not a purely "local" problem, i.e., nodes that are far from each other have an impact on each other (and the chromatic number). Therefore, it is not surprising that all previous work has targeted to compute a $\Delta+1$ coloring in

[1] This Technical Report (ID: TIK- Report-335) contains an omitted proof of the 18th International Colloquium on Structural Information and Communication Complexity (SIROCCO), June 26-29, 2011 conference paper.

general graphs as fast as possible (or resorted to very restricted graph classes). However, this somehow overlooks the original goal of the coloring problem, i.e., use as little colors as possible. Though in distributed computing the focus is often on communication, in many cases keeping the number of colors low outweighs the importance of minimizing communication. For example, a TDMA schedule can be derived from a (2-hop) coloring. The length of the schedule (and thus the throughput of the network) is determined by the number of employed colors.

In this paper, we also consider fast distributed computation of $\Delta + 1$ colorings in the first part. In the second part we are interested in both using less than $\Delta + 1$ colors and efficient computation. For sparse graphs, such as trees and planar graphs, as well as for dense graphs, e.g. cliques and unit disk graphs (UDG), efficient distributed algorithms are known that have both "good" time complexity and "good" approximation ratio of the chromatic number. Sparse graphs typically restrict the number of edges to be linear in the number of nodes. Unit disk graphs restrict the number of independent nodes within distance i to be bounded by a polynomial function $f(i)$. Our requirements on the graph are much less stringent than for UDGs, i.e., we do not restrict the number of independent nodes to grow dependent on the distance only. We allow for growth of the neighborhood dependent on the distance and also on Δ, i.e., n. For illustration, if the number of nodes within distance i is bounded by $\Delta^{1+(i-1)/100}$ our deterministic algorithm improves on the state-of-the-art algorithms running in linear time in Δ by more than a factor of $\Delta^{1/10}$.[2] Note, for any graph the size of the neighborhood within distance i is bounded by Δ^i. Additionally, if the size of the neighborhood within distance i of a graph is lower bounded by $\Delta^{c \cdot i}$ for an aribtrary constant c then the graph can have only small diameter, i.e. $O(\log \Delta)$. In such a case a trivial algorithm collecting the whole graph would allow for a coloring exponentially faster than the current state of the art deterministic algorithms running in time $O(\Delta + \log^* n)$ for small Δ already. Therefore, we believe that for many graphs that are considered "difficult" to color we significantly improve on the best known algorithms. The guarantee on the number of used colors is the same as in previous work, i.e., $\Delta + 1$. Despite the hardness of the coloring problem, intuitively, it should be possible to color a graph with small chromatic number with fewer colors and also a lot faster than a graph with large chromatic number. Our second (randomized) algorithm shows that this in indeed the case. The algorithm works without knowledge of the chromatic number χ.

2 Model and Definitions

Communication among nodes is done in synchronous rounds without collisions, i.e., each node can exchange one distinct message with each neighbor. Nodes start the algorithm concurrently. The communication network is modeled with a graph $G = (V, E)$. The distance between two nodes u, v is given by the length of the shortest path between nodes u and v. For a node v its neighborhood $N^r(v)$

[2] Generally, for a graph with $\Delta_{5c+2} \cdot (\Delta_5)^{2/c}/(\Delta_1)^\epsilon + \log^* n = (\Delta_1)^\epsilon$ for any choice of parameters ϵ and c, the best known run time (linear in Δ) is cut by a factor $\Delta^{1-\epsilon}$.

represents all nodes within distance r of v (not including v itself). By $N(v)$ we denote $N^1(v)$ and by $N_+(v) := N(v) \cup v$. The degree $d(v)$ of a node v is defined as $|N(v)|$, $d_+(v) := |N_+(v)|$, $\Delta := \max_{u \in V} d(u)$ and $\Delta_i := \max_{u \in V} |N^i(u)|$. By $G^i = (V, E^i)$ of $G = (V, E)$ we denote the graph where for each node $v \in V$ there is an edge to each node $u \in N^i(v)$. In a (vertex) coloring any two adjacent nodes u, v have a different color. A set $T \subseteq V$ is said to be independent in G if no two nodes $u, v \in T$ are neighbors. A set $R \subseteq V$ is (α, β)-ruling if every two nodes in the set R have distance at least α and any node not in the set R has a node in the set within distance β. The function $\log^* n$ states how often one has to take the (iterated) logarithm to get at most 1, i.e., $\log^{(\log^* n)} n \leq 1$. The term " with high probability" abbreviated by w.h.p. denotes a number $1 - 1/n^c$ for an arbitrary constant $c > 1$.

Our algorithm is non-uniform, i.e., every node knows an upper bound on the total number of nodes n and the maximal degree Δ. We also use the following Chernoff bound:

Theorem 1 *The probability that the number X of occurred independent events $X_i \in \{0, 1\}$, i.e., $X := \sum X_i$, is less than $(1 - \delta)$ times a with $a \leq \mathbb{E}[X]$ can be bounded by $Pr(X < (1 - \delta)a) < e^{-a\delta^2/2}$. The probability that the sum is more than $(1+\delta)b$ with $b \geq \mathbb{E}[X]$ with $\delta \in [0, 1]$ can be bounded by $Pr(X > (1+\delta)b) < e^{-b\delta^2/3}$.*

Corollary 2 *The probability that the number X of occurred independent events $X_i \in \{0, 1\}$, i.e., $X := \sum X_i$, is less than $\mathbb{E}[X]/2$ is at most $e^{-\mathbb{E}[X]/8}$ and the probability that is more than $3\mathbb{E}[X]/2$ is bounded by $e^{-\mathbb{E}[X]/12}$.*

3 Related Work

Distributed coloring is a well studied problem in general graphs in the message passing model e.g. [3, 4, 15, 13, 12, 11]. There is a tradeoff between the number of used colors and the running time of an algorithm. Even allowing a constant factor more colors can have a dramatic influence on the running time of a coloring algorithm, i.e., in [15] the gap between the running time of an $O(\Delta)$ and an $\Delta + 1$ coloring algorithm can be more than exponential for randomized algorithms. More precisely, a $\Delta + 1$ coloring is computed in time $O(\log \Delta + \sqrt{\log n})$ and an $\mathrm{O}(\Delta + \log^{1+1/\log^* n} n)$ coloring in time $\mathrm{O}(\log^* n)$. When using $O(\Delta^2)$ colors, a coloring can be computed in time $\mathrm{O}(\log^* n)$ [12], which is asymptotically optimal for constant degree graphs due to a lower bound of time $\Omega(\log^* n)$ for three coloring of an n-cycle. Using $\mathrm{O}(\Delta^{1+o(1)})$ colors [4] gives a deterministic algorithm running in time $\mathrm{O}(f(\Delta) \log \Delta \log n)$ where $f(\Delta) = \omega(1)$ is an arbitrarily slow growing function in Δ. To this date, the fastest deterministic algorithm to compute a $\Delta + 1$ coloring in general graphs requires $\mathrm{O}(\Delta + \log^* n)$ [3, 10] or $n^{O(1)/\sqrt{\log n}}$ time [13]. Algorithm [13] computes graph decompositions recursively until the maximum degree in the graph is sufficiently small. To deal with large degree vertices, a ruling forest is computed for each decomposition

and each tree is collapsed into a single vertex. The algorithms [3, 10] improved on [11] by a factor of $\log \Delta$ through employing defective colorings, i.e., several nodes initially choose the same color. However, through multiple iterations the number of adjacent nodes with the same color is reduced until a proper coloring is achieved. In [4] defective colorings were combined with tree decompositions [2]. In comparison, our deterministic algorithm improves the linear running time in Δ by a factor Δ^d for a constant d for a large class of graphs by iteratively computing ruling sets, such that a node in the ruling set can color its two hop neighborhood.

Overall $\Delta+1$ coloring has probably attracted more attention than employing $O(\Delta)$ or more colors. Using less than $\Delta + 1$ colors is not possible for complete graphs – not even in a centralized setting, where the entire graph is known. An algorithm in [9] parallelizes Brooks' sequential algorithm to obtain a Δ coloring from a $\Delta + 1$ coloring. In a centralized setting the authors of [1] showed how to approximate a three-colorable graph using $O(n^{0.2111})$ colors. Some centralized algorithms iteratively compute large independent sets, e.g. [5]. It seems tempting to apply the same ideas in a distributed setting, e.g. a parallel minimum greedy algorithm for computing large independent sets is given in [8]. It has approximation ratio $(\Delta + 2)/3$. However, the algorithm runs in time polynomial in Δ and logarithmic in n and thus is far from efficient. For some restricted graph classes, there are algorithms that allow for better approximations in a distributed setting. A Δ/k coloring for $\Delta \in O(\log^{1+c} n)$ for a constant c with $k \leq c_1(c) \log \Delta$ where constant c_1 depends on c is given in [7]. It works for quite restricted graphs (only), i.e., graphs that are Δ-regular, triangle free and $\Delta \in O(\log^{1+c} n)$. Throughout the algorithm a node increases its probability to be active. An active node picks a color uniformly at random. The algorithm runs in $O(k+\log n/\log \Delta)$ rounds.Constant approximations of the chromatic number are achieved for growth bounded graphs (e.g. unit disk graphs) [14] and for many types of sparse graphs [2]. In [6] the existence of graphs of arbitrarily high girth was shown such that $\chi \in \Omega(\Delta/\log \Delta)$. Since graphs of high girth locally look like trees and trees can be colored with two colors only, this implies that coloring is a non-local phenomenon. Thus, a distributed algorithm that only knows parts of the graph and is unaware of global parameters such as χ, has a clear disadvantage compared to a centralized algorithm.

We give a randomized coloring algorithm in terms of the chromatic number of a graph which uses ideas from [15]. Given a set of colors $\{0, 1, \ldots, f(\Delta)\}$ for an arbitrary function f with $f(x) \geq x$ [15] computes an $f(x) + 1$ coloring. The run time depends on f, i.e. for $f(\Delta) := \Delta$ Algorithm *DeltaPlus1Coloring*[15] takes time $O(\log \Delta + \sqrt{\log n})$. For $f(\Delta) := O(\Delta + \log^{1+1/\log^* n} n)$ Algorithm *ConstDeltaColoring*[15] takes only $O(\log^* n)$ time. Both Algorithms from [15] operate analogously: In each communication round a node chooses a subset of all available colors and keeps one of the colors, if no neighbor has chosen the same color. For our deterministic coloring we employ the (deterministic) Algorithm *CoordinateTrials*[15] for computing ruling sets from [15] as a subroutine. Due to Theorem 16 in [15] a $(2, c)$-ruling set is computed using *CoordinateTrials(d,c)* in

time $2^c d^{1/c}$ from an initial coloring $\{0, 1, \ldots, d\}$. The algorithm partitions the digits of a node's label, e.g. color, into c equal parts (with the same number of digits). A node v computes a rank $Rank(v)$ consisting of c bits in each round j, where bit i is 1 if the ith part equals round j and 0 otherwise. Based on the rank a node either continues the algorithm (and eventually joins the ruling set) or stops. Nodes with rank larger 0 compete to continue and force other nodes to stop the algorithm. More precisely, a node v tells its neighbors with distinct rank to halt the algorithm, if in the kth round of the competition its $Rank(v)$ equals k.

4 Deterministic Coloring

For coloring one can either let each node decide itself on a color or decompose the graph into (disjoint) clusters and elect a leader to coordinate the coloring in a cluster. Our deterministic algorithm follows the later strategy by iteratively computing ruling sets. Each node in the set can color itself and (some) nodes up to distance 2 from it (in a greedy manner). To make fast progress, only nodes can join the ruling set that color many nodes. Once, no node has sufficiently many nodes to color within distance two, i.e., less than Δ^ϵ (for a parameter ϵ of the algorithm), the nodes switch to another algorithm [3, 10].

When a node v is in the ruling set, it gets to know all nodes $N^3(v)$ and assigns colors to all uncolored nodes $N^2(v)$ by taking into account previously assigned colors. Node v can assign colors, for instance, in a greedy manner, i.e., it looks at a node $u \in N^2(v)$ and picks the smallest color that is not already given to a neighbor of $w \in N(u)$. Potentially, two nearby nodes u, v in the ruling set might concurrently assign the same colors two adjacent nodes, e.g. node u assigns 1 to x and node v assigns 1 to y and x, y are neighbors. To prevent this problem any two nodes u, v in the ruling-set must have distance at least 6 to avoid that neighbors are potentially assigned the same color. Thus, the algorithm computes a $(6, 5k)$- ruling set, where k is a parameter of the ruling-set algorithm [15]. In fact, the ruling-set algorithm from [15] computes only a $(2, k)$-ruling set for the graph G. However, if the ruling set is computed on the graph G^i then we get a $(1 + i, ik)$-ruling set. Note that any algorithm working on the graph G^i can be run by using the graph G at the price of prolonging the algorithm by a factor of i and, potentially, requiring larger messages. This is because a message between two adjacent nodes u, v in G^i might have to be forwarded along up to i edges in G and a single node might have to forward several messages at a time from its neighbors. Computing a $(6, 5k)$-ruling set in turn demands that two nodes u, v within distance 5 have distinct labels, therefore we start out by computing an $O(|N^5(v)|^2)$ coloring in the graph G^5 using [12] (or [4] to compute an $O(|N^5(v)|)$ coloring).

After the initial coloring, a node participates in iteratively computing $(6, 5k)$-ruling sets until it has color less than $\Delta + 1$, or it and all neighbors have less than Δ^ϵ neighbors with color larger than Δ for some parameter ϵ. The remaining nodes (with color larger Δ) are taken care of using [3, 10].

Algorithm *RulingColoring* for arbitrary ϵ and any integer k

1: $col(v)$:= Compute an $O(|N^5(v)|^2)$ coloring in the graph G^5 using [12]
2: **while** $col(v) > \Delta \wedge \exists u \in N_+(v)$ with $|\{w \in N(u) | col(w) > \Delta\}| > \Delta^\epsilon$ **do**
3: Compute a $(6, 5k)$-ruling set R using [15]
4: **if** $v \in$ ruling set R **then**
5: **for all** $w \in N_+(v)$ with $|\{x \in N(w) | col(x) > \Delta\}| > \Delta^\epsilon$ **do**
6: (Greedily) color all nodes $u \in N(w)$ for which $col(u) > \Delta$
7: **end for**
8: **end if**
9: **end while**
10: Execute Algorithm [3] if $col(v) > \Delta$

Lemma 1. *A $(6, 5c)$-ruling set is computed in time $O(|N^5(v)|^{2/c})$ (Line 1 Algorithm* RulingColoring*).*

Proof. Due to Theorem 16 in [15] a $(2, c)$-ruling set can be computed deterministically using Algorithm *CoordinateTrials(d,c)* in time $2^c d^{1/c}$ in a graph G from an initial $d + 1$ coloring. The algorithm only requires that two adjacent nodes have distinct colors. Since a node v has a distinct color from any node $u \in N^5(v)$, any two adjacent nodes in G^5 have distinct colors. Thus we can run *CoordinateTrials($|N^5(v)|^2, c$)* to compute a $(2, c)$-ruling set in G^5, i.e., a $(6, 5c)$-ruling set in G, in time $O(c2^c|N^5(v)|^{2/c}) = O(|N^5(v)|^{2/c})$, since c is a constant. Two nodes that are adjacent in G^5 have distance 1 to 5 in G. If two nodes are at distance 2 in G^5 then they are at distance 6 to 10 in G. Thus, we get a $(6, 5c)$-ruling set, since any two nodes in G^5 at distance 2 have distance at least 6 in G^5 and any two nodes at distance c in G^5 have distance at most $5c$ in G.

Theorem 3 *Algorithm* RulingColoring *computes a $\Delta + 1$ coloring in time $O(\Delta_{5c+2} \cdot (\Delta_5)^{2/c}/\Delta^\epsilon + \Delta^\epsilon + \log^* n)$.*

Proof. Any node v that participates in computing a ruling set, can color at least Δ^ϵ nodes within distance $5c + 2$. This is because a node only takes part in the computation, if there is a node $u \in N_+(v)$ with at least Δ^ϵ uncolored neighbors. If a node enters the ruling set it can color itself and all nodes $w \in N(u)$. By definition of a $(6, 5c)$-ruling set any two nodes have distance 6. Therefore, if all nodes in the ruling set color nodes at distance 2 from them, there will not be any color conflicts. Furthermore, every node gets a node in the ruling set within distance $5c$. Any node that has more than Δ^ϵ uncolored neighbors, gets at least Δ^ϵ colored nodes within distance $5c + 2$ by computing one ruling set. The total time until node v gets colored or has less than Δ^ϵ colored neighbors is given by the following expression: The total number of nodes $|N^{5c+2}(v)| \leq \Delta_{5c+2}$ within distance $5c + 2$ divided by the number of nodes that get colored for a computation of a ruling set, i.e., at least Δ^ϵ, multiplied by the time it takes to color the nodes and to compute one ruling set, i.e., $O((\Delta_5)^{2/c})$ (see Lemma 1). In total this results in time $O(\Delta_{5c+2}/\Delta^\epsilon \cdot (\Delta_5)^{2/c})$. To color the remaining nodes, i.e., at most Δ^ϵ neighbors of each node, using [3] (or [10]) takes time

$O(\Delta^\epsilon + \log^* n)$. Computing an $O(|N^5(v)|^2)$ coloring in graph G^5 using [12] takes time $O(\log^* n)$.

Depending on the maximal degree Δ, it might be better to compute an (initial) $O(|N^5(v)|)$ coloring in time $O((\Delta_5)^{c_0} \log \Delta_5 \log n)$ for an arbitrary small constant c_0 using [4]. The time complexity changes to $O(\Delta_{5c+2} \cdot (\Delta_5)^{1/c}/\Delta^\epsilon + \Delta^\epsilon + (\Delta_5)^d \log \Delta_5 \log n)$ for arbitrary small d.

5 Coloring Depending on the Chromatic Number

The algorithm (but not the analysis) itself is straight forward without many novel ideas. In the first two rounds a node attempts to get a color from a set with less than Δ colors. Then, (essentially) coloring algorithms from [15] are used to color the remaining nodes.

Let $\Delta_0 := \Delta$ be the maximal size of a neighborhood in the graph, where all nodes are uncolored, and let $N(v)$ be all uncolored neighbors of node v (in the current iteration). The algorithm lets an uncolored node v be a active twice with a fixed constant probability $1/c_1$. An active node chooses (or picks) a random color from all available colors in the interval $[0, \Delta_0/2-1]$. Node v obtains its color and exits the algorithm, if none of its neighbors $N(v)$ has chosen the same color. After the initial two attempts to get colored each node v computes how many neighbors have been colored and how many colors have been used. The difference denotes the number of "conserved" colors $n_c(v)$. The algorithm can use the conserved colors to either speed up the running time, since more available colors render the problem simpler, e.g. allow for easier symmetry breaking, or to reduce the number of used colors as much as possible. In Algorithm *FastRandColoring* we spend half of the conserved colors for fast execution and preserve the other half to compute a coloring using $\Delta_0 + 1 - n_c(v)/2$ colors. A node v repeatedly picks uniformly at random an available color from $[0, \Delta_0 + 1 - n_c(v)/2]$ using Algorithm *DeltaPlus1Coloring* until the number of available colors is at least a factor two larger than the number of uncolored neighbors. Afterwards it executes Algorithm *ConstantDeltaColoring* [15] using 2Δ colors.

If an event occurs with high probability then conditioning on the fact that the event actually took place does not alter the probability of other likely events much as shown in the next theorem. It is a (slight) generalization of Theorem 2 from [15].

Theorem 4 *For n^{k_0} (dependent) events E_i with $i \in \{0, ..., n^{k_0}-1\}$ and constant k_0, such that each event E_i occurs with probability $Pr(E_i) \geq 1 - 1/n^k$ for $k > k_0 + 2$, the probability that all events occur is at least $1 - 1/n^{k-k_0-2}$.*

Proof. For an event E_i let $S_{E_i} \subseteq \Omega$ be all elementary events from all possible events Ω for which E_i cannot happen. By definition $Pr(E_i) = 1 - \sum_{F \in S_{E_i}} Pr(F)$). We want to show that $Pr(\wedge_{i \in \{0,...,n^{k_0}-1\}} E_i) \geq 1 - 1/n^{k-k_0-2}$. Since $Pr(\wedge_{i \in \{0,...,n^{k_0}-1\}} E_i) = Pr(E_0) \cdot Pr(E_1|E_0) \cdot Pr(E_2|E_0 \wedge E_1) \cdot ... \cdot$

Algorithm FastRandColoring

1: $col(v) := none$
2: **for** $i = 0..1$ **do**
3: $choice(v) :=$ With probability $1/c_1$ random color from interval $[0, \Delta_0/2 - 1]$ else $none$
4: **if** $choice(v) \neq none \wedge \nexists u \in N(v), s.t. choice(u) = choice(v) \vee col(u) = choice(v)$ **then** $col(v) := choice(v)$ and exit **end if**
5: **end for**
6: $n_c(v) := (\Delta - |N(v)|) - |\{col(u) | u \in N(v)\}|$
7: $C(v) := [0, \max(3\Delta_0/4, \Delta_0 + 1 - n_c(v)/2] \setminus \{col(u) | u \in N(v)\}$ {available colors}
8: Execute Algorithm DeltaPlus1Coloring [15] until $C(v) \geq 2|N(v)|$
9: Execute Algorithm ConstDeltaColoring [15]

$Pr(E_{n^{k_0}-1} | \wedge_{i \in \{0,\ldots,n^{k_0}-2\}} E_i)$ we can also derive lower bounds for the conditional probabilities $Pr(E_i | \wedge_{i \in T, T \subset \in \{0,\ldots,n^{k_0}-2\}} E_i)$. We assume a worst case correlation among events E_i, i.e. the occurrence of an event E_i has the worst impact on the probability that another event E_j occurs. Given that the event E_i occurs, all elementary events S_{E_i} for which E_i cannot happen, are known not to occur. They can be excluded from Ω, when computing the probability of another event E_j, i.e. $Pr(E_j|E_i)$. Thus, the elementary events that can occur given E_i are $\Omega_{|i} := \Omega \setminus S_{E_i}$. Since $Pr(E_j|E_i)$ is a probability distribution, the sum of the probabilities $Pr(F|E_i)$ of all elementary events $F \in \Omega_{|i}$ must be one, i.e. $\sum_{F \in \Omega_{|i}} Pr(F|E_i) = 1$. We have that $Pr(F|E_i) = Pr(F) \cdot 1/(1 - \sum_{F \in S_{E_i}} Pr(F)) \leq Pr(F) \cdot 1/(1 - 1/n^k)$, since the occurrence of event E_i only removes the set S_{E_i} with $\sum_{F \in S_{E_i}} Pr(F) \leq 1/n^k$ from Ω, but does not make one elementary event $F_1 \in \Omega_{|i}$ (relatively) more favorable to another $F_2 \in \Omega_{|i}$, i.e. the probabilities $Pr(F|E_i)$ of all remaining events $F \in \Omega_{|i}$ are increased by the same factor $1/(1 - \sum_{F \in S_{E_i}} Pr(F)) \leq 1/(1 - 1/n^k)$. To compute $Pr(E_j|E_i)$ assuming a worst case correlation, all excluded elementary events S_{E_i} cause E_j to occur and $S_{E_i} \cap S_{E_j} = \{\}$, i.e. all elementary elements S_{E_j} are excluded for $Pr(E_j|E_i)$ to occur. Thus, all elementary events $\Omega_{|i} \setminus S_{E_j}$ cause E_j to occur and $Pr(E_j|E_i) = 1 - \sum_{F \in S_{E_j}} Pr(F|E_i) \geq 1 - \sum_{F \in S_{E_j}} Pr(F) \cdot 1/(1 - 1/n^k) \geq 1 - 1/n^k \cdot 1/(1 - 1/n^k) = 1 - 1/(n^k - 1)$. The argument can be generalized to bound any conditional probability $Pr(E_j | \wedge_{i \in T, T \subseteq \{1,\ldots,n^{k_0}-1\}} E_i)$ for any $0 \leq j \leq n^{k_0} - 1$. The remaining possible events for E_j to occur given all events E_i with $i \in T$ happen is $\Omega_{|T} := \Omega \setminus \cup_{i \in T} S_{E_i}$. For the probability of an elementary event $F \in S_{E_j}$ holds $Pr(F | \wedge_{i \in T} E_i) = Pr(F) \cdot 1/(1 - \sum_{F \in \cup_{i \in T} S_{E_i}} Pr(F)) \leq Pr(F) \cdot 1/(1 - |T|/n^k)$. The last inequality follows since all sets S_{E_i} are assumed to be disjunct, i.e. $S_{E_i} \cap S_{E_l} = \{\}$, to maximize the probability that an elementary event $F \in S_{E_j}$ occurs. Therefore, $Pr(E_j | \wedge_{i \in T} E_i) = 1 - \sum_{F \in S_{E_j}} Pr(F | \wedge_{i \in T} E_i) \geq 1 - \sum_{F \in S_{E_j}} Pr(F) \cdot 1/(1 - |T|/n^k) = 1 - 1/(1 - |T|/n^k) \sum_{F \in S_{E_j}} Pr(F) \geq 1 - 1/(1 - n^{k_0}/n^k) \cdot 1/n^k = 1 - 1/(n^k - n^{k_0})$. Using the bound of the conditional probabilities and $Pr(E_i) \geq 1 - 1/n^k \geq 1 - 1/(n^k - n^{k_0})$ (the first inequal-

ity is by assumption), we obtain: $Pr\left(\wedge_{i\in\{0,\ldots,n^{k_0}-1\}}E_i\right) = Pr\left(E_0\right)\cdot Pr\left(E_1|E_0\right)\cdot Pr\left(E_2|\wedge_{i\in\{0,1\}}E_i\right)\cdot Pr\left(E_3|\wedge_{i\in\{0,1,2\}}E_i\right)\cdot\ldots\cdot Pr\left(E_{n^{k_0}-1}|\wedge_{i\in\{0,\ldots,n^{k_0}-2\}}E_i\right) \geq \prod_{i\in\{0,\ldots,n^{k_0}-1\}}\left(1-1/(n^k-n^{k_0})\right) = \left(1-1/(n^k-n^{k_0})\right)^{n^{k_0}} \geq \left(1-2/n^k\right)^{n^{k_0}} \geq 1-1/n^{k-k_0-2}$.

Consider any coloring of the graph G using the minimal number of colors χ. Let S_c be a set of nodes having color $c \in [0,\chi-1]$ for this coloring. For a node v with color c for this optimal coloring, we have $v \in S_c$. Let choice $i \geq 0$ (of colors) be the $(i+1)$-st possibility where a node could have picked a color, i.e., iteration i of the for-loop of Algorithm *FastRandColoring*. Let the set C_S^i be all distinct colors that have been obtained by a set of nodes S for any choice $j \leq i$, i.e. $C_S^i := \{c|\exists u \in S,$ s.t. $col(u) = c$ after iteration $i\}$. We do not use multisets here, i.e. a color c can only occur once in C_S^i. Let P_S^i be all nodes in S that make a choice in iteration i,i.e., $P_S^i := \{c|\exists u \in S,$ s.t. $choice(u) = c$ in iteration $i\}$. Let CP_S^i be all colors that have been chosen (but not yet obtained) by a set of nodes S in iteration i, i.e., $CP_S^i := \{c|\exists u \in P_S^i,$ s.t. $choice(u) = c$ in iteration $i\}$. By definition, $|CP_S^i| \leq |P_S^i|$.

To deal with the interdependence of nodes we follow the idea of *stochastic domination*. If X is a sum of random binary variables $X_i \in \{0,1\}$, i.e., $X := \sum_i X_i$, with probability distributions A, B and $Pr_A(X_i = 1|X_0 = x_0, X_1 = x_1, ..., X_{i-1} = x_{i-1}) \geq Pr_B(X_i = 1|X_0 = x_0, X_1 = x_1, ..., X_{i-1} = x_{i-1}) = p$ for any values of $x_0, x_1, ..., x_{i-1}$, we can apply a Chernoff bound to lower bound $Pr_A(X \geq x)$ by a sum of independent random variables X_i, where $X_i = 1$ with probability p.

Theorem 5 *After choice $i \in [0,1]$ for every node v holds w.h.p.: The colored nodes C_S^1 of any set $S \in N(v)\cup\{S_c\cap N(v)|c \in [0,\chi-1]\}$ with $|S| \geq c_2 \log n$ fulfill $|S|/(16c_1) \leq |C_S^1| \leq 3|S|/c_1$ with $c_1 > 32$. The number of nodes $|P_S^1|$ making a choice is at least $|S|/(4c_1)$ and at most $3|S|/(2c_1)$.*

Proof. Consider a set $S \in \{S_c \cap N(v)|c \in [0,\chi-1]\}$ of nodes for some node v. For i choices we expect (up to) $i|S|/c_1$ nodes to make a choice. Using the Chernoff bound from Corollary 2 the number of nodes that pick a color deviates by no more than one half of the expectation with probability $1-2^{i/8|S|/c_1} \geq 1-2^{i/8c_2\log n/c_1} = 1-1/n^{c_3}$ for a constant $c_3 := ic_2/(8c_1)$. Thus, at most $3i|S|/(2c_1)$ neighbors of v make a choice and potentially get colored with probability $1-1/n^{c_3}$. Using Theorem 4 this holds for all nodes with probability $1-1/n^{c_3-3}$, which yields the bounds $|P_S^1| \leq 3|S|/(2c_1)$ and $|C_S^1| \leq 3|S|/c_1$.

For choice i w.h.p. the number of nodes that make a choice is therefore in $[a,b] := [1/2\cdot(1-3i/(2c_1))\cdot|S|/c_1, 3|S|/(2c_1)]$. The lower bound, i.e. $a \leq |P_S^i|$, follows if we assume that for each choice $j < i$ at most $3|S|/(2c_1)$ nodes get colored, which happens w.h.p.. Thus, after $i-1$ choices at least $(1-3i/(2c_1))\cdot|S|$ nodes can make a choice, i.e. are uncolored. We expect a fraction of $1/c_1$ to choose a color. Using Corollary 2 the nodes that make a choice is at least half the expected number w.h.p.. Thus, for choice 1 we have for $c_1 > 32$ and $a := (1-3/(2c_1))/(2c_1)\cdot|S|$: $|S|/(4c_1) \leq a \leq |P_S^1|$.

Consider an arbitrary order $w_0, w_1, ..., w_{|S|-1}$ of nodes S. We compute the probability that node $w_k \in S$ obtains a distinct color for choice i from all previous nodes $w_0, w_1, ..., w_{k-1} \in S$. The probability is minimized, if all $k-1$ nodes have distinct colors and k is large. Since $k \leq b = 3|S|/(2c_1)$ we have $p(col(w_k) \in [0, \Delta_0/2] \setminus S_{w_0,w_1,...,w_{k-1}}) \geq p(col(w_k) \in [0, \Delta_0/2] \setminus S_{w_0,w_1,...,w_{b-1}}) \geq 1/c_1 \cdot (1 - b/(\Delta_0/2) \geq (1 - 3/\Delta_0/(2c_1)/(\Delta_0/2))/c_1 = 1/c_4$ with constant $c_4 := 1/c_1 \cdot (1 - 3/c_1)$. The lower bound for the probability of $1/c_4$ holds for any $k \in [0, b-1]$ and any outcome for nodes $S_{w_0,w_1,...,w_{k-1}}$. Thus, to lower bound the number of distinct colors $|C_S|$ that are obtained by nodes in S we assume that the number of nodes that make a choice is only a and that each node that makes a choice gets a color with probability $1/c_4$ (independent of the choices of all other nodes). Using the Chernoff bound from Corollary 2 gives the desired result for a set S. In total there are n nodes and we have to consider at most $1 + \chi \leq n + 2$ sets per node. Using Theorem 4 for $n \cdot (n+1)$ events each occurring w.h.p. completes the proof.

Next we consider a node v and prove that for the second attempt of all uncolored nodes $u \in S_c \cap N(v)$ a constant fraction of colors taken by independent nodes $w \in S_c \cap N(v) \setminus \{u\}$ from u are not taken (or chosen) by its neighbors $y \in N(u)$.

Theorem 6 *For the second choice let $E(c)$ be the event that for a node v for each uncolored node $u \in N(v) \cap S_c$ holds $|(CP^1_{N(u)} \cup C^1_{N(u)}) \cap C^1_{N(v)\cap S_c}| \leq 3/4|C_{N(v)\cap S_c}|$ for $|N(v) \cap S_c| \geq c_2 \log n$. Event $E(c)$ occurs given $\bigwedge_{c_1 \in X \subseteq [0,\chi], |N(v)\cap S_{c_1}| \geq c_2 \log n} E(c_1)$ w.h.p. for an arbitrary set $X \subseteq [0, \chi]$.*

Proof. Consider a colored node $w \in S_c \cap N(v)$ for some node v, i.e. $col(w) \neq$ none. We compute an upper bound on the probability that a node $y \in N(u)$ gets (or chooses) color $col(w)$, i.e., $p(\exists y \in N(u), col(y) = col(w) \vee choice(y) = col(w)) = p(\vee_{y \in N(u)} col(y) = col(w) \vee choice(y) = col(w)) \leq \sum_{y \in N(u)} p(col(y) = col(w) \vee choice(y) = col(w))$. The latter inequality follows from the inclusion-exclusion principle: For two events A, B we have $p(A \cup B) = p(A) + p(B) - p(A \cap B) \leq p(A) + p(B)$. We consider the worst case topology and worst case order in which nodes make their choices to maximize $\sum_{y \in N(u)} p(col(y) = col(w) \vee choice(y) = col(w))$. Due to Theorem 5 for every node $y \in N(u)$ at most $|P^0_{N(y)}| + |P^1_{N(y)}| \leq 3d(y)/c_1 \leq 3\Delta_0/c_1$ neighbors $z \in N(y)$ make a choice during the first two attempts $i \in [0, 1]$. To maximize the chance that some node y obtains (or chooses) color $col(w)$, we can minimize the number of available colors for y and the probability that some neighbor $z \in N(y)$ chooses color $col(w)$, since when making choice i we have $p(choice(y) = col(w)) \leq 1/(c_1|C(y)|)$ because each available color is chosen with the same probability. To minimize $|C(y)|$ the number of colored nodes $z \in N(y)$ should be maximized and at the same time each node $z \in N(y)$ should have a neighbor itself with color $col(w)$. The latter holds, if $z \in N(y)$ is adjacent to node w. Thus, to upper bound $p(col(y) = col(w))$ we assume that node w and each node $y \in N(u)$ share the same neighborhood (except u), i.e., $N(y) \setminus \{u\} = N(w)$, and the maximal number of nodes in $N(y)$ given our

initial assumption are colored or make a choice, i.e., $3d(y)/c_1 \leq 3\Delta_0/c_1$. This, yields $p(col(y) = col(w)) \leq 1/(c_1|C(y)|) \leq 1/(c_1(\Delta_0/2 - 3\Delta_0/c_1)) \leq 8/(c_1\Delta_0)$ (for $c_1 > 32$) and therefore $p(\exists y \in N(u), col(y) = col(w)) = p(\vee_{y\in N(u)} col(y) = col(w)) \leq \sum_{y\in N(u)} p(col(y) = col(w)) \leq 3\Delta_0/c_1 \cdot 8/(c_1\Delta_0) \leq 1/c_1$ (for $c_1 > 32$). In other words, the probability that some node $y \in N(u)$ has obtained color $col(w)$ or chooses $col(w)$ is bounded by $1/c_1$.

Let us estimate the probability that some neighbor $y \in N(u)$ gets the same color as a node $w_1 \in N(v) \cap S_c$ given that no (or some) node in $y \in N(u)$ has chosen or obtained $col(w_0)$ for some node $w_0 \in C_{N(v)\cap S_c} \setminus \{w_1\}$. To minimize $|C(y)|$ we assume that $|C(y)|$ is reduced by 1 for every colored node $w_0 \in C_{N(v)\cap S_c} \setminus \{w_1\}$. Since at most $3/2d(y)/c_1 \leq 3/2\Delta_0/c_1$ neighbors make a choice concurrently, the event reduces our (unconditioned) estimate of the size of $|C(y)|$ by at most $3/2\Delta_0/c_1$. Using the same calculations as above with $|C(y)| \leq \Delta_0/2 - 9/2\Delta_0/c_1$, the probability that some node $y \in N(u)$ has obtained color $col(w)$ or chooses $col(w)$ given the outcome for any set of colored nodes $W \subseteq N(v)\cap S_c$ is at most $1/2$. Thus, we expect at most $|C_{N(v)\cap S_c}|/2$ colors from $C_{N(v)\cap S_c}$ to occur in node u's neighborhood. Using the Chernoff bound from Corollary 2, we get that the deviation is at most $1/2$ the expectation with probability $1 - 2^{-|C_{N(v)\cap S_c}|/8}$ for node u, i.e., the probability $p(E(u, c))$ of the event $E(u, c)$ that for a node $u \in N(v)$ at most $3|C_{N(v)\cap S_c}|/4$ colors from $N(v) \cap S_c$ are also taken or chosen by its neighbors $y \in N(u)$ is at least $1 - 2^{-|C_{N(v)\cap S_c}|/8}$. Using Theorem 5 for $S = N(v) \cap S_c$ we have $|C_{N(v)\cap S_c}| \geq |S|/(16c_1) = |N(v) \cap S_c|/(16c_1) \geq c_2 \log n/(16c_1)$. Therefore, $p(E(u, c)) \geq 1 - 1/n^{c_2/(16c_1)}$. Due to Theorem 4 the event $E(c) := \bigwedge_{u\in N(v)} E(u, c)$ occurs with probability $1 - 1/n^{c_2/(16c_1)-3}$.

Theorem 7 *After the first two choices for a node v with initial degree $d(v) \geq \Delta_0/2$ there exists a subset $N_c \subseteq N(v)$ with $|N_c| \geq (\Delta + 1)/(c_5\chi)$ that has been colored with $(\Delta + 1)/(2c_5\chi)$ colors for a constant $c_5 := 2048c_1^2$ w.h.p. for $\Delta \in \Omega(\log^{1+1/\log^* n} n)$ and $\chi \in O(\Delta/\log n)$.*

Proof. By assumption $\chi \in O(\Delta/\log n)$, i.e., $\chi < 1/(4c_3)\Delta/\log n$. At least half of all neighbors $u \in N(v)$ with $u \in S_c \cap N(v)$ must be in sets $|S_c \cap N(v)| \geq c_3 \log n$. This follows, since the maximum number of nodes in sets $|S_c \cap N(v)| < c_3 \log n$ is bounded by $\chi \cdot c_3 \log n \leq \Delta_0/4$. Assume that all statements of Theorem 5 that happen w.h.p. have actually taken place. Consider a node v and a set $N(v) \cap S_c$ with $|S_c \cap N(v)| \geq c_3 \log n$ given there are at most $3/2\Delta_0$ colored neighbors $u \in N(v)$. For a node $u \in N(v) \cap S_c$ the probability that it obtains the same color of another node $N(v) \setminus \{u\} \cap S_c$ is given by the probability that it chooses a color $col(w)$ taken by node $w \in N(v) \setminus \{u\} \cap S_c$ that is not chosen by any of u's neighbors $x \in N(u)$. Due to Theorem 6 $|C_{N(v)\cap S_c}|/4$ colors exist that are taken by some node $w \in N(v) \cap S_c$ but not taken (or chosen for the second choice) by a neighbor $x \in N(u)$. Due to Theorem 5 we have $|C_{N(v)\cap S_c}|/4 \geq |N(v)\cap S_c|/(64c_1)$. Additionally, the theorem yields $|P^1_{N(v)\cap S_c}| \geq |N(v) \cap S_c|/(4c_1)$.

The probability for a node $u \in P^1_{N(v)\cap S_c}$ to obtain (not only pick) a color in $C_{N(v)\cap S_c}$ becomes the number of "good" colors, i.e., $|N(v) \cap S_c|/(64c_1)$, divided by the total number of available colors, i.e., $1/(\Delta_0/2)$, yielding $|N(v)\cap S_c|/(32c_1 \cdot \Delta_0)$. This holds irrespectively of the behavior of other nodes $w \in P^1_{N(v)\cap S_c}$ and other sets $N(v) \cap S_d$ with $d \in [0, \chi - 1] \setminus \{c\}$. The reason is that a node u makes its decision what color to pick independently of its neighbors $y \in N(u)$ and Theorems 5 and 6 already account for the worst case behavior of neighbors $y \in N(u)$ to bound the probability that node u gets a chosen color.

Thus, for a set of $|P^1_{N(v)\cap S_c}| \geq |N(v) \cap S_c|/(4c_1)$ nodes we expect that for at least $|N(v)\cap S_c|^2/(128c_1^2 \cdot \Delta_0)$ nodes u there exists another node $w \in (N(v)\cap S_c)\setminus \{u\}$ with the same color. The expectation $|N(v) \cap S_c|^2/(128c_1^2 \cdot \Delta_0)$ is minimized if all sets $|N(v) \cap S_c| \geq c_3 \log n$ are of equal size and as small as possible, i.e., $\Delta_0/(4\chi)$ since at least $\Delta_0/4$ nodes are in sets $|N(v) \cap S_c| \geq c_3 \log n$ for some c. This gives $\sum_{c\in[0,\chi-1]} |N(v) \cap S_c|^2/(128c_1^2 \cdot \Delta_0) \geq \sum_{c\in[0,\chi-1]} (\Delta_0)^2/(2048c_1^2 \cdot \Delta_0 \cdot \chi^2) = \Delta_0/(c_5 \cdot \chi)$ for $c_5 = 2048c_1^2$. Since by assumption $\chi \in O(\Delta_0/\log n)$ using Corollary 2 the actual number deviates by at most $1/2$ of its expectation with probability $1 - 1/n^{c_4}$ for an arbitrary constant c_4. Therefore, for at least $\Delta_0/(c_5 \cdot \chi)$ nodes $u \in N(v) \cap S_c$ there exists another node $w \in N(v) \setminus \{u\} \cap S_c$ with the same color. Thus to color all of these $\Delta_0/(c_5 \cdot \chi)$ nodes only $\Delta_0/(2c_5 \cdot \chi)$ colors are used.

Theorem 8 *If* $\Delta \in \Omega(\log^{1+1/\log^* n} n)$ *and* $\chi \in O(\Delta/\log^{1+1/\log^* n} n)$ *then Algorithm* FastRandColoring *computes a* $(1 - 1/O(\chi))\Delta$ *coloring in time* $O(\log \chi + \log^* n)$ *w.h.p..*

Proof. Extending Theorem 7 to all nodes using Theorem 4 we have w.h.p. that each node v with $d(v) \geq \Delta_0/2$ has at most $(\Delta_0 + 1) \cdot (1 - 1/(c_5\chi))$ uncolored neighbors after the first two choices. However, node v is allowed to use $d(v) + 1$ colors and, additionally, half of the conserved colors, i.e., $\Delta_0/(8c_2\chi) \geq \log^{1+1/\log^* n} n/(4c_5)$ (see Theorem 7), colors to get a color itself. Nodes with initial degree $d(v) < \Delta_0/2$ can use much more colors, i.e., at least $3\Delta_0/4$. When executing Algorithm *DeltaPlus1Coloring* [15] the maximum degree is reduced by a factor 2 in $O(1)$ rounds as long as it is larger than $\Omega(\log n)$ due to Theorem 8 in [15]. Thus, since $\Delta_0/\chi \in O(\log^{1+1/\log^* n} n)$ the time until the maximum degree Δ is less than $O(\max(\Delta_0/\chi, \log n))$ is given by $O(\log \Delta_0 - \log(\Delta_0/\chi)) = O(\log \chi)$. Thus, we have at least 2Δ colors available, i.e., at least $\log^{1+1/\log^* n} n/(4c_5)$ additional colors, when calling Algorithm *ConstDeltaColoring*[15]. Therefore, the remaining nodes are colored in time $O(\log^* n)$ using Corollary 14 [15].

6 Conclusion

It is still an open problem, whether deterministic $\Delta + 1$ coloring in a general graph is possible in time $\Delta^{1-\epsilon} + \log^* n$ for a constant ϵ. Our algorithm indicates

that this might well be the case, since we broke the bound for a wide class of graphs.

Though it is hard in a distributed setting – and sometimes not even possible – to use less than $\Delta + 1$ colors, we feel that one should also keep an eye on the original definition of the coloring problem in a distributed environment: Color a graph with as little colors as possible. To strive for a $\Delta + 1$ coloring is of much appeal and gives interesting insights but as we have shown (in many cases) better bounds regarding the number of used colors and the required time complexity can be achieved by taking the chromatic number of the graph into account.

References

1. S. Arora and E. Chlamtac. New approximation guarantee for chromatic number. In *Symp. on Theory of computing(STOC)*, 2006.
2. L. Barenboim and M. Elkin. Sublogarithmic distributed MIS algorithm for sparse graphs using nash-williams decomposition. In *PODC*, 2008.
3. L. Barenboim and M. Elkin. Distributed $(\delta + 1)$-coloring in linear (in δ) time. In *Symp. on Theory of computing(STOC)*, 2009.
4. L. Barenboim and M. Elkin. Deterministic distributed vertex coloring in polylogarithmic time. In *Symp. on Principles of distributed computing(PODC)*, 2010.
5. A. Blum. New approximation algorithms for graph coloring. *Journal of the ACM*, 41:470–516, 1994.
6. B. Bollobas. Chromatic nubmer, girth and maximal degree. *Discrete Math.*, 24:311–314, 1978.
7. D. A. Grable and A. Panconesi. Fast distributed algorithms for Brooks-Vizing colorings. *J. Algorithms*, 37(1):85–120, 2000.
8. M. M. Halldórsson and J. Radhakrishnan. Greed is good: approximating independent sets in sparse and bounded-degree graphs. In *STOC*, 1994.
9. M. Karchmer and J. Naor. A fast parallel algorithm to color a graph with delta colors. *J. Algorithms*, 9(1):83–91, 1988.
10. F. Kuhn. Weak Graph Coloring: Distributed Algorithms and Applications. In *Symp. on Parallelism in Algorithms and Architectures (SPAA)*, 2009.
11. F. Kuhn and R. Wattenhofer. On the Complexity of Distributed Graph Coloring. In *Symp. on Principles of Distributed Computing (PODC)*, 2006.
12. N. Linial. Locality in Distributed Graph Algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.
13. A. Panconesi and A. Srinivasan. Improved distributed algorithms for coloring and network decomposition problems. In *Symp. on Theory of computing (STOC)*, 1992.
14. J. Schneider and R. Wattenhofer. A Log-Star Distributed Maximal Independent Set Algorithm for Growth-Bounded Graphs. In *Symp. on Principles of Distributed Computing(PODC)*, 2008.
15. J. Schneider and R. Wattenhofer. A New Technique For Distributed Symmetry Breaking. In *Symp. on Principles of Distributed Computing(PODC)*, 2010.