On the Complexity of Distributed Graph Coloring

Fabian Kuhn
Microsoft Research – Silicon Valley
1065 La Avenida
Mountain View, CA 94043, USA
kuhn@microsoft.com

Roger Wattenhofer
Computer Engineering and Networks Laboratory
ETH Zurich
8092 Zurich, Switzerland
wattenhofer@tik.ee.ethz.ch

ABSTRACT

Coloring the nodes of a graph with a small number of colors is one of the most fundamental problems in theoretical computer science. In this paper, we study graph coloring in a distributed setting. Processors of a distributed system are nodes of an undirected graph G. There is an edge between two nodes whenever the corresponding processors can directly communicate with each other. We assume that distributed coloring algorithms start with an initial m-coloring of G. In the paper, we prove new strong lower bounds for two special kinds of coloring algorithms. For algorithms which run for a single communication round—i.e., every node of the network can only send its initial color to all its neighbors—, we show that the number of colors of the computed coloring has to be at least $\Omega(\Delta^2/\log^2\Delta + \log\log m)$. If such one-round algorithms are iteratively applied to reduce the number of colors step-by-step, we prove a time lower bound of $\Omega(\Delta/\log^2\Delta + \log^* m)$ to obtain an $O(\Delta)$-coloring. The best previous lower bounds for the two types of algorithms are $\Omega(\log\log m)$ and $\Omega(\log^* m)$, respectively.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*computations on discrete structures*;
G.2.2 [**Discrete Mathematics**]: Graph Theory—*graph algorithms*;
G.2.2 [**Discrete Mathematics**]: Graph Theory—*network problems*

General Terms

Algorithms, Theory

Keywords

chromatic number, distributed algorithms, graph coloring, locality, neighborhood graph, symmetry breaking

PODC'06, July 22-26, 2006, Denver, Colorado, USA.
Copyright 2006 ACM 1-59593-384-0/06/0007 ...$5.00.

1. INTRODUCTION

Many problems which are simple in standard sequential systems become much harder in a distributed setting. A classic example is the problem of *coloring* the nodes of a graph with $O(\Delta)$ colors where Δ is the largest degree Δ of the graph. While in sequential systems, a $(\Delta+1)$-coloring can be obtained by a simple greedy algorithm, the complexity of finding such a coloring in a distributed way is unknown and considered to be an important open problem. Apart from being interesting from a theoretical perspective, the problem is important because many network coordination primitives are based on colorings of the nodes of the network. For example, the assignment of frequencies or time slots in wireless networks are classical applications of minimum graph coloring [12, 24]. In contrast to many other distributed graph-theoretic problems [7, 13, 16], not much progress has been made on distributed coloring in the last few years. The goal of this paper is to improve this situation and to make a step towards understanding the distributed complexity of graph coloring.

Throughout the paper, we study the standard *message passing* model for distributed computing [17, 23]. Processors of a distributed system are represented by nodes of an undirected graph $G = (V, E)$. Two processors can communicate with each other directly if and only if they are connected by an edge in the graph. We assume that the processors operate in synchrony[1], that is, communication is round-based. In each round, every node can send a message to all its neighbors in the graph, receive the messages sent by its neighbors, and perform some local computation based on the information contained in the received messages. Since in distributed systems the running time of an algorithm is mainly determined by the time needed for the communication, we make no restrictions on local computations. Further we assume that all processors start an algorithm at the same time. The *time complexity* of a distributed algorithm then is the maximum possible number of rounds needed until every node has completed its computation.

A proper s-coloring of a graph $G = (V, E)$ is an assignment $\gamma : V \to \{1\ldots, s\}$ of colors between 1 and s to nodes such that adjacent nodes have different colors, that is, $(u, v) \in E \Rightarrow \gamma(u) \neq \gamma(v)$. The minimum number of colors which are needed to properly color the vertices of a given graph G is called the chromatic number $\chi(G)$ of G. In the distributed setting, the goal is to color the network graph

[1] As long as we only care about running time and not about the number of messages that have to be sent, synchronous and asynchronous message passing models are equivalent.

G. We assume that the distributed computation starts with an initial proper m-coloring of G.[2] Initially, every node only knows its own color. By exchanging messages, the nodes of G have to learn about the colors of other nodes and compute a new color based on this information.

In this paper, we look at two special kinds of distributed coloring algorithms. We first study the most simple case by considering algorithms which run for one round only. In one round, every node can learn the initial colors of its direct neighbors. Based on this information, each node has to choose a new color such that the colors chosen by adjacent nodes are different. In [17], it has been shown that given an initial coloring with m colors, it is possible to deterministically compute an $\mathrm{O}(\Delta^2 \log m)$-coloring in a single round. The best known lower bound is also from [17] where it is shown that $\Omega(\log\log m)$ colors are needed. We show that if $m \in \Omega(\Delta^2/\log\Delta)$, a coloring computed by a one-round algorithm must consist of least $\Omega(\Delta^2/\log^2\Delta + \log\log m)$ colors. In Section 6, we show that for appropriate values of Δ, m, and n—the number of nodes of G—, this lower bound can be broken by using randomization. We present a randomized one-round algorithm which computes a $\mathrm{O}(\Delta \log n \log m)$-coloring with high probability.

The second kind of algorithms that we address are distributed coloring methods which are based on iterative applications of one-round coloring algorithms to reduce the number of colors step by step, a technique which is applied by many known algorithms [2, 11, 17]. We show that in order to obtain an $\mathrm{O}(\Delta)$-coloring, any such algorithm needs at least $\Omega(\Delta/\log^2\Delta + \log^* m)$ rounds. The log-star function $\log^* m$ denotes the number of logarithms one has to apply to m in order to get a value smaller than 1. The best previous lower bound is proved in [17] where it is shown that every distributed coloring algorithm needs at least $\Omega(\log^* m)$ rounds to obtain an $\mathrm{O}(\Delta)$-coloring of the ring. In addition to the lower bound, we also slightly improve the best upper bound for computing a $(\Delta+1)$-coloring from $\mathrm{O}(\Delta \log m)$ [2] and $\mathrm{O}(\Delta^2 + \log^* m)$ [11] to $\mathrm{O}(\Delta\log\Delta + \log^* m)$ rounds. Using randomization, the number of one-round steps needed to obtain a $(\Delta+1)$-coloring is $\mathrm{O}(\Delta \log\log n)$ with high probability.

The remainder of the paper is organized as follows. In Section 2, we summarize important previous work. Section 3 introduces the concept of neighborhood graphs which we will need to analyze one-round coloring algorithms. Neighborhood graphs have first been used in [17]. In Sections 4 and 5, we describe our results about deterministic one-round algorithms and iterative applications of one-round color reductions, respectively. Section 6 shows that randomization can help in some situations. Finally, Section 7 concludes the paper.

2. RELATED WORK

Finding a coloring with $\chi(G)$ colors is one of the first problems which was shown to be NP-hard [10, 14]. A lot of progress has been made since then, showing that for general graphs, we cannot hope to find reasonably good colorings in polynomial time. In particular, unless P $=$ NP, for every constant $\varepsilon > 0$, minimum graph coloring cannot be approximated better than $|V|^{1/7-\varepsilon}$ [3]. If NP $\neq$ ZPP, that is, if NP-problems cannot be solved by a randomized algorithm in expected polynomial time, one can even show that minimum graph coloring cannot be approximated better than $\Omega(|V|^{1-\varepsilon})$ [9]. Clearly, when considering distributed coloring algorithms, our goal cannot be to achieve better colorings than in a non-distributed scenario.[3] We therefore usually do not compare an obtained solution with a global optimal solution. We rather compare a computed coloring to what we can achieve by a sequential algorithm. Therefore, for distributed algorithms, mostly, the ultimate goal is to achieve a $(\Delta+1)$-coloring or even just an $\mathrm{O}(\Delta)$-coloring of the network graph.

There is an intriguing relation between the distributed complexities of computing $(\Delta+1)$-colorings and maximal independent sets (MIS). On the one hand, the best coloring algorithms for general graphs are based on MIS algorithms. On the other hand, the best MIS algorithms for small-degree graphs are based on coloring algorithms.

The best distributed MIS algorithm for general graphs is a simple randomized algorithm with expected time complexity $\mathrm{O}(\log n)$ [1, 19]. In [17], a nice reduction from $(\Delta+1)$-coloring to the MIS problem is described. For a given graph G which we want to color with $\Delta+1$ colors, a graph G' is constructed as follows. We make $\Delta+1$ copies $v_0, \ldots, v_\Delta$ for every node v of G. All $\Delta+1$ copies are connected to form a clique. Two nodes u_i and v_j of G' are connected if $(u,v) \in E(G)$ and if $i = j$. Then an MIS of G' is computed by a given distributed algorithm. For every node v of G, exactly one of the $\Delta+1$ copies $v_0, \ldots, v_\Delta$ is in an MIS of G'. If it is v_i, we assign color i to node v. Because $(u,v) \in E(G)$ implies that u_i and v_i cannot both be in the MIS, this gives a $(\Delta+1)$-coloring of the original graph. Applying this reduction with the algorithm of [1, 19] results in a randomized $(\Delta+1)$-coloring algorithm with expected time complexity $\mathrm{O}(\log n)$. Note that the maximal message size grows from $\mathrm{O}(\log n)$ to $\mathrm{O}(\Delta \log n)$ when applying the described reduction.

The problem of finding an MIS or a $(\Delta+1)$-coloring with a deterministic algorithm turns out to be a lot harder. For general graphs, the best algorithms are based on computing a small-diameter network decomposition as described in [2, 21]. With this technique, it is possible to compute both structures in time $n^{\mathrm{O}(1/\sqrt{\log n})}$. There are special graphs for which extremely efficient deterministic algorithms exist. On a ring or on a rooted tree, it is possible to compute an MIS or a 3-coloring in time $\mathrm{O}(\log^* n)$ [5, 11]. Using the algorithm of [5, 11], it is also possible to $(\Delta+1)$-color a constant-degree graph in time $\mathrm{O}(\log^* n)$. Because any k-coloring can be converted into an MIS in k rounds, the time complexity for computing an MIS in bounded degree graphs is also $\mathrm{O}(\log^* n)$. These upper bounds are matched by a lower bound in [17], stating that $\Omega(\log^* n)$ rounds are needed to compute an MIS

[2] Instead of having an initial coloring, it is usually assumed that all nodes have a unique identifier. Note that this can be seen as a special initial coloring where the colors of all nodes are different.

[3] Because we do not bound local computations, the described distributed model in principle allows to achieve better approximation ratios. In fact, if we do not restrict the maximum message size and local computations, it can be shown that there is a randomized $\mathrm{O}(\log n)$-round algorithm with an expected approximation ratio of $\mathrm{O}(\log n)$ [15, 18]. However, it certainly seems unreasonable to study a distributed problem where exponential local computations are unavoidable.

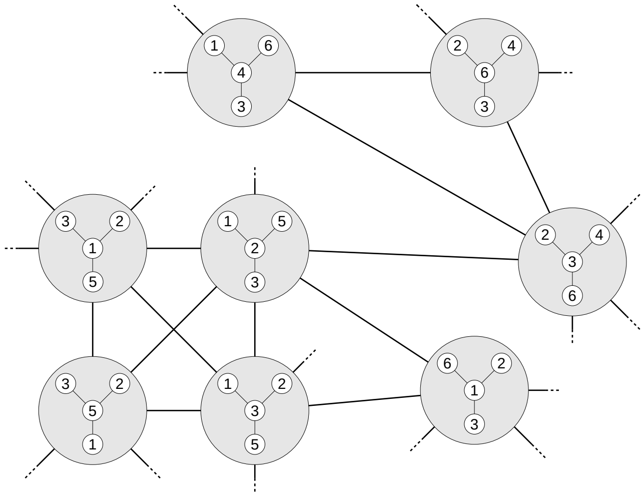

Figure 1: Looking into the one-round neighborhood graph $\mathcal{N}_1(6, 3)$ of 6-colored degree 3 graphs. Nodes $(x, \{x_1, x_2, x_3\})$ are depicted as star graphs with center color x and leaf colors x_1, x_2, and x_3.

or a coloring with a constant number of colors for the ring. Also interesting for graphs with moderate degrees is an algorithm which finds a $(\Delta+1)$-coloring of an arbitrary graph in time $O(\Delta \log n)$ [2, 11, 23]. In [17], a deterministic algorithm which colors any graph with $O(\Delta^2)$ colors in $O(\log^* n)$ rounds is presented. Because any t-coloring can be turned into a t' coloring in $t - t'$ rounds, the algorithm implies an $O(\Delta^2 + \log^* n)$-time algorithm for computing a $(\Delta+1)$-coloring, a result which has also been presented in [11]. New interesting ideas for the problem of coloring a graph by a local algorithm have been presented in [6]. In [6], it is claimed that the presented algorithm colors any graph with $O(\Delta)$ colors in $O(\log^* n)$ rounds. Unfortunately, there is an error in the analysis [22]. With a correct analysis, we can only show that the given algorithm achieves an $O(\Delta^2)$-coloring in $O(\log^* n)$ time, that is, asymptotically, the algorithm is not better than [17].

3. THE NEIGHBORHOOD GRAPH

For the analysis of deterministic algorithms, we use the concept of *neighborhood graphs* which was introduced for ring networks in [17]. The nodes of a k-neighborhood graph are all possible distance k views of the nodes of a network. Two nodes of a neighborhood graph are adjacent if the corresponding views can occur at adjacent nodes of a given network. For distributed coloring algorithms, neighborhood graphs are a means to formalize the possible views on which nodes have to base their decisions and the neighborhood relationships between different views. If two views are adjacent in the neighborhood graph, the corresponding nodes have to choose different colors because otherwise two adjacent nodes having the given views would choose the same color. If two nodes have views which are not adjacent in the neighborhood graph. they may choose the same color.

In a one-round algorithm, every node can send its initial color to all its neighbors. All nodes then have to decide on a new color. Hence, every node has to choose a color based on its own color and the colors of its neighbors only. Therefore after one round, the view of each node is a pair $(x_v, \Gamma_x(v))$ where x_v is the initial color of v and where $\Gamma_x(v)$ is the multi-set of the colors of the neighbors of v.

The value of x_v and all elements of $\Gamma_x(v)$ are integers between 1 and m. If we only consider graphs with degrees at most Δ, we have $|\Gamma_x(v)| \leq \Delta$ and because we start with a valid coloring, we also have $x_v \notin \Gamma_x(v)$. There is a network graph with initial coloring in which two one-hop views $(x_u, \Gamma_x(u))$ and $(x_v, \Gamma_x(u))$ are the one-hop views of adjacent nodes if and only if $x_u \in \Gamma_x(v)$ and $x_v \in \Gamma_x(u)$. It turns out that we can w.l.o.g. assume that all nodes have exactly degree Δ and that all neighbors of any node v have different colors (cf. Lemma 3.1). The neighborhood graph $\mathcal{N}_1(m, \Delta)$ for one-round algorithms on m-colored graphs of degree at most Δ can therefore be defined as follows.

DEFINITION 3.1. **(Neighborhood Graph)** *Let $\Delta \geq 1$ and $m > \Delta$ be two integer parameters. The node set of the neighborhood graph $\mathcal{N}_1(m, \Delta)$ consists of all pairs (x, Γ_x) for which $x \in \{1, \ldots, m\}$, $\Gamma_x \subset \{1, \ldots, m\}$, $|\Gamma_x| = \Delta$, and $x \notin \Gamma_x$. There is an edge between two nodes (x, Γ_x) and (y, Γ_y) if and only if $x \in \Gamma_y$ and $y \in \Gamma_x$.*

Figure 1 illustrates Definition 3.1 by showing a small part of $\mathcal{N}_1(6, 3)$. The fundamental relation between the neighborhood graph $\mathcal{N}_1(m, \Delta)$ and one-round coloring algorithms is given by the following lemma.

LEMMA 3.1. *Let $\mathcal{G}$ denote the class of m-colored graphs with maximum degree Δ. In one round, a deterministic distributed algorithm can color a graph of $\mathcal{G}$ with exactly $q = \chi\big(\mathcal{N}_1(m, \Delta)\big)$ colors. In other words, there is an algorithm which colors every graph of $\mathcal{G}$ with at most q colors and for every algorithm, there is a graph in $\mathcal{G}$ for which the resulting number of colors is at least q.*

PROOF. We first prove that there is an algorithm which colors every properly m-colored graph G with q colors in one round. Slightly more general, we show that every t-coloring of $\mathcal{N}_1(m, \Delta)$ can be turned into a one-round algorithm for t-coloring arbitrary m-colored graphs with maximum degree Δ. Assume that we have a t-coloring which is globally known, that is, all nodes of G know the coloring. Note that this is no problem since $\mathcal{N}_1(m, \Delta)$ does only depend on m and Δ and is independent of a particular network graph G. Let v be a node of G and let x_v be the initial color of v. Further, let $\Gamma_x(v)$ denote the set of initial colors of the neighbors of v. If $|\Gamma_x(v)| = \Delta$, v selects the color of $(x_v, \Gamma_x(v))$ in the t-coloring of $\mathcal{N}_1(m, \Delta)$ as its new color. If $|\Gamma_x(v)| < \Delta$, v first adds $\Delta - |\Gamma_x(v)|$ arbitrary additional colors to $\Gamma_x(v)$ and then select the new color in the described fashion. If u and v are neighbors in G, we must have $x_v \in \Gamma_x(u)$ and $x_u \in \Gamma_x(v)$. By the definition of $\mathcal{N}_1(m, \Delta)$ and because t is a valid coloring of $\mathcal{N}_1(m, \Delta)$, the new colors selected by u and v are different.

It remains to show that q is also a lower bound on the number of colors which a one-round algorithm can achieve. Every deterministic one-round t-coloring algorithm can be seen as a function from the possible one-hop views $(x_v, \Gamma_x(v))$ of a node v to a number in $\{1, \ldots, t\}$. This implies a function γ from the nodes of $\mathcal{N}_1(m, \Delta)$ to $\{1, \ldots, t\}$. If $t < q$, there must be two adjacent nodes (x, Γ_x) and (y, Γ_y) of $\mathcal{N}_1(m, \Delta)$ to which γ assigns the same value. However, by the definition of $\mathcal{N}_1(m, \Delta)$ we can construct a graph G where (x, Γ_x)

and (y, Γ_y) occur as the one-hop views of two adjacent nodes u and v of G. As a consequence, the algorithm corresponding to the function γ assigns the same color to u and v. □

Lemma 3.1 shows that in order to study the potential of one-round coloring algorithms and coloring algorithms based on iterative one-round color reduction steps, it suffices to analyze the chromatic number of $\mathcal{N}_1(m, \Delta)$ for general m and Δ. Before starting a more detailed discussion, let us look at a few simple properties of $\chi\big(\mathcal{N}_1(m, \Delta)\big)$. From the definition of $\mathcal{N}_1(m, \Delta)$, it is immediate that $\chi\big(\mathcal{N}_1(m, \Delta)\big) \geq \chi\big(\mathcal{N}_1(m-1, \Delta)\big)$ and $\chi\big(\mathcal{N}_1(m, \Delta)\big) \geq \chi\big(\mathcal{N}_1(m, \Delta - 1)\big)$. Further, if $m > \Delta + 1$, it can be shown that $\chi\big(\mathcal{N}_1(m, \Delta)\big) \leq m - 1$, that is, as long as there are more than $\Delta + 1$ colors, we can always make progress by applying a one-round algorithm.

4. DETERMINISTIC ONE-ROUND ALGORITHMS

The best known upper and lower bounds for one-round coloring algorithms come from [17] where it is proved that $\chi\big(\mathcal{N}_1(m, \Delta)\big) \in \mathrm{O}(\Delta^2 \log m)$ and $\chi\big(\mathcal{N}_1(m, 2)\big) \in \Omega(\log \log m)$. The upper bound is based on the existence of the following set system. For some $t \in \mathrm{O}(\Delta^2 \log m)$, there are m subsets $S_1, \dots, S_m$ of $\{1, \dots, t\}$ such that for every $(\Delta + 1)$-tuple of sets $S_{i_0}, \dots, S_{i_\Delta}$ for $i_j \in \{1, \dots, m\}$, we have

$$S_{i_0} \setminus \bigcup_{j=1}^{\Delta} S_{i_j} \neq \emptyset.$$

A node v of color i_0 with neighbors of colors $i_1, \dots, i_\Delta$ then chooses a value which is in S_{i_0} but not in S_{i_j} for $j \in \{1, \dots, \Delta\}$ as its new color. Based on a result in [8], it is also proved in [17] that with such a set system, we cannot obtain a coloring with less than $\Omega(\Delta^2)$ colors. In this section, we extend this result to general one-round algorithms proving that for large enough m, $\chi\big(\mathcal{N}_1(m, \Delta)\big) \in \Omega(\Delta^2 / \log^2 \Delta)$.

We start our analysis of the chromatic number of the neighborhood graph $\mathcal{N}_1(m, \Delta)$ by looking at the structure of independent sets of the $\mathcal{N}_1(m, \Delta)$. Because in a proper coloring of a graph, the nodes of each individual color form an independent set, finding a minimum coloring can equivalently be formulated as finding a minimum number of independent sets such that each node is contained in at least one independent set.

The above observations allow us to define a relation $\lhd_S$ among the colors $1, \dots, m$ for each independent set S of $\mathcal{N}_1(m, \Delta)$. For an independent set S and two colors $x, y \in \{1, \dots, m\}$, we define $x \lhd_S y$ if and only if there is a node $(x, \Gamma_x) \in S$ for which $y \in \Gamma_x$. By the definition of $\mathcal{N}_1(m, \Delta)$ and because S is an independent set, $\lhd_S$ is antisymmetric: $x \lhd_S y \implies \neg(y \lhd_S x)$. Note that $\lhd_S$ does not have to be a partial order. In particular, $\lhd_S$ can contain cycles. For convenience, we also define the complementary relation of $\lhd_S$ as $x \not\lhd_S y := \neg(x \lhd_S y)$. In the described manner, it is not only possible to define an antisymmetric relation $\lhd_S$ for each independent set S, we can also find an independent set $S_\lhd$ for each antisymmetric relation $\lhd$. The independent set $S_\lhd$ consists of all nodes (x, Γ_x) for which $x \lhd y$ for all $y \in \Gamma_x$. The following lemma shows that if $\lhd$ is a total order, $S_\lhd$ is a maximum independent set.

Lemma 4.1. *Let N be the number of nodes of $\mathcal{N}_1(m, \Delta)$ and let $\prec$ be a total order on $\{1, \dots, m\}$. The resulting independent set $S_\prec$ is a maximum independent set of size $N/(\Delta + 1)$ of $\mathcal{N}_1(m, \Delta)$.*

Proof. Let D be a $(\Delta+1)$-subset of $\{1, \dots, m\}$. For each $x \in D$, there is a node $(x, D \setminus \{x\})$ in $\mathcal{N}_1(m, \Delta)$. For a given total order $\prec$ on $\{1, \dots, m\}$, exactly one of the $\Delta + 1$ colors is the smallest w.r.t. $\prec$. Hence, exactly one of the $\Delta + 1$ nodes with colors in D is in $S_\prec$. Consequently, a $1/(\Delta + 1)$-fraction of all nodes of $\mathcal{N}_1(m, \Delta)$ is in $S_\prec$. Because the nodes of $\mathcal{N}_1(m, \Delta)$ corresponding to the one-round views of the $\Delta + 1$ nodes of a complete graph $K_{\Delta+1}$ form a clique of size $\Delta + 1$ in $\mathcal{N}_1(m, \Delta)$, every node of $\mathcal{N}_1(m, \Delta)$ is in a $(\Delta + 1)$-clique. Therefore $S_\prec$ is a maximum independent set. □

In fact, it can even be shown that the maximum independent sets defined by total orders in the described way are the only maximum independent sets of $\mathcal{N}_1(m, \Delta)$. A direct implication of the structure of maximum independent sets of $\mathcal{N}_1(m, \Delta)$ is given by the following Corollary 4.2. The fractional chromatic number of a graph G is defined as the size of the smallest fractional covering of the nodes of G with independent sets of G. That is, we have to assign a weight x_S to every independent set such that for every node v, the sum of the weights of independent sets containing v is at least 1 and such that the total sum of x_S is minimized.

Corollary 4.2. *For $m > \Delta$, the fractional chromatic number of $\mathcal{N}_1(m, \Delta)$ is $\chi_f\big(\mathcal{N}_1(m, \Delta)\big) = \Delta + 1$.*

Proof. By Lemma 4.1 and by symmetry, every node of $\mathcal{N}_1(m, \Delta)$ is in $m!/(\Delta+1)$ independent sets of size $N/(\Delta+1)$ defined by the $m!$ possible total orders on $\{1, \dots, m\}$. To obtain a fractional covering of the nodes with independent sets, we have to assign a weight x_S to each independent set S of G such that for each node v of G, the sum of the weights of all independent sets containing v is at least 1. By assigning $x_S = (\Delta + 1)/m!$ to each of the $m!$ maximum independent sets of $\mathcal{N}_1(m, \Delta)$, we obtain a fractional covering of size $\Delta + 1$. □

As a consequence of Corollary 4.2, we obtain the following upper bound on $\chi\big(\mathcal{N}(m, \Delta)\big)$.

Theorem 4.3. *For all $m > \Delta$, the chromatic number of $\mathcal{N}_1(m, \Delta)$ is $\chi\big(\mathcal{N}_1(m, \Delta)\big) \leq (\Delta + 1)^2(\ln m + 1)$.*

Proof. The chromatic number $\chi(G)$ of a graph G equals the number of independent sets needed to cover all nodes of G. Hence, $\chi(G)$ is the solution of a minimum set cover instance. The integrality gap (maximal ratio between an optimal integer solution and an optimal fractional solution) of minimum set cover is at most $\ln s + 1$, where s is the size of the largest set. We thus have $\chi(G) \leq \big(\ln(\alpha(G)) + 1\big)\chi_f(G)$ where $\alpha(G)$ is the size of a maximum independent set of G. The theorem follows because

$$\alpha\big(\mathcal{N}_1(m, \Delta)\big) \; = \; \binom{m}{\Delta + 1} \; < \; m^{\Delta+1}.$$

□

Remark: In [17], it has also been shown that the chromatic number $\chi\big(\mathcal{N}_1(m,\Delta)\big) \in \mathrm{O}(\Delta^2 \log m)$. Theorem 4.3 is a small constant improvement over the result of [17]. Up to lower-order terms, it is better by a factor of 4.

In [17], it is shown that even for degree 2 graphs, every one-round coloring algorithm needs at least $\Omega(\log\log m)$ colors. Together with Corollary 4.2, we obtain $\Omega(\Delta+\log\log m)$ as a lower bound on $\chi\big(\mathcal{N}_1(m,\Delta)\big)$. In the following, we will significantly improve this lower bound showing that the $\Omega(\Delta^2)$ lower bound for algorithms based on the technique described in [17] is almost tight for general one-round algorithms.

We start with an outline of the lower bound proof. Let $S_1,\dots,S_q$ be q independent sets of $\mathcal{N}_1(m,\Delta)$ for some given m and Δ. If every node of $\mathcal{N}_1(m,\Delta)$ is in at least one of the q independent sets, the chromatic number of $\mathcal{N}_1(m,\Delta)$ is at most q. To prove a lower bound, the goal therefore is to show that if q is small enough, we can find at least one node which is not in any of the independent sets. To argue about the role of a color x in independent set S, we define $\Phi_S(x) := \big\{y \,|\, x \not\lhd_S y\big\}$ and $\varphi_S(x) := \big|\Phi_S(x)\big|$. Hence, $\Phi_S(x)$ denotes the set of colors y for which $y \notin \Gamma_x$ for all nodes $(x,\Gamma_x) \in S$. Consequently $\varphi_S(x)$ is the number of colors which do not occur in Γ_x for $(x,\Gamma_x) \in S$. Our goal is to show that for small enough q, we can find $\Delta+1$ colors $x, y_1,\dots,y_\Delta \in \{1,\dots,m\}$ such that

$$\forall i \in \{1,\dots,q\},\ \exists j \in \{1,\dots,\Delta\} \ :\ y_j \in \Phi_{S_i}(x) \qquad (1)$$

for any possible choice of independent sets $S_1,\dots,S_q$. If we can find such colors, it follows that the node $(x,\{y_1,\dots,y_\Delta\})$ is not in any of the q independent sets $S_1,\dots,S_q$ and does therefore not get one of the q colors. Hence, these q independent sets do not define a valid coloring of $\mathcal{N}_1(m,\Delta)$. For a given color x, the problem to find y_j for which Condition (1) is satisfied can be interpreted as an instance of minimum set cover. The colors $\{1,\dots,m\}\setminus\{x\}$ define the sets and the independent sets S_i define the elements. An independent set S_i is covered by a color y_j if $y_j \in \Phi_{S_i}(x)$. To prove that a one-round coloring algorithm needs more than q colors, we show that there is an $x \in \{1,\dots,m\}$ for which the optimal solution of the described set cover problem is at most Δ. In order to find such an x, we need the following lemma.

Lemma 4.4. *Let $A \subseteq \{1,\dots,m\}$ be a set of colors. For every independent set S of $\mathcal{N}_1(m,\Delta)$, we have*

$$\sum_{x\in A} \varphi_S(x) \ \geq\ \binom{|A|}{2}.$$

Proof. Because $\lhd_S$ is an antisymmetric relation, for any two colors $x_1, x_2 \in A$, we have

$$(x_1 \not\lhd_S x_2) \ \vee\ (x_2 \not\lhd_S x_1). \qquad (2)$$

By definition, $\varphi_S(x)$ is the number of colors y for which $x \not\lhd_S y$. Hence, Equation (2) implies that for any two colors $x_1, x_2 \in A$, either $\varphi_S(x_1)$ or $\varphi_S(x_2)$ is increased by 1. Because the number of pairs in A is $\binom{|A|}{2}$, the claim follows. □

Lemma 4.4 implies that for each independent set S and each set A of colors, for at least half of the $|A|$ colors, $\varphi_S(x) \in \Omega(|A|)$. Hence, for each independent set S, many colors x are bad center colors because there are many colors which cannot occur as peripheral colors. In the following, we show that for small enough q, there must be a color x for which $\varphi_{S_i}(x)$ is large for almost all independent sets S_i, $i \in \{1,\dots,q\}$. To do so, for every x, we sort the values $\varphi_{S_i}(x)$ $(i = 1,\dots,q)$ in increasing order. For all $j \in \{1,\dots,q\}$, we denote the j^{th} value $\varphi_{S_i}(x)$ in this sorted order by $h_j(x)$. Ties are broken arbitrarily. The next lemma shows that there is an x for which $h_j(x)$ grows linearly with j.

Lemma 4.5. *Let t be an integer. If $t\big((m-q)t-q\big) > 2q(m-1)$, there is an $x \in \{1,\dots,m\}$ for which*

$$\sum_{i=1}^{t} h_i(x) \ \geq\ m \quad \text{and} \quad h_1(x) > 0.$$

Proof. Let $Q \subset \{1,\dots,m\}$ be the set of colors x for which there is an $i \in \{1,\dots,q\}$ such that $\varphi_{S_i}(x) = 0$, that is, $Q = \{x \in \{1,\dots,m\} \big| h_1(x) = 0\}$. For each independent set S, $\varphi_S(x) = 0$ for at most one color x. If $\varphi_S(x) = \varphi_S(y) = 0$ for two different colors $x \neq y$, this would imply that $x \lhd_S y \wedge y \lhd_S x$ which is not possible because $\lhd_S$ is antisymmetric. We therefore have $|Q| \leq q$. Let P be the complementary color set of Q, that is, $P = \{1,\dots,m\}\setminus Q$. We want to show that if $t\big((m-q)t-q\big) > 2q(m-1)$, there is an $x \in P$ for which $\sum_{i=1}^{t} h_i(x) \geq m$. For the sake of contradiction, assume that this is not the case and thus

$$\forall x \in P : \sum_{i=1}^{t} h_i(x) \leq m-1$$

$$\Longrightarrow \quad \sum_{x\in P}\sum_{i=1}^{t} h_i(x) \leq |P|(m-1). \qquad (3)$$

Let us take a closer look at the double sum in the right inequality of (3). The sum is over $|P|\cdot t$ different $h_i(x)$ values and therefore also over $|P|\cdot t$ different $\varphi_S(x)$ values. Let us denote the number of $\varphi_{S_j}(x)$ values for a particular independent set S_j in the double sum of Inequality (3) by $a(S_j)$. By Lemma 4.4, we have

$$\sum_{x\in|P|}\sum_{i=1}^{t} h_i(x) \ \geq\ \sum_{j=1}^{q}\binom{a(S_j)}{2},$$

$$\text{where} \quad \sum_{j=1}^{q} a(S_j) = |P|\cdot t. \qquad (4)$$

For two integers A and B with $A < B$, we have $\binom{A}{2} + \binom{B}{2} \geq \binom{A+1}{2} + \binom{B-1}{2}$. Combining with Inequality (4), we obtain

$$\begin{aligned}\sum_{x\in|P|}\sum_{i=1}^{t} h_i(x) \quad &\geq \quad \sum_{j=1}^{q}\binom{a(S_j)}{2} \\ &\geq \quad q\cdot\binom{|P|t/q}{2} \ = \ \frac{|P|t\cdot(|P|t-q)}{2q}.\end{aligned}$$

Note that the above inequality also holds if $|P|t/q$ is not integral. Combined with Inequality (3), we therefore have $2q(m-1) \geq t(|P|t-q) \geq t\big((m-q)t-q\big)$ which is a contradiction because we assumed that $t\big((m-q)t-q\big) > 2q(m-1)$. □

As described above, for each center color x, finding a node of $\mathcal{N}_1(m,\Delta)$ which is not covered by a given set of q independent sets of $\mathcal{N}_1(m,\Delta)$ can be seen as a minimum set cover problem where the colors $\{1,\dots,m\}$ are the sets and the q independent sets are the elements. Using Lemma 4.5, we can now prove that there is a color for which the above described set cover has a small fractional solution.

Lemma 4.6. *If $t\big((m-q)t-q\big) > 2q(m-1)$, there is a color $x \in \{1,\dots,m\}$ for which the above described minimum set cover problem has a fractional solution of size at most t.*

Proof. We have to assign a fractional value λ_y to each color $y \in \{1,\dots,m\} \setminus \{x\}$ such that

$$\forall i \in \{1,\dots,q\} : \sum_{y \in \Phi_{S_i}(x)} \lambda_y \ \geq\ 1. \qquad (5)$$

We define $\lambda_y := 1/\big(\min\{\varphi_{S_i}(x) \,\big|\, y \in \Phi_{S_i}(x)\}\big)$. Clearly, this definition satisfies Condition (5). The value of the given fractional set cover solution is $\sum_y \lambda_y$. By the definition of $h_1(x)$, $h_1(x)$ colors y have a value $\lambda_y = 1/h_1(x)$. Further, at most $h_i(x)$ colors y have a value $\lambda_y = 1/h_i(x)$. Because of Lemma 4.5, there is a color x for which

$$\sum_y \lambda_y \ \leq\ \sum_{i=1}^{t} h_i(x)\frac{1}{h_i(x)} \ =\ t.$$

which concludes the proof. Note that we need that $h_1(x) > 0$ because the independent set S for which $\varphi_S(x) = 0$ contains all nodes (i.e., stars) with center color x. □

Based on Lemma 4.6, we can now also find a color x for which the described set cover problem has a small integer solution. This allows to derive the next lemma.

Lemma 4.7. *We have $\chi\big(\mathcal{N}_1(m,\Delta)\big) > q$ if the following condition holds:*

$$\Delta\left(\Delta - \frac{q\ln(eq)}{m-q}\right) \ >\ \frac{2(m-1)q\ln^2(eq)}{m-q}. \qquad (6)$$

Proof. Let s be the size of the largest set of some given minimum set cover problem. By the analysis of the set cover greedy algorithm [4], the integrality gap of the minimum set cover problem is at most $H(s) \leq \ln s + 1 = \ln(es)$. The largest set of the set cover instance considered in Lemma 4.6 has size less than q. The integrality gap of the considered set cover problem therefore is bounded by $\ln(eq)$. We have seen that if there is some center color x for which there is a solution of size Δ for the problem of covering independent sets with colors, at least one node of $\mathcal{N}_1(m,\Delta)$ is not covered by any of the independent sets. It therefore follows from Lemma 4.6 that $\chi\big(\mathcal{N}_1(m,\Delta)\big) > q$ if

$$\Delta \geq t\ln(eq) \quad \text{and} \quad t\big((m-q)t-q\big) > 2q(m-1). \qquad (7)$$

Plugging the first inequality of (7) into the second inequality of (7), yields the desired inequality. □

This allows us to compute a lower bound for one-round coloring algorithms as given by the following theorem.

Theorem 4.8. *For $m \in \Omega(\Delta^2/\log\Delta)$, the chromatic number of the one-round neighborhood graph is*

$$\chi\big(\mathcal{N}_1(m,\Delta)\big) \ \in\ \Omega\left(\frac{\Delta^2}{\log^2\Delta} + \log\log m\right).$$

Proof. We have to show that Inequality (6) of Lemma 4.7 holds for $q \in \Omega(\Delta^2/\log^2\Delta)$ if $m \in \Omega(\Delta^2/\log\Delta)$. Hence, we assume that $m = d\Delta^2/\ln\Delta$ for some $d \in \Omega(1)$ and show how to find a $c \in \Omega(1)$ such that Inequality (6) holds for $q = c\Delta^2/\ln^2\Delta$. For $c < d$, we have

$$m - q \ =\ \frac{\Delta^2}{\ln\Delta}\cdot\left(d - \frac{c}{\ln\Delta}\right) \ \geq\ \frac{\Delta^2}{\ln\Delta}\cdot\frac{\ln\Delta - 1}{\ln\Delta}.$$

Hence, there is a small enough constant c for which the left-hand side of Inequality (6) is bounded by

$$\begin{aligned}
&\Delta\left(\Delta - \frac{q\ln(eq)}{m-q}\right)\\
&\geq\ \Delta\left(\Delta - \frac{c\Delta^2(\ln(ec) + 2\ln\Delta - 2\ln\ln\Delta)\ln^2\Delta}{\ln^2\Delta\cdot\Delta^2(\ln\Delta-1)}\right)\\
&>\ \Delta(\Delta-1).
\end{aligned}$$

There is a constant c' such that

$$\frac{m}{m-q} \ \leq\ \frac{d\Delta^2\ln^2\Delta}{\ln\Delta\cdot\Delta^2(\ln\Delta-1)} \ \leq\ c'.$$

For the right-hand side of Inequality (6), we therefore obtain

$$\begin{aligned}
\frac{2(m-1)q\ln^2(eq)}{m-q} \quad &\leq \quad \frac{2c'c\Delta^2(\ln(ec) + 2\ln\Delta - 2\ln\ln\Delta)^2}{\ln^2\Delta}\\
&< \quad \Delta(\Delta-1)
\end{aligned}$$

if the constant c is chosen small enough. The $\Omega(\log\log m)$ lower bound follows from the proof of Theorem 2.1 in [17]. □

5. ITERATIVE ONE-ROUND COLOR REDUCTION

The bounds on $\chi\big(\mathcal{N}_1(m,\Delta)\big)$ given by Theorems 4.3 and 4.8 are strong if m is at least in the order of $\Theta(\Delta^2)$. However, if we want lower bound the number of one-round color reduction steps needed to for example achieve an $O(\Delta)$-coloring, we have to bound $\chi(\mathcal{N}_1(m,\Delta))$ for small m. The following theorem gives an upper bound on the chromatic number of the neighborhood graph.

Theorem 5.1. *For all m, the chromatic number of the neighborhood graph is at most*

$$\chi\big(\mathcal{N}_1(m,\Delta)\big) \ \leq\ \left\lceil m\cdot\frac{\Delta+1}{\Delta+2}\right\rceil \ =\ \left\lceil m\cdot\left(1 - \frac{1}{\Delta+2}\right)\right\rceil.$$

Proof. Let G be an m-colored graph of maximum degree Δ. We can construct a q-coloring for G for any q satisfying $q + q/(\Delta+1) = q\cdot(\Delta+2)/(\Delta+1) \geq m$ as follows. Every node v with color $x_v \leq q$ keeps its color. We now still have to assign a color from $\{1,\dots,q\}$ to all nodes having a color greater than q. Let the number of such colors be $t = m - q$. From the above condition, we have $t \leq \lfloor q/(\Delta+1)\rfloor$. We number those colors from x_0 to x_{t-1}, that is, we can for example set $x_i = m - i$. A node v with color x_i chooses a color from the set $\{i(\Delta+1)+1,\dots,(i+1)(\Delta+1)\}$ which is not equal to the original color of any of v's neighbors. Because v can choose among $\Delta+1$ colors, such a color exists. Because nodes having colors x_i and x_j for $i \neq j$ choose their colors from disjoint color ranges, the obtained q-coloring is valid. The given upper bound on $\chi\big(\mathcal{N}_1(m,\Delta)\big)$ satisfies the condition for q because

$$\left\lceil m\cdot\frac{\Delta+1}{\Delta+2}\right\rceil\cdot\frac{\Delta+2}{\Delta+1} \ \geq\ m\cdot\frac{\Delta+1}{\Delta+2}\cdot\frac{\Delta+2}{\Delta+1} \ =\ m.$$

□

By Theorem 5.1, it is possible to transform an m coloring into a $\Delta+1$-coloring by $\mathrm{O}(\Delta\log(m/\Delta))$ consecutive one-round color reduction steps. Additionally, the theorem shows that $\chi\big(\mathcal{N}_1(m,\Delta)\big) \leq m-1$ as long as $m \geq \Delta+2$. Combined with the $\mathrm{O}(\log^* m)$-time, $\mathrm{O}(\Delta^2)$-coloring algorithm of [17]—the algorithm of [17] is also based on iterative one-round color reduction steps—, Theorem 5.1 implies the following corollary.

Corollary 5.2. *Starting with an arbitrary m-coloring, it is possible to obtain a $(\Delta+1)$-coloring by at most $\mathrm{O}(\Delta\log\Delta + \log^* m)$ iterative one-round color reduction steps.*

Note that by combining the algorithm of [17] with the techniques from [2, 11, 23] used to $(\Delta+1)$-color a graph in $\mathrm{O}(\Delta\log n)$ rounds, it would also possible to obtain an $\mathrm{O}(\Delta\log\Delta+\log^*)$-time, $(\Delta+1)$-coloring algorithm. The next theorem shows that one can also bound $\chi\big(\mathcal{N}_1(m,\Delta)\big)$ from below for arbitrary m.

Theorem 5.3. *Assume that $m = \rho\Delta$ for an arbitrary parameter $\rho > 1$. We have*

$$\chi\big(\mathcal{N}_1(m,\Delta)\big) \;\geq\; \left(1 - \mathrm{O}\left(\frac{\rho\log^2\Delta}{\Delta}\right)\right)\cdot m.$$

Proof. By Lemma 4.7, we have $\chi\big(\mathcal{N}_1(m,\Delta)\big) > q$ if

$$\Delta\left(\Delta - \frac{q\ln(eq)}{m-q}\right) \;>\; \frac{2(m-1)q\ln^2(eq)}{m-q}.$$

By substituting $\rho\Delta$ for m and $(1-\varepsilon)m$ for q, we obtain

$$\begin{aligned}\varepsilon\Delta\big(\Delta-(1-\varepsilon)\ln(e(1-\varepsilon)\rho\Delta)\big) \;>&\\ 2(\rho\Delta-1)(1-\varepsilon)\ln^2(e(1-\varepsilon)\rho\Delta)&\end{aligned} \qquad (8)$$

W.l.o.g., we can assume that $\rho \in \mathrm{O}(\Delta/\log^2\Delta)$. We thus have $\ln^2(e(1-\varepsilon)\rho\Delta) \leq c\ln\Delta$ for some constant c. The left-hand side and the right-hand side of Inequality 8 can be bounded as follows:

$$\begin{aligned}\varepsilon\Delta\big(\Delta-(1-\varepsilon)\ln(e(1-\varepsilon)\rho\Delta)\big) &\;\geq\; \varepsilon(\Delta^2 - c\Delta\ln\Delta),\\ 2(\rho\Delta-1)(1-\varepsilon)\ln^2(e(1-\varepsilon)\rho\Delta) &\;\leq\; 2c\rho\Delta\ln^2\Delta.\end{aligned}$$

Inequality 8 therefore holds if

$$\varepsilon \;\geq\; \frac{2c\rho\Delta\ln^2\Delta}{\Delta^2 - c\Delta\ln\Delta} \;\in\; \mathrm{O}\left(\frac{\rho\log^2\Delta}{\Delta}\right)$$

which concludes the proof. □

Theorem 5.3 shows that if we start with a $\rho\Delta$-coloring, at least $\Omega\big(\Delta/(\rho\log^2\Delta)\big)$ rounds are needed to reduce the number of colors by a constant factor. Combining with the $\Omega(\log^* m)$-lower bound of [17], we obtain the following corollary.

Corollary 5.4. *Assume that $m \geq c\beta(\Delta+1)$ for some constant $c > 1$ and some $\beta > 1$. The number of one-round color reduction steps needed to obtain a $\beta(\Delta+1)$-coloring is at least $\Omega\big(\Delta/(\beta\log^2\Delta) + \log^* m\big)$. In particular, any $\mathrm{O}(\Delta)$-coloring algorithm which is based on iterative applications of one-round color reduction steps, needs at least $\Omega(\Delta/\log^2\Delta + \log^* m)$ rounds.*

Algorithm 1 Randomized coloring in one round (code for node v)

```
1: choose color $t_v$ uniformly at random from $\{1,\ldots,\lceil\Delta/\ln n\rceil\}$;
2: send ID($v$) and $t_v$ to all neighbors;
3: let $G_{t_v}$ be the graph induced by all nodes $u$ with $t_u = t_v$;
4: let $\Delta_{t_v}$ be the maximum degree of $G_{t_v}$;
5: compute $\mathrm{O}(\Delta_{t_v}^2\log m)$-coloring of $G_{t_v} \Longrightarrow$ color $y_v$;
6: color $x_v := y_v\lceil\Delta/\ln n\rceil + t_v - 1$
```

6. RANDOMIZED DISTRIBUTED COLORING

Up to this point, we have focused on deterministic coloring algorithms. Let us now explore the potentials of randomized coloring algorithms. In [17], it has been proved that randomization does not help for distributed coloring algorithms. However, in the proof of [17], it is implicitly assumed that a k-round randomized coloring algorithm colors a graph with s colors if the algorithm always stops with a proper s-coloring after at most k rounds. The proof of [17] does not hold if it suffices that the number of rounds is k or the number of colors is s in expectation or with high probability. If we allow this, depending on k, m, Δ, and the number of nodes n, it might be possible that randomized algorithms are strictly better than deterministic ones. Intuitively, randomization does not help if the number of rounds k and the number of initial colors m are fixed and independent of n. Assume that we are given an m-colored graph G for which some algorithm $\mathcal{A}$ has some possibly very small failure probability. We can construct a graph G' consisting of sufficiently many independent copies of G for which the failure probability of $\mathcal{A}$ is arbitrarily close to 1. A formal proof of this argument can be found in [20] where it is shown that randomization does not help to locally (in constant time) construct any labeling consisting of a constant number of different labels.

If we assume that m is a function of n, randomization can help. We will now show that for certain Δ, m, and n randomized one-round coloring algorithms can beat the one-round lower bound for deterministic algorithms given by Theorem 4.8. For deterministic algorithms, we assumed that in one round, every node can learn the labels of all its neighbors. In a randomized algorithm, it is additionally possible to collect all random decisions (i.e., all random bits) of the neighbors. Algorithm 1 describes a randomized algorithm which colors a given network graph in one round. The following theorem shows that Algorithm 1 computes a small, proper coloring. We assume that all nodes know m, Δ, and n.

Theorem 6.1. *For any constant c and with probability $1-1/n^c$, Algorithm 1 computes a proper coloring of the network graph G with $\mathrm{O}(\Delta\log n\log m)$ colors in one round. The choice of the constant c only influences the number of colors by a constant factor.*

Proof. Let Δ_T be $\max\big\{\Delta_t | t \in \{1,\ldots,\lceil\Delta/\ln n\rceil\}\big\}$. We first show that Algorithm 1 computes a valid coloring with $\mathrm{O}(\Delta\Delta_T^2\log m/\log n)$ colors in a single round. The only places where something has to be computed are Lines 1, 5, and 6. The only problem occurs in Line 5. If v knew Δ_{t_v}, an $\mathrm{O}(\Delta_{t_v}^2\log m)$-coloring could be computed by using

Theorem 4.3 or by applying the algorithm of [17]. If we are willing to pay a small constant factor in the number of colors, the described algorithms can be adapted to the case where the maximum degree is not known. Let $\Delta_i := 2^i$. Assume that we are given $\mathrm{O}(\Delta_i^2 \log m)$-colorings of $\mathcal{N}_1(\Delta_i, m)$ for all i such that different colors are used for different Δ_i. A node v with degree $\delta(v)$ can choose its color according to the respective color of the neighborhood graph for the smallest $\Delta_i \geq \delta(v)$.

In order to complete the proof, it therefore suffices to show that $\Delta_T \in \mathrm{O}(\log n)$ with probability $1 - 1/n^c$. To do so, we compute a high probability upper bound for the degree $\delta_{t_v}(v)$ of v in G_{t_v} using Chernoff. Let $Q := \lceil \Delta / \ln n \rceil$. The probability that a neighbor u of v chooses the same color in Line 1 (i.e., $t_u = t_v$) is $1/Q$. The expected number of neighbors u of v for which $t_u = t_v$ is therefore at most $\Delta/Q \leq \ln n$. By applying the Chernoff bound for upper tails, we get

$$\Pr[\delta_{t_v}(v) \geq \kappa e \ln n] \;<\; \left(\frac{e^{\kappa e - 1}}{(\kappa e)^{(\kappa e)}}\right)^{\ln n} \;=\; \frac{1}{n^{1+\ln\kappa\cdot\kappa e}}.$$

We then have $\Pr[\Delta_T \geq \kappa e \ln n] \leq n \cdot \Pr[\delta_{t_v}(v) \geq \kappa e \ln n] < 1/(n^{\ln\kappa\cdot\kappa e})$. Choosing κ such that $c = \ln\kappa \cdot \kappa e$ completes the proof. □

If $\log n \ll \Delta$, Theorem 6.1 together with the lower bound of Theorem 4.8 shows that randomization can help in distributed coloring.

In Line 5, we apply a deterministic one-round algorithm which colors an m-colored graph with maximum degree Δ with $\mathrm{O}(\Delta^2 \log m)$ colors. If we replace Line 5 of Algorithm 1 by the $\mathrm{O}(\log^* m)$-time, $\mathrm{O}(\Delta^2)$-coloring algorithm, we can color any m-colored graph G with maximum degree Δ with $\mathrm{O}(\Delta \log n)$ colors in $\mathrm{O}(\log^* m)$ rounds with probability $1 - 1/n^c$. Combined with Theorem 5.1, we obtain the following result.

Corollary 6.2. *For an arbitrary constant c, it is possible to color any graph with* $\Delta + 1$ *colors in* $\mathrm{O}(\Delta \log\log n)$ *rounds with probability* $1 - 1/n^c$.

7. CONCLUSIONS

We have proved new lower bounds and slightly improved upper bounds for one-round coloring algorithms and coloring algorithms based on iterative one-round color reduction steps. Beyond being the simplest non-trivial case for general graphs, there are several reasons why it is interesting to look at these types of coloring algorithms. In the context of emerging dynamic and mobile distributed systems such as peer-to-peer, ad hoc, or sensor networks, it is often desirable to keep the time complexity as small as possible even at the cost of globally not optimal global solutions. For such networks, constant-time algorithms are especially interesting. Further, most known distributed coloring algorithms are iterative applications of one-round color reduction schemes [5, 11, 17]. From a practical point of view, this technique is especially interesting because it results in communication-efficient algorithms. In every round, each node just sends its current own color, a message of size $\mathrm{O}(\log m)$ to all its neighbors. Sending the same message to all neighbors is particularly interesting for wireless networks where typically, every message is automatically sent to all neighbors.

8. REFERENCES

[1] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7(4):567–583, 1986.

[2] B. Awerbuch, A. V. Goldberg, M. Luby, and S. A. Plotkin. Network decomposition and locality in distributed computation. In *Proc. of the 30th Symposium on Foundations of Computer Science (FOCS)*, pages 364–369, 1989.

[3] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability – towards tight results. *SIAM Journal on Computing*, 27:804–915, 1998.

[4] V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3), 1979.

[5] R. Cole and U. Vishkin. Deterministic coin tossing with applications to optimal parallel list ranking. *Information and Control*, 70(1):32–53, 1986.

[6] G. De Marco and A. Pelc. Fast distributed graph coloring with O(Δ) colors. In *Proc. of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 630–635, 2001.

[7] M. Elkin. Distributed approximation: A survey. *ACM SIGACT News*, 35(4):40–57, 2004.

[8] P. Erdős, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel Journal of Mathematics*, 51:79–89, 1985.

[9] U. Feige and J. Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57:187–199, 1998.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[11] A. Goldberg, S. Plotkin, and G. Shannon. Parallel symmetry-breaking in sparse graphs. *SIAM Journal on Discrete Mathematics*, 1(4):434–446, 1988.

[12] T. Hermann and S. Tixeuil. A distributed TDMA slot assignment algorithm for wireless sensor networks. In *Proc. of 1st Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)*, volume 3121 of *Lecture Notes in Computer Science*, pages 45–58, 2004.

[13] L. Jia, R. Rajaraman, and R. Suel. An efficient distributed algorithm for constructing small dominating sets. *Distributed Computing*, 15(4):193–205, 2002.

[14] R. M. Karp. Reducibility among combinatorial problems. In *Proc. of a Symposium on the Complexity of Computer Computations*, pages 85–103, 1972.

[15] F. Kuhn. *The Price of Locality: Exploring the Complexity of Distributed Coordination Primitives.* PhD thesis, ETH Zurich, August 2005.

[16] F. Kuhn, T. Moscibroda, and R. Wattenhofer. The price of being near-sighted. In *Proc. of the 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.

[17] N. Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, February 1992.

[18] N. Linial and M. Saks. Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993.

[19] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15:1036–1053, 1986.

[20] M. Naor and L. Stockmeyer. What can be computed locally? In *Proc. of the 25th Annual ACM Symposium on Theory of Computing (STOC)*, pages 184–193, 1993.

[21] A. Panconesi and A. Srinivasan. On the complexity of distributed network decomposition. *Journal of Algorithms*, 20(2):581–592, 1995.

[22] A. Pelc. Personal communication.

[23] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.

[24] S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5:81–94, 1999.