Weak Graph Colorings: Distributed Algorithms and Applications

Fabian Kuhn
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
fkuhn@csail.mit.edu

ABSTRACT

We study deterministic, distributed algorithms for two weak variants of the standard graph coloring problem. We consider defective colorings, i.e., colorings where nodes of a color class may induce a graph of maximum degree d for some parameter $d > 0$. We also look at colorings where a minimum number of multi-chromatic edges is required. For an integer $k > 0$, we call a coloring k-partially proper if every node v has at least $\min\{k, \deg(v)\}$ neighbors with a different color We show that for all $d \in \{1, \ldots, \Delta\}$, it is possible to compute a $\mathcal{O}(\Delta^2/d^2)$-coloring with defect d in time $\mathcal{O}(\log^* n)$ where Δ is the largest degree of the network graph. Similarly, for all $k \in \{1, \ldots, \Delta\}$, a k-partially proper $\mathcal{O}(k^2)$-coloring can be computed in $\mathcal{O}(\log^* n)$ rounds.

As an application of our weak defective coloring algorithm, we obtain a faster deterministic algorithm for the standard vertex coloring problem on graphs with moderate degrees. We show that in time $\mathcal{O}(\Delta + \log^* n)$, a $(\Delta + 1)$-coloring can be computed, a task for which the best previous algorithm required time $\mathcal{O}(\Delta \log \Delta + \log^* n)$. The same result holds for the problem of computing a maximal independent set.

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

distributed algorithms, graph coloring, locality, deterministic symmetry breaking

SPAA'09, August 11–13, 2009, Calgary, Alberta, Canada.

1. INTRODUCTION

When devising distributed algorithms to compute network-wide structures in large decentralized systems, one of the core problems in most cases is to break symmetries. Even though various nodes in the network may run the same algorithm and despite possible symmetries in the topology of the network, different nodes generally need to terminate an algorithm with different outcomes, such as e.g. different colors in the case of a coloring algorithm. While using randomization is often a convenient and efficient way to break symmetries, deterministic symmetry breaking is typically harder. In the latter case, node identifiers or some other a priori labeling of the network have to be used to achieve the goal.

Properly coloring the nodes of a graph with a small number of colors exemplifies the challenges arising in the context of symmetry breaking. As a consequence of this and because of the general graph-theoretic interest in vertex colorings, there is a large history of work on distributed solutions for the coloring problem (cf. Section 2). Apart from being interesting from a graph-theoretic and distributed complexity point of view, coloring the network graph also has direct practical applications. In particular, colorings can be used to coordinate access to the wireless medium in ad hoc or sensor networks. Nodes in such networks communicate with each other via wireless radio and therefore have to use a medium access scheme to avoid interference. Colorings can for example be used to establish time schedules in a time division multiple access (TDMA) scheme or to obtain frequency (FDMA) or code (CDMA) assignments.

In the present paper, we consider distributed algorithms for two relaxations of the classic vertex coloring problem which are interesting as a step towards understanding the complexity of distributed coloring in general, and allow to parameterize the extent to which symmetries have to be broken. A t-coloring of a graph $G = (V, E)$ is a partition of the nodes of G into t color classes $V_1 \dot{\cup} \ldots \dot{\cup} V_t = V$. A t-coloring is called proper if no two adjacent nodes are in the same color class. Specifically, we study the following two weak coloring variants.

Definition 1.1 (Defective Coloring). *The defect d of a t-coloring is the maximum degree of any graph induced by one of the t color classes. A coloring with defect d is said to be d-defective.*

Definition 1.2 (Partially Proper Coloring). *We call a t-coloring of a graph $G = (V, E)$ k-partially proper if every node $v \in V$ has at least* $\min\{\deg(v), k\}$ *neighbors with different colors where* $\deg(v)$ *denotes the degree of v.*

Defective colorings have been introduced in [1, 7, 13] and e.g. studied in [8, 11, 25]. As a consequence of a result from the 1960s, it is known that for every k, every graph G with maximum degree Δ has k-coloring with defect at most $\lfloor \Delta/k \rfloor$ [18].

We develop local distributed algorithms with constant or log-star running times for defective and partially proper colorings. In both cases, we establish natural trade-offs between the number of colors of the final coloring and the extent to which the standard coloring problem has to be relaxed (i.e., the remaining symmetry breaking requirement). In particular, we show that in $\log^* n$ time[1], an $\mathcal{O}(\Delta^2/d^2)$-coloring with defect at most d can be computed. Here, as well as throughout the remainder of the paper, n denotes the number of nodes and Δ denotes the largest degree of the network graph. Analogously, for $k \geq 1$, a k-partially proper $\mathcal{O}(k^2)$-coloring can be computed in time $\mathcal{O}(\log^* n)$. The best known algorithm for the classical vertex coloring problem with the same time complexity requires $\mathcal{O}(\Delta^2)$ colors [17]. In addition, we show that in a single communication round (if every node just learns the identifiers of all its neighbors), a d-defective $\mathcal{O}((\Delta^2/d^2) \cdot \log n)$-coloring and a k-partially proper $\mathcal{O}(k^2 \cdot \log n)$-coloring can be computed.

Based on the algorithm for d-defective $\mathcal{O}(\Delta^2/d^2)$-colorings, it is possible to obtain a simple recursive deterministic algorithm for the standard vertex coloring algorithm that computes a $\lambda \cdot (\Delta+1)$-coloring in time $\mathcal{O}(\Delta/\lambda + \log^* n)$ for every $\lambda \geq 1$. This improves the best previous $(\Delta+1)$-coloring algorithm for moderate degrees[2] which requires $\mathcal{O}(\Delta \log \Delta + \log^* n)$ time [16].

Recently, the author of this paper has learned that independently, Barenboim and Elkin have developed a deterministic distributed algorithm that computes a $\lambda \cdot (\Delta+1)$-coloring in time $\mathcal{O}(\Delta/\lambda) + \log^*(n)/2$ for $1 \leq \lambda \leq \Delta^{1-\varepsilon}$ and any constant $\varepsilon > 0$ [5]. Similarly to the algorithm described in the present paper, the algorithm in [5] is based on the distributed computation of defective colorings. For parameters $1 \leq p \leq \Delta$, $p^2 < q$, and $q < c\Delta$ for a positive constant c, the defective coloring algorithm of Barenboim and Elkin computes an $\mathcal{O}\big(\frac{\log \Delta}{\log(q/p^2)} \cdot \Delta/p\big)$-defective p^2-coloring in time $\mathcal{O}\big(\frac{\log \Delta}{\log(q/p^2)} \cdot q\big) + \log^*(n)/2$. Rephrased in terms of a similar parameter p, the defective coloring algorithm of this paper computes an $\mathcal{O}(\Delta/p)$-defective p^2-coloring in time $\mathcal{O}(\log^* n)$ and is thus strictly stronger for all values of p.[3] Having the stronger defective coloring algorithm of the present paper allows us to use a significantly simpler algorithm to compute a proper $\lambda \cdot (\Delta+1)$-coloring of the network graph for all $\lambda \geq 1$.

[1]See Section 3 for a definition of the log-star function.

[2]There is a deterministic distributed $(\Delta+1)$-coloring algorithm with time complexity $2^{\mathcal{O}(\sqrt{\log n})}$ [21].

[3]As described, the algorithm of the present paper is only strictly stronger as long as the $\mathcal{O}(\frac{\log \Delta}{\log(q/p^2)} \cdot q)$ term in the time complexity of [5] dominates the $\log^*(n)/2$ term. There is a straight-forward way to strengthen all algorithms in this paper in the same way. Note that none of the algorithm needs unique identifiers, the bounds all hold if some initial proper n-coloring is given. As it is known that an $\mathcal{O}(\Delta^2)$-coloring can be computed in time $\log^*(n)/2 + \mathcal{O}(1)$ [24], we can compute such a coloring before executing each of the algorithms. All $\log^* n$ terms in the time complexities of this paper then become $\log^* \Delta$ and we have to add a $\log^*(n)/2$ term (outside of the $\mathcal{O}(\cdot)$) to each time bound.

The remainder of this paper is organized as follows. The next section discusses existing work on distributed algorithms for coloring and related problems. In Section 3, we describe the communication and computation model that we use and introduce some necessary definitions. The basic algorithms for defective and partially proper colorings are developed and analyzed in Section 4, our new distributed vertex coloring algorithm is described in Section 5.

2. PREVIOUS WORK

Not surprisingly, there is a large body of previous work on distributed coloring algorithms (e.g. [3, 4, 6, 12, 14, 15, 16, 17, 21, 23, 24]). The work on distributed coloring was started with a seminal paper by Linial [17] where among other results, it is shown that coloring a ring with a constant number of colors requires $\log^*(n)/2$ rounds. If one is willing to use a rather large number of colors, this lower bound is matched by an upper bound that computes an $\mathcal{O}(\Delta^2)$-coloring in time $\log^*(n)/2 + \mathcal{O}(1)$ on general graphs [17, 24]. In case, one needs a coloring with significantly less colors, the best algorithms for general graphs are randomized.[4] In [14], an algorithm that allows to compute an $\mathcal{O}(\Delta)$ coloring in $\mathcal{O}(\sqrt{\log n})$ time w.h.p. is given. The best randomized algorithm to compute a $(\Delta+1)$-coloring (generally the ultimate goal in distributed coloring) needs $\mathcal{O}(\log n)$ time and is based on an algorithm to compute a maximal independent set (MIS) by Luby [19] and a simple reduction described by Linial in [17].

Computing a coloring deterministically turns out to be significantly harder. For networks with large degrees, the best algorithm is based on techniques to decompose the network into clusters of small diameter and requires time $2^{\mathcal{O}(\sqrt{\log n})}$ to compute a $(\Delta+1)$-coloring [3, 21]. For networks with a small or moderate maximal degree, the best known algorithm previous to this paper and the work of Barenboim and Elkin [5] needed $\mathcal{O}(\Delta \log \Delta + \log^* n)$ time to compute a $(\Delta+1)$-coloring [16].

Distributed algorithms for relaxations of the standard vertex coloring problem have hardly been studied. Most relevant to our work is a paper by Naor and Stockmeyer [20]. As a problem with minimal symmetry breaking requirements, in [20] colorings where every node needs to have at least one neighbor with a different color (i.e., 1-partially proper colorings) are studied. Together with [17], the paper effectively started the research on local distributed algorithm.

3. MODEL AND PRELIMINARIES

We model the network as an undirected graph $G = (V, E)$. For convenience, we denote the number of nodes by $n = |V|$ and the maximum degree of G by Δ. For simplicity, we assume that all nodes know n and Δ. Wherever n appears in our time bounds, we assume that the nodes of G have a unique identifier of size at $\mathcal{O}(\log n)$.[5] We assume a standard synchronous message passing model, i.e., time is divided into

[4]There is a paper claiming a deterministic $\mathcal{O}(\log^* n)$-time $\mathcal{O}(\Delta)$-coloring algorithm [9]. However, the proof in [9] has a serious flaw. With a correct analysis, the bound on the number of colors becomes $\mathcal{O}(\Delta^2)$ and is thus not better than existing results [22].

[5]In fact, whenever n only appears in a $\log^* n$ term, it is even sufficient to assume that for the size N of the ID space, $\log^* N = \log^* n + \mathcal{O}(1)$.

Algorithm 1 Computing a coloring with defect at most d

Input: color $x \in [M]$, neighbor colors $y_1, \dots, y_\delta \in [M]$, parameter d
Output: a new color
1: search $\alpha \in \mathcal{A}$ such that $\big|\{i \in [\delta] : \varphi_x(\alpha) = \varphi_{y_i}(\alpha)\}\big| \le d$
2: color := $\big(\alpha, \varphi_x(\alpha)\big)$

Algorithm 2 Computing a k-partially proper coloring

Input: color $x \in [M]$, neighbor colors $y_1, \dots, y_\delta \in [M]$, parameter k
Output: a new color
1: search $\alpha \in \mathcal{A}$ such that $\big|\{i \in [\delta] : \varphi_x(\alpha) \neq \varphi_{y_i}(\alpha)\}\big| \ge \min\{k, \delta\}$
2: color := $\big(\alpha, \varphi_x(\alpha)\big)$

rounds, in every round, each node can perform some local computations, send a message to each neighbor, and receive messages from all neighbors. Note that it is well-known that it is possible to run a synchronous algorithm in an asynchronous system with the same asymptotic time complexity but at the cost of a few synchronization messages [2]. We do not assume that there is a bound on the allowed message size, however, all our algorithms only need to send messages of size $\mathcal{O}(\log n)$. In some cases, it is however necessary to send different messages to different neighbors. We assume that all nodes start a computation synchronously. The time complexity of an algorithm is the number of rounds from the start until the last node terminates.

Let us conclude this section with a few conventions and definitions. For an integer $k \ge 1$, we will frequently make use of the following common abbreviation: $[k] := \{1, \dots, k\}$. Further, unless otherwise stated, $\log(x)$ denotes the base 2 logarithm of x, whereas $\ln(x)$ is the natural logarithm of x. For an integer $i \ge 0$, the iterative log-functions $\log^{(i)}(x)$ and $\ln^{(i)}(x)$ are defined recursively as follows. We have $\log^{(0)}(x) = \ln^{(0)}(x) = x$ and for all $i \ge 1$:

$$\log^{(i)}(x) = \log\big(\log^{(i-1)}(x)\big),\ \ln^{(i)}(x) = \ln\big(\ln^{(i-1)}(x)\big).$$

Finally, for $x > 0$, the log-star function $\log^*(x)$ is defined as

$$\log^*(x) := \min\big\{i \ge 0 \,:\, \log^{(i)}(x) \le 2\big\}.$$

4. WEAK COLORING ALGORITHMS

We first describe an algorithm to reduce the number of colors in a single round. Assume that we start with an M-coloring of G (possibly with defect > 0). W.l.o.g., assume that the M colors of this coloring are $1, \dots, M$. Our coloring algorithm generalizes techniques described in [17]. It is based on a mapping from the color set $[M]$ to functions from a finite set $\mathcal{A}$ to a finite set $\mathcal{B}$. The new colors are chosen from the set $\mathcal{A} \times \mathcal{B}$. Let $v \in V$ be a node with degree $\delta \le \Delta$ and color $x \in [M]$ and assume that the δ neighbors of v have colors $y_1, \dots, y_\delta \in [M]$. Further, let φ_x be the function assigned to a color $x \in [M]$. The basic idea is to choose a value $\alpha \in \mathcal{A}$ such that the number of values y_i for which $\varphi_x(\alpha) = \varphi_{y_i}(\alpha)$ is sufficiently small. The details for computing d-defective and k-partially proper colorings are given by Algorithms 1 and 2, respectively. The following two lemmas are the basis of the analyzes of the two algorithms.

Lemma 4.1. *Assume that we are given an M-coloring of G with defect at most $d' \le d$. For a value $\kappa > 0$, let the functions φ_x for $x \in [M]$ be chosen such that for any two distinct colors $x, y \in [M]$, there are at most κ values $\alpha \in \mathcal{A}$ for which $\varphi_x(\alpha) = \varphi_y(\alpha)$ and such that $|\mathcal{A}| > \kappa\cdot(\Delta - d')/(d - d' + 1)$. Then, Algorithm 1 computes a $|\mathcal{A}| \cdot |\mathcal{B}|$-coloring with defect at most d.*

Proof. We prove the lemma in two steps. We first show that if all nodes choose a color $(\alpha, \beta) \in \mathcal{A} \times \mathcal{B}$ (i.e., if there is an $\alpha \in \mathcal{A}$ that satisfies the condition in line 1), the defect of the computed coloring is at most d. We then show that there are functions φ_x such that every node chooses a color.

Consider a node v with color $x \in [M]$ and degree $\delta \le \Delta$ that has neighbors with colors $y_1, \dots, y_\delta \in [M]$. Assume that v chooses a color $\big(\alpha, \varphi_x(\alpha)\big)$. Let u be a neighbor with a color $y \in [M]$ for which $\varphi_y(\alpha) \neq \varphi_x(\alpha)$. Assume that u chooses color (α', β'). If $\alpha = \alpha'$, we have $\varphi_y(\alpha) = \beta' \neq \beta = \varphi_x(\alpha)$. Therefore, either $\alpha \neq \alpha'$ or $\beta \neq \beta'$ and thus, u chooses a different color than v. Because there are at most d neighbors u' with a color $y' \in [M]$ for which $\varphi_{y'}(\alpha) = \varphi_x(\alpha)$, the defect of the computed coloring is therefore at most d.

It remains to prove that every node can choose a color. Because additional neighbors can certainly not increase the number of available colors, we can w.l.o.g. assume that the degree of v is $\delta = \Delta$. We definitely have $\varphi_x(\alpha) = \varphi_{y_i}(\alpha)$ for all α if $x = y_i$. Let $\ell \le d'$ be the number of values $i \in [\delta]$ for which this is the case. Let $S = \{i \in [\delta] : y_i \neq x\}$ be the indices of the neighbors with a different initial color. We need to show that there is an $\alpha \in \mathcal{A}$ such that $|\{i \in S : \varphi_x(\alpha) = \varphi_{y_i}(\alpha)\}| \le d - \ell$. For the sake of contradiction, assume that for every $\alpha \in \mathcal{A}$, there are at least $d - \ell + 1$ values $i \in S$ for which $\varphi_x(\alpha) = \varphi_{y_i}(\alpha)$. Because for every $i \in S$, $\varphi_x(\alpha) = \varphi_{y_i}(\alpha)$ for at most κ values $\alpha \in \mathcal{A}$, we then have

$$\begin{aligned}(\Delta - \ell)\cdot\kappa &\ge \sum_{i \in S}\big|\{\alpha \in \mathcal{A} : \varphi_x(\alpha) = \varphi_{y_i}(\alpha)\}\big| \\ &= \sum_{\alpha \in \mathcal{A}}\big|\{i \in S : \varphi_x(\alpha) = \varphi_{y_i}(\alpha)\}\big| \\ &\ge |\mathcal{A}|\cdot(d - \ell + 1). \end{aligned} \tag{1}$$

Because computing a Δ-defective coloring is trivial, we can certainly assume that $d < \Delta$. Inequality (1) then implies that

$$|\mathcal{A}| \le \frac{\Delta - \ell}{d + 1 - \ell} \le \frac{\Delta - d'}{d - d' + 1}.$$

This is a contradiction to the assumption $|\mathcal{A}| > \kappa \cdot (\Delta - d')/(d + 1 - d')$ and thus proves the lemma. ☐

Lemma 4.2. *Assume that we are given a k'-partially proper M-coloring of G for $k' \ge k$. For a value $\kappa > 0$, let the functions φ_x for $x \in [M]$ be chosen such that for any two distinct colors $x, y \in [M]$, there are at most κ values $\alpha \in \mathcal{A}$ for which $\varphi_x(\alpha) = \varphi_y(\alpha)$ and such that $|\mathcal{A}| > \kappa \cdot k$. Then, Algorithm 2 computes a k-partially proper $|\mathcal{A}| \cdot |\mathcal{B}|$-coloring.*

Proof. Similarly to the last lemma, we first prove that the computed coloring is k-partially proper and we then prove that all nodes can choose a color. The proof that Algorithm 2 computes a k-partially proper coloring is analogous to the proof that the defect of the coloring computed by Algorithm 1 is at most d. Let (α, β) be the color chosen by v. Neighbors with an initial color y for which $\varphi_x(\alpha) \neq \varphi_y(\alpha)$

choose a different color. By the condition according to which α is chosen in line 1 of Algorithm 2, there are thus at least k neighbors that choose a different color.

It remains to show that each node can choose a color. The set $S = \{i \in [\delta] : y_i \neq x\}$ has cardinality at least $\min\{k', \delta\}$. Let S' be a subset of S of size $|S'| = \min\{k, \delta\}$. We want to prove that there is an $\alpha \in \mathcal{A}$ such that $\varphi_x(\alpha) \neq \varphi_{y_i}(\alpha)$ for all $i \in S'$. Let C be the set of α's for which this is not the case, i.e., $C = \{\alpha \in \mathcal{A} : \exists i \in S' : \varphi_x(\alpha) = \varphi_{y_i}(\alpha)\}$. Because for every two distinct $x, y \in [M]$, $\varphi_x(\alpha) = \varphi_y(\alpha)$ for at most κ values $\alpha \in \mathcal{A}$, we have $|C| \leq \kappa \cdot |S'| \leq \kappa \cdot k$. The assumption $|\mathcal{A}| > \kappa \cdot k$ hence implies that C does not contain all $\alpha \in \mathcal{A}$ and that therefore there is an $\alpha \in \mathcal{A}$ such that $\varphi_x(\alpha) \neq \varphi_{y_i}(\alpha)$ for all $i \in S'$. This completes the proof. □

In order to apply Lemmas 4.1 and 4.2, we need a function $\varphi_x : \mathcal{A} \to \mathcal{B}$ for every $x \in [M]$ such that for any two distinct colors $x, y \in [M]$, there are at most κ values $\alpha \in \mathcal{A}$ for which $\varphi_x(\alpha) = \varphi_y(\alpha)$. To obtain efficient algorithms, we want the values $|\mathcal{B}|$ and κ to be as small as possible. The following lemma proves the existence of efficient functions.

LEMMA 4.3. *Let the set $\mathcal{A}$ be fixed, let $\mathcal{B}$ be a set of cardinality $|\mathcal{B}| \geq |\mathcal{A}|/(2\ln M)$, and let $\kappa = \lfloor 2e \ln M \rfloor$. For $x \in [M]$, there are functions φ_x such that for any two distinct $x, y \in [M]$ $\varphi_x(\alpha) = \varphi_y(\alpha)$ for at most κ values $\alpha \in \mathcal{A}$.*

PROOF. We prove the lemma by using the probabilistic method. We show that choosing the functions independently and uniformly at random from all functions from $\mathcal{A}$ to at set sufficiently large $\mathcal{B}$ leads to functions that satisfy the required conditions with positive probability. This then proves that there exist functions φ_x for $x \in [M]$ as claimed by the lemma.

Let us therefore assume that the functions φ_x for $x \in [M]$ are chosen independently and uniformly at random. Let p be the probability that for two distinct colors $x, y \in [M]$ and a value $\alpha \in \mathcal{A}$, we have $\varphi_x(\alpha) = \varphi_y(\alpha)$. Because $\varphi_x(\alpha)$ and $\varphi_y(\alpha)$ are independent, random elements of $\mathcal{B}$, we have $p = 1/|\mathcal{B}|$. Let Z be the number of values $\alpha \in \mathcal{A}$ for which $\varphi_x(\alpha) = \varphi_y(\alpha)$ for two distinct values $x, y \in [M]$. We have $\mathbb{E}[Z] = |\mathcal{A}| \cdot p = |\mathcal{A}|/|\mathcal{B}| \leq 2\ln M$. Applying a Chernoff bound gives

$$\mathbb{P}[Z > 2e \ln M] < \left(\frac{e^{e-1}}{e^e}\right)^{2\ln M} = \frac{1}{e^{2\ln M}} = \frac{1}{M^2}.$$

By a union bound, we therefore obtain that

$$\mathbb{P}\left[\max_{x \neq y \in [M]} \left|\{\alpha \in \mathcal{A} : \varphi_x(\alpha) = \varphi_y(\alpha)\}\right| > \kappa\right] = \mathbb{P}\left[\max_{x \neq y \in [M]} \left|\{\alpha \in \mathcal{A} : \varphi_x(\alpha) = \varphi_y(\alpha)\}\right| > 2e \ln M\right] < 1.$$

□

Combining Lemmas 4.1 and 4.2 with 4.3 allows to quantify the progress that can be achieved in a single communication round.

THEOREM 4.4. *Assume that we are given an M-coloring of G with defect at most $d' \leq d < \Delta$. There is a constant $C_D > 0$ such that a $C_D \cdot (\Delta - d')^2/(d+1-d')^2 \cdot \ln M$-coloring with defect at most d can be computed in a single communication round.*

PROOF. By Lemmas 4.1 and 4.3, we can choose $\kappa = \lfloor 2e \ln M \rfloor$, and $\mathcal{A}$ and $\mathcal{B}$ such that

$$|\mathcal{A}| = \left\lfloor \frac{(\Delta - d')2e \ln M}{d+1-d'} + 1 \right\rfloor > \frac{\kappa(\Delta - d')}{d+1-d'},$$
$$|\mathcal{B}| = \left\lceil \frac{|\mathcal{A}|}{2\ln M} \right\rceil.$$

The resulting number of colors then is

$$|\mathcal{A}| \cdot |\mathcal{B}| = \mathcal{O}\left(\frac{|\mathcal{A}|^2}{\ln M}\right) = \mathcal{O}\left(\left(\frac{\Delta - d'}{d+1-d'}\right)^2 \cdot \ln M\right).$$

□

THEOREM 4.5. *Assume that we are given a k-partially proper M-coloring of G. There is a constant $C_I > 0$ such that a k-partially proper $C_I \cdot k^2 \cdot \ln M$-coloring can be computed in a single communication round.*

PROOF. Analogously to above, by Lemmas 4.2 and 4.3, we can choose $\kappa = \lfloor 2e \ln M \rfloor$, and $\mathcal{A}$ and $\mathcal{B}$ such that

$$|\mathcal{A}| = \lfloor 1 + 2e \cdot k \cdot \ln M \rfloor > \kappa \cdot k \quad \text{and} \quad |\mathcal{B}| = \left\lceil \frac{|\mathcal{A}|}{2\ln M} \right\rceil.$$

We then obtain $|\mathcal{A}| \cdot |\mathcal{B}| = \mathcal{O}\left(|\mathcal{A}|^2/\ln M\right) = \mathcal{O}\left(k^2 \cdot \ln M\right)$ as the resulting number of colors. □

Unfortunately, Lemma 4.3 only proves the existence of functions φ_x for $x \in [M]$ with the given guarantees. The lemma does not give an explicit way to construct such functions. In the following, we show an explicit algebraic construction that achieves similar guarantees. The same construction has been described as an explicit way to construct families of sets such that no set is contained in the union of k other sets for some parameter k in [10]. Such set systems have been used to obtain distributed algorithms for the standard coloring problem in [17]. For a prime power q, let $\mathcal{P}(q, \kappa)$ be the set of all $q^{\kappa+1}$ polynomials of degree at most κ in the polynomial ring $\mathbb{F}_q[z]$, where $\mathbb{F}_q$ is the finite field or order q. It is well known that two polynomials of degree at most κ can be equal at at most κ positions. We can therefore choose the functions φ_x from $\mathcal{P}(q, \kappa)$. The details are given by the following two theorems.

THEOREM 4.6. *Assume that we are given an M-coloring of G with defect at most $d' \leq d < \Delta$. There are explicit functions φ_x for $x \in [M]$ and a constant $\overline{C_D} > 0$ such that Algorithm 1 computes a $\overline{C_D} \cdot (\Upsilon \log_\Upsilon M)^2$-coloring with defect at most d where $\Upsilon = (\Delta - d')/(d+1-d')$.*

PROOF. Choosing the functions φ_x from $\mathcal{P}(q, \kappa)$ gives $\mathcal{A} = \mathcal{B} = \mathbb{F}_q$. By Lemma 4.1, we therefore need $q > \kappa \cdot \Upsilon$. Because we need to assign different polynomials to every $x \in [M]$, we also need $|\mathcal{P}(q, \kappa)| = q^{\kappa+1} \geq M$. We choose

$$\kappa = \lceil \log_\Upsilon M \rceil \quad \text{and} \quad \lfloor \kappa\Upsilon + 1 \rfloor < q \leq 2\lfloor \kappa\Upsilon + 1 \rfloor.$$

Note that there must be a prime power q in the given interval. Choosing the parameters like this definitely guarantees that $q > \kappa\Upsilon$. We can certainly assume that $M \geq \Upsilon$ as otherwise, the theorem becomes trivial. This implies

$$|\mathcal{P}(q, \kappa)| = q^{\kappa+1} > (\Upsilon \log_\Upsilon M)^{\log_\Upsilon M} = e^{\ln(M)/\ln(\Upsilon) \cdot (\ln \Upsilon + \ln(\log_\Upsilon M))} \geq e^{\ln M} = M$$

and thus concludes the proof. □

THEOREM 4.7. *Assume that we are given a k-partially proper M-coloring of G. There are explicit functions φ_x for $x \in [M]$ and a constant $\overline{C_I} > 0$ such that Algorithm 2 computes a k-partially proper $\overline{C_D} \cdot (k \log_k M)^2$-coloring.*

PROOF. The proof is analogous to the proof of Theorem 4.6 where all occurrences of Υ are replaced by k. □

Algorithms 1 and 2 reduce the number of colors in a single round. One can obtain better defective and partially proper colorings by iterative applications of the two algorithms. Because the analysis is significantly simpler, we start with an algorithm to compute partially proper colorings.

THEOREM 4.8. *Assume that we are given a k-partially proper M-coloring of G. By iteratively applying Algorithm 2, a k-partially proper $\mathcal{O}(k^2)$-coloring can be computed in $\mathcal{O}(\log^* M)$ rounds.*

PROOF. The theorem is a direct consequence of Theorem 4.7. □

THEOREM 4.9. *Assume that we are given an M-coloring of G with defect at most $d' \leq d < \Delta$. Iteratively applying Algorithm 1, an $\mathcal{O}((\Delta - d')^2/(d+1-d')^2)$-coloring with defect at most d can be computed in $\mathcal{O}(\log^* M)$ rounds.*

PROOF. The case is more involved than the iterative application of Algorithm 2 because we cannot choose the same value for d throughout the algorithm. If we always use the same value for d in each iterative application of Algorithm 1, we can compute an $\mathcal{O}((\Delta - d')^2)$-coloring with defect at most d (always choosing the same value d only gives an $\mathcal{O}((\Delta - d')^2)$-coloring even for large values of $d - d'$). Note that an $\mathcal{O}((\Delta-d')^2)$-coloring is good enough if $d-d' = \mathcal{O}(1)$. We can therefore w.l.o.g. assume that $d - d'$ is sufficiently large.

We iteratively apply the algorithm T times for an integer $T \geq 1$ that will be determined below. W.l.o.g., we can assume that $d = d' + 2^h$ for some integer $h \geq 0$. For $i \geq 1$, let d_i be the value for i that is used in the i^{th} iterative application. We choose $d_i = d' + \lfloor (d-d')/2^{T-i} \rfloor$ and use Algorithm 1 with polynomial functions as analyzed in Theorem 4.6. For convenience, we also define $d_0 := d'$. Further, let M_i be the number of colors after the i^{th} iterative application of Algorithm 1. We choose T to be the smallest positive integer such that

$$\ln^{(T-1)} M \;<\; 16 \cdot \sqrt{\overline{C_D}} \cdot \frac{\Delta - d'}{d - d'}.$$

For $i \in \{2, \dots, T-1\}$, this implies that

$$\begin{aligned} 4\sqrt{\overline{C_D}} \cdot \frac{\Delta - d'}{d_i + 1 - d_{i-1}} &= 4\sqrt{\overline{C_D}} \cdot \frac{\Delta - d'}{1 + \lfloor \frac{d-d'}{2^{T-i}} \rfloor - \lfloor \frac{d-d'}{2 \cdot 2^{T-i}} \rfloor} \\ &\leq 4\sqrt{\overline{C_D}} \cdot \frac{\Delta - d'}{1 + \frac{d-d'}{2^{T-i}} - 1 - \frac{d-d'}{2 \cdot 2^{T-i}}} \\ &= 16\sqrt{\overline{C_D}} \cdot 2^{T-1-i} \cdot \frac{\Delta - d'}{d - d'} \\ &\leq 2^{T-1-i} \cdot \ln^{(T-2)} M \\ &\leq \ln^{(i-1)} M. \end{aligned} \tag{2}$$

The first inequality on the second line follows from the choice of T. The last inequality because $2 \ln x < x$ for all $x > 0$ and from the definition of T. W.l.o.g., we can assume that $\overline{C_D} \geq e$. We use induction to show that for all $i \leq T-1$, we then have

$$M_i \;\leq\; 16 \cdot \overline{C_D} \cdot \left(\frac{\Delta - d'}{d_i + 1 - d_{i-1}} \cdot \ln^{(i)} M \right)^2 \tag{3}$$

For $i = 1$, Inequality (3) is true by Theorem 4.6. For $1 < i \leq T-1$, we get

$$\begin{aligned} M_i &\leq \overline{C_D} \cdot \left(\frac{\Delta - d_{i-1}}{d_i + 1 - d_{i-1}} \cdot \frac{\ln M_{i-1}}{\ln\left(\overline{C_D} \frac{\Delta - d_i}{d_i + 1 - d_{i-1}} \right)} \right)^2 \\ &\leq \overline{C_D} \cdot \left(\frac{\Delta - d'}{d_i + 1 - d_{i-1}} \cdot \ln M_{i-1} \right)^2 \\ &\leq \overline{C_D} \cdot \Bigg(\frac{\Delta - d'}{d_i + 1 - d_{i-1}} \cdot \\ &\qquad \ln\left[16 \cdot \overline{C_D} \cdot \left(\frac{\Delta - d'}{d_i + 1 - d_{i-1}} \cdot \ln^{(i-1)} M \right)^2 \right] \Bigg)^2 \\ &\leq 16 \cdot \overline{C_D} \cdot \left(\frac{\Delta - d'}{d_i + 1 - d_{i-1}} \cdot \ln^{(i)} M \right)^2. \end{aligned}$$

The first inequality follows because of the assumption that $\overline{C_D} \geq e$. The last inequality follows from Inequality (2). Applying Inequality (3) for $i = T-1$ yields

$$\begin{aligned} M_{T-1} &\leq 16 \cdot \overline{C_D} \cdot \left(\frac{\Delta - d'}{d_{T-1} + 1 - d_{T-2}} \cdot \ln^{(T-1)} M \right)^2 \\ &\leq 16 \cdot \overline{C_D} \cdot \left(4 \cdot \frac{\Delta - d'}{d - d'} \cdot 16\sqrt{\overline{C_D}} \cdot \frac{\Delta - d'}{d - d'} \right)^2 \\ &= \left(16 \cdot \sqrt{\overline{C_D}} \cdot \frac{\Delta - d'}{d - d'} \right)^4. \end{aligned}$$

The theorem now follows by applying Algorithm 1 (and thus Theorem 4.6) one more time. □

Note that Theorem 4.9 in particular implies that when starting with unique identifiers as initial coloring, for every $d \leq \Delta$, a d-defective $\mathcal{O}(\Delta^2/d^2)$-coloring can be computed in $\mathcal{O}(\log^* n)$ rounds.

4.1 Defective Edge Colorings

We conclude this section by presenting a simple but interesting one-round algorithm to compute defective edge colorings. Assume that there is an integer parameter $i \geq 1$. Every node numbers its adjacent edges with numbers from $\{1, \dots, \lceil \Delta/i \rceil\}$ such that no number is assigned to more than i edges. Note that because the degree of each node is at most Δ, this is always possible. Then, each node, notifies all neighbors of the number assigned to the respective edge. Each edge $e = \{u, v\}$ gets assigned two numbers u_e, v_e in this way. Let the set $\{u_e, v_e\}$ be the computed color of the edge. The following theorem states the properties of this simple algorithm.

THEOREM 4.10. *For all integers $i \geq 1$, the above algorithm computes an edge coloring with defect at most $4i - 2$ in one round. The number of colors of the coloring is at most $\binom{\lceil \Delta/i \rceil + 1}{2}$.*

PROOF. As every node only needs to send a single message to each neighbor, the algorithm can be executed in a

Algorithm 3 $(\Delta+1)$-coloring algorithm

Input: M-coloring of the graph
Output: Call to Color(Δ) returns $(\Delta+1)$-coloring

```
 1: procedure Color(max_deg):
 2:   if deg = 1 then
 3:     compute 2-coloring in 1 round.
 4:   else
 5:     d := ⌊deg/2⌋
 6:     compute d-defective C-coloring in time O(log* M)
 7:     for all colors c ∈ [C] in parallel do
 8:       Call Color(d) on sub-graph induced by color c
 9:     Combine colors to get C · (d + 1)-coloring
10:     Reduce to (deg + 1)-coloring in time O(deg · log C)
11:   return computed coloring
```

single round. The number of possible colors is

$$\binom{\lceil \Delta/i \rceil}{2} + \lceil \Delta/i \rceil = \binom{\lceil \Delta/i \rceil + 1}{2}.$$

It therefore remains to prove that the defect of the coloring is at most $4i-2$. Consider an edge $e = \{u, v\}$ and let u_e and v_e be the numbers assigned to edge e by nodes u and v, respectively. A different edge e' adjacent to node u can only obtain the same color if u assigns one of the colors u_e or v_e to the edge e'. There are at most $i+(i-1)$ edges (in addition to edge e) to which u assigns one of the two colors. Hence, u and v can both have at most $2i-1$ additional adjacent edges with the same assigned color. This concludes the proof. □

5. IMPROVED COLORING ALGORITHM

We will now discuss how to use our defective coloring algorithm (Theorem 4.9) to obtain an algorithm for the standard graph coloring problem. In addition to Theorem 4.9, we require the following two results.

LEMMA 5.1 ([17, 24]). *An $\mathcal{O}(\Delta^2)$-coloring can be computed in $\log^*(n)/2 + \mathcal{O}(1)$ rounds.*

LEMMA 5.2 ([16]). *Let A and B be integers such that $B > A \geq \Delta + 1$. When starting with a B-coloring, an A-coloring can be computed in $\mathcal{O}(\Delta \cdot \log(B/A))$ rounds.*

The details of our $(\Delta+1)$-coloring algorithm are given by Algorithm 3. The core of the algorithm is the procedure Color(deg) which is used on a sub-graph with maximum degree at most deg and computes a (max_deg + 1)-coloring of the sub-graph. The procedure first partitions the graph into sub-graphs of maximum degree at most $d \leq \mathsf{deg}/2$ by computing a d-defective coloring C-coloring and by using the C colors to partition the graph. The procedure Color$(\cdot)$ is then called recursively for each of the C sub-graphs. Note that these C recursive calls can be done in parallel. Every node then has a color between 1 and C from the defective coloring and a color between 1 and $d+1$ from the recursive call to Color$(\cdot)$. Combining the two colors gives a proper $C \cdot (d+1)$-coloring of the graph on which Color(deg) was called. Using Lemma 5.2, a $(\mathsf{deg}+1)$-coloring can then be computed in $\mathcal{O}(\mathsf{deg} \cdot \log C)$ rounds. The following lemma bounds the time needed to execute procedure Color(deg).

LEMMA 5.3. *Let $T(\mathsf{deg})$ be the number of rounds needed to execute procedure* Color(deg)*. For all $\mathsf{deg} \geq 0$, we have*

$$T(\mathsf{deg}) = \mathcal{O}(\mathsf{deg} + \log(\mathsf{deg}) \cdot \log^* M).$$

PROOF. For $\mathsf{deg} = 1$, a 2-coloring can be computed in one round, as the sub-graph consists of at most 2 nodes. We therefore have $T(1) = 1$. For $\mathsf{deg} \geq 2$, we first show that $T(\mathsf{deg})$ can be computed recursively as follows. There is a positive constant α such that for all $\mathsf{deg} > 0$,

$$T(\mathsf{deg}) \leq T(\lfloor \mathsf{deg}/2 \rfloor) + \alpha \cdot \mathsf{deg} + \log^* M. \quad (4)$$

By Theorem 4.9, the number of colors C computed in line 6 is $\mathcal{O}(\mathsf{deg}/d) = O(1)$. As a consequence, the number of rounds needed to reduce the colors in line 10 is $\mathcal{O}(\mathsf{deg})$. Inequality (4) now directly follows because all the recursive calls in lines 8 can be executed in parallel. $T(\mathsf{deg}) \leq 2\alpha \cdot \mathsf{deg} + \log(\mathsf{deg}) \cdot \log^* M$ and therefore also the lemma now follows by induction on deg. □

As a consequence of Lemma 5.3, we can state the main theorem of this section.

THEOREM 5.4. *For every $\lambda \geq 1$, a proper $\lambda \cdot (\Delta+1)$-coloring can be computed in $\mathcal{O}(\Delta/\lambda + \log^* n)$ rounds.*

PROOF. We prove the theorem by constructing an algorithm that achieves this result. We start by applying Lemma 5.1 and compute an $\mathcal{O}(\Delta^2)$-coloring in $\log^*(n)/2 + \mathcal{O}(1)$ rounds. It now remains to show that based on an $\mathcal{O}(\Delta^2)$-coloring, a $\lambda \cdot (\Delta+1)$-coloring can be computed in time $\mathcal{O}(\Delta/\lambda)$. We first show how to obtain a $(\Delta+1)$-coloring in time $\mathcal{O}(\Delta)$. Because we already have an $\mathcal{O}(\Delta^2)$-coloring, we can apply procedure Color(Δ) to the whole graph with this initial coloring and can thus compute a $(\Delta+1)$-coloring in time $\mathcal{O}(\Delta + \log(\Delta) \cdot \log^* \Delta) = \mathcal{O}(\Delta)$ by Lemma 5.3.

Let us now look at the general case. Because we can compute a $(\Delta+1)$-coloring in time $\mathcal{O}(\Delta)$ and because we already start with an $\mathcal{O}(\Delta^2)$-coloring, we can w.l.o.g. assume that $\alpha \leq \lambda \leq \Delta/\alpha$ for a sufficiently large constant α. By Theorem 4.9 and the assumption that we already have an $\mathcal{O}(\Delta^2)$-coloring, there is a positive constant β such that for every integer $d \geq 1$, a d-defective $\lfloor \beta \cdot \Delta^2/(d+1)^2 \rfloor$-coloring can be computed in time $\mathcal{O}(\log^* \Delta)$. Let $\mathsf{deg} < \Delta$ be the smallest integer such that

$$\beta \cdot \frac{\Delta^2}{(\mathsf{deg}+1)^2} \cdot (\mathsf{deg}+1) = \beta \cdot \frac{\Delta^2}{\mathsf{deg}+1} \leq \lambda \cdot (\Delta+1). \quad (5)$$

We define $c = \lfloor \beta \cdot \Delta^2/(\mathsf{deg}+1)^2 \rfloor$. If the constant α is chosen sufficiently large, we have $\mathsf{deg} \geq 1$ and $\mathsf{deg} < \Delta$. Further, by Inequality (5), we have $\mathsf{deg} = \Theta(\Delta/\lambda)$. The algorithm first computes a deg-defective c-coloring in time $\mathcal{O}(\log^* \Delta)$. For each of the c color classes, in parallel a $(\mathsf{deg}+1)$-coloring is computed in time $\mathcal{O}(\mathsf{deg} + \log^* \Delta) = \mathcal{O}(\Delta/\lambda + \log^* \Delta)$. In combination with the c colors from the defective coloring, this allows to compute a $c \cdot (\mathsf{deg}+1) \leq \lambda \cdot (\Delta+1)$-coloring of the network graph. □

Remark: Besides coloring, computing a maximal independent set (MIS) also is a fundamental problem that is often used to model the algorithmic challenges arising in the context of symmetry breaking. As there is a simple and well-known reduction to convert a proper χ-coloring into an MIS in χ rounds, Theorem 5.4 also directly implies an $\mathcal{O}(\Delta + \log^* n)$ time deterministic algorithm for computing an MIS.

6. REFERENCES

[1] J. Andrews and M. Jacobson. On a generalization of chromatic number. *Congressus Numerantium*, 47:33–48, 1985.

[2] B. Awerbuch. Complexity of network synchronization. *Journal of the ACM*, 32(4):804–823, 1985.

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

[4] L. Barenboim and M. Elkin. Sublogarithmic distributed mis algorithm for sparse graphs using nash-williams decomposition. In *Proc. of 27th ACM Symposium on Principles of Distributed Computing (PODC)*, 2008.

[5] L. Barenboim and M. Elkin. Distributed $(\Delta + 1)$-coloring in linear (in Δ) time. In *Proc. of the 41st ACM Symposium on Theory of Computing (STOC)*, 2009.

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

[7] L. Cowen, R. Cowen, and D. Woodall. Defective colorings of graphs in surfaces: Partitions into subgraphs of bounded valence. *Journal of Graph Theory*, 10:187–195, 1986.

[8] L. Cowen, W. Goddard, and C. Jesurum. Defective coloring revisitied. *Journal of Graph Theory*, 24(3):205–219, 1997.

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

[10] 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.

[11] M. Frick. A survery of (m, k)-colorings. *Annals of Discrete Mathematics*, 55:45–58, 1993.

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

[13] F. Harary and K. Jones. Conditional colorability II: Bipartite variations. *Congressus Numerantium*, 50:205–218, 1985.

[14] K. Kothapalli, M. Onus, C. Scheideler, and C. Schindelhauer. Distributed coloring in $o(\sqrt{\log n})$ bit rounds. In *Proc. of 20th IEEE Int. Parallel and Distributed Processing Symposium (IPDPS)*, 2006.

[15] F. Kuhn. Local multicoloring algorithms: Computing a nearly-optimal TDMA schedule in constant time. In *Proc. of 26th Symp. on Theoretical Aspects of Computer Science (STACS)*, 2009.

[16] F. Kuhn and R. Wattenhofer. On the complexity of distributed graph coloring. In *Proc. of 25th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 7–15, 2006.

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

[18] L. Lovász. On decompositions of graphs. *Studia Sci. Math. Hungar.*, 1:237–238, 1966.

[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 25th 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] J. Schneider and R. Wattenhofer. A log-star distributed maximal independent set algorithm for growth-bounded graphs. In *Proc. of 27th ACM Symposium on Principles of Distributed Computing (PODC)*, 2008.

[24] M. Szegedy and S. Vishwanathan. Locality based graph coloring. In *Proc. of the 25th ACM Symposium on Theory of Computing (STOC)*, pages 201–207, 1993.

[25] D. Woodall. Improper colourings of graphs. In R. Nelson and R. Wilson, editors, *Graph Colourings*. Longman Scientific and Technical, 1990.