A New Technique For Distributed Symmetry Breaking

Johannes Schneider
Computer Engineering and Networks Laboratory
ETH Zurich
8092 Zurich, Switzerland
jschneid@tik.ee.ethz.ch

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

Abstract

We introduce Multi-Trials, a new technique for symmetry breaking for distributed algorithms and apply it to various problems in general graphs. For instance, we present three randomized algorithms for distributed (vertex or edge) coloring improving on previous algorithms and showing a time/color trade-off. To get a $\Delta + 1$ coloring takes time $O(\log \Delta + \sqrt{\log n})$. To obtain an $O(\Delta + \log^{1+1/\log^* n} n)$ coloring takes time $O(\log^* n)$. This is more than an exponential improvement in time for graphs of polylogarithmic degree. Our fastest algorithm works in constant time using $O(\Delta \log^{(c)} n + \log^{1+1/c} n)$ colors, where c denotes an arbitrary constant and $\log^{(c)} n$ denotes the c times (recursively) applied logarithm to n.

We also use the Multi-Trials technique to compute network decompositions and to compute maximal independent set (MIS), obtaining new results for several graph classes.

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*;
General Terms Algorithms, Theory
Keywords Local Algorithms, Parallel Algorithms, Coloring, Symmetry Breaking, Network Decomposition

1. INTRODUCTION

Symmetry breaking is a fundamental problem in distributed computing, and as such immensely studied. Its applications range from resource scheduling for parallel threads in a multi-core environment (i.e. MIS) to transmission scheduling in wireless networks (i.e. coloring) on to network decompositions.

When multiple unorganized units have to be coordinated, communication costs usually outweigh the costs for (local) computation. Thus, in the standard *message passing model* the complexity of an algorithm is measured in the number of communication rounds needed. Per round a node can exchange an arbitrary message with each of its neighbors.

So far, per message exchange all techniques for computing a maximal independent set (MIS) or a coloring have performed one attempt to obtain either a color or to join the MIS. Our technique transcends this approach and allows to perform multiple trials per communication round, hence Multi-Trials. Our constant time coloring algorithm shows how to get a color by approximating the result of multiple communication rounds through one communication round and local computation. Our MIS algorithm increases the number of possibilities a node can choose from to break symmetry from one (i.e. join the MIS or not) to many.

The application of our technique leads to a variety of interesting findings. For an overview of the results related to coloring see Figure 1. We also apply the Multi-Trials method to compute ruling sets. A set is (α, β)-ruling if every two nodes in the set have distance at least α and any node not in the set has a node in the set within distance β. Ruling sets are a natural way to obtain network decompositions, i.e. each node attaches itself to a closest node in the ruling set. Network decompositions in turn are fundamental to exploit the locality of a problem. That is to say, to efficiently distribute a task to several components of the network, which typically solve a subproblem and later combine the partial solutions. What kind of decomposition is best, depends on the task at hand. However, the diameter of a network component is usually an important parameter, since it determines the time for the exchange of information among all nodes in a component. In many cases it is natural to ask for components of equal diameter. We are the first to achieve network decompositions of similar diameter, i.e. of constant ratio of α and β, in sub-logarithmic time. We describe deterministic and randomized variants of an algorithm trading time for distance to a node in the ruling set. For example, we give a randomized algorithm computing a $(2k, 2k(c+1))$-ruling set in time $O(k \cdot 2^c \cdot \log^{1/c} n)$ for any integers $k, c > 0$. Our novel approach is based on partitioning the ID of a node. Our algorithm uses each partition as a separate trial.

By definition a MIS is the same as a (2,1)-ruling set. For graphs of sub-logarithmic degree our method is an improvement of prior randomized algorithms for the MIS problem. For graphs, where the size of a maximum IS within distance r of a node is bounded by $f(r)$, we are the first to achieve running time linear in $f(r)$.

PODC'10, July 25–28, 2010, Zurich, Switzerland.

Previous work		This paper	
Colors	Time	Colors	Time
$\Delta+1$	$O(\log n)$ [19, 1]	$\Delta+1$	$O(\log\Delta+\sqrt{\log n})$
	$O(\Delta+\log^* n)$ [14, 5]		
$O(\Delta)$	$O(\sqrt{\log n})$ [13]	$O(\Delta+\log n)$	$O(\log\log n)$
$O(\Delta+n^{d/\log\log n})$	$O(\log\log n)$ [11]	$O(\Delta+\log^{1+1/\log^* n} n)$	$O(\log^* n)$
$O(\Delta\log^2 n)$	$O(1)$ [17]	$O(\Delta\log^{(c)} n+\log^{1+1/c} n)$	$O(1)$
$O(\Delta^2)$	$O(\log^* n)$ [18, 22]		

Table 1: Comparison of coloring algorithms, where c is an arbitrary constant and d is a fixed constant

2. RELATED WORK

For symmetry breaking a number of techniques exist. For example, [21] extends a technique called the "deterministic coin tossing" by [7], which uses *ID*'s for symmetry breaking. A node iteratively computes a new serial number based on its own serial number (initially, its *ID*) and a serial number of a neighbor. Roughly speaking, the new number is the position where the two numbers differ. In [5, 4, 20] network decompositions are used, i.e. a graph is partitioned into subgraphs and then in each subgraph (one after the other or in parallel) the problem (i.e. coloring [4] and MIS [20] using ruling sets) is solved. For defective colorings [5, 14, 6] several nodes initially choose the same color. However, through multiple iterations the number is reduced until a proper coloring is achieved. Other algorithms [1, 12, 19] are also of iterative nature and let each node attempt to join the MIS with some probability in every round, or randomly choose a color until it is distinct from all neighbors. Sometimes several schemes are combined, e.g. [13, 6]. Our approach can be seen as an improvement of the iterative technique. All problems studied in this paper are simple in a centralized setting and all allow for straight-forward sequential greedy algorithms running in linear time. However, coming up with sub-linear algorithms is not easy, even for a seemingly simple problem such as an $O(\Delta)$ coloring in the weak message passing model, where a node can concurrently send and receive a (distinct) message to each of its neighbors. Deterministically, a $\Delta+1$ coloring is achievable in time $O(\Delta^2+\log^* n)$ [10] or in time $O(\Delta+\log^* n)$ [5, 14] or in time $O(n^{O(1)/\sqrt{\log n}})$ [20]. Using randomization, it is computable in time $O(\log n)$ [19, 1].[1] In [19] every node iteratively picks a random color of all still available colors and keeps it, if no neighbor has chosen the same color. For growth-bounded graphs (GBG)[15], where the size of a maximum independent set within distance r is bounded by $f(r)$ (e.g. Unit Disk Graphs), only time $\Theta(\log^* n)$ is needed to compute a $\Delta+1$ coloring [21]. For arbitrary graphs, an $O(\Delta)$ coloring can be computed in a randomized way in $O(\sqrt{\log n})$ [13]. Using $O(\Delta^{1+o(1)})$ colors [6] gives an algorithm running in time $O(f(\Delta)\log\Delta\log n)$ where $f(\Delta)=\omega(1)$ is an arbitrary slow growing function in Δ. Using $O(\Delta^2)$ colors [18] gives a deterministic algorithm running in time $O(\log^* n)$ and also a lower bound of time $\Omega(\log^* n)$ for three coloring of an n-cycle.

The upper bound iteratively computes an $O(\Delta^2\log^2 k)$ coloring [8], where k is the number of colors used in the current coloring. In [18] it is also proven that even for d-regular trees any algorithm running in time $O(\log_d n)$ uses at least $\Omega(\sqrt{d})$ colors.

Edge and vertex coloring are closely related, i.e. a vertex coloring algorithm on a line graph of G corresponds to an edge coloring of G. In the line graph of G all edges in G are nodes and two nodes are adjacent, if their corresponding edges share a common vertex in G. The randomized edge coloring in [11] is efficient for graphs of relatively large degree, i.e. $\Delta\in\Omega(n^{c/\log\log n})$, it runs in time $O(\log\log n)$ using $O(\Delta)$ colors. For graphs of smaller degree but still much larger than polylogarithmic, i.e. $\Delta\in\Omega(n^{1/\sqrt{\log n}})$ it requires time $O(\sqrt{\log n})$.

A MIS, i.e. a (2,1)-ruling set, can be computed in time $O(\log n)$ in general graphs using simple randomized algorithms [1, 12, 19]. A node marks itself with probability inversely proportional to its degree in every round. A marked node communicates with its neighbors to find out if all other nodes are either unmarked or have lower degree. If so, it joins the MIS. The fastest known deterministic distributed algorithm [20] is in $O(n^{\sqrt{c/\log n}})$ with constant c. In [14, 5] the running time for deterministic algorithms is improved for graphs with $\Delta\in o(2^{\sqrt{\log n}})$ to $O(\Delta)$. In general graphs every algorithm requires at least $\Omega(\sqrt{\log n/\log\log n})$ or $\Omega(\log\Delta/\log\log\Delta)$ communication rounds for computing a MIS [16]. For GBGs [9] gives an algorithm running in time $O(f(7)f(3)\log\log n\log^* n)$ and the algorithm in [21] runs in $O(f(f(2)+3)\log^* n)$ time. Our algorithm runs in time only linear in f, i.e. $O(f(2(c+1))\cdot 2^c\log^{1/c} n))$ for arbitrary constant $c>0$. Thus, if f also depends on n, i.e. $f(r,n)$, our algorithm improves [21, 9] for a large class of functions f.

A (α,β)-ruling set [3] induces a network decomposition, such that any component has diameter at least α and at most β. In [3] it is shown how to compute a $(k,k\log n)$-ruling set in time $O(k\log n)$. In [9] a $(1,\log\log\Delta)$-ruling set is computed in time $O(\log\log\Delta)$ such that each node in the ruling set has at most $O(\log^5 n)$ neighbors also in the ruling set. We target sparser sets, i.e. independent ruling sets. However, different quality measures exist, e.g. in [2] several types of network decompositions (and covers) are considered. Our strength lies primarily in getting network decompositions of balanced and small (sub-logarithmic) diameter efficiently. Our notion is in particular of interest if the main concern is communication time among all nodes within a cluster.

3. MODEL AND DEFINITIONS

The communication network is modeled with a graph $G=(V,E)$. For a node v its neighborhood $N^r(v)$ represents all nodes within r hops of v (not including v itself). By $N(v)$ we

[1] In [22] it is claimed that an $O(\Delta)$ coloring is computed in time $O(\log^*(n/\Delta))$. However, there is an error in the analysis pointed out by [17], increasing the number of colors to $O(\Delta^2)$ and time to $O(\log^* n)$.

Algorithm ColorTrials(Available colors $C(v)$ for node v)

1: $S(v) := \{(c,r)|\forall c \in C(v)$ choose a number $r \in [0,\Delta_{N(v)}]$ uniformly at random$)\}$
2: Transmit $S(v)$ to all uncolored neighbors $u \in N(v)$
3: **for each** $(c_v, r_v) \in S(v)$ **do**
4: **if** $r_v > \max\{r|(c_v,r) \in S(u), u \in N(v)\}$ **then** $color(v) := c_v$ **end if**
5: **end for each**

Algorithm ConstDeltaColoring, i.e. $(1+\epsilon)\Delta$ for $\epsilon > 1/2^{\log^* n}$

1: $color(v) := none$
2: $C(v) := \{0, 1, ..., (1+\epsilon)\Delta_{N(v)} + \log^{1+1/\log^* n} n\}$
3: **repeat**
4: *ColorTrials*$(C(v))$
5: $N(v) := \{u|u \in N(v) \wedge color(u) = none\}$
6: $C(v) := C(v) \setminus \{color(u)|u \in N(v)\}$
7: **until** $color(v) \neq none$

denote $N^1(v)$ and by $N_+(v)$ the neighborhood of v including v, i.e. $N(v) \cup v$. The degree $d(v)$ of a node v is defined as $|N(v)|$. $d_+(v)$ denotes $|N_+(v)|$, $\Delta := \max_{u \in V} d(u)$ and $\Delta_{N_+(v)} := \max_{u \in N_+(v)} d(u)$. In a (vertex) coloring any two neighboring nodes u, v have a different color. A set $T \subseteq V$ is said to be independent in G if no two nodes $u, v \in T$ are neighbors. A set $S \subseteq V$ is a maximal independent set (MIS), if it is independent and there exists no independent superset $T \supset S$. A MIS S of maximum cardinality, i.e. $|S| \geq \max_{\text{MIS } T} |T|$, is called a maximum independent set (MaxIS).

Our algorithm is non-uniform, i.e. every node knows an upper bound on the total number of nodes n. However, the maximum degree Δ in the graph is unknown. Communication among nodes is done in synchronous rounds without collisions, i.e. each node can exchange one distinct message of poly-logarithmic size with each neighbor. Nodes start executing the algorithm concurrently.

The term $\log^b n$ denotes $(\log n)^b$. The term $\log^{(b)} n$ equals $\underbrace{\log\log\ldots\log}_{b \text{ times}} n$. The function $\log^* n$ states how often one has to take the (iterated) logarithm to get 1, i.e. $\log^{(\log^* n)} n = 1$. The tetration ${}^b a$ expresses $\underbrace{a^{a^{\cdot^{\cdot^a}}}}_{b \text{ times}}$.

4. SYMMETRY BREAKING TECHNIQUE

We apply our MULTI-TRIALS technique to solve coloring and MIS problems as well as to compute ruling sets. We show several ways how to allow for many trials per communication round, even when only few seem possible.

4.1 Coloring

Instead of randomly choosing a single color and exchanging the color with its neighbors, a node gives a (random) preference for each color and transmits all its preferences at once, see Algorithm *ColorTrials*. A preference is a random number in $[0, \Delta_{N(v)}]$ for a node v, where $\Delta_{N(v)}$ is the size of the largest neighborhood of a neighbor $u \in N(v)$. This implies that the number of selectable preferences for a color is always at least as large as the number of preferences chosen for a color by the neighbors of node v. This ensures that the probability that node v has a neighbor with the same random preference for a color is small (i.e. constant). If node v's preference is unique and largest for a color, it can take the color. However, it only keeps one (arbitrary) color, it is allowed to take. A colored node informs its neighbors about its obtained color and stops Algorithm *ColorTrials*. Colored nodes and their incident edges are removed from the graph $G = (V, E)$ and, therefore, $N(v)$ denotes all uncolored neighbors upon calling *ColorTrials*.

Algorithms *ConstDeltaColoring* and *ConstTimeColoring*

Algorithm CoordinateTrials (maxColor d, parameter c)

1: $j := 0$; $coord(v)[i] :=$ the number given by bits $[i(\log d)/c, (i+1)(\log d)/c]$ of $color(v)$
2: **repeat**
3: $Rank(v) := [x_0, x_1, \ldots, x_{c-1}]$, s.t. $(x_i = 1 \Leftrightarrow coord(v)[i] = j) \wedge (x_i = 0 \Leftrightarrow coord(v)[i] \neq 0)$
4: **for** $k = 1..2^c$ **do**
5: **if** $Rank(v) = k$ **then** Transmit *Stop*
6: **elsif** received Stop **then** Exit **end if**
7: **end for**
8: $j := j + 1$
9: **until** $\nexists u \in N_+(v)$ executing algorithm
10: Join the ruling set

simply repeat *ColorTrials*. The only difference is that for constant time the number of initially available colors is larger, i.e. $\{0, 1, ..., \Delta_{N(v)} \log^{(c)} n + \log^{1+1/c} n\}$ for some constant c and some node v. For both algorithms the number of unused colors, i.e. colors not taken by a neighbor, is always larger than logarithmic. Thus any node performs more than a logarithmic number of trials to get a color in any communication round. For computing a $\Delta + 1$ coloring, Algorithm *DeltaPlus1Coloring* also iteratively calls Algorithm *ColorTrials* with an initial set of colors $C(v) := \{0, 1, ..., \Delta\}$ until only few unused colors left (more precisely, $|C(v)| < \sqrt{\log n}$). The remaining uncolored nodes can be colored by our MULTI-TRIALS technique as shown in Algorithm *RankingTrials* in Section 4.2 (or defective colorings[14, 5]).

For the sake of simplicity, we have described the algorithms using large messages. In the Analysis Section we show that messages of polylogarithmic size suffice.

4.2 Algorithm RankingTrials

In Algorithm *RankingTrials* we assume that each node has at most $\Delta \leq \sqrt{\log n}$ uncolored neighbors. The goal is to color all nodes with $\Delta + 1$ colors. To make effective use of the MULTI-TRIALS technique, we must ensure that a node can perform many trials in one communication round, i.e. even if Δ is much smaller than $\sqrt{\log n}$ a node should perform at least $\sqrt{\log n}$ trials per round. One idea is to create a ranking of the nodes using $2\sqrt{\log n} + 1$ ranks, i.e. each node can try to get any of the at least $\sqrt{\log n} + 1$ ranks that are not used by any of its neighbors. Then nodes pick a color depending on this ordering, i.e. once every node v has a rank $Rank(v)$, the uncolored node with smallest rank among its neighbors chooses a color until all nodes are colored. However, it is possible to perform even more trials per round, i.e. in order to obtain a rank, a node must ob-

Subroutine IncProb(threshold t)

1: Transmit p_v to all nodes $u \in N^2(v)$
2: **while** $\forall u \in N_+(v) | \sum_{w \in N_+(u)} p_w < 1/t$ **do**
3: $p_v := p_v \cdot t$
4: Transmit p_v to all nodes $u \in N^2(v)$
5: **end while**

Algorithm RandRulingSet(parameter c)

1: $p_v := 1/n$; $color(v) := none$
2: **for** $i = 1..c$ **do** IncProb($2^{(\log n)^{1-i/c}}$) **end for**
3: Participate in computing an $O(\log^{1+1/\log^* n} n)$ coloring with probability $\min(1, 64 p_v \log n)$
4: **if** $color(v) \neq none$ **then**
5: CoordinateTrials($O(\log^{1+1/\log^* n} n)$,$c+1$) **end if**

tain a priority $RP \in [0, 2\sqrt{\log n}]$ for that rank. The rank priorities RP determine which rank a node gets. Thus, a node could perform about $4 \log n$ trials per round. However, to keep messages small we perform fewer trials (but each having higher success probability). A node must get a rank priority for at least $\sqrt{\log n} + 1$ ranks and all rank priorities must be distinct. After obtaining the rank priorities, starting from smallest priority 0 on to priority $2\sqrt{\log n}$ a node keeps the first rank, where its priority is smallest (among its neighbors) and which is not taken by a neighbor.

4.3 Ruling Set And Maximal Independent Set

A simple but slow algorithm to compute a ruling set lets a node v with color (or ID) i join the ruling set in the ith round if none of its neighbors $u \in N(v)$ is already in the set. To speed up the process, we split up the the digits of a node's color into c equal parts (with the same number of digits), i.e. $coord(v)[0], ..., coord(v)[c-1]$, to perform multiple trials. A node v computes a summary of all trials of an attempt to join the ruling set, i.e. a rank $Rank(v)$ consisting of c bits, where bit i is 1 if and only if the ith coordinate equals the current attempt j, i.e. $coord(v)[i] = j$, and 0 otherwise. Based on the rank a node either continues the algorithm (and eventually joins the ruling set) or stops. After a computation of a rank all nodes with rank larger 0 compete to continue and force other nodes to stop the algorithm. More precisely, a node v continues and forces its neighbors with distinct rank to exit the algorithm, if in the kth round of the competition its $Rank(v)$ equals k.

Algorithm *CoordinateTrials* with arguments d and c requires a coloring for a subset $U \subseteq V$ of nodes using colors $\{0, 1, \ldots, d-1\}$. It determines a ruling set with at least one node within distance c for each colored node. A color $color(v) \in [0, d-1]$ of a node v is seen as a point in the c dimensional space $[0, d^{1/c} - 1]^c$, i.e. $color(v) = (coord(v)[0], ..., coord(v)[c-1])$. In the Analysis Section we describe modifications of the algorithm trading communication time for distance to a node in the ruling set.

To compute a $(2, c)$-ruling set deterministically, we start from an initial $O(\Delta^2)$[18] or $O(\Delta^{1+o(1)})$ [6] coloring before calling Algorithm *CoordinateTrials*. Our randomized Algorithm *RandRulingSet(c)* computes a subset of nodes $U \subseteq V$ which gets colored as follows: Each node v starts with a small value $p_v = 1/n$. This value is raised repeatedly by calling Subroutine *IncProb(t)* until the node v or a neighbor satisfies the condition that the sum of the values p_w of nodes $w \in N_+(v)$ exceeds one half. Then with probability $\min(1, 64 p_v \log n)$ a node joins set U, i.e. participates in computing an $O(\log^{1+1/\log^* n} n)$ coloring. Afterwards Algorithm *CoordinateTrials(*$O(\log^{1+1/\log^* n} n)$*,c)* is called by every colored node.

To compute a MIS for graphs where the maximum size of an independent set within distance r for every node v is bounded by $f(r)$ one simply iterates algorithm *RandRulingSet(c)* to compute a ruling set R_i in iteration i. After the ith iteration all nodes from the ruling set R_i join the MIS and all neighbors $u \in N_+(v), v \in R_i$ are removed from the graph. Due to Theorem 18, in $O(2^c \log^{1/c} n)$ time, for every node v at least one node $u \in N_+^{2(c+1)}(v)$ joins the MIS. Since, the size of any MIS within distance $2(c+1)$ is

Algorithm RankingTrials

1: $Rank(v) := none$, $RP(v)[i] := none, 0 \leq i \leq 2\sqrt{\log n}$ {$RP[i]$ equals priority for Rank i}
2: $RP_{free}(v)[i] := \{0, 1, ..., 2\sqrt{\log n}\}, 0 \leq i \leq 2\sqrt{\log n}$ {Available rank priorities for rank i}
3: **repeat**
4: $S(v) := \{(i, r) | RP(v)[i] = none$ and r is chosen randomly s.t.: {Pick a random priority for each available rank}
5: a) $r \in RP_{free}(v)[i]$ {Choose rank priority for rank i not taken by a neighbor}
6: b) $r \notin \{RP(v)[j] |$ for $i \neq j\}$ {Do not take a rank priority already taken by oneself}
7: c) $(i, r) \neq (j, r) \in S(v)$ for $i \neq j$ {Do not try to get the same rank priorities for different ranks}
8: Transmit $S(v)$ to all uncolored nodes $N(v)$
9: **for each** $(c_v, r_v) \in S(v)$ **do**
10: **if** $r_v \notin \{r | (c_v, r) \in S(u), u \in N(v)\}$ **then**
11: $RP(v)[c_v] := r_v$ {Keep priority if noone else wants it}
12: **end if**
13: **endfor**
14: Transmit $RP(v)$ to all nodes $N(v)$
15: $RP_{free}(v)[i] := RP_{free}(v)[i] \setminus \{RP(u)[i] | u \in N(v)\}$, for $0 \leq i \leq 2\sqrt{\log n}$
16: **until** $|\{i | RP(v)[i] \neq none\}| > \sqrt{\log n}$
17: Transmit *Ready* {Inform neighbors and wait for them}
18: **Wait until** recv *Ready* from all uncolored $u \in N(v)$
19: **for** $i = 0...2\sqrt{\log n}$ **do**
20: **if** $Rank(v) = none \wedge \exists l, RP(v)[l] = i$ **then**
21: $Rank(v) := l$; Transmit $Rank(v)$
22: **else**
23: $\forall$received $Rank(u)$ do $RP(v)[Rank(u)] := none$
24: **end if**
25: **end for**
26: **for** $i = 0..2\sqrt{\log n}$ **do**
27: **if** $Rank(v) = i$ **then**
28: $color(v) :=$ arbitrary $c \in C(v)$
29: Transmit $color(v)$ to all nodes $u \in N(v)$
30: **end if**
31: $C(v) := C(v) \setminus \{color(u) |$ received $color(u)\}$
32: **end for**

at most $f(2(c+1))$, the time complexity to compute a MIS is $O(f(2(c+1)) \cdot 2^c \log^{1/c} n)$.

5. ANALYSIS

5.1 Coloring

5.1.1 Notation and Prerequisites

Recall that $G = (V, E)$ is the graph given by all *uncolored* nodes before calling *ColorTrials*. $\tilde{N}(v) \subseteq N(v)$ denotes the still uncolored neighbors of node v during the foreach loop of *ColorTrials* and $\tilde{d}(v) := |\tilde{N}(v)|$. Let $E(v, c) = x$ denote the event for node v that x uncolored nodes $u \in \tilde{N}_+(v)$ take color $c \in C(v)$ during the execution of *ColorTrials*. The probability of event $E(v, c) = x$ is $Pr(E(v, c) = x)$. We also use the following Chernoff bound:

THEOREM 1. *The probability that the number X of occurred independent events $X_i \in \{0, 1\}$, i.e. $X := \sum X_i$, is less than $(1 - \delta)$ times the expectation $\mathbb{E}[X]$ can be bounded by $Pr(X < (1 - \delta)\mathbb{E}[X]) < e^{-\mathbb{E}[X]\delta^2/2}$.*

Observe that the two events $E(u, c)$ and $E(v, c)$ are dependent. For instance, consider four nodes u, v, w, x in a line, i.e. with edges (u, v), (v, w), (w, x), and assume that nodes u and x already obtained a color before calling *ColorTrials*. In this case, for event $E(u, c) > 0$ to occur node v's random number must be larger than w's and the other way round for event $E(x, c) > 0$ to happen. Thus, it is not possible that both events $E(u, c) > 0$ and $E(x, c) > 0$ occur, i.e. $Pr(E(u, c) > 0 | E(x, c) > 0) = 0$, since either v's random number is larger than w's or the other way round or both are equal. In general, the correlation might also be positive, e.g. if nodes form a clique.

To deal with the dependencies among nodes, we derive a bound for n (slightly) dependent events. Let Ω denote the union of all possible (elementary) events F, i.e. $\sum_{F \in \Omega} Pr(F) = 1$. For an event E_i to occur with $0 \leq i \leq n-1$, let certain elementary events $S_{E_i} \subseteq \Omega$ be infeasible for event E_i, i.e. $Pr(E_i | F \in S_{E_i}) = 0$, and let all others $\Omega \setminus S_{E_i}$ cause E_i to happen, i.e. the probability $Pr(E_i)$ of event E_i is $\sum_{F \in \Omega \setminus S_{E_i}} Pr(F) = \sum_{F \in \Omega} Pr(F) - \sum_{F \in S_{E_i}} Pr(F) = 1 - \sum_{F \in S_{E_i}} Pr(F)$.

THEOREM 2. *For n (dependent) events E_i with $i \in \{0, ..., n-1\}$, such that each event E_i occurs with probability $Pr(E_i) = 1 - \sum_{F \in S_{E_i}} Pr(F) \geq 1 - 1/n^k$, the probability that all events occur is at least $1 - 1/n^{k-3}$.*

PROOF. We want to show that $Pr(\wedge_{i \in \{0,...,n-1\}} E_i) \geq 1 - 1/n^{k-3}$. Since $Pr(\wedge_{i \in \{0,...,n-1\}} E_i) = Pr(E_0) \cdot Pr(E_1 | E_0) \cdot Pr(E_2 | E_0 \wedge E_1) \cdot ... \cdot Pr(E_{n-1} | \wedge_{i \in \{0,...,n-2\}} E_i)$ we can also derive lower bounds for the conditional probabilities $Pr(E_i | \wedge_{i \in T, T \subset \in \{0,...,n-2\}} E_i)$. We assume a worst case correlation among events E_i, i.e. the occurrence of an event E_i has the worst impact on the probability that another event E_j occurs. Given that the event E_i occurs, all elementary events S_{E_i} for which E_i cannot happen, are known not to occur. They can be excluded from Ω, when computing the probability of another event E_j, i.e. $Pr(E_j | E_i)$. Thus, the elementary events that can occur given E_i are $\Omega_{|i} := \Omega \setminus S_{E_i}$. Since $Pr(E_j | E_i)$ is a probability distribution, the sum of the probabilities $Pr(F | E_i)$ of all elementary events $F \in \Omega_{|i}$ must be one, i.e. $\sum_{F \in \Omega_{|i}} Pr(F | E_i) = 1$. We have that $Pr(F | E_i) = Pr(F) \cdot 1/(1 - \sum_{F \in S_{E_i}} Pr(F)) \leq Pr(F) \cdot 1/(1 - 1/n^k)$, since the occurrence of event E_i only removes the set S_{E_i} with $\sum_{F \in S_{E_i}} Pr(F) \leq 1/n^k$ from Ω, but does not make one elementary event $F_1 \in \Omega_{|i}$ (relatively) more favorable to another $F_2 \in \Omega_{|i}$, i.e. the probabilities $Pr(F | E_i)$ of all remaining events $F \in \Omega_{|i}$ are increased by the same factor $1/(1 - \sum_{F \in S_{E_i}} Pr(F)) \leq 1/(1 - 1/n^k)$. To compute $Pr(E_j | E_i)$ assuming a worst case correlation, all excluded elementary events S_{E_i} cause E_j to occur and $S_{E_i} \cap S_{E_j} = \{\}$, i.e. all elementary elements S_{E_j} are excluded for $Pr(E_j | E_i)$ to occur. Thus, all elementary events $\Omega_{|i} \setminus S_{E_j}$ cause E_j to occur and $Pr(E_j | E_i) = 1 - \sum_{F \in S_{E_j}} Pr(F | E_i) \geq 1 - \sum_{F \in S_{E_j}} Pr(F) \cdot 1/(1 - 1/n^k) \geq 1 - 1/n^k \cdot 1/(1 - 1/n^k) = 1 - 1/(n^k - 1)$. The argument can be generalized to bound any conditional probability $Pr(E_j | \wedge_{i \in T, T \subseteq \{1,...,n-1\}} E_i)$ for any $0 \leq j \leq n-1$. The remaining possible events for E_j to occur given all events E_i with $i \in T$ happen is $\Omega_{|T} := \Omega \setminus \cup_{i \in T} S_{E_i}$. For the probability of an elementary event $F \in S_{E_j}$ holds $Pr(F | \wedge_{i \in T} E_i) = Pr(F) \cdot 1/(1 - \sum_{F \in \cup_{i \in T} S_{E_i}} Pr(F)) \leq Pr(F) \cdot 1/(1 - |T|/n^k)$. The last inequality follows since all sets S_{E_i} are assumed to be disjunct, i.e. $S_{E_i} \cap S_{E_l} = \{\}$, to maximize the probability that an elementary event $F \in S_{E_j}$ occurs . Therefore, $Pr(E_j | \wedge_{i \in T} E_i) = 1 - \sum_{F \in S_{E_j}} Pr(F | \wedge_{i \in T} E_i) \geq 1 - \sum_{F \in S_{E_j}} Pr(F) \cdot 1/(1 - |T|/n^k) = 1 - 1/(1 - |T|/n^k) \sum_{F \in S_{E_j}} Pr(F) \geq 1 - 1/(1 - n/n^k) \cdot 1/n^k = 1 - 1/(n^k - n)$.

Using the bound of the conditional probabilities and $Pr(E_i) \geq 1 - 1/n^k \geq 1 - 1/(n^k - n)$ (the first inequality is by assumption), we obtain: $Pr\left(\wedge_{i \in \{0,...,n-1\}} E_i\right) = Pr\left(E_0\right) \cdot Pr\left(E_1 | E_0\right) \cdot Pr\left(E_2 | \wedge_{i \in \{0,1\}} E_i\right) \cdot Pr\left(E_3 | \wedge_{i \in \{0,1,2\}} E_i\right) \cdot ... \cdot Pr\left(E_{n-1} | \wedge_{i \in \{0,...,n-2\}} E_i\right) \geq \prod_{i \in \{0,...,n-1\}} \left(1 - 1/(n^k - n)\right) = \left(1 - 1/(n^k - n)\right)^n \geq 1 - 1/n^{k-3}$ □

Moreover, the probability $Pr(E(v, c) > 0)$ that a node v or some of its uncolored neighbors $\tilde{N}(v)$ get a certain color $c \in C(v)$ depends on the topology of the graph, i.e. on each neighborhood $N(u)$ of every uncolored node $u \in \tilde{N}(v)$. In contrast to the probability $Pr(E(v, c) > 0)$, the probability that an uncolored node u gets color $c \in C(u)$ does not depend on the current number of uncolored neighbors $\tilde{d}(u)$, but only depends on $d(u)$, i.e. the number of uncolored neighbors before executing *ColorTrials*. This is because u is unaware whether a neighbor $w \in N(u)$ got a color $c_1 < c$ while executing *ColorTrials*. Thus, node u must consider all random numbers from all nodes $w \in N(u)$ and must be larger than all of them, even those that already got a smaller color.

Since $\tilde{d}(v)$ might change after every considered color $c \in C(v)$, two events $E(v, c_1)$ and $E(v, c_2)$ are *not* independent for arbitrary colors c_1, c_2. For example, consider a star graph with v in the center, i.e. a tree with v as root and $n - 1$ leaves. Say c_1 is the first considered color and c_2 is the second. Assume that all initially uncolored neighbors $N(v)$ of v get the first color c_1. Then the probability of event $E(v, c_2) > 0$ is roughly $1/|N_+(v)| = 1/n$,[2] since node v's

[2] $Pr(E(v, c_2) > 0) = 1/n$ would hold if all nodes were known to draw distinct random numbers.

random number for color c_2 must be larger than the choice of all its neighbors $N(v)$ executing *ColorTrials*. If none of v's neighbors got the first color, the only situation where $E(v, c_2)$ equals 0, is when v drew a random number that is maximum among all nodes $u \in N(v)$ and at least one neighbor chose the same number. This happens with probability less than $1/|N_+(v)| = 1/n$. Thus, in this case, a lower bound of $Pr(E(v, c_2) > 0)$ is $1 - 1/|N_+(v)| = 1 - 1/n$. Therefore, the probability of $Pr(E(v, c_2) > 0)$ depends on the outcome of the first color $E(v, c_1)$.

To deal with the interdependence for different colors we follow the idea of *stochastic domination*. A probability distribution A, where $Pr_A(X = x)$ denotes the probability of outcome x, dominates a probability distribution B, if for any outcome x, A gives a higher probability of receiving an outcome equal to or better than x under B, i.e. $Pr_A(X \geq x) \geq Pr_B(X \geq x)$. More precisely, we use the following basic theorem.

THEOREM 3. *For t dependent events $E_i \in \{0, 1\}$ with $i \in [0, t-1]$, such that each event E_i occurs with probability $Pr(E_i = 1 | \sum_{j=0}^{i} E_j \leq y) \geq p$, the probability that at least $\min(y, tp/2)$ events occur is at least $1 - e^{-tp/8}$.*

PROOF. Consider the random variable $X_E = \sum_{j=0}^{t} E_j$. We want to compute $Pr(X_E \leq y)$. Consider any event F, i.e. a sequence of events $(E_0 = x_0, E_1 = x_1, ..., E_t = x_t) =: F$, with $X_E \leq y$. For any such event F holds that each event $E_i = 1$ occurs with probability at least p independently of all others, since by assumption $Pr(E_i = 1 | X_E \leq y) \geq p$ for all $i \in [0, t-1]$. Consider the random variable $X_I = \sum_{j=0}^{t} X_j$ which is the sum of t independent events X_i such that each event occurs with probability p. We have $Pr(X_E \geq x) \geq Pr(X_I \geq x)$ for all $x \leq y$, since $Pr(E_i = 1 | X_E \leq y) \geq p = Pr(X_i = 1)$ for all $i \in [0, t-1]$. Thus, to upper bound $Pr(X_E \leq \min(y, tp/2))$ we can use $Pr(X_I \leq \min(y, tp/2))$. Using Theorem 1 with $\delta = 1/2$ and $\mathbb{E}[X_I] = tp$ yields a bound $Pr(X_I \leq tp/2) < e^{-tp/8}$. Therefore, $Pr(X_E \leq \min(y, tp/2)) \leq Pr(X_I \leq \min(y, tp/2)) \leq Pr(X_I \leq tp/2) < e^{-tp/8}$. □

In our case the probability of a certain outcome is defined by Algorithm *ColorTrials* and the given graph G. To derive bounds for the probabilities of events $Pr(E(v, c) = x)$ of a node v we consider a special topology $G'_v = (V', E')$ which is dominated by G, meaning that it is more likely for node v to get a certain number of colored neighbors in G than in G'_v. For such a graph $G'_v = (V', E')$ all terms $E'(v, c)$, $\tilde{d}'(v)$, $N'(v)$ etc. are defined in the same manner as for G.

5.1.2 Algorithm ColorTrials

Given a node $v \in V$, let $G'_v = (V', E')$ be the graph obtained by enhancing G, i.e. we begin with $V' = V$ and $E' = E$ and add edges and nodes to G'_v. If $|V'| < \Delta + d(v) + 1$ then add an arbitrary set S of nodes to V', i.e. $V' = V \cup S$ such that $|V'| = \Delta + d(v) + 1$. Each node $u \in N(v)$ is connected to all other nodes $w \in N(v) \setminus u$, i.e. the neighbors $N(v)$ form a clique. Furthermore, each node $u \in N(v)$ is connected to Δ arbitrary nodes $V' \setminus N_+(v)$, i.e. node u has degree $\Delta + d(v)$.

Thus node v has the same neighbors in G and G'_v, i.e. $N(v) = N'(v)$. But the degree of a node $u \in N'(v)$ in G'_v is larger than that of any node $w \in N(v)$ in G and all neighbors $u \in N'(v)$ are connected among themselves. Note, that G'_v is defined for a single node $v \in V$ and can be seen as a worse topology than G for node v.

LEMMA 4. *For a node v in a graph $G'_v = (V', E')$ we have that $Pr(\sum_{c \in C(v)} E(v, c) > y) \geq Pr(\sum_{c \in C(v)} E'(v, c) > y)$ for any integer $y \geq 0$.*

PROOF. Since node v's neighbors form a clique in G'_v only one node $u \in N_+(v)$ can get a color $c \in C(v)$. Thus $E'(v, c) \in \{0, 1\}$, i.e. $Pr(E'(v, c) = i) = 0$ for $i > 1$.[3] Thus, $Pr(E(v, c) > i) \geq Pr(E'(v, c) > i) = 0$ holds $\forall 1 \leq i \leq n$. We show that it also holds for $i = 0$, i.e. $Pr(E(v, c) > 0) \geq Pr(E'(v, c) > 0)$ given $\tilde{d}(v) = \tilde{d}'(v)$. Since $Pr(E'(v, c) > 0) = 1 - Pr(E'(v, c) = 0)$ we have $Pr(E(v, c) > 0) \geq Pr(E'(v, c) > 0) \Leftrightarrow 1 - Pr(E(v, c) = 0) \geq 1 - Pr(E'(v, c) = 0) \Leftrightarrow Pr(E(v, c) = 0) \leq Pr(E'(v, c) = 0)$. To upper bound $Pr(E(v, c) = 0)$ in G, we use the observation that the larger the neighborhood $N(u)$ of a neighbor $u \in N(v)$ (with nodes $w \in N(u)$ being at distance 2 from v), the lower are the chances for u to get colored and, therefore, the higher are the chances for $Pr(E(v, c) = 0)$. This is because u's random number must be larger than that of all its neighbors $w \in N(u)$ for u to get colored and event $E(v, c) > 0$ to occur. Thus, we assume that each neighbor $u \in N(v)$ has Δ neighbors $w \in N(u)$ at distance 2 from v. The probability of $Pr(E(v, c) = 0)$ does not decrease, if, additionally, nodes $u \in N(v)$ are interconnected. Thus we assume that all nodes $u \in N(v)$ form a clique. The described topology giving an upper bound for $Pr(E(v, c) = 0)$ is the topology of v for G'_v used to compute $Pr(E'(v, c) = 0)$ and thus $Pr(E(v, c) = 0) \leq Pr(E'(v, c) = 0)$ and we have shown $Pr(E(v, c) > i) \geq Pr(E'(v, c) > i)$ holds $\forall 0 \leq i \leq n$ given $\tilde{d}(v) = \tilde{d}'(v)$.

Therefore, the number of uncolored nodes $\tilde{d}(v)$ in G decreases at least as fast as $\tilde{d}'(v)$ in G'_v with every considered color c for $\tilde{d}(v) = \tilde{d}'(v)$. When looking at the first color we have $d(v) = \tilde{d}(v) = \tilde{d}'(v)$. Thus $\tilde{d}(v)$ is expected to be less or equal $\tilde{d}'(v)$ after examining all colors $c \in C(v)$, i.e. for the number of newly colored nodes (after the last color $c \in C(v)$) holds $Pr\left(d(v) - \tilde{d}(v) > y\right) \geq Pr\left(d(v) - \tilde{d}'(v) > y\right)$. The argument is analogous to the following: Given two functions $f(x)$ and $g(x)$ with $f(0) = g(0)$ and $f'(x) \leq g'(x) \leq 0$ for $x \geq 0$. Then $f(y) \leq g(y)$ for any $y \geq 0$.

Since for the number of newly colored nodes holds: $d(v) - \tilde{d}(v) = \sum_{c \in C(v)} E(v, c)$, we have: $Pr\left(d(v) - \tilde{d}(v) > y\right) \geq Pr\left(d(v) - \tilde{d}'(v) > y\right) \Rightarrow Pr\left(\sum_{c \in C(v)} E(v, c) > y\right) \geq Pr\left(\sum_{c \in C(v)} E'(v, c) > y\right)$ □

Now, that we have shown that the probability distribution for the number of obtained colored neighbors given by the topology G'_v for node v (and Algorithm *ColorTrials*) is indeed dominated by the one for G (and Algorithm *ColorTrials*), we derive a bound on the probability $Pr(E'(v, c) > 0)$ for the topology G'_v for an arbitrary color $c \in C(v)$. Note, that $Pr(E'(v, c) > 0) = Pr(E'(v, c) = 1)$, since nodes $u \in N'(v)$ form a clique and thus only one node $u \in N'(v)$ can get a color c.

[3] In contrast, for an event $E(v, c)$ we have that it might also have a non-zero probability that more than one node gets colored, e.g. for a tree $Pr(E(v, c) = x) > 0$ for $x \in \{0, 1, ..., \tilde{d}(v)\}$.

LEMMA 5. *The probability of event $E'(v,c)=1$ is at least $\tilde{d}'_+(v)/(24\Delta)$.*

PROOF. If we choose $d'(u) = \Delta + d(v) \leq 2\Delta$ random numbers in $[0,\Delta_{N(u)}] = [0,d'(u)] \subseteq [0,2\Delta]$ then the probability that no random number is 0 is $(1-1/d'_+(u))^{d'(u)} \geq 1/e$. The probability that node v or a neighbor chose one random number equal to 0 is $\tilde{d}'_+(v)\cdot 1/d'_+(u)\cdot(1-1/d'_+(u))^{\tilde{d}'_+(v)-1} \geq \tilde{d}'_+(v)/d'_+(u)\cdot(1-1/d'_+(u))^{\Delta} \geq \tilde{d}'_+(v)/(2\Delta+1)\cdot(1-1/(2\Delta+1))^{\Delta} \geq \tilde{d}'_+(v)/(3e\Delta)$.

The probability of event E_1 that for color c exactly one node $u \in \tilde{N}_+(v)$ chooses a random number in $[0,d'(u)]$ is at least $\tilde{d}_+(v)/(3e\Delta)$. The probability of the event E_2 that none of u's neighbors $w \in N(u)$ chooses a random number in this interval is $1/e$. Events E_1 and E_2 are not independent since nodes u and v have (some) common neighbors, i.e. $N(v) \cap N(u) = N(v)$. The probability $Pr(E_2|E_1)$ of event E_2 given event E_1 is at least the probability $Pr(E_2)$ of event E_2, since due to event E_1 node u is the only node in $N_+(v)$ with a random number equal 0 and all other nodes $N_+(v)\setminus u$ are known not to have a random number equal to 0. Therefore, for event E_2 to occur only the random numbers of nodes $N(u)\setminus N(v)$ instead of those of all nodes $N(u)$ must be shown to be distinct from 0. Thus the chance that both events happen is $Pr(E_1 \wedge E_2) = Pr(E_2|E_1)\cdot Pr(E_1) \geq Pr(E_2)Pr(E_1) \geq 1/e\cdot\tilde{d}_+(v)\cdot 1/(3e\Delta) \geq \tilde{d}_+(v)/(24\Delta)$. □

As quantified in Lemma 5, the more uncolored neighbors $\tilde{N}(v)$ node v has, the larger is the probability of event $E'(v,c)=1$. As long as a node has a certain number of uncolored neighbors we can guarantee a reasonable minimum probability for an event $E'(v,c)=1$. This observation is used in the proof of the following theorem showing that the number of uncolored nodes reduces drastically when executing *ColorTrials* using more colors than the maximum size Δ of any uncolored neighborhood.

THEOREM 6. *The probability $Pr(E_v)$ of event E_v for node v that $\tilde{d}(v) \leq \max(d(v)/2^s, \log n)$ after executing* ColorTrials *is at least $1-1/n^k$ for arbitrary constant k using color set $C(v)$ with $|C(v)| \geq k_0 s\Delta$ and $k_0 \geq 384(k+1)$.*

PROOF. Consider a node v in a graph G'_v and a sequence of $k_0\Delta$ colors that nodes $u \in \tilde{N}_+(v)$ attempt to get during *ColorTrials*. Assume that before processing a sequence of $k_0\Delta$ the number of uncolored neighbors is $\tilde{d_0}'(v)$. In Lemma 5 the probability of $Pr(E'(v,c)=1)$ was shown to be at least $\tilde{d}'(v)/(24\Delta)$. In particular as long as there are at most $\tilde{d_0}'(v)/2$ colors assigned the probability is at least $\tilde{d_0}'(v)/(48\Delta)$ for event $E'(v,c)=1$, i.e. $Pr(E'(v,c)=1|\sum_{i=0}^{c-1}E'(v,c) \leq \tilde{d_0}'(v)/2) \geq \tilde{d_0}'(v)/(48\Delta)$. Thus we can use Theorem 3 with $t=k_0\Delta$, $p=\tilde{d_0}'(v)/(48\Delta)$ and $y=\tilde{d_0}'(v)/2 \leq tp/2$ (for k_0 sufficiently large), yielding a bound of $e^{k_0/48\cdot\tilde{d}'_0(v)/8} = e^{k_0/384\cdot\tilde{d_0}'(v)} \geq e^{k_0/384\cdot\log n} = 1-1/n^{k_0/384}$. Therefore, the probability that for any sequence of $k_0\Delta$ colors and any initial degree $\tilde{d_0}(v) \geq \log n$ at least half of the neighbors get colored is $1-1/n^{k_0/384}$. For s distinct sequences the probability $Pr(E'_v)$ in G'_v that the degree of node v for $\tilde{d}'(v) > \log n$ (for each sequence) got halved $s < \log n$ times in G'_v becomes $\left(1-1/n^{k_0/384}\right)^s > 1-1/n^{k_0/384-1}$. Due to Lemma 4 the lower bound of probability $Pr(E'_v)$ is also a lower bound for the event $Pr(E_v)$ in G. □

To derive a bound for all nodes, we must take dependencies among nodes into account. The goal is to show that Theorem 6 holds for all nodes concurrently, i.e. to show that $Pr(\wedge_{v\in V}E_v) \geq 1-1/n^k$ for an arbitrary constant k.

THEOREM 7. *All nodes $v \in V$ have $\tilde{d}(v) \leq \max(d(v)/2^s, \log n)$ uncolored neighbors with probability $1-1/n^k$ for arbitrary constant k after executing* ColorTrials *using color set $C(v)$ with $|C(v)| \geq k_0 s\Delta$ and sufficiently large constant k_0.*

PROOF. Let an elementary event F be all random choices made by all nodes v during *ColorTrials* for a graph G, i.e. the random preferences for each node $v \in V$ for each color $c \in C(v)$. Let only the set S_{E_v} of elementary events cause the event E_v not to occur, i.e. $Pr(E_v) = 1-\sum_{F\in S_{E_v}} Pr(F) \geq 1-1/n^k$, where the last inequality is due to Theorem 6. Thus we can use Theorem 2 to bound the probability that all events E_v with $v \in V$ occur by $Pr(\wedge_{v\in V}E_v) \geq 1-1/n^{k-4}$. □

Let us discuss the message size. A node transmits a number of at most $O(\log\Delta)$ bits for every unused color $C(v)$ with $|C(v)| = k_0 s\Delta$ to its neighbors, yielding messages of size $O(|C(v)|s\log\Delta) = O(s\Delta\log\Delta)$. In a modification of Algorithm *ColorTrials* a node picks one color uniformly at random for each sequence $[2i\Delta_{N_+(v)}, 2(i+1)\Delta_{N_+(v)}]$ of $\Delta_{N_+(v)}$ colors for $0 \leq i < k_0 s$ and transmits these $k_0 s$ colors of size $\log\Delta$ resulting in a message of size $O(s\log\Delta)$. The probability that a node v gets a color out of a sequence of $2\Delta_{N_+(v)}$ colors is at least $1/2$ independently of the choices of its neighbors, since any node has at most $\Delta_{N_+(v)}$ uncolored neighbors, but chooses one color among $2\Delta_{N_+(v)}$ many. In case $|C(v)| \leq 2\Delta+1$ with $\Delta \geq \log n$ a node executes the modified version (described above) of *ColorTrials* c times for some constant c. It picks $c/4$ executions out of the c executions, where it picks one color uniformly at random, i.e. it stays inactive for the other $3c/4$ executions. An execution is a success, if at most $\Delta/2$ neighbors participated with node v. Using Theorem 1 it can be shown that we have a success with probability $1-1/n^k$ for arbitrary k (and sufficiently large c). For a successful execution a node gets colored with probability at least $1/2$ since it picks one color out of $\Delta+1$ and there are at most $\Delta/2$ colors picked by neighbors. Therefore, the required message size is only $O(s\log\Delta)$.

Theorem 6 only applies for nodes of degree $d(v) \in \Omega(\log n)$. The next theorem gives a bound on the reduction of an arbitrary degree $d(v)$ for more than one execution of *ColorTrials*. The proof uses similar ideas to the proof of Theorem 6.

THEOREM 8. *After $O(r)$* ColorTrials *all nodes $v \in V$ have $d(v) \leq \Delta_0/2$, where Δ_0 is the maximum size of any uncolored neighborhood before the first execution, with probability $1-1/n^k$ for arbitrary constant k and $k_0 \geq 384(k+3)$, if $r\cdot\Delta_0 \geq \log n$.*

PROOF. Consider a node v for a graph G'_v. Let $\Delta_0/2 \leq d'_0(v) \leq \Delta_0$ be the initial degree of v before the first execution of *ColorTrials*. Due to Lemma 5 the probability $Pr(E'(v,c)=1)$ to get a certain color c is at

least $\tilde{d}'(v)/(24\Delta) \geq 1/48$ as long as $\tilde{d}'(v) > d'_0(v)/2$, i.e. $Pr(E'(v,c) = 1 | \sum_{i=0}^{c} E'(v,i)) \leq d'_0(v)/2)$. We have $|C_i(v)| > \Delta$ for all $i \in [0, k_0 r - 1]$ (otherwise node v might not be able to compute a coloring, e.g. if it is in a clique). Thus, we can use Theorem 3 with $t = k_0 r \Delta$ executions of *ColorTrials*, $p = 1/48$ and $y = d'_0(v)/2 \leq tp/2$ (for k_0 sufficiently large), yielding a bound of $e^{k_0/48 \cdot \tilde{d}'_0(v)/8} = e^{k_0/384 \cdot \tilde{d}_0{}'(v)} \geq e^{k_0/384 \cdot \log n} = 1 - 1/n^{k_0/384}$. Due to Theorem 4 this probability bounds also the probability of the event E_v for a node $v \in G$ that the number of uncolored neighbors is halved. Using Theorem 2 the number of all uncolored neighbors is at most $\Delta_0/2$ for all nodes $v \in V$ with probability $1 - 1/n^{k_0/384-3}$. □

5.1.3 Algorithm DeltaPlus1Coloring

First we consider Algorithm *RankingTrials* before discussing the run time and message size required to compute a $\Delta + 1$ coloring.

THEOREM 9. *Within time $O(\sqrt{\log n})$ Algorithm* RankingTrials *computes a $\Delta + 1$ coloring with $\Delta \leq \sqrt{\log n}$ with probability $1 - \frac{1}{n^c}$ for some constant c.*

PROOF. Consider a node $v \in V$ having less than $\sqrt{\log n}$ uncolored neighbors, i.e. $d(v) \leq \sqrt{\log n}$. Initially, it does not have any rank priority, i.e. $RP(v)[i] = none$ for all ranks i. For each rank i there are initially $2\sqrt{\log n} + 1$ available priorities. All rank priorities must be distinct, i.e. $RP(v)[i] \neq RP(v)[j]$ for $i \neq j$ and a node picks at most $\sqrt{\log n}$ rank priorities and at most one per rank i. Since $d(v) \leq \sqrt{\log n}$ at most $\sqrt{\log n}$ priorities get picked by neighbors per rank i. Thus we have for any rank i at least $\sqrt{\log n}+1$ unused priorities throughout the execution of the algorithm. Therefore, the probability that for some rank i within one message exchange a node v chooses a priority out of $\sqrt{\log n}+1$ such that no other uncolored neighbor of at most $\sqrt{\log n}$ many chooses the same random priority is at least $(1 - 1/(\sqrt{\log n} + 1))^{d(v)} \geq 1/e$ independent of the choices of the rank priorities for ranks $j < i$ of all adjacent nodes. Once a node got a priority for some rank i, it does not apply to get another priority for the same rank. Still, at any time during the algorithm a node v has at least $\sqrt{\log n}+1$ different ranks to try to get a rank priority, since the total number of ranks is $2\sqrt{\log n} + 1$ and whenever v tries to get a rank priority it has acquired at most $\sqrt{\log n}$ yet. Thus when performing one message exchange a node performs $\sqrt{\log n} + 1$ trials. After $O(\sqrt{\log n})$ exchanges we have performed $O(\log n)$ trials. Each has a success probability of at least $1/e$ independent of prior trials. Thus the probability for a single node v to get $\sqrt{\log n}$ rank priorities is $1 - 2^{-O(\log n)}$ using Theorem 1. Using Theorem 2 all nodes obtained $\sqrt{\log n}$ rank priorities with probability $1 - 2^{-O(\log n)}$.

All priorities are distinct and are intended for distinct ranks. When using the rank priorities to get a rank then for each chosen rank priority $RP(v)[i] \neq none$ either a node v gets the rank i or a neighbor has already got it, since a node gets only one rank and a node v has picked $\sqrt{\log n}+1$ rank priorities but has at most $\sqrt{\log n}$ uncolored neighbors, node v gets a rank distinct from its neighbors. Afterwards, when nodes attempt to join according to their ranks, every node get a color. □

THEOREM 10. *Within time $O(\log \Delta + \sqrt{\log n})$ a $\Delta + 1$ coloring is computed with probability $1 - 1/n^k$ for arbitrary constant k.*

PROOF. Due to Theorem 8 after O(1) executions of Algorithm *ColorTrials* any node v halfs its uncolored neighbors as long as $\Delta \geq \log n$. Thus after $O(\log \Delta)$ executions we have $\Delta \leq \log n$. Assume $\sqrt{\log n} \leq d(v) < \log n$. Assume a node v with $d(v) \geq \Delta_0/2$ performed a sequence of $k_0 \log n/d(v)$ executions of *ColorTrials* for constant k_0, where Δ_0 equals the maximum degree Δ before the first execution of the sequence. Using Theorem 8 with $r = k_0 \log n/\Delta$ the degree $d(v)$ for each node $v \in V$ must be less than $\Delta/2$ with probability at least $1 - 1/n^{k_1+1}$ for an arbitrary constant k_1 (and sufficiently large constant k_0).

The total time until $\tilde{d}(v) < \sqrt{\log n}$ is bounded by $\sum_{i=\log\log n/2}^{\log\log n} O(1) \cdot \log n \cdot (1/2)^i = \sum_{i=0}^{\log\log n/2} O(\sqrt{\log n}) \cdot (1/2)^i = O(\sqrt{\log n})$. The chance that all $O(\log \Delta + \log\log n)$ divisions succeed for all nodes is $(1 - 1/n^{k_1+1})^{O(\log \Delta + \log\log n)} = 1 - 1/n^{k_1}$. Using Theorem 9 the remaining nodes can be colored in time $O(\sqrt{\log n})$. □

Messages of size $O(\sqrt{\log n} \cdot \log\log n)$ are sufficient during Algorithm *RankingTrials*, since $O(\sqrt{\log n})$ numbers have to be transmitted using one message. Each number is in $[0, 2\sqrt{\log n}]$ and can be encoded with $O(\log\log n)$ bits.

For Algorithm *DeltaPlus1coloring* messages of size $O(\log \Delta + \sqrt{\log n} \cdot \log\log n)$ are sufficient. As long as a node has more than $\sqrt{\log n}$ uncolored neighbors, i.e. $\tilde{d}(v) > \sqrt{\log n}$ it suffices to choose randomly one color of $O(\log \Delta)$ bits for each round and transmit it as shown in the remark after Theorem 7.

5.1.4 Algorithms ConstDeltaCo. And ConstTimeCo.

After establishing a bound on the running time we investigate the amount of transmitted information.

LEMMA 11. *Using at least $(1 + 1/2^c)\Delta$ colors, within time $O(c)$ the number of uncolored neighbors is at most $\max(\Delta/({}^c 2), \log n)$ for every node with probability $1 - 1/n^k$ for arbitrary constant k.*

PROOF. Let Δ_0 be the maximum degree Δ before the first execution. Due to Theorem 8 after O(1) executions of Algorithm *ColorTrials* any node v halfs its uncolored neighbors as long as $\Delta \geq \log n$. After $O(c)$ rounds the fraction of uncolored neighbors is less than $1/2^{k_1 c}$ (for an arbitrary constant k_1), i.e. the maximum number of uncolored neighbors is $\Delta \leq \Delta_0/2^{k_1 c}$. Since we use $(1 + 1/2^c)\Delta_0$ colors and the neighbors $N(v)$ of a node v use up at most Δ_0 colors, at least $1/2^c \Delta_0$ colors are available for a node v in every execution of *ColorTrials*.

Using Theorem 7 with $s = 1/2^c \Delta_0/(\Delta_0/2^{k_1 c}) = 2^{(k_1-1)c}$ after one execution of *ColorTrials* the maximum number of uncolored neighbors for node v becomes $\Delta/2^{2^{(k_1-1)c}}$. In an analogous derivation for an initial maximum degree of $\Delta/x \leq \Delta/2^{2^{(k_1-1)c}}$, we can reduce the maximum degree to $\Delta/2^{x/k_0}$ with one message exchange with probability at least $1 - 1/n^{k+1}$ for any constant k and any sufficiently large constant $k_0 \geq 384(k + 2)$ due to Theorem 7. Thus after two message exchanges the maximum number of uncolored neighbors is reduced from Δ to $\Delta/2^{2^{(x/k_0)/k_0}} < \Delta/2^x$ for $x \geq (k_0)^2$. Thus after $O(c)$ rounds all nodes have less than

$\Delta/\binom{c}{2} \leq \log n$ uncolored neighbors with probability $(1 - 1/n^{k+1})^{O(c)} > 1 - 1/n^k$. □

THEOREM 12. *If the number of uncolored neighbors is at most $\log n$ for all nodes, i.e. $\Delta \leq \log n$, then using $O(\log^{1+1/c} n)$ colors within time $O(c)$ every node gets colored with probability $1 - 1/n^k$ for arbitrary constant k.*

PROOF. Consider a node $v \in G'_v$ with $(\Delta)^{1-1/c} < \tilde{d}'(v) \leq \Delta$. Due to Lemma 5 the probability $Pr(E'(v,c) = 1)$ to get a certain color c is at least $Pr(E'(v,c) = 1) \geq \Delta^{1-1/c}/(24\Delta) = 1/(24\Delta^{1/c}) \geq 1/(24\log^{1/c} n)$ as long as $\tilde{d}'(v) \geq (\Delta)^{1-1/c}$, i.e. $Pr(E'(v,c) = 1 | \sum_{i=0}^{c} E'(v,c) \leq d'_0(v) - (\Delta)^{1-1/c})$. Thus we can use Theorem 3 with $t = k_0 \log^{1+1/c} n$, $p = 1/(24\log^{1/c} n)$ and $y = d'(v) - (\Delta)^{1-1/c} \leq tp/2$ (for k_0 sufficiently large), yielding a bound of $e^{k_0/48 \cdot \log n/8} = e^{k_0/384 \cdot \log n} = 1 - 1/n^{k_0/384}$. Thus after one round the number of remaining uncolored nodes $d'(v) - y = d'(v) - (d'(v) - (\Delta)^{1-1/c}) = (\Delta)^{1-1/c})$. Using Theorem 2 the probability that event E_v occurs for all nodes is at least $1 - 1/n^{k_0/384-4}$. After c iterations of the loop every node is colored with probability $(1 - 1/n^{k_0/384-4})^c \geq (1 - 1/n^{k_0/384-5})$. □

LEMMA 13. *Using $(1 + 1/2^c)\Delta + O(\log^{1+1/c} n)$ colors within time $O(c + \log^* n)$ every node gets colored with probability $1 - 1/n^k$ for arbitrary constant k.*

PROOF. Since $\Delta \leq n$ and ${}^{\log^* n}2 = n$ using Theorem 11 yields that after $O(\log^* n)$ time the maximum number of uncolored neighbors Δ is at most $\log n$ with probability $1-1/n^k$. Using Theorem 12 yields that all nodes are colored with probability $1 - 1/n^k$. The probability that both eventuate is $(1 - 1/n^k)^2 \geq 1 - 1/n^{k-1}$ for an arbitrary constant k. □

COROLLARY 14. *Using $(1 + 1/2^{\log^* n})\Delta + O(\log^{1+1/\log^* n} n)$ colors within time $O(\log^* n)$ every node gets colored with probability $1 - 1/n^k$ for arbitrary constant k.*

THEOREM 15. *Within time $O(c)$ every node gets a color out of $O(\Delta \log^{(c)} n + \log^{1+1/c} n)$ colors with probability $1 - 1/n^k$ for arbitrary constant k.*

PROOF. Using Theorem 8 with $s = k_0 \log^{(c)} n$ and constant k_0 the maximum number of uncolored neighbors Δ is reduced by a factor $2^{\log^{(c)} n} = \log^{(c-1)} n$ with probability $1-1/n^k$ for all nodes $v \in V$ after one execution of *ColorTrials*. Using Theorem 6 again with $s = k_0 \log^{(c)} n \log^{(c-1)} n$ the maximum number of still uncolored neighbors Δ is reduced by a factor $2^{\log^{(c)} n \log^{(c-1)} n} > \log^{(c-2)} n$. Thus after c rounds the reduction is by a factor $\log n$ and in the round $c + 1$ it is by a factor up to n, such that Δ is at most $\log n$ (Theorem 8 holds only for $\Delta \geq \log n$). Using Theorem 12 all nodes with $\Delta \leq \log n$ are colored in time $O(c)$ with probability $1 - 1/n^k$ yielding an overall probability of $(1 - 1/n^k)^2 \geq 1 - 1/n^{k-1}$ for arbitrary constant k. □

Let us discuss the amount of transmitted information for algorithm *ConstDeltaColoring*. A node transmits a number for every unused color in every round to its neighbors. However, as shown in the remark after Theorem 7 it is sufficient to choose one color for every sequence of $2\Delta_{N_+(v)}$ colors. Thus, choosing $O(\log n)$ colors each with $O(\log \Delta)$ bits yields a probability of $1-1/n^k$ for an arbitrary constant k, yielding messages of size $O(\log \Delta \log n)$ during the execution of *ColorTrials*. Still, Theorem 7 is only valid for $\tilde{d}(v) \geq \log n$. But once $\tilde{d}(v) < \log n$ the size of a color is only $\Omega(\log \log n)$ bits (e.g. we might add $O(\log^{1+1/c} n)$ to all assigned colors and, thereby, we can assign colors $\{0, 1, ..., \Omega(\log^{1+1/c} n)\}$ to the remaining nodes). Thus the maximum message size in one round is $O(\log n(\log \Delta + \log \log n \log^{1/c} n) = O(\log \Delta \log n)$. The reason for Algorithm *ConstTimeColoring* is the same except that the number of colors is $|C(v)| \in O(\Delta \log^{(c)} n + \log^{1+1/c} n)$ and thus a color requires $O(\log \Delta + \log^{(c+1)} n)$ bits instead of $O(\log \Delta)$.

5.2 Ruling Set

THEOREM 16. *Within time $2^c d^{1/c}$ Algorithm* CoordinateTrials(d, c) *computes a $(2, c)$-ruling set.*

PROOF. If node v stops attempting to join the ruling set, it was forced to stop by some neighbor $u \in N(v)$ with $Rank(u)$ larger 0 that continues itself, since u forced all its neighbors $N(u)$ (except those with the same rank) to stop and therefore is not stopped itself by any neighbor. Thus, all nodes $U \subseteq V$ that continue and are reachable from u by nodes in U only, must have the same rank, i.e. all nodes $u \cup U$ must have some coordinate i that is equal for all of them. If this was not the case, then there must have been two nodes $u, w \in U$ with distinct ranks that both continued. However, due to the algorithm either u forced w to stop or the other way around. Since some coordinate i has the same value for all nodes in U, we know that bit i of $Rank(u)$ must remain 0 from now on for all nodes $u \cup U$, i.e. we say that coordinate i is *done* and has no influence on the further computation.

Any two neighbors u, v attempting to join the ruling set must have one coordinate that is distinct for both of them throughout the algorithm, i.e. $\exists i \in [0, c-1]$ s.t. $coord(v)[i] \neq coord(u)[i]$. Initially, this holds since otherwise they have the same color. If two adjacent nodes u, v consider a coordinate i as done and still execute the algorithm then both must have the same value for coordinate i. Thus, at least one distinct coordinate that is not done remains. Thus, if a node v considers $c-1$ coordinates as done and is still executing the algorithm then for the last coordinate it is either forced to stop by a neighbor u that joins the ruling set or it joins the ruling set itself. This also implies that nodes joining the ruling sets are independent, since otherwise the last coordinate(s) that has not been considered as done would have to be equal.

Consider a node $v \in U \subseteq V$. When node v stops some neighbor w proceeds and and one coordinate for all continuing nodes $w \in W \subseteq U$ is done. Since there are only c coordinates within distance c of v a node joins the ruling set. One iteration out of $d^{1/c}$ many of the repeat loop (i.e. an increment of j) takes time 2^c. Thus the time complexity is $2^c \cdot d^{1/c}$. □

We describe three ways to improve the time complexity of Algorithm *CoordinateTrials* – two of them at the price of having larger distance to a node in the ruling set. First, it is not necessary to check a rank for all values in $[0, 2^c]$ if some

coordinates, e.g. i, j, are *done* (see Proof of Theorem 16), i.e. the bits i, j of a rank are fixed to 0. If k coordinates are done, then a rank of a node v (and its neighbors) has k bits fixed to 0 and the rank must be one of 2^{c-k} distinct values. Second, if a node and all its neighbors have rank 0, there is no need to go through all possible 2^{c-k} values for a rank. If a node v only iterates through all possible rank values, if itself or a neighbor has rank larger 0 then at least one neighbor within distance 2 will consider a coordinate as done. Thus, within distance $2c$ a node joins the ruling set within time $d^{1/c} + \sum_{k=0..c} 2^{c-k} = d^{1/c} + 2^{c+1}$, i.e. we can compute a $(2, 2c)$ ruling set in time $d^{1/c} + 2^{c+1}$. Third, a node can join the ruling set whenever its rank is strictly larger than that of all its neighbors and stop the algorithm whenever some neighbor has a rank larger than itself. In this case it does not have to iterate through the rank values at all. This results in a running time of $d^{1/c}$ to compute a $(2, 2^{c+1})$ ruling set.

Let $f_{trials}(d)$ be the time complexity depending on the number of colors of Algorithm *CoordinateTrials*. Let $c_{trials}(c)$ be the distance from any node to a node in the ruling set depending on the parameter c.

THEOREM 17. *Within time* $O(c \log^{1/c} n + f_{trials}(\log^{1+1/\log^* n} n))$ *Algorithm* RandRulingSet(c) *computes a* $(2, (c+1) + c_{trials}(c+1))$*-ruling set for* $c > 0$ *with probability* $1 - 1/n^3$.

PROOF. We show that within time $O(c \log^{1/c} n)$ every node gets a colored node within distance $c + 1$, such that all colored nodes form a proper coloring. Once this coloring is computed we call Algorithm *CoordinateTrials*$((\log^{1+1/\log^* n} n, c+1)$.

Initially, a node v starts with $p_v := 1/n = 2^{-\log n}$. After the first iteration of the for loop for node v or one of its neighbors holds that the sum of all values p_u of nodes $u \in N_+(v)$ is at least $2^{(-\log n^{1-1/c})}$ and at most 1. Since the maximum degree of a node is $n - 1$ and a node begins with joining probability $1/n$, the sum of the values p_u of node $u \in N_+(v)$ is at most $1/n \cdot n = 1$. Due to Subroutine *IncProb(t)* a value p is not multiplied by t if the sum is larger than $1/t$. Thus the sum never exceeds 1. Consider a node v with a sum of at least $2^{(-\log n^{1-1/c})}$. After the second iteration node v itself or a neighbor has sum at least $2^{(-\log n^{1-2/c})}$. Thus after c iterations a node v must have a node $u \in N^c(v)$, i.e. within distance c, having the sum p_w with $w \in N_+(u)$ being at most 1 and at least 1/2. The duration of the first iteration is given by $\log n/(\log n^{1-1/c}) = \log n^{1/c}$ and in general, the duration of the ith iteration is $\log n^{1-(i-1)/c}/(\log n^{1-i/c}) = \log n^{1/c}$. Since one iteration takes $\log^{1/c} n$ time and we need c iterations, the time complexity is $c \log^{1/c} n$. When node v participates in computing a coloring with probability $\min(1, 64 p_v \log n)$ (independently of all other nodes), the probability that more than $1024 \log n$ neighbors or none participate is $1/n^8$ using a Chernoff bound (Theorem 1). Using Theorem 2 with probability $1 - 1/n^4$ all nodes $u \in V$ have at least 1 and at most $1024 \log n$ neighbors that participate in computing the coloring.

An $O(\log^{1+1/\log^* n})$ coloring can be computed with probability at least $1 - 1/n^4$ as shown in Corollary 14. The overall success probability is $(1 - 1/n^4) \cdot (1 - 1/n^4) > 1 - 1/n^3$.

□

THEOREM 18. *Within time* $O(2^c \log^{1/c} n)$ *Algorithm* RandRulingSet(c) *computes a* $(2, 2(c+1))$*-ruling set for* $c > 0$ *with probability* $1 - 1/n^3$.

PROOF. Due to Theorem 16, Algorithm *CoordinateTrials*$(\log^{1+1/\log^* n} n, c+1)$ computes a $(2, c+1)$-ruling set in time $O(2^{c+1}(\log^{1+1/\log^* n} n)^{1/c+1}) < O(2^c \log^{1/c} n)$. Using Theorem 17 completes the proof. □

In Subroutine *IncProb(t)* a node v transmits its value p_v to all two hop neighbors, resulting in messages of size $O(\Delta \log n)$. Alternatively, a node might transmit p_v to its neighbors and any node $u \in V$ having $\sum_{w \in N_+(u)} p_w \geq 1/t$ informs its neighbors. Thus messages of size $O(\log n)$ are sufficient.

6. REFERENCES

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

[2] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg. Fast distributed network decompositions and covers. *J. Parallel Distrib. Comput.*, 39(2):105–114, 1996.

[3] B. Awerbuch, A. V. Goldberg, M. Luby, and S. A. Plotkin. Network decomposition and locality in distributed computation. In *Symp. on Foundations of Computer Science (FOCS)*, 1989.

[4] L. Barenboim and M. Elkin. Sublogarithmic distributed mis algorithm for sparse graphs using nash-williams decomposition. In *Symp. on Principles of distributed computing*, 2008.

[5] L. Barenboim and M. Elkin. Distributed $(\delta + 1)$-coloring in linear (in δ) time. In *Symp. on Theory of computing*, 2009.

[6] L. Barenboim and M. Elkin. Deterministic distributed vertex coloring in polylogarithmic time. In *PODC*, 2010.

[7] R. Cole and U. Vishkin. Deterministic Coin Tossing with Applications to Optimal Parallel List Ranking. *Inf. Control*, 70(1):32–54, 1986.

[8] P. L. Erdös, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of two others. *J. Comb. Theory, Ser. A*, 33(2):158–166, 1982.

[9] B. Gfeller and E. Vicari. A Randomized Distributed Algorithm for the Maximal Independent Set Problem in Growth-Bounded Graphs. In *Sy. on Principles of Distributed Computing*, 2007.

[10] A. Goldberg, S. Plotkin, and G. Shannon. Parallel Symmetry-Breaking in Sparse Graphs. *SIAM Journal on Discrete Mathematics (SIDMA)*, 1(4):434–446, 1988.

[11] D. A. Grable and A. Panconesi. Nearly optimal distributed edge colouring in o(log log n) rounds. In *SODA*, 1997.

[12] A. Israeli and A. Itai. A Fast and Simple Randomized Parallel Algorithm for Maximal Matching. *Information Processing Letters*, 22:77–80, 1986.

[13] K. Kothapalli, C. Scheideler, M. Onus, and C. Schindelhauer. Distributed coloring in $O(\log^{1/2} n)$ bit rounds. In *International Parallel & Distributed Processing Symp. (IPDPS)*, 2006.

[14] F. Kuhn. Weak Graph Coloring: Distributed Algorithms and Applications. In *SPAA*, 2009.

[15] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer. Local Approximation Schemes for Ad Hoc and Sensor Networks. In *DIALM-POMC*, 2005.

[16] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What Cannot Be Computed Locally! In *PODC*, 2005.

[17] F. Kuhn and R. Wattenhofer. On the Complexity of Distributed Graph Coloring. In *PODC*, 2006.

[18] N. Linial. Locality in Distributed Graph Algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

[19] M. Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM Journal on Computing*, 1986.

[20] A. Panconesi and A. Srinivasan. Improved distributed algorithms for coloring and network decomposition problems. In *Symp. on Theory of computing (STOC)*, 1992.

[21] J. Schneider and R. Wattenhofer. A Log-Star Distributed Maximal Independent Set Algorithm for Growth-Bounded Graphs. In *Symp. on Principles of Distributed Computing(PODC)*, 2008.

[22] G. De Marco and A. Pelc. Fast distributed graph coloring with $O(\Delta)$ colors. In *SODA*, 2001.