An Optimal Distributed $(\Delta + 1)$-Coloring Algorithm?*

Yi-Jun Chang
University of Michigan
USA
cyijun@umich.edu

Wenzheng Li
The Institute for Theoretical Computer Science (ITCS), Institute for Interdisciplinary Information Sciences, Tsinghua University
China
wz-li14@mails.tsinghua.edu.cn

Seth Pettie
University of Michigan
USA
pettie@umich.edu

ABSTRACT

Vertex coloring is one of the classic symmetry breaking problems studied in distributed computing. In this paper we present a new algorithm for $(\Delta+1)$-list coloring in the randomized LOCAL model running in $O(\log^* n + \mathrm{Det}_d(\mathrm{poly}\log n))$ time, where $\mathrm{Det}_d(n')$ is the deterministic complexity of $(\deg+1)$-list coloring on n'-vertex graphs. This improves upon a previous randomized algorithm of Harris, Schneider, and Su (STOC 2016) with complexity $O(\sqrt{\log \Delta} + \log\log n + \mathrm{Det}_d(\mathrm{poly}\log n))$, and (when Δ is sufficiently large) is much faster than the best known deterministic algorithm of Fraigniaud, Heinrich, and Kosowski (FOCS 2016), whose time complexity is $O(\sqrt{\Delta}\log^{2.5}\Delta + \log^* n)$ time.

Our algorithm *appears* to be optimal. It matches the $\Omega(\log^* n)$ randomized lower bound, due to Naor (SIDMA 1991) and *sort of* matches the $\Omega(\mathrm{Det}(\mathrm{poly}\log n))$ randomized lower bound due to Chang, Kopelowitz, and Pettie (FOCS 2016), where Det is the deterministic complexity of $(\Delta+1)$-list coloring. The best known upper bounds on $\mathrm{Det}_d(n')$ and $\mathrm{Det}(n')$ are both $2^{O(\sqrt{\log n'})}$ (Panconesi and Srinivasan (J. Algor 1996)) and it is quite plausible that the complexities of both problems are the same, asymptotically.

CCS CONCEPTS

• **Theory of computation → Distributed algorithms**;

KEYWORDS

Distributed algorithms, Local model, Vertex Coloring

ACM Reference Format:
Yi-Jun Chang, Wenzheng Li, and Seth Pettie. 2018. An Optimal Distributed $(\Delta+1)$-Coloring Algorithm?. In *Proceedings of 50th Annual ACM SIGACT Symposium on the Theory of Computing (STOC'18).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3188745.3188964

*This work is supported by NSF grants CCF-1514383 and CCF-1637546. A full version of this paper is available at [7].

STOC'18, June 25–29, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5559-9/18/06…$15.00
https://doi.org/10.1145/3188745.3188964

1 INTRODUCTION

Much of what we know about the LOCAL model has emerged from studying the complexity of four canonical symmetry breaking problems and their variants: maximal independent set (MIS), $(\Delta+1)$-vertex coloring, maximal matching, and $(2\Delta-1)$-edge coloring. The palette sizes "$\Delta+1$" and "$2\Delta-1$" are minimal to still admit a greedy sequential solution; here Δ is the maximum degree.

Early work [1, 2, 24–27] showed that all the problems are reducible to MIS, all four problems require $\Omega(\log^* n)$ time, even with randomization; all can be solved in $O(\mathrm{poly}(\Delta) + \log^* n)$ time (optimal when Δ is constant), or in $2^{O(\sqrt{\log n})}$ time for any Δ. Until recently, it was actually consistent with known results that all four problems had the same complexity.

Kuhn, Moscibroda, and Wattenhofer (KWM) [22] proved that the "independent set" problems (MIS and maximal matching) require $\Omega\left(\min\{\frac{\log\Delta}{\log\log\Delta}, \sqrt{\frac{\log n}{\log\log n}}\}\right)$ time, with or without randomization, via a reduction from $O(1)$-approximate minimum vertex cover. This lower bound provably separated MIS/maximal matching from simpler symmetry-breaking problems like $O(\Delta^2)$-coloring, which can be solved in $O(\log^* n)$ time [24].

We now know the KMW lower bounds cannot be extended to the canonical coloring problems, nor to variants of MIS like $(2,t)$-ruling sets, for $t \geq 2$ [5, 6, 16]. Elkin, Pettie, and Su [12] proved that $(2\Delta-1)$-list edge coloring can be solved by a randomized algorithm in $O(\log\log n + \mathrm{Det}(\mathrm{poly}\log n))$ time, which shows that neither the $\Omega\left(\frac{\log\Delta}{\log\log\Delta}\right)$ nor $\Omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)$ KMW lower bound applied to this problem. Here $\mathrm{Det}(n')$ represents the *deterministic* complexity of the problem in question on n'-vertex graphs. Improving on [5, 29], Harris, Schneider, and Su [19] proved a similar separation for $(\Delta+1)$-vertex coloring. Their randomized algorithm solves the problem in $O(\sqrt{\log\Delta} + \log\log n + \mathrm{Det}_d(\mathrm{poly}\log n))$ time, where Det_d is the complexity of $(\deg+1)$-list coloring.

The "$\mathrm{Det}(\mathrm{poly}\log n)$" terms in the running times of [12, 19] are a consequence of the *graph shattering* technique applied to distributed symmetry breaking. Barenboim, Elkin, Pettie, and Schneider [5] showed that all the classic symmetry breaking problems could be reduced in $O(\log\Delta)$ or $O(\log^2\Delta)$ time, w.h.p., to a situation where we have independent subproblems of size $\mathrm{poly}\log(n)$, which can then be solved with the best available deterministic algorithm.[1]

[1]In the case of MIS, the subproblems actually have size $\mathrm{poly}(\Delta)\log n$, but satisfy the additional property that they contain distance-5 dominating sets of size $O(\log n)$, which is often just as good as having $\mathrm{poly}\log(n)$ size. See [5, §3] or [16, §4] for more discussion of this.

Later, Chang, Kopelowitz, and Pettie (CKP) [8] gave a simple proof illustrating *why* graph shattering is inherent to the LOCAL model: the randomized complexity of any locally checkable problem[2] is at least its deterministic complexity on $\sqrt{\log n}$-size instances.

The CKP lower bound explains why the state-of-the-art randomized symmetry breaking algorithms have such strange stated running times: they all depend on a randomized graph shattering routine (Rand.) and a deterministic (Det.) algorithm.

- $O(\log \Delta + 2^{O(\sqrt{\log \log n})})$ for MIS (Rand. due to [16] and Det. to [27]),
- $O(\sqrt{\log \Delta} + 2^{O(\sqrt{\log \log n})})$ for $(\Delta + 1)$-vertex coloring (Rand. due to [19] and Det. to [27]),
- $O(\log \Delta + (\log \log n)^3)$ for maximal matching (Rand. due to [5] and Det. to [13]),
- $O((\log \log n)^8)$ for $(2\Delta - 1)$-edge coloring (Rand. due to [12] and Det. to [14]).

In each, the term that depends on n is the complexity of the best deterministic algorithm, scaled down to poly $\log(n)$-size instances. In general, improvements in the deterministic complexities of these problems imply improvements to their randomized complexities, but only if the running times are improved in terms of "n" rather than "Δ." For example, a recent line of research has improved the complexity of $(\Delta+1)$-coloring in terms of Δ, from $O(\Delta + \log^* n)$ [4], to $\tilde{O}(\Delta^{3/4}) + \log^* n$ [3], to the state-of-the-art bound of $\tilde{O}(\sqrt{\Delta}) + \log^* n$, due to Fraigniaud, Heinrich, and Kosowski [15]. These improvements do not have consequences for randomized coloring algorithms using graph shattering [5, 19] since we can only assume $\Delta = (\log n)^{\Omega(1)}$ in the shattered instances.

A Technical History of Randomized $(\Delta + 1)$-Coloring. In this paper we prove that $(\Delta + 1)$-list coloring can be solved in $O(\log^* n + \mathrm{Det}_d(\mathrm{poly} \log n))$ time w.h.p., which is always $2^{O(\sqrt{\log \log n})}$, given the best known bound on $\mathrm{Det}_d(n') = 2^{O(\sqrt{\log n'})}$ [27]. Our algorithm seems to come close to the $\Omega(\log^* n + \mathrm{Det}(\mathrm{poly} \log n))$ lower bound implied by [8, 24, 26], where Det is the deterministic complexity of $(\Delta + 1)$-list coloring. Intellectually, our algorithm builds on a succession of breakthroughs by Schneider and Wattenhofer [29], Barenboim, Elkin, Pettie, and Schneider [5], Elkin, Pettie, and Su, [12], and Harris, Schneider, and Su [19], which we shall now review.

Schneider and Wattenhofer [29] gave the first evidence that the canonical coloring problems may not be subject to the KMW lower bounds. They showed that when the palette size is $(1 + \epsilon)\Delta$, where $\epsilon = \Omega(1)$ and $\Delta > \mathrm{poly} \log n$ is sufficiently large, that vertex coloring could be solved in just $O(\log^* n)$ time, w.h.p. The key observation is that the number of *excess colors* (current palette size minus number of uncolored neighbors) is non-decreasing over time. After $O(\log \epsilon^{-1})$ rounds of a standard coloring routine, the number of excess colors $(\epsilon\Delta)$ becomes larger than the uncolored degree. At this point there is a dramatic transition, and the probability that a vertex remains uncolored is reduced exponentially in each successive round: $O(\log^* n)$ more rounds suffice. Of course, in the $(\Delta+1)$-coloring problem there *is just one excess color* initially, so the problem is how to *create* them. Elkin, Pettie, and Su [12] observed that if the graph is "$(1 - \epsilon)$-locally sparse," that after *one* iteration of a random coloring routine, a significant number $(\Omega(\epsilon\Delta))$ of *pairs* of vertices in the neighborhood $N(v)$ get assigned the same color, thereby creating $\Omega(\epsilon\Delta)$ excess colors at v.[3] The notion of local sparsity is especially useful for addressing the $(2\Delta-1)$-edge coloring problem [12], since it can be phrased as $(\Delta' + 1)$-vertex coloring the *line graph* $(\Delta' = 2\Delta - 2)$, which is $(1/2 + o(1))$-locally sparse.

Of course, in the vertex coloring problem we cannot count on any kind of local sparsity, so the next challenge is to make local *density* also work to our advantage. Harris, Schneider, and Su [19] developed a remarkable new graph decomposition that can be computed in $O(1)$ rounds of communication. The decomposition takes a parameter ϵ, and partitions the vertices into an "ϵ-*sparse*" set, and several vertex-disjoint "ϵ-*dense*" components, each with weak diameter 2. The sparse set can be colored in $O(\log \epsilon^{-1} + \log \log n + \mathrm{Det}_d(\mathrm{poly} \log n))$ time[4] using [12] and [5]. Harris et al. [19] proved that by coordinating the coloring decisions within each dense component, it takes only $O(\log_{1/\epsilon} \Delta + \log \log n + \mathrm{Det}_d(\mathrm{poly} \log n))$ time to color the dense sets, i.e., the bound *improves* as $\epsilon \to 0$. The time for the overall algorithm is minimized by choosing $\epsilon = \exp(-\Theta(\sqrt{\log \Delta}))$.

1.1 New Results and Technical Overview

In this paper we give a fast randomized algorithm for $(\Delta + 1)$-vertex coloring. It is based on a hierarchical version of the Harris-Schneider-Su clustering with roughly $\log \log \Delta$ levels determined by an increasing sequence of sparsity thresholds $(\epsilon_1, \ldots, \epsilon_\ell)$, with $\epsilon_i = \sqrt{\epsilon_{i-1}}$. Following [19], we begin with a single iteration of a procedure OneShotColoring, in which a constant fraction of the vertices are colored. The guarantee of this procedure is that any vertex v at the ith level (which is ϵ_i-dense but ϵ_{i-1}-sparse), has $\Omega(\epsilon_{i-1}^2 \Delta)$ *pairs* of vertices in its neighborhood $N(v)$ assigned the same color, thereby creating that many excess colors in the palette of v.

At this point, the most natural way to proceed is to apply a Harris-Schneider-Su style coloring procedure to each level, one by one, with the hope that each will take roughly constant time. The reason is that $O(\log_{1/\epsilon_i} 1/\mathrm{poly}(\epsilon_{i-1})) = O(1)$, so in constant time we should be able to create a situation where any uncolored vertices have $O(\mathrm{poly}(\epsilon_{i-1})\Delta)$ uncolored neighbors but $\Omega(\epsilon_{i-1}^2 \Delta)$ excess colors in their palette. With such a large gap, a Schneider-Wattenhofer style coloring algorithm should complete in $O(1)$ additional steps. This approach does not seem to work. Moreover, doing the layers one by one takes $\Omega(\log \log \Delta)$ time.

In order to color ϵ_i-dense components efficiently we need to maintain relatively large *lower bounds* on the available palette and relatively small *upper bounds* on the number of external neighbors (outside the ϵ_i-dense component). Thus, it is important that when we first consider a vertex, we have not already colored too many of its neighbors. Our algorithm partitions the vertices at level i into *large* and *small* blocks, depending on how many vertices of

[2] See Naor and Stockmeyer [26] or Chang and Pettie [9] for a formal definition of the class of locally checkable labeling (LCL) problems.

[3] A graph is $(1 - \epsilon)$-locally sparse if, for every v, the subgraph induced by $N(v)$ has at most $(1 - \epsilon)\binom{\Delta}{2}$ edges.

[4] It is $(1 - \epsilon')$-locally sparse according to Elkin et al.'s definition [12], for some ϵ' depending on ϵ.

Table 1: Development of lower and upper bounds for distributed $(\Delta+1)$-list coloring in the LOCAL model. The terms $\mathrm{Det}(n')$ and $\mathrm{Det}_d(n')$ are the deterministic complexities of $(\Delta+1)$-list coloring and $(\deg+1)$-list coloring on n'-vertex graphs. All algorithms listed, except for [19] and ours, also solve the $(\deg+1)$-list coloring problem.

	Randomized		Deterministic	
	$O(\log^* n + \mathrm{Det}_d(\mathrm{poly}\log n))$	**new**	$O(\sqrt{\Delta}\log^{2.5}\Delta + \log^* n)$	[15]
	$O(\sqrt{\log\Delta} + \log\log n + \mathrm{Det}_d(\mathrm{poly}\log n))$	[19]	$O(\Delta^{3/4}\log\Delta + \log^* n)$	[3]
	$O(\log\Delta + \mathrm{Det}_d(\mathrm{poly}\log n))$	[5]	$O(\Delta + \log^* n)$	[4]
Upper	$O(\log\Delta + \sqrt{\log n})$	[29]	$O(\Delta\log\Delta + \log^* n)$	[23]
Bounds	$O(\Delta\log\log n)$	[23]	$O(\Delta\log n)$	[2]
	$O(\log n)$	[1, 21, 25]	$O(\Delta^2 + \log^* n)$	[17, 24]
			$O(\Delta^{O(\Delta)} + \log^* n)$	[18]
			$2^{O(\sqrt{\log n})}$	[27]
			$2^{O(\sqrt{\log n\log\log n})}$	[2]
Lower	$\Omega(\log^* n)$	[26]	$\Omega(\log^* n)$	[24]
Bounds	$\Omega(\mathrm{Det}(\sqrt{\log n}))$	[8]		

their ϵ_i-dense components stay at level i (because they are ϵ_{i-1}-sparse). It also partitions the layers themselves into $\log^*(\Delta)$ *strata*. We show that by coloring the small blocks in each stratum, one stratum at a time, and then the large blocks, that we can always guarantee a sufficiently large palette at each vertex when it is first considered. Each of these coloring steps takes $O(1)$ rounds of communication but may not color all vertices. The vertices left uncolored are put in $O(1)$ classes, some of which induce constant degree graphs, which are colored in $O(\log^* n)$ time, and some induce $\mathrm{poly}\log n$-size components, which are colored in $\mathrm{Det}_d(\mathrm{poly}\log n)$ time.

1.2 The LOCAL Model

The undirected input graph $G = (V, E)$ and communications network are identical. Each $v \in V$ hosts a processor that initially knows $\deg(v)$, a unique $\Theta(\log n)$-bit $\mathrm{ID}(v)$, and global graph parameters $n = |V|$ and $\Delta = \max_{v\in V}\deg(v)$. In the $(\Delta + 1)$-list coloring problem each vertex v also has a set $\Psi(v)$ of allowable colors, with $|\Psi(v)| \geq \Delta + 1$. As vertices progressively commit to their final color, we also use $\Psi(v)$ to denote v's available palette, excluding colors taken by its neighbors in $N(v)$. Each processor is allowed unbounded computation and has access to a stream of private unbiased random bits. *Time* is partitioned into synchronized rounds of communication, in which each processor sends one unbounded message to each neighbor. At the end of the algorithm, each v declares its output label, which in our case is a color from $\Psi(v)$ that is distinct from colors declared by all neighbors in $N(v)$. Refer to [24, 28] for more on the LOCAL model and variants.

1.3 Organization

In Section 2 we define a hierarchical decomposition based on [19] and a certain partition of the vertices into $\log\log\Delta$ layers and $\log^*\Delta$ strata. Section 3 gives a high-level description of the algorithm, which uses a variety of coloring routines whose guarantees are summarized in Lemmas 3.2–3.7. Lemma 3.2 (cf. [5, 19]) shows that a procedure OneShotColoring creates many excess colors; it is proved in Section 4. Lemma 3.7 (cf. [12, 29]) analyzes a procedure ColorBidding, which is a generalization of the Schneider-Wattenhofer coloring routing; it is proved in Section 5. Lemmas 3.3–3.6 analyze two versions of an algorithm DenseColoringStep, which is a generalization of the Harris-Schneider-Su routine [19]; they are proved in Section 6. Appendix A reviews some standard concentration inequalities.

2 HIERARCHICAL DECOMPOSITION

In this section, we extend the work of Harris, Schneider, and Su [19] to define a hierarchical decomposition of the vertices based on local sparsity. Let $G = (V, E)$ be the input graph, Δ be the maximum degree, and $\epsilon \in (0, 1)$ be a parameter. An edge $e = \{u, v\}$ is an *ϵ-friend edge* if $|N(u) \cap N(v)| \geq (1 - \epsilon)\Delta$. We call u an *ϵ-friend* of v if $\{u, v\}$ is an *ϵ-friend edge*. A vertex v is *ϵ-dense* if v has at least $(1 - \epsilon)\Delta$ ϵ-friends, otherwise it is *ϵ-sparse*.

We write V^{s}_ϵ (and V^{d}_ϵ) to be the set of ϵ-sparse (and ϵ-dense) vertices. Let v be a vertex in a set $S \subseteq V$ and $V' \subseteq V$. Define $\bar{d}_{S,V'}(v) = |(N(v) \cap V') \setminus S|$ to be the *external degree* of v with respect to S and V', and $a_S(v) = |S \setminus (N(v) \cup \{v\})|$ to be the *anti-degree* of v with respect to S. A connected component C of the subgraph induced by the ϵ-dense vertices and the ϵ-friend edges is called an *ϵ-almost clique*. The following lemma summarizes some properties of ϵ-almost cliques from [19].

Lemma 2.1 ([19]). *Fix any $\epsilon < 1/5$. The following conditions are met for each ϵ-almost clique C, and each vertex $v \in C$. (i) $\bar{d}_{C,V^{\mathrm{d}}_\epsilon}(v) \leq \epsilon\Delta$, (ii) $a_C(v) \leq 3\epsilon\Delta$, (iii) $|C| \leq (1 + 3\epsilon)\Delta$, and (iv) $\mathrm{dist}_G(u, v) \leq 2$ for each $u, v \in C$, i.e., C has weak diameter 2.*

2.1 A Hierarchy of Almost Cliques

Throughout this section, we fix some increasing sequence of sparsity parameters $(\epsilon_1, \ldots, \epsilon_\ell)$ and a subset of vertices $V^\star \subseteq V$, whose meaning will be explained shortly. The sequence $(\epsilon_1, \ldots, \epsilon_\ell)$ always adheres to Definition 2.2.

Definition 2.2. A sequence $(\epsilon_1, \ldots, \epsilon_\ell)$ is a valid *sparsity sequence* if the following conditions are met: (i) $\epsilon_i = \sqrt{\epsilon_{i-1}} = \epsilon_1^{2^{-(i-1)}}$, and (ii) $\frac{1}{\epsilon_\ell} \geq K$ for some large enough constant K.

Layers. Define $V_1 = V^\star \cap V^{\mathsf{d}}_{\epsilon_1}$, and $V_i = V^\star \cap (V^{\mathsf{d}}_{\epsilon_i} \setminus V^{\mathsf{d}}_{\epsilon_{i-1}})$, for $i > 1$. Define $V_{\mathsf{sp}} = V^\star \cap V^{s}_{\epsilon_\ell} = V^\star \setminus (V_1 \cup \cdots \cup V_\ell)$. It is clear that $(V_1, \ldots, V_\ell, V_{\mathsf{sp}})$ is a partition of $V^\star$. We call V_i the *layer-i* vertices, and call V_{sp} the *sparse vertices*. In other words, V_i is the subset of $V^\star$ that are ϵ_i-dense but ϵ_{i-1}-sparse. Remember that the definition of sparsity is with respect to the entire graph $G = (V, E)$ not the subgraph induced by $V^\star$.

Strata. Define $\xi_1 = \epsilon_1$, and $\xi_k = 1/\log(1/\xi_{k-1})$ for $k > 1$. By definition, the 1st stratum is $W_1 = V_1$. The kth stratum $W_k = \bigcup_{i\,:\,\epsilon_i \in (\xi_{k-1}, \xi_k]} V_i$ spans those layers whose sparsity parameter lies in $(\xi_{k-1}, \xi_k]$. Define $s \leq \log^*(1/\epsilon_1)$ to be the index of the last stratum W_s.

Blocks. The layer-i vertices V_i are partitioned into *blocks* as follows. Let $\{C_1, C_2, \ldots\}$ be the set of ϵ_i-almost cliques, and let $B_j = C_j \cap V_i$. Then $(B_1, B_2, \ldots)$ is a partition of V_i, and we call each B_j a *layer-i block*. If layer i is in stratum k, then B_j is also called a *stratum-k block*.

A layer-i block B is a *descendant* of a layer-i' block B', $i < i'$, if B and B' are both subsets of the same $\epsilon_{i'}$-almost clique. Therefore, the set of all blocks in all layers naturally forms a rooted tree $\mathcal{T}$. (The root represents V_{sp}; every other node represents a block in some layer.)

Definition 2.3. A stratum-k block B is a *large block* if $|B| \geq \frac{\Delta}{\log^2(1/\xi_k)}$ <u>and</u> there is no other stratum-k' block B' ($k' \geq k$) such that B' is ancestral to B in $\mathcal{T}$ and $|B'| \geq \frac{\Delta}{\log^2(1/\xi_{k'})}$. Otherwise B is a *small block*.

Notice that the threshold $\frac{\Delta}{\log^2(1/\xi_k)}$ in the above definition depends on the stratum in which the block B resides. By definition, for any two blocks B and B' in different layers, if B is a descendant of B', then B and B' cannot both be large.

Define V^{S}_i, V^{L}_i, W^{S}_k, and W^{L}_k to be, respectively, the sets of all vertices in layer-i small blocks, layer-i large blocks, stratum-k small blocks, and stratum-k large blocks. Notice that $(V^{\mathsf{S}}_i, V^{\mathsf{L}}_i)$ is a partition of V_i and $(W^{\mathsf{S}}_k, W^{\mathsf{L}}_k)$ is a partition of W_k.

Super-blocks. Suppose stratum k spans layers $i', i'+1, \ldots, i$. Let $\{C_1, C_2, \ldots\}$ be the set of ϵ_i-almost cliques, and let $R_j = C_j \cap W_k$. Then $(R_1, R_2, \ldots)$ is a partition of W_k, and we call each R_j a *stratum-k super-block*.

Overview of Our Algorithms. The decomposition and $\mathcal{T}$ are trivially computed in $O(1)$ rounds of communication. Let us briefly explain how our algorithm uses this hierarchical decomposition. The first step is to execute an $O(1)$-round coloring procedure (OneShotColoring) which colors a small constant fraction of the vertices in G. Let $V^\star$ be the remaining uncolored vertices. The set $V^\star$ is partitioned into subsets

$$(W^{\mathsf{S}}_1, \ldots, W^{\mathsf{S}}_s, W^{\mathsf{L}}_1, \ldots, W^{\mathsf{L}}_s, V_{\mathsf{sp}})$$

based on the hierarchical decomposition with respect to a particular sparsity sequence $(\epsilon_1, \ldots, \epsilon_\ell)$. We color the vertices of $V^\star \setminus V_{\mathsf{sp}}$ in $s + 2 = O(\log^*(1/\epsilon_1))$ stages according to the ordering $(W^{\mathsf{S}}_s, \ldots, W^{\mathsf{S}}_1, W', W^{\mathsf{L}}_1)$, where W' is defined as $W^{\mathsf{L}}_2 \cup \cdots \cup W^{\mathsf{L}}_s$. At the end of this process a small portion of vertices $U \subseteq V^\star \setminus V_{\mathsf{sp}}$ may remain uncolored. However, they all have sufficiently large palettes such that $U \cup V_{\mathsf{sp}}$ can be colored efficiently in $O(\log^* n)$ time.

The purpose of processing the small blocks before the large blocks is to ensure that the vertices in small blocks still have an adequate number of colors in their palettes when they are considered. Lemma 2.4 specifies exactly what an *adequate* number of colors is. That is, regardless of how $(W^{\mathsf{S}}_s, W^{\mathsf{S}}_{s-1}, \ldots, W^{\mathsf{S}}_{k+1})$ are colored, each $v \in W^{\mathsf{S}}_k$ still has at least $\Delta/2\log^2(1/\xi_k)$ excess colors in its palette, beyond those needed to color W^{S}_k.

Lemma 2.4. *Suppose that $|N(v) \cap V^\star| \geq \Delta/3$. For each $k \in [1, s]$ and each $v \in W^{\mathsf{S}}_k$, we have $|N(v) \cap (W^{\mathsf{S}}_1 \cup \cdots \cup W^{\mathsf{S}}_{k-1} \cup W^{\mathsf{L}}_1 \cup \cdots \cup W^{\mathsf{L}}_s \cup V_{\mathsf{sp}})| \geq \Delta/2\log^2(1/\xi_k)$.*

Before proving Lemma 2.4 we first establishing a useful property of the block hierarchy $\mathcal{T}$.

Lemma 2.5. *Let C be an ϵ_i-almost clique and $C_1, \ldots, C_l$ be the ϵ_{i-1}-almost cliques contained in C. Either $l = 1$ or $\sum_{j=1}^{l} |C_j| \leq 2(3\epsilon_i + \epsilon_{i-1})\Delta$. In particular, if B is the layer-i block contained in C, either B has one child in $\mathcal{T}$ or the number of vertices in all strict descendants of B is at most $2(3\epsilon_i + \epsilon_{i-1})\Delta < 7\epsilon_i\Delta$.*

Proof. Suppose, for the purpose of obtaining a contradiction, that $l \geq 2$ and $\sum_{j=1}^{l} |C_j| > 2(3\epsilon_i + \epsilon_{i-1})\Delta$. W.l.o.g. suppose C_1 is smallest, so $\sum_{j=2}^{l} |C_j| > (3\epsilon_i + \epsilon_{i-1})\Delta$. Any $v \in C_1$ is ϵ_{i-1}-dense and therefore has at least $(1 - \epsilon_{i-1})\Delta$ neighbors that are ϵ_{i-1}-friends. By definition any ϵ_{i-1}-friend is also an ϵ_i friend, so this set is contained in C. By Lemma 2.1, $|C| \leq (1 + 3\epsilon_i)\Delta$. Thus, by the pigeonhole principle, some ϵ_{i-1}-friend of v must be in $C_2 \cup \cdots \cup C_l$, contradicting the fact that C_1 is a connected component in the subgraph induced by ϵ_{i-1}-dense vertices and ϵ_{i-1}-friend edges. □

Proof of Lemma 2.4. Recall that $v \in W^{\mathsf{S}}_k$ lies in stratum k and that by assumption, $|N(v) \cap V^\star| \geq \Delta/3$. Suppose stratum k spans layers $[i_0, i_1]$ and let $v \in B$ where B is a layer-i small block, $i \in [i_0, i_1]$. We put the neighbors of v into one of several groups.

(1) Neighbors in $W_1 \cup \cdots \cup W_{k-1} \cup V_{\mathsf{sp}}$.
(2) The remaining neighbors in blocks that are neither in ancestors nor descendants of B in $\mathcal{T}$.
(3) Neighbors in all ancestors of B and those stratum-k descendants of B.

Define

$$A_1 = |N(v) \cap (W_1 \cup \cdots \cup W_{k-1} \cup V_{\mathsf{sp}})|.$$

If $A_1 \geq \Delta/2\log^2(1/\xi_k)$ then the conclusion of the lemma already holds, so assume otherwise. Let A_2 be the number of neighbors in blocks that are neither in ancestors nor descendants of v. By Lemma 2.1, $A_2 \le \sum_{j=1}^{\ell} \epsilon_j \Delta \le 2\epsilon_\ell \Delta$.

We now turn to A_3. Let C_{i_1} be the ϵ_{i_1}-almost clique containing B. It follows from Lemma 2.5 that there is some index $i^\star \in [i_0, i_1]$ such that

(i) $C_{i_1} \supseteq C_{i_1-1} \supseteq \cdots \supseteq C_{i^\star}$.
(ii) Each C_j, $j \in [i^\star, i_1 - 1]$, is an ϵ_j-almost clique and is the *only* such almost clique contained in C_{j+1}.
(iii) Either $i^\star = i_0$ or $C_{i^\star}$ contains at least two $\epsilon_{i^\star-1}$-almost cliques.

If $i^\star \ne i_0$ then, by Lemma 2.5 again, the number of vertices in $C_{i^\star}$ that are in $\epsilon_{i^\star-1}$-almost cliques is at most $7\epsilon_{i^\star}\Delta$. Define $B_{i^\star}$ to be the block contained in $C_{i^\star}$ and let $\mathcal{B}$ be the set of all blocks that are ancestors of $B_{i^\star}$ in $\mathcal{T}$.

We first entertain the possibility that all blocks in $\mathcal{B}$ are small. Note that since $\mathcal{B}$ spans many strata, the definition of *small* is different for each stratum. It follows from Definition 2.2 that there are fewer than $\log\log(1/\xi_{k'-1})$ layers in stratum k'. Thus, the maximum number of neighbors that v has in $\mathcal{B}$ is

$$\begin{aligned}&\sum_{k'=k}^{s} \frac{\Delta}{\log^2(1/\xi_{k'})} \cdot \log\log(1/\xi_{k'-1}) \\ &= \sum_{k'=k}^{s} \frac{\Delta}{\log(1/\xi_{k'})} \\ &\le 2\Delta/\log(1/\xi_s) \ \le\ 2\Delta/\log\log(1/\epsilon_\ell)\end{aligned}$$

In this case, the the number of neighbors contributed by group 3 is at most $A_3 = 7\epsilon_{i^\star}\Delta + 2\Delta/\log\log(1/\epsilon_\ell)$. Thus, the number of v's neighbors in $V^\star$ is at most

$$\begin{aligned}&A_1 + A_2 + A_3 \\ &\le \Delta/\log^2(1/\xi_k) + 2\epsilon_\ell\Delta + 7\epsilon_{i^\star}\Delta + 2\Delta/\log\log(1/\epsilon_\ell) \\ &\ll \Delta/3,\end{aligned}$$

contradicting the assumption of the lemma. (Recall that $\epsilon_\ell < 1/K$ for some sufficiently large constant K.) Thus, there must exist some stratum-k' block $B' \in \mathcal{B}$ containing at least $\Delta/\log^2(1/\xi_{k'})$ neighbors of v. According to Definition 2.3, this implies that either B' or a strict ancestor of B' is large. Let B'' be the (unique) large ancestor of B', and suppose it is in layer i'' and stratum k''. According to Lemma 2.1, the number of neighbors of v in B'' is at least

$$\begin{aligned}|B''| - 3\epsilon_{i''}\Delta &\ge \frac{\Delta}{\log^2(1/\xi_{k''})} - 3\epsilon_{i''}\Delta \\ &\ge \frac{\Delta}{\log^2(1/\xi_{k''})} - 3\xi_{k''}\Delta \\ &\ge \frac{\Delta}{2\log^2(1/\xi_{k''})} \\ &\ge \frac{\Delta}{2\log^2(1/\xi_k)}.\end{aligned}$$

Thus $N(v) \cap W^{\mathsf{L}}_{k''} \ge \Delta/2\log^2(1/\xi_k)$. □

3 MAIN ALGORITHM

Our algorithm follows the *graph shattering* framework [5]. In each step of the algorithm, we specify an invariant that all vertices must satisfy in order to continue to participate. Those *bad vertices* that violate the invariant are removed from consideration; they form connected components of size $\operatorname{poly}\log n$ w.h.p., so we can color them later in $\mathsf{Det}_d(\operatorname{poly}\log n)$ time. More precisely, the emergence of the small components is due to the following lemma [5, 13].

Lemma 3.1 (The Shattering Lemma). *Consider a randomized procedure that generates a subset of vertices $B \subseteq V$. Suppose that for each $v \in V$, we have $\Pr[v \in B] \le \Delta^{-(2c+1)}$, and this holds even if the random bits not in $N^c(v)$ are determined adversarially. Then, with probability at least $1 - n^{-c'}$, each connected component in the graph induced by B has size at most $O(c'\Delta^{2c}\log n)$.*

Since our algorithm consists of $t = O(\log^* \Delta)$ steps, whether a vertex v is bad actually depends on random bits in its distance-t neighborhood. Nonetheless, we are still able to apply Lemma 3.1. The reason is that we are able to show that each vertex v becomes bad in one particular step with probability at most Δ^{-x} (for any specified constant x), and this is true regardless of the outcomes in all previous steps and the choices of random bits outside of a constant-radius of v.

The sparsity sequence for our algorithm is defined by $\epsilon_1 = \Delta^{-1/10}$, $\epsilon_i = \sqrt{\epsilon_{i-1}}$ for $i > 1$, and ℓ is the largest index such that $\frac{1}{\epsilon_\ell} \ge K$ for some sufficiently large K.

3.1 Initial Coloring Step

At any point in time, the number of *excess colors* at v is the size of v's remaining palette minus the number of v's uncolored neighbors. This quantity is obviously non-decreasing over time. We first show that in $O(1)$ time, we can color a portion of the vertices such that each remaining uncolored vertex has a certain number of excess colors, which depends on its local sparsity. Refer to Section 4 for proof.

Lemma 3.2. *There is an $O(1)$-round algorithm that colors a subset of vertices such that each ϵ-sparse vertex v with $\deg(v) \ge 0.9\Delta$ satisfies the following conditions.*

- *With probability $1 - \exp(-\Omega(\Delta))$, the number of uncolored neighbors of v is at least $\Delta/2$.*
- *With probability $1 - \exp(-\Omega(\epsilon^2\Delta))$, v has at least $\Omega(\epsilon^2\Delta)$ excess colors.*

We execute the algorithm of Lemma 3.2. In order to proceed a vertex must satisfy both of the following conditions: (i) if v is ϵ_ℓ-dense, the number of uncolored neighbors of v is at least $\Delta/2$; (ii) if v is ϵ_i-dense but ϵ_{i-1}-sparse, v must have $\Omega(\epsilon_{i-1}^2\Delta)$ excess colors. If either fails to hold, v is put in the set V_{bad}.

Define $V^\star$ to be the set of uncolored vertices that are not in V_{bad}. We compute the partition $V^\star = W_1^{\mathsf{S}} \cup \cdots \cup W_s^{\mathsf{S}} \cup W_1^{\mathsf{L}} \cup \cdots \cup W_s^{\mathsf{L}} \cup V_{\mathsf{sp}}$. Notice that we invoke the conditions of Lemma 3.2 only with $\epsilon \ge \epsilon_1 = \Delta^{-1/10}$. Thus, if $\Delta = \Omega(\log^2 n)$, then with high probability (i.e., $1 - 1/\operatorname{poly}(n)$), $V_{\mathsf{bad}} = \emptyset$. Otherwise, each component of V_{bad} must, by Lemma 3.1, have size $O(\operatorname{poly}(\Delta) \cdot \log n) = O(\operatorname{poly}\log n)$, w.h.p. We do not invoke a deterministic algorithm to color V_{bad} just yet. In subsequent steps of the algorithm, we continue to add "bad vertices" to V_{bad}. These vertices are colored at the end of the algorithm.

3.2 Coloring Vertices by Stratum

In this section, we show how we can color most of the vertices in $W_1, \ldots, W_s$, leaving a small portion of uncolored vertices U, each having a large number of excess colors. We do the coloring in $s+2$ stages in this order (i) $s-1$ stages of small blocks: $W_s^{\mathsf{S}}, \ldots, W_2^{\mathsf{S}}$, (ii) first layer small blocks W_1^{S}, (iii) large blocks $W' = W_2^{\mathsf{L}} \cup \cdots \cup W_h^{\mathsf{L}}$, and (iv) first layer large blocks W_1^{L}.

Notice that due to Lemma 2.4, at the time we begin to process W_k^{S}, each vertex $v \in W_k^{\mathsf{S}}$ must have at least $\Delta/2\log^2(1/\xi_k)$ *excess colors w.r.t.* W_k^{S}. That is, its palette size minus the number of its neighbors in W_k^{S} is large. If the condition $|N(v) \cap V^\star| \geq \Delta/3$ in Lemma 2.4 is not met, then it means that at least $(\Delta/2 - \Delta/3)$ neighbors of v were included in V_{bad} after the initial coloring step, and so v automatically has at least $\Delta/6 > \Delta/2\log^2(1/\xi_k)$ excess colors w.r.t. W_k^{S}. Refer to Section 6 for proofs of Lemmas 3.3–3.6.

Lemma 3.3 (Small blocks; strata other than 1). *Suppose that each vertex $v \in W_k^{\mathsf{S}}$ has at least $\Delta/2\log^2(1/\xi_k)$ excess colors w.r.t. W_k^{S}. There is an $O(1)$-time algorithm that colors a subset of W_k^{S} meeting the following condition. For each $v \in V^\star$ and each layer i in stratum k, with probability at least $1-\exp(-\Omega(\text{poly}(\Delta)))$, the number of uncolored layer-i neighbors of v in V_i^{S} is at most $\epsilon_i^5\Delta$. Vertices that violate this property join the set V_{bad}.*

Lemma 3.4 (Small blocks; stratum 1). *Suppose that each vertex $v \in W_1^{\mathsf{S}}$ has at least $\Delta/2\log^2(1/\xi_1)$ excess colors w.r.t. W_1^{S}. There is an $O(1)$-time algorithm that colors a subset of W_1^{S} meeting the following conditions, for any specified constant c. If $\Delta = O(\log^4 n)$, then each $v \in W_1^{\mathsf{S}}$ is colored with probability at least $1-\Delta^{-c}$, and all uncolored vertices in W_1^{S} joins V_{bad}. If $\Delta = \Omega(\log^4 n)$, then, with probability at at least $1-n^{-c}$, the remaining uncolored vertices of W_1^{S} are partitioned into 2 sets X and R such that (i) the subgraph induced by R has maximum degree $O(1)$, (ii) each connected component in the graph induced by X has size at most* poly log n.

Lemma 3.5 (Large blocks; strata other than 1). *There is an $O(1)$-time algorithm that colors a subset of W' meeting the following condition. For each $v \in V^\star$ and each layer $i \in [2,\ell]$, with probability at least $1-\exp(-\Omega(\text{poly}(\Delta)))$, the number of uncolored layer-i neighbors of v in V_i^{L} is at most $\epsilon_i^5\Delta$. Vertices that violate this property join the set V_{bad}.*

Lemma 3.6 (Large blocks; stratum 1). *Let α be a sufficiently large constant, and let c be any constant. There is an $O(1)$-time algorithm that colors a subset of W_1^{L} and puts the remaining uncolored vertices in one of X_1, X_2, R or V_{bad}. It is required that the subgraph induced by R has constant degree, and every component in the subgraph induced by X_1 and the subgraph induced by X_2 has size at most* poly log n. *If $\Delta \leq \log^\alpha n$, then $X_1 = X_2 = \emptyset$, and each $v \in W_1^{\mathsf{L}}$ is added to V_{bad} with probability at most Δ^{-c}. If $\Delta \geq \log^\alpha n$, with probability $1-1/n^{-c}$, no vertex in W_1^{L} is added to V_{bad}.*

We apply Lemmas 3.3–3.6 to color the vertices in $V^\star \setminus V_{\text{sp}}$. The subgraph induced by R (Lemma 3.4 and Lemma 3.6) has constant degree. We immediately color these vertices using any $O(\log^* n)$-time algorithm. All vertices in X (Lemma 3.4), X_1, or X_2 (Lemma 3.6) are colored in time $\text{Det}_d(\text{poly}\log n)$ using a deterministic algorithm. *The vertices in X, X_1, X_2 do not join V_{bad}.*

Any vertex in $V^\star$ that violates at least one condition specified in the lemmas is added to the set V_{bad}. All remaining uncolored vertices join the set U. In other words, U is the set of all vertices in $V^\star \setminus (V_{\text{sp}} \cup X \cup X_1 \cup X_2 \cup V_{\text{bad}})$ that remain uncolored after applying Lemmas 3.3–3.6.

3.3 Coloring the Remaining Vertices

At this point all uncolored vertices are in $U \cup V_{\text{sp}} \cup V_{\text{bad}}$. We show that $U \cup V_{\text{sp}}$ can be colored efficiently in $O(\log^* \Delta)$ time.

We first consider the set U. Let G' be the directed acyclic graph induced by U, where all edges are oriented from the sparser to the denser endpoint. In particular, an edge $e = \{u, u'\}$ is oriented as (u, u') if u is at layer i, u' at layer i', and $i > i'$, or if u and u' are at the same layer i and $\text{ID}(u) > \text{ID}(u')$. We write $N_{\text{out}}(v)$ to denote the set of out-neighbors of v.

For each layer-i vertex v in G', and each layer-j, the number of layer-j neighbors of v in G' is at most $O(\epsilon_j^5\Delta)$, due to Lemmas 3.3 and 3.5. The out-degree of v is therefore at most $\sum_{j=1}^{i}\epsilon_j^5\Delta = O(\epsilon_i^5\Delta) = O(\epsilon_{i-1}^{2.5}\Delta)$. The number of excess colors at v is at least $\Omega(\epsilon_{i-1}^2\Delta)$. Thus, there is an $\Omega(1/\sqrt{\epsilon_{i-1}})$-factor gap between the palette size of v and the out-degree of v.

We write $\Psi(v)$ to denote the set of available colors of v. There exists a constant $\eta > 0$ such that, for each $i \in [2,\ell]$ and each layer-i vertex v in G', we have $|\Psi(v)| - \text{outdeg}(v) \geq \eta\epsilon_{i-1}^2\Delta \stackrel{\text{def}}{=} p_v$. There is a constant $C > 0$ such that for each $i \in [2,\ell]$ and each layer-i vertex $v \in U$ satisfies

$$\sum_{u \in N_{\text{out}}(v)} 1/p_u \leq \sum_{j=1}^{i} O\left(\frac{\epsilon_{j-1}^{2.5}\Delta}{\epsilon_{j-1}^2\Delta}\right) = \sum_{j=1}^{i} O(\epsilon_{j-1}^{0.5}) < 1/C.$$

Lemma 3.7 is applied to color nearly all vertices in U in $O(\log^* \Delta)$ time, with any remaining uncolored vertices added to V_{bad}. Notice that in our setting, the parameters of Lemma 3.7 are $p^\star \geq \eta\epsilon_1^2\Delta = \Omega(\Delta^{8/10})$ and $d^\star \leq \Delta$. Thus, the probability that a vertex is bad is $\exp(-\Omega(\sqrt{p^\star}))+d^\star\exp(-\Omega(p^\star)) = \exp(-\Omega(\Delta^{2/5}))$ (by Lemma 3.7). Refer to Section 5 for proof of Lemma 3.7.

Lemma 3.7. *Consider a directed acyclic graph, where vertex v is associated with a parameter $p_v \leq |\Psi(v)| - \text{outdeg}(v)$. We write $p^\star = \min_{v \in V} p_v$. Suppose that there is a constant $C > 0$ such that all vertices v satisfy $\sum_{u \in N_{out}(v)} 1/p_u \leq 1/C$. Let $d^\star$ be the maximum out-degree of the graph. There is an $O(\log^*(p^\star))$-time algorithm achieving the following. Each vertex v remains uncolored with probability at most $\exp(-\Omega(\sqrt{p^\star})) + d^\star\exp(-\Omega(p^\star))$. This is true even if the random bits generated outside a constant radius around v are determined adversarially.*

The set V_{sp} can be colored in a similar way using the above lemma. We let G'' be any acyclic orientation of the graph induced by V_{sp} (e.g., orienting each edge $\{u, v\}$ to the vertex v such that $\text{ID}(v) > \text{ID}(u)$). The number of available colors of each $v \in V_{\text{sp}}$ minus its out-degree is at least $\Omega(\epsilon_\ell^2\Delta)$, which is at least $\gamma\Delta$, for some constant $\gamma > 0$ (according to the way we select the sparsity sequence). We define $p_v = \gamma\Delta < |\Psi(v)| - \text{outdeg}(v)$. We have $\sum_{u \in N_{\text{out}}(v)}(1/p_u) \leq \text{outdeg}(v)/(\gamma\Delta) \leq 1/\gamma$. Thus, we can apply Lemma 3.7 with $C = \gamma$. Notice that both $p^\star$ and $d^\star$ are $\Theta(\Delta)$, and so the probability that a vertex is bad is $\exp(-\Omega(\sqrt{\Delta}))$.

We add all remaining uncolored vertices in $V_{\mathrm{sp}} \cup U$ to V_{bad}. We are now ready to color V_{bad}. If $\Delta \geq \log^{\alpha} n$, then $V_{\mathrm{bad}} = \emptyset$, w.h.p., in view of the probabilities stated in Lemmas 3.3–3.7. Otherwise, $\Delta \leq \log^{\alpha} n$, and by Lemma 3.1, each connected component of V_{bad} has size at most $\mathrm{poly}(\Delta) \cdot \mathrm{poly} \log n = \mathrm{poly} \log n$. Thus, it takes $\mathrm{Det}_d(\mathrm{poly} \log n)$ to color all bad vertices V_{bad}.

3.4 Time Complexity

The time for the initial coloring step is $O(1)$. The time for processing each of $W_s^{\mathrm{S}}, \ldots, W_2^{\mathrm{S}}, W_1^{\mathrm{S}}, W', W_1^{\mathrm{L}}$ is also $O(1)$, or $O(s) = O(\log^* \Delta)$ in total. The time to color the vertices of $U \cup V_{\mathrm{sp}}$ not marked *bad* is $O(\log^* \Delta)$. In addition, we invoke $O(1)$ times

(i) an $O(\log^* n)$-time algorithm for coloring a bounded degree graph (i.e., the set R in Lemma 3.4 and Lemma 3.6), and

(ii) a $\mathrm{Det}_d(\mathrm{poly} \log n)$-time algorithm for coloring components of size $\mathrm{poly} \log(n)$ (i.e., the sets X, X_1, and X_2 in Lemma 3.4 and Lemma 3.6, and the bad vertices V_{bad}).

Thus, the total time complexity is $O(\log^* n + \mathrm{Det}_d(\mathrm{poly} \log n))$.

Theorem 3.8. *There is an algorithm that computes a $(\Delta + 1)$-list coloring, with high probability, in $O(\log^* n + \mathrm{Det}_d(\mathrm{poly} \log n))$ time.*

There is a universal constant c such that the size of each connected component of V_{bad}, X, X_1, and X_2 is at most $\log^c n$, w.h.p. If each vertex is allowed to have $\log^c n$ extra colors, then we can invoke the $O(\log^* \Delta)$-time algorithm of Lemma 3.7 to color them (rather than spending $\mathrm{Det}_d(\mathrm{poly} \log n)$ time), thereby improving the time complexity greatly. Notice that if every vertex is ϵ-sparse, with $\epsilon^2 \Delta$ sufficiently large, then the algorithm of Lemma 3.2 gives *every* vertex $\Omega(\epsilon^2 \Delta)$ excess colors, w.h.p. Thus, we have the following theorems.

Theorem 3.9. *There is a universal constant c such that there is a randomized algorithm that, w.h.p., computes a $(\Delta + \log^c n)$-list coloring in $O(\log^* n)$ time.*

Theorem 3.10. *There is a universal constant c such that the following holds. Suppose each vertex is ϵ-sparse, and $\epsilon^2 \Delta = \log^c n$. There is a randomized algorithm that, w.h.p., computes a $(\Delta + 1)$-list coloring in $O(\log^* n)$ time.*

Remark. Notice that Theorem 3.10 insists on every vertex being ϵ-sparse, as defined in Section 2. It is straightforward to show connections between this definition of sparsity and others standard measures from the literature. For example, such a graph is $(1 - \epsilon')$-*locally sparse* (according to the definition of [12]), where $\epsilon' = \Omega(\epsilon^2)$. Similarly, any $(1 - \epsilon')$-locally sparse graph is $\Omega(\epsilon')$-sparse. Graphs of *degeneracy* $d \leq (1 - \epsilon')\Delta$ or *arboricity* $\lambda \leq (1/2 - \epsilon')\Delta$ are trivially $(1 - \Omega(\epsilon'))$-locally sparse [5].

4 ANALYSIS OF ONESHOTCOLORING — PROOF OF LEMMA 3.2

Fix a constant parameter $p \in (0, 1/4)$. The procedure OneShotColoring is a simple $O(1)$-round coloring procedure that breaks ties by ID. Define $N^*(v) = \{u \in N(v) \mid \mathrm{ID}(u) < \mathrm{ID}(v)\}$ to be the neighbors of v with higher priority than v. We assume that each vertex v is associated with a palette $\Psi(v)$ of size $\Delta + 1$, and this is used implicitly in the proofs of the lemmas in this section.

The procedure OneShotColoring is as follows.

1. Each uncolored vertex v decides to participate independently with probability p.
2. Each participating vertex v selects a color $c(v)$ from its palette $\Psi(v)$ uniformly at random.
3. A participating vertex v successfully colors itself if $c(v)$ is not chosen by any vertex in $N^*(v)$.

After OneShotColoring, each vertex v removes all colors from $\Psi(v)$ that are taken by some neighbor $u \in N(v)$. The number of *excess colors* at v is the size of v's remaining palette minus the number of uncolored neighbors of v. We prove one part of Lemma 3.2 by showing that after a call to OneShotColoring, the number of excess colors at any ϵ-sparse v is $\Omega(\epsilon^2 \Delta)$, with probability $1 - \exp(-\Omega(\epsilon^2 \Delta))$. Similar but (slightly) weaker lemmas were proved in [12, 19]. The corresponding lemma from [12] does not apply to *list* coloring, and the corresponding lemma from [19] obtains a high probability bound only if $\epsilon^4 \Delta = \Omega(\log n)$. Optimizing this requirement is of importance, since this is the threshold about how locally sparse a vertex needs to be in order to obtain excess colors from OneShotColoring. The remainder of this section constitutes a proof of Lemma 3.2.

Consider an execution of OneShotColoring with any constant $p \in (0, 1/4)$. Recall that we assume $1/\epsilon \geq K$, for some large enough constant K. Let v be an ϵ-sparse vertex. Define the following two numbers.

$f_1(v)$: the number of vertices $u \in N(v)$ that successfully color themselves by some $c \notin \Psi(v)$.

$f_2(v)$: the number of colors $c \in \Psi(v)$ such that at least two vertices in $N(v)$ successfully color themselves c.

It is clear that $f_1(v) + f_2(v)$ is a lower bound on the number of excess colors at v after OneShotColoring. Our first goal is to show that $f_1(v) + f_2(v) = \Omega(\epsilon^2 \Delta)$ with probability at least $1 - \exp(-\Omega(\epsilon^2 \Delta))$. We divide the analysis into two cases (Lemma 4.3 and Lemma 4.4), depending on whether $f_1(v)$ or $f_2(v)$ is likely to be the dominant term. For any v, the preconditions of either Lemma 4.3 or Lemma 4.4 are satisfied. Our second goal is to show that for each vertex v of degree at least 0.9Δ, with high probability, at least $(1 - 1.1p)|N(v)| > \Delta/2$ neighbors of v remain uncolored after after OneShotColoring. This is done in Lemma 4.5.

Lemmas 4.1 and 4.2 establish some generally useful facts of OneShotColoring, which are used in the proofs of Lemma 4.3 and 4.4.

Lemma 4.1. *Let Q be any set of colors, and let S be any set of vertices with size at most 2Δ. The number of colors in Q that are selected by some vertices in S is less than $|Q|/2$ with probability at least $1 - \exp(-\Omega(|Q|))$.*

Proof. Let E_c denote the event that color c is selected by at least one vertex in S. Then $\Pr[E_c] \leq \frac{p|S|}{\Delta+1} < 2p < 1/2$, since $p < 1/4$ and $|S| \leq 2\Delta$. Moreover, the collection of events $\{E_c\}$ are negatively correlated [11].

Let X denote the number of colors in Q that are selected by some vertices in S. By linearity of expectation, $\mu = \mathrm{E}[X] < 2p \cdot |Q|$. We apply a Chernoff bound with $\delta = \frac{(1/2) - 2p}{2p}$. Recall that $0 < p < 1/4$, and so $\delta > 0$. For the case of $\delta \in [0, 1]$, we have:

$$\Pr[X \geq (1 + \delta)\mu = |Q|/2] \leq \exp(-\delta^2 \mu / 3) = \exp(-\Omega(|Q|)).$$

Similarly, if $\delta > 1$, we still have:

$$\Pr[X \geq (1+\delta)\mu = |Q|/2] \leq \exp(-\delta\mu/3) = \exp(-\Omega(|Q|)). \quad \square$$

Lemma 4.2. *Fix a sufficiently small $\epsilon > 0$. Consider a set of vertices $S = \{u_1, \ldots, u_k\}$ with cardinality $\epsilon\Delta/2$. Let Q be a set of colors such that each $u_i \in S$ satisfies $|\Psi(u_i) \cap Q| \geq (1-\epsilon/2)(\Delta+1)$. Moreover, each $u_i \in S$ is associated with a vertex set R_i such that (i) $S \cap R_i = \emptyset$, and (ii) $|R_i| \leq 2\Delta$. Then, with probability at least $1 - \exp(-\Omega(\epsilon^2\Delta))$, there are at least $p\epsilon\Delta/8$ vertices $u_i \in S$ such that the color c selected by u_i satisfies (i) $c \in Q$, and (ii) c is not selected by any vertex in $R_i \cup S \setminus \{u_i\}$.*

Proof. Define $Q_i = \Psi(u_i) \cap Q$. We call a vertex u_i *happy* if u_i selects some color $c \in Q$ and c is not selected by any vertex in $R_i \cup S \setminus \{u_i\}$. Define the following events.

- E_i^{good}: u_i selects a color $c \in Q_i$ such that c is not selected by any vertices in R_i.
- E_i^{bad}: the number of colors in Q_i that are selected by some vertices in R_i is at least $|Q_i|/2$.
- E_i^{repeat}: the color selected by u_i is also selected by some vertices in $\{u_1, \ldots, u_{i-1}\}$.

Let X_i be the indicator random variable that *either* E_i^{good} or E_i^{bad} occurs, and let $X = \sum_{i=1}^k X_i$. Let Y_i be the indicator random variable that E_i^{repeat} occurs, and let $Y = \sum_{i=1}^k Y_i$. *Assuming that E_i^{bad} does not occur* for each $i \in [1,k]$, it follows that $X - 2Y$ is a lower bound on the number of happy vertices. Notice that by Lemma 4.1, $\Pr[E_i^{\text{bad}}] = \exp(-\Omega(|Q_i|)) = \exp(-\Omega(\Delta))$. Thus, assuming that no E_i^{bad} occurs merely distorts our probability estimates by a negligible $\exp(-\Omega(\Delta))$. We prove concentration bounds on X and Y, which together imply the lemma.

We show that $X \geq p\epsilon\Delta/7$ with probability $1 - \exp(-\Omega(\epsilon\Delta))$. It is clear that

$$\Pr[X_i = 1] \geq \Pr[E_i^{\text{good}} \mid \overline{E_i^{\text{bad}}}] \geq \frac{p \cdot |Q_i|/2}{\Delta+1} \geq \frac{p(1-\epsilon/2)}{2} > \frac{p}{3}.$$

Moreover, since $\Pr[X_i = 1 \mid E_i^{\text{bad}}] = 1$, the above inequality also holds, when conditioned on *any* colors selected by vertices in R_i. Thus, $\Pr[X \leq t]$ is upper bounded by $\Pr[\text{Binomial}(n', p') \leq t]$ with $n' = |S| = \epsilon\Delta/2$ and $p' = \frac{p}{3}$. We set $t = p\epsilon\Delta/7$. Notice that $n'p' = p\epsilon\Delta/6 > t$. Thus, according to a tail bound of binomial distribution, $\Pr[X \leq t] \leq \exp(\frac{-(n'p'-t)^2}{2n'p'}) = \exp(-\Omega(\epsilon\Delta))$.

We show that $Y \leq p\epsilon^2\Delta/2$ with probability $1 - \exp(-\Omega(\epsilon^2\Delta))$. It is clear that $\Pr[Y_i = 1] \leq \frac{p(i-1)}{\Delta+1} \leq \frac{p\epsilon}{2}$, even if we condition on arbitrary colors selected by vertices in $\{u_1, \ldots, u_{i-1}\}$. We have $\mu = \mathrm{E}[Y] \leq \frac{p\epsilon}{2} \cdot |S| = \frac{p\epsilon^2\Delta}{4}$. Thus, by a Chernoff bound (with $\delta = 1$), $\Pr[Y \geq p\epsilon^2\Delta/2] \leq \Pr[Y \geq (1+\delta)\mu] \leq \exp(-\delta^2\mu/3) = \exp(-\Omega(\epsilon^2\Delta))$.

To summarize, with probability at least $1 - \exp(-\Omega(\epsilon^2\Delta))$, we have $X - 2Y \geq p\epsilon\Delta/7 - 2p\epsilon^2\Delta/2 > p\epsilon\Delta/8$. $\square$

Lemma 4.3 considers the case when a large fraction of v's neighbors are likely to color themselves with colors outside the palette of v, and therefore be counted by $f_1(v)$. This lemma holds regardless of whether v is ϵ-sparse or not.

Lemma 4.3. *Suppose that there is a subset $S \subseteq N(v)$ such that $|S| = \epsilon\Delta/5$, and for each $u \in S$, $|\Psi(u) \setminus \Psi(v)| \geq \epsilon(\Delta+1)/5$. Then $f_1(v) \geq \frac{p\epsilon^2\Delta}{100}$ with probability at least $1 - \exp(-\Omega(\epsilon^2\Delta))$.*

Proof. Let $S = (u_1, \ldots, u_k)$ be sorted in increasing order by ID. Define $R_i = N^*(u_i)$, and $Q_i = \Psi(u_i) \setminus \Psi(v)$. Notice that $|Q_i| \geq \epsilon\Delta/5$. Define the following events.

- E_i^{good}: u_i selects a color $c \in Q_i$ and c is not selected by any vertex in R_i.
- E_i^{bad}: the number of colors in Q_i that are selected by vertices in R_i is more than $|Q_i|/2$.

Let X_i be the indicator random variable that either E_i^{good} or E_i^{bad} occurs, and let $X = \sum_{i=1}^k X_i$. Given that the events E_i^{bad} for all $i \in [1,k]$ do not occur, we have $X \leq f_1(v)$, since if E_i^{good} occurs, then u_i successfully colors itself by some color $c \notin \Psi(v)$. By Lemma 4.1, $\Pr[E_i^{\text{bad}}] = \exp(-\Omega(|Q_i|)) = \exp(-\Omega(\epsilon\Delta))$. Thus, up to this negligible error, we can assume that E_i^{bad} does not occur, for each $i \in [1,k]$.

We show that $X \geq \epsilon^2\Delta/100$ with probability $1 - \exp(-\Omega(\epsilon^2\Delta))$. It is clear that $\Pr[X_i = 1] \geq \Pr[E_i^{\text{good}} \mid \overline{E_i^{\text{bad}}}] \geq \frac{p|Q_i|/2}{\Delta+1} \geq \frac{p\epsilon}{10}$, and this inequality holds even when conditioning on *any* colors selected by vertices in R_i and $\bigcup_{1 \leq j < i} R_j \cup \{u_j\}$ (since $S = (u_1, \ldots, u_k)$ is sorted in increasing order by ID, $u_i \notin R_j = N^*(u_j)$ for any $1 \leq j < i$). Thus, $\Pr[X \geq t]$ is upper bounded by $\Pr[\text{Binomial}(n', p') \leq t]$ with $n' = |S| = \epsilon\Delta/5$ and $p' = \frac{p\epsilon}{10}$. We set $t = \frac{n'p'}{2} = \frac{p\epsilon^2\Delta}{100}$. Thus, according to a lower tail of the binomial distribution, $\Pr[X \leq t] \leq \exp(\frac{-(n'p'-t)^2}{2n'p'}) = \exp(-\Omega(\epsilon^2\Delta))$. $\square$

Lemma 4.4 considers the case that many pairs of neighbors of v are likely to color themselves the same color, and contribute to $f_2(v)$. Note that any ϵ-sparse vertex that does not satisfy the preconditions of Lemma 4.3 *does* satisfy the preconditions of Lemma 4.4.

Lemma 4.4. *Let v be an ϵ-sparse vertex. Suppose that there is a subset $S \subseteq N(v)$ such that $|S| \geq (1-\epsilon/5)\Delta$, and for each $u \in S$, $|\Psi(u) \cap \Psi(v)| \geq (1-\epsilon/5)(\Delta+1)$. Then $f_2(v) \geq p^3\epsilon^2\Delta/2000$ with probability $1 - \exp(-\Omega(\epsilon^2\Delta))$.*

Proof. Let S' be any subset of S such that (i) $|S'| = \frac{p\epsilon\Delta}{100}$, (ii) each $u_i \in S'$ is associated with a set $S_i \subseteq S \setminus (S' \cup N(u_i))$ of size $\frac{\epsilon\Delta}{2}$. The existence of $S', S_1, \ldots, S_{|S'|}$ is guaranteed by the ϵ-sparseness of v. In particular, S must contain at least $\epsilon\Delta - \epsilon\Delta/5 > p\epsilon\Delta/100 = |S'|$ *non*-friends of v, and for each such non-friend $u_i \in S'$, $|S \setminus (S' \cup N(u_i))| \geq \Delta(1 - \epsilon/5 - p\epsilon/100 - (1-\epsilon)) > \epsilon\Delta/2$.

Order the set $S' = \{u_1, \ldots, u_k\}$ in increasing order by vertex ID. Define $R_i = \{u_1, \ldots, u_{i-1}\} \cup N^*(u_i)$, and $Q_i = \Psi(u_i) \cap \Psi(v)$. Define Q_i^{good} as the subset of colors $c \in Q_i$ such that c is selected by some vertex $w \in S_i$, but c is not selected by any vertex in $(N^*(w) \cup N^*(u_i)) \setminus S'$. Define the following events.

- E_i^{good}: u_i selects a color $c \in Q_i^{\text{good}}$.
- E_i^{bad}: the number of colors in Q_i^{good} is less than $p\epsilon\Delta/8$.
- E_i^{repeat}: the color selected by u_i is also selected by some vertices in $\{u_1, \ldots, u_{i-1}\}$.

Let X_i be the indicator random variable that either E_i^{good} or E_i^{bad} occurs, and let $X = \sum_{i=1}^{k} X_i$. Let Y_i be the indicator random variable that E_i^{repeat} occurs, and let $Y = \sum_{i=1}^{k} Y_i$. Suppose that E_i^{good} occurs, and the color c selected by u_i is not selected by any vertex in $S \setminus \{u_i\}$. Then there must exist a vertex $w \in S_i$ such that both u_i and w successfully color themselves c. Notice that w and u_i are not adjacent. Thus, $X - Y \le f_2(v)$, given that E_i^{bad} does not occur, for each $i \in [1, k]$. Notice that $\Pr[E_i^{\text{bad}}] = \exp(-\Omega(\epsilon^2\Delta))$ (by Lemma 4.2 and the definition of Q_i^{good}), and thus indeed we can assume that E_i^{bad} does not occur. In what follows, we prove concentration bounds on X and Y, which together imply the lemma.

We show that $X \ge \frac{p^3\epsilon^2\Delta}{1000}$ with probability $1 - \exp(-\Omega(\epsilon^2\Delta))$. It is clear that $\Pr[X_i = 1] \ge p \cdot \frac{p\epsilon\Delta/8}{\Delta+1} > \frac{p^2\epsilon}{8}$, regardless of the colors selected by vertices in R_i. Thus, $\Pr[X \le t]$ is upper bounded by $\Pr[\text{Binomial}(n', p') \le t]$ with $n' = |S'| = \frac{p\epsilon\Delta}{100}$ and $p' = \frac{p^2\epsilon}{8}$. We set $t = \frac{p^3\epsilon^2\Delta}{1000} < n'p'$. According to a tail bound of binomial distribution, $\Pr[X \le t] \le \exp(\frac{-(n'p'-t)^2}{2n'p'}) = \exp(-\Omega(\epsilon^2\Delta))$.

We show that $Y \le \frac{p^3\epsilon^2\Delta}{2000}$ with probability $1 - \exp(-\Omega(\epsilon^2\Delta))$. It is clear that $\Pr[Y_i = 1] \le p \cdot \frac{(i-1)}{\Delta+1} \le \frac{p^2\epsilon}{100}$ regardless of the colors selected by vertices in $\{u_1, \ldots, u_{i-1}\}$. We have $\mu = \mathrm{E}[Y] \le \frac{p^2\epsilon}{100} \cdot |S'| = \frac{p^3\epsilon^2\Delta}{10{,}000}$. Thus, by a Chernoff bound (with $\delta = 4$), $\Pr[Y \ge \frac{p^3\epsilon^2\Delta}{2000}] \le \Pr[Y \ge (1+\delta)\mu] \le \exp(-\delta\mu/3) = \exp(-\Omega(\epsilon^2\Delta))$.

To summarize, with probability at least $1 - \exp(-\Omega(\epsilon^2\Delta))$, we have $X - Y \ge p^3\epsilon^2\Delta/1000 - p^3\epsilon^2\Delta/2000 > p^3\epsilon^2\Delta/2000$. □

Lemma 4.5. *The number of vertices in $N(v)$ that remain uncolored after* OneShotColoring *is at least $(1 - 1.1p)|N(v)|$, with probability at least $1 - \exp(-\Omega(|N(v)|))$.*

Proof. Let X be the number of vertices in $N(v)$ participating in OneShotColoring. It suffices to show that $X \le 1.1p|N(v)|$ with probability $1 - \exp(-\Omega(|N(v)|))$. Since a vertex participates with probability p, we have

$$\Pr[X \ge (1 + 0.1)p|N(v)|] \le \exp(-\frac{(0.1)^2 p|N(v)|}{3}) = \exp(-\Omega(|N(v)|))$$

by Chernoff bound with $\delta = 0.1$. □

5 ANALYSIS OF COLORBIDDING — PROOF OF LEMMA 3.7

Consider a directed acyclic graph $G = (V, E)$, where each vertex v has a palette $\Psi(v)$. Recall that each vertex v is associated with a parameter $p_v \le |\Psi(v)| - \text{outdeg}(v)$, and we write $p^\star = \min_{v \in V} p_v$. The maximum out-degree is denoted as $d^\star$. There is a number $C > 0$ such that all vertices v satisfy $\sum_{u \in N_{\text{out}}(v)} 1/p_u \le 1/C$. Intuitively, the term $\sum_{u \in N_{\text{out}}(v)} 1/p_u$ measures the amount of "contention" at a vertex v (in ColorBidding, u selects each color $c \in \Psi(u)$ with probability $\frac{C}{2p_u}$, which is proportional to $1/p_u$). All vertices agree on the value of C.

The procedure ColorBidding is as follows.

(1) Each color $c \in \Psi(v)$ is added to S_v with probability $\frac{C}{2p_v}$ independently.
(2) If there exists a color $c^\star \in S_v$ that is not selected by any vertex in $N_{\text{out}}(v)$, v colors itself $c^\star$.

In Lemma 5.1 we present an analysis of ColorBidding. We show that after an iteration of ColorBidding, the amount of "contention" at a vertex v decreases by (roughly) an $\exp(C/6)$-factor, with very high probability. See [8, 12, 29] for proofs of similar claims. The main technical difficulty of our setting is that we need to deal with vertices with different out-degrees, and the guarantee of the number of excess colors of a vertex depends on its out-degree (rather than the global parameter Δ), and so we cannot simply use out-degree as the measure of contention.

Lemma 5.1. *Consider an execution of* ColorBidding. *Let v be any vertex. Let d be the summation of $1/p_u$ over all vertices u in $N_{\text{out}}(v)$ that remain uncolored after* ColorBidding. *Then the following holds.*

$$\Pr[\ v \text{ remains uncolored}\ ] \le \exp(-C/6) + \exp(-\Omega(p^\star)).$$

$$\Pr[d \ge (1+\lambda)\exp(-C/6)/C] \le \exp\left(-2\lambda^2 p^\star \exp(-C/6)/C\right) + d^\star \exp(-\Omega(p^\star)).$$

Proof. For each vertex u, we define the following two events.

E_u^{good} : u selects a color that is not selected by any vertex in $N_{\text{out}}(u)$.
E_u^{bad} : number of colors in $\Psi(u)$ that are selected by some vertices in $N_{\text{out}}(u)$ is at least $\frac{2}{3} \cdot |\Psi(u)|$.

Notice that E_u^{good} is the event where u successfully colors itself. We show that $\Pr[E_u^{\text{bad}}] = \exp(-\Omega(p^\star))$. Fix a color $c \in \Psi(u)$. The probability that c is selected by some vertex in $N_{\text{out}}(u)$ is

$$1 - \prod_{w \in N_{\text{out}}(u)} \left(1 - \frac{C}{2p_w}\right) \le \sum_{w \in N_{\text{out}}(u)} \frac{C}{2p_w d} \le \frac{1}{2}.$$

Thus, $\Pr[E_u^{\text{bad}}] \le \Pr[\text{Binomial}(n', p') \ge \frac{2n'}{3}]$ with $n' = |\Psi(u)| \ge p_u$ and $p' = \frac{1}{2}$. By a Chernoff bound, we have:

$$\Pr[E_u^{\text{bad}}] \le \exp(-\Omega(n'p')) = \exp(-\Omega(p^\star)).$$

Conditioned on $\overline{E_u^{\text{bad}}}$, u will color itself unless it fails to choose *any* of $|\Psi(u)|/3$ specific colors from its palette. Thus,

$$\Pr[\overline{E_u^{\text{good}}} \mid \overline{E_u^{\text{bad}}}] \le \left(1 - \frac{C}{2p_u}\right)^{|\Psi(u)|/3} \le \left(1 - \frac{C}{2p_u}\right)^{\frac{p_u}{3}} \le \exp\left(\frac{-C}{6}\right).$$

We are now in a position to prove the first inequality. The probability that v remains uncolored is at most $\Pr[E_i^{\text{bad}}] + \Pr[\overline{E_u^{\text{good}}} \mid \overline{E_u^{\text{bad}}}]$, which is at most $\exp(\frac{-C}{6}) + \exp(-\Omega(p^\star))$.

Next, we prove the second inequality. Let $N_{\text{out}}(v) = (u_1, \ldots, u_k)$. Let E_i^{bad} and E_i^{good} be short for $E_{u_i}^{\text{bad}}$ and $E_{u_i}^{\text{good}}$. By a union bound,

$$\Pr\left[\bigcup_{i=1}^{k} E_i^{\text{bad}}\right] \le \text{outdeg}(v) \cdot \exp(-\Omega(p^\star)) \le d^\star \cdot \exp(-\Omega(p^\star)).$$

Let $X_i = 1/p_{u_i}$ if *either* $\overline{E_i^{\text{good}}}$ or E_i^{bad} occurs, and $X_i = 0$ otherwise. Let $X = \sum_{i=1}^{k} X_i$. Notice that if E_i^{bad} does not occur, for all $i \in [1, k]$, we have $X = d$.

Notice that $\mu \stackrel{\text{def}}{=} \mathrm{E}[X] \le \exp(-C/6)/C$, since $\Pr[\overline{E_i^{\text{good}}} \mid \overline{E_i^{\text{bad}}}] \le \exp(\frac{-C}{6})$. Each variable X_i is within the range $[a_i, b_i]$, where $a_i = 0$ and $b_i = 1/p_{u_i}$. We have $\sum_{i=1}^{k}(b_i - a_i)^2 \le \sum_{u \in N_{\text{out}}(v)} 1/(p_u \cdot p^\star) \le 1/(Cp^\star)$. By Hoeffding's inequality,[5] we have

$$\begin{aligned}\Pr[X \ge (1+\lambda)\exp(-C/6)/C] &\le \Pr[X \ge (1+\lambda)\mu] \\ &\le \exp\left(\frac{-2(\lambda\mu)^2}{\sum_{i=1}^{k}(b_i - a_i)^2}\right) \\ &\le \exp\left(-2(\lambda\exp(-C/6)/C)^2(p^\star C)\right) \\ &= \exp\left(-2\lambda^2 p^\star \exp(-C/6)/C\right).\end{aligned}$$

Thus,

$$\begin{aligned}&\Pr[d \ge (1+\lambda)\exp(-C/6)/C] \\ &\le \exp\left(-2\lambda^2 p^\star \exp(-C/6)/C\right) + d^\star \exp(-\Omega(p^\star)).\end{aligned}$$ □

Proof of Lemma 3.7. In what follows, we show how Lemma 5.1 can be used to derive Lemma 3.7. Our plan is to apply ColorBidding for $O(\log^*(p^\star))$ iterations. For the ith iteration we use the parameter C_i, which is defined as follows: $C_1 = C$, and $C_l = \min\{\sqrt{p^\star}, \frac{C_{l-1}}{(1+\lambda)\exp(\frac{-C_{l-1}}{6})}\}$. Here λ must be selected to be sufficiently small such that $(1+\lambda)\exp(-C_{l-1}/6) < 1$ so the sequence increases. For example, if $C \ge 6$ initially, we can fix $\lambda = 1$ throughout.

In each iteration each vertex v use the same parameter p_v, since the number of excess colors never decrease. The last iteration $l^\star = O(\log^*(p^\star))$ is the minimum index l such that $C_l = \sqrt{p^\star}$.

At the end of the lth iteration ($1 \le l \le l^\star$), we have the following invariants $\mathcal{H}_l$ that we expect all vertices to satisfy: If $1 \le l < l^\star$, for each uncolored vertex v after the lth iteration, we require the summation of $1/p_u$ over all uncolored vertices u in $N_{\text{out}}(v)$ to be less than $1/C_{l+1}$; if $l = l^\star$, all vertices are colored at the end of the lth iteration. The purpose of the invariant $\mathcal{H}_l$ ($1 \le l < l^\star$) is to guarantee that the parameter C_{l+1} is a valid parameter for the $(l+1)$th iteration.

We remove from consideration all vertices v violating $\mathcal{H}_l$ at the end of the lth iteration, and add them to the set V_{bad}. Our goal is to show that with probability at most $\exp(-\Omega(\sqrt{p^\star})) + d^\star\exp(-\Omega(p^\star))$, a vertex v is removed, and this is true even if the randomness outside constant distance to v is determined adversarially. By definition of $\mathcal{H}_{l^\star}$, all vertices that are not removed must be colored.

By Lemma 5.1 the probability that a vertex v is removed at the end of the lth iteration, where $1 \le l < l^\star$, is at most

$$\begin{aligned}&\exp(\Omega(p^\star/C_{l+1})) + d^\star\exp(-\Omega(p^\star)) \\ &\le \exp(-\Omega(\sqrt{p^\star})) + d^\star\exp(-\Omega(p^\star))\end{aligned}$$

and the probability that a vertex v is removed at the end of the $l^\star$th iteration is at most $\exp(-C_{l^\star}/6) + \exp(-\Omega(p^\star)) \le \exp(-\Omega(\sqrt{p^\star}))$. □

[5] The variables $\{X_1, \ldots, X_k\}$ are not independent, but we are still able to apply Hoeffding's inequality. The reason is as follows. Assume that $N_{\text{out}}(v) = (u_1, \ldots, u_k)$ is sorted in reverse topological order, and so for each $1 \le j \le k$, we have $N_{\text{out}}(u_j) \cap \{u_j, \ldots, u_k\} = \emptyset$. Thus, conditioning on (i) $\overline{E_i^{\text{bad}}}$ and (ii) *any* colors selected by vertices in $\bigcup_{1 \le j < i} N_{\text{out}}(u_j) \cup \{u_j\}$, the probability that $\overline{E_i^{\text{good}}}$ occurs is still at most $\exp(\frac{-C}{6})$.

6 COLORING ϵ-DENSE VERTICES

Consider the following setting. We are given a graph $G = (V, E)$, where a subset of vertices are already colored. Let S be a *subset* of the uncolored vertices, and suppose S is partitioned into g disjoint sets (clusters) $S_1, \ldots, S_g$, each with weak diameter 2. Our goal is to color a large fraction of the vertices in S in only constant time.

In Section 6.1 we describe a procedure DenseColoringStep (version 1) that is efficient when each vertex has many excess colors w.r.t. S. It is analyzed in Lemma 6.1, which is then used to prove Lemmas 3.3 and 3.4.

For Lemmas 3.5 and 3.6, we have to deal with the case where no excess color is available, and so we need another version of DenseColoringStep. The proof of these two lemmas and the description of DenseColoringStep (version 2) are omitted in the conference proceeding; they can be found in the full version of the paper [7].

6.1 Version 1 of DenseColoringStep — Many Excess Colors are Available

All vertices in S agree on a parameter Z_{ex}, which is a lower bound on the number of excess colors w.r.t. S. That is, for each $v \in S$, the palette size of v minus the number of neighbors of v in S is at least Z_{ex}.

Each vertex $v \in S_j$ is associated with a parameter D_v. We prioritize vertices by D_v-value, breaking ties by ID. Define $N'(v) = \{u \in N(v) \mid D_u < D_v \text{ or } D_u = D_v \text{ and } \mathrm{ID}(u) < \mathrm{ID}(v)\}$ to be the neighbors of v with higher priority. For each $v \in S_j$, we assume that the choice of the parameter D_v satisfies $|N'(v) \cap (S \setminus S_j)| \le D_v$. Define $\delta_v = D_v / Z_{\text{ex}}$.

The procedure DenseColoringStep (version 1) is as follows.

(1) Let $\pi : \{1, \ldots, |S_j|\} \to S_j$ be the permutation that lists S_j in increasing order by D-value, breaking ties by ID. For q from 1 to $|S_j|$, the vertex $\pi(q)$ selects a color $c(\pi(q))$ uniformly at random from $\Psi(\pi(q)) \setminus \{c(\pi(q')) \mid q' < q \text{ and } \{\pi(q), \pi(q')\} \in E(G)\}$.
(2) Each $v \in S_j$ permanently colors itself $c(v)$ if $c(v)$ is not selected by any vertices in $N'(v)$.

Observe that because each S_j has weak diameter 2, Step 1 of DenseColoringStep takes only $O(1)$ rounds of communication. Intuitively, the probability that a vertex $v \in S$ remains uncolored after DenseColoringStep (version 1) is at most δ_v. The following lemma gives us the probabilistic guarantee of the DenseColoringStep (version 1).

Lemma 6.1. *Consider an execution of* DenseColoringStep *(version 1). Let T be any subset of S, and let $\delta = \max_{v \in T} \delta_v$. For any t, the number of uncolored vertices in T is at least t with probability at most* $\Pr[\text{Binomial}(|T|, \delta) \ge t]$.

Proof. Let $T = \{v_1, \ldots, v_{|T|}\}$ be listed by priority: in increasing order by D-value, breaking ties by vertex ID. (Remember that vertices in T can be spread across multiple clusters in S.) Imagine exposing the color choices of all vertices in S, one by one, in order of priority. The vertex v_l will successfully color itself if it chooses any color not already selected by a vertex in $N'(v_l) \cap (S \setminus S_j)$. Since $|N'(v_l) \cap (S \setminus S_j)| \le D_v$ and v_l has at least Z_{ex} colors to choose from, the probability that it fails to be colored is at most $D_v/Z_{\text{ex}} = \delta_v \le \delta$, *independent of the choices made by higher priority vertices.* Thus, for any t, the number of uncolored vertices in T is at least t with probability at most $\Pr[\text{Binomial}(|T|, \delta) \ge t]$. □

Next, we prove Lemmas 3.3 and 3.4. The basic setup of these two proofs are similar. We let $S = W_k^{\mathsf{S}}$ ($k = 1$ for Lemma 3.4), and let $S_1, \ldots, S_g$ be the *super*-blocks constituting S. According to Lemma 2.4 we can set $Z_{\text{ex}} = \Delta/2\log^2(1/\xi_k)$ and according to Lemma 2.1's bound on the external degree we can set $D_v = \epsilon_i\Delta$ if v is a layer-i vertex. Our algorithm consists of several iterations of DenseColoringStep (version 1) on $S = S_1 \cup \cdots \cup S_g$.

6.2 Proof of Lemma 3.3

We execute DenseColoringStep (version 1) for 6 iterations using the same parameters Z_{ex} and D_v for all iterations. Consider any vertex $v \in V^\star$, and a layer i that is within stratum k. Let T be the set of layer-i neighbors of v in S. Then $\delta = \max_{u\in T}\{\delta_u\} = \frac{\epsilon_i\Delta}{Z_{\text{ex}}} = 2\epsilon_i\log^2(1/\xi_k) \le 2\epsilon_i\log^2(1/\epsilon_i)$. Define $t_0 = |T|$, and $t_l = \max\{2\delta t_{l-1}, \epsilon_i^5\Delta\}$. Since $(2\delta)^6|T| \le \epsilon_i^5\Delta$, we have $t_6 = \epsilon_i^5\Delta$.

Assume that at the beginning of the lth iteration, the number of uncolored vertices in T is at most t_{l-1}. By Lemma 6.1 and a Chernoff bound, after the lth iteration, with probability at most $\exp(-\Omega(t_l)) \le \exp(-\Omega(\epsilon_i^5\Delta))$, the number of uncolored vertices in T is more than t_l. Thus, with probability $1 - \exp(-\Omega(\text{poly}(\Delta)))$, after 6 iterations the number of uncolored layer-i neighbors of v in W_k^{S} is at most $\epsilon_i^5\Delta$.

6.3 Proof of Lemma 3.4

Notice that the parameter $\delta_v = D_v/Z_{\text{ex}}$ is always at most

$$2\epsilon_1\log^2(1/\epsilon_1) \ll \Delta^{-1/20}.$$

Thus, we define $\bar{\delta} = \Delta^{-1/20}$ as an upper bound on δ_v. Let x be a number to be determined. Consider the following invariants that all vertices $v \in S$ and all clusters S_j should satisfy after the lth iteration:

> **Invariant $\mathcal{H}_l(v)$:** the number of uncolored vertices of $(N(v) \cap S)$ is at most $\max\{x, \bar{\delta}^l\Delta\}$.
> **Invariant $\mathcal{H}_l(S_j)$:** the number of uncolored vertices of S_j is at most $\max\{x, \bar{\delta}^l\Delta\}$.

Let $l^\star$ be minimum such that $\bar{\delta}^{l^\star}\Delta \le x$. We run DenseColoringStep (version 1) for $l^\star$ iterations. Again, we use the same parameters Z_{ex} and D_v (as defined above). Via Lemma 6.1, it is straightforward to prove the following probabilistic bounds using a Chernoff bound.

$$\Pr[\mathcal{H}_1(v)] = 1 - \exp(-\Omega(\text{poly}\Delta)).$$
$$\Pr[\mathcal{H}_l(v) \mid \mathcal{H}_{l-1}(v)] = 1 - \exp(-\Omega(x)), \text{ for } 1 < l \le l^\star.$$
$$\Pr[\mathcal{H}_1(S_j)] = 1 - \exp(-\Omega(\text{poly}\Delta)).$$
$$\Pr[\mathcal{H}_l(S_j) \mid \mathcal{H}_{l-1}(S_j)] = 1 - \exp(-\Omega(x)), \text{ for } 1 < l \le l^\star.$$

For any $l \in [1, l^\star]$, any uncolored vertex $v \in S_j$ such that $\mathcal{H}_l(v)$ or $\mathcal{H}_l(S_j)$ is violated is removed from further consideration at the end of the lth iteration, and included in V_{bad}. Thus, by the end of the $l^\star$th iteration, we have x as an upper bound on the cluster size and the maximum degree of the remaining uncolored vertices.

Case: $\Delta = O(\log^4 n)$. For this case, we set $x = \Delta^{1/20}$. We do one additional iteration of DenseColoringStep (version 1), aiming to reduce the maximum degree of the uncolored vertices to $O(1)$. For this iteration, we set $D_v = \Delta^{1/20}$, and $Z_{\text{ex}} = \Delta/2\log^2(1/\xi_1) = \Theta(\Delta/\log^2\Delta)$. Thus, we have the shrinking rate

$$\delta_v = O(\Delta^{-19/20}\log^2\Delta).$$

Let $v \in S$ be any vertex. By Lemma 6.1, the probability that there exist at least t uncolored neighbors of v in S is at most

$$\Pr[\text{Binomial}(|T|, \delta) \ge t],$$

where $|T| = \Delta^{1/20}$ and $\delta = O(\Delta^{-19/20}\log^2\Delta)$. Thus,

$$\Pr[\text{Binomial}(|T|, \delta) \ge t] \le |T|^t\delta^t = \Delta^{-\Omega(t)}.$$

We choose $t = \Theta(c) = O(1)$ in such a way that $|T|^t\delta^t \ll \Delta^{-c}$. Let $v \in S$ be an uncolored vertex. If there exist at least t uncolored neighbors of v in S, then we add v to V_{bad}; otherwise, v is added to R. It is clear that the subgraph induced by R has maximum degree $O(1)$.

Case: $\Delta = \Omega(\log^4 n)$. We now turn to the case where $\Delta = \Omega(\log^4 n)$. We set $x = \Theta(\log n)$. Clearly, with high probability (i.e., $1 - 1/\text{poly}(n)$) all invariants $\mathcal{H}_1(v)$ and $\mathcal{H}_1(S_j)$ are met for each iteration. But we still need to reduce the maximum degree from $\Theta(\log n)$ to $O(1)$.

We do DenseColoringStep (version 1) for one extra iteration. This time, for each vertex v, we use the parameter $D_v = D = x = \Theta(\log n)$, and so the shrinking rate is $\delta_v = \delta = D/Z_{\text{ex}} = O\left(\frac{\log n}{\Delta/\log^2\Delta}\right) \ll 1/\log^2 n$, due to the assumption $\Delta = \Omega(\log^4 n)$.

Consider any uncolored vertex v, and let T be the set of uncolored neighbors of v just before this iteration. Notice that $|T| \le x = O(\log n)$. By Lemma 6.1, after this iteration, the number of uncolored vertices in T is at least t with probability at most

$$\Pr[\text{Binomial}(|T|, \delta) \ge t] \le |T|^t\delta^t = (O(1/\log n))^t.$$

After this iteration, we partition the uncolored vertices into two subsets X and R, where X consists of all vertices whose number of uncolored neighbors are at least t. Thus, the subgraph induced by R has maximum degree $O(1)$.

Using Lemma 3.1, we argue that if t is set to be sufficiently large, then vertices in X form connected components of size at most poly $\log n$, with high probability. Consider the graph G' induced by the vertices that are uncolored at the beginning of this iteration, together with additional edges added to G' making (the uncolored vertices of) each cluster a clique. Due to the $O(\log n)$ upper bound on the maximum degree and the cluster size, the maximum degree of G' is also $\Delta' = O(\log n)$. Recall that a vertex v is added to X with probability at most $(O(1/\log n))^t$, and this is true regardless of random bits of vertices outside of a constant radius of v in G'. Thus, if t is a sufficiently large constant, then w.h.p., each connected component of X in G' has size at most $O(\text{poly}(\Delta') \cdot \log n) = \text{poly}\log n$. Thus, w.h.p., each connected component of X (in the original graph G) has size poly $\log n$.

Notice that in the above analysis, to argue that the component size is small, it is crucial that we use Lemma 3.1 w.r.t. a graph whose maximum degree is poly $\log n$, and this is the reason that we define the graph G'. This also explains the reason for having a separate set X (rather than adding all these vertices to V_{bad}). In general, the size of a component in $V_{\text{bad}} \cup X$ could be super-polylogarithmic.

7 CONCLUSION

We have presented a new randomized $(\Delta + 1)$-list coloring algorithm that requires $O(\log^* n + \text{Det}_d(\text{poly} \log n))$ rounds of communication, which comes close to the $\Omega(\log^* n + \text{Det}(\sqrt{\log n}))$ lower bound implied by Naor [26] ($\Omega(\log^* n)$) and Chang, Kopelowitz, and Pettie [8] ($\Omega(\text{Det}(\sqrt{\log n}))$).[6] When Δ is unbounded (relative to n), the best known algorithms for $(\Delta + 1)$- and $(\deg +1)$-list coloring are the same: they use Panconesi and Srinivasan's [27] $2^{O(\sqrt{\log n})}$-time construction of network decompositions. Even if optimal $(O(\log n), O(\log n))$-network decompositions could be computed *for free*, we still do not know how to solve $(\Delta+1)$-list coloring faster than $O(\log^2 n)$ time. Thus, reducing the $\text{Det}_d(\text{poly} \log n)$ term in our running time below $O((\log \log n)^2)$ will require a radically new approach to the problem.

It is an open problem to generalize our algorithm (or that of [19]) to solve the $(\deg +1)$-coloring problem. The main difficulty is to extend the definition of "ϵ-friend" to account for neighbors of different degrees, while still preserving the useful properties of ϵ-dense clusters from Lemma 2.1.

A CONCENTRATION BOUNDS

We make use of the following standard tail bounds [10]. Let X be binomially distributed with parameters (n, p), i.e., it is the sum of n independent 0-1 variables with mean p. We have the following bound on the lower tail of X:

$$\Pr[X \le t] \le \exp\left(\frac{-(\mu - t)^2}{2\mu}\right), \quad \text{where } t < \mu = np.$$

Chernoff bounds also hold when X is the sum of n *negatively correlated* 0-1 random variables [10, 11] with mean p, i.e., total independent is not required. We use a bound on the upper tail of X with mean $\mu = np$.

$$\Pr[X \ge (1 + \delta)\mu] \le \begin{cases} \exp(\frac{-\delta^2\mu}{3}) & \text{if } \delta \in [0, 1] \\ \exp(\frac{-\delta\mu}{3}) & \text{if } \delta > 1. \end{cases}$$

Consider the scenario where $X = \sum_{i=1}^{n} X_i$, and each X_i is an independent random variable bounded by the interval $[a_i, b_i]$. Let $\mu = \mathrm{E}[X]$. Then we have the following concentration bound (Hoeffding's inequality) [20].

$$\Pr[X \ge (1 + \delta)\mu] \le \exp\left(\frac{-2(\delta\mu)^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

[6] Recall that Det and Det_d are the deterministic complexities of $(\Delta + 1)$-list coloring and $(\deg +1)$-list coloring.

REFERENCES

[1] N. Alon, L. Babai, and A. Itai. 1986. A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem. *J. Algor.* 7 (1986), 567–583.

[2] B. Awerbuch, A. V. Goldberg, M. Luby, and S. A. Plotkin. 1989. Network Decomposition and Locality in Distributed Computation. In *Proceedings 30th IEEE Symposium on Foundations of Computer Science (FOCS)*. 364–369.

[3] L. Barenboim. 2015. Deterministic $(\Delta + 1)$-Coloring in Sublinear (in Δ) Time in Static, Dynamic and Faulty Networks. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing (PODC)*. 345–354. https://doi.org/10.1145/2767386.2767410

[4] L. Barenboim, M. Elkin, and F. Kuhn. 2014. Distributed $(\Delta + 1)$-Coloring in Linear (in Δ) Time. *SIAM J. Comput.* 43, 1 (2014), 72–95.

[5] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider. 2016. The Locality of Distributed Symmetry Breaking. *J. ACM* 63, 3, Article 20 (2016), 20:1–20:45 pages.

[6] T. Bisht, K. Kothapalli, and S. V. Pemmaraju. 2014. Brief announcement: Super-fast t-ruling sets. In *Proceedings 33rd ACM Symposium on Principles of Distributed Computing (PODC)*. 379–381.

[7] Y.-J. Chang, W. Li, and S. Pettie. 2017. An Optimal Distributed $(\Delta+1)$-Coloring Algorithm? *CoRR* abs/1711.01361 (2017). arXiv:1711.01361 http://arxiv.org/abs/1711.01361

[8] Y.-J. Chang, T. Kopelowitz, and S. Pettie. 2016. An Exponential Separation Between Randomized and Deterministic Complexity in the LOCAL Model. In *Proceedings 57th IEEE Symposium on Foundations of Computer Science (FOCS)*. 615–624.

[9] Y.-J. Chang and S. Pettie. 2017. A Time Hierarchy Theorem for the LOCAL Model. In *Proceedings 58th IEEE Symposium on Foundations of Computer Science (FOCS)*.

[10] D. P. Dubhashi and A. Panconesi. 2009. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press.

[11] D. P. Dubhashi and D. Ranjan. 1998. Balls and bins: A study in negative dependence. *J. Random Structures and Algs.* 13, 2 (1998), 99–124.

[12] M. Elkin, S. Pettie, and H.-H. Su. 2015. $(2\Delta - 1)$-Edge Coloring is Much Easier than Maximal Matching in the Distributed Setting. In *Proceedings 26th ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 355–370.

[13] M. Fischer and M. Ghaffari. 2017. Sublogarithmic Distributed Algorithms for Lovász Local Lemma with Implications on Complexity Hierarchies. In *Proceedings 31st International Symposium on Distributed Computing (DISC)*. 18:1–18:16.

[14] M. Fischer, M. Ghaffari, and F. Kuhn. 2017. Deterministic Distributed Edge Coloring via Hypergraph Maximal Matching. In *Proceedings 58th IEEE Symposium on Foundations of Computer Science (FOCS)*.

[15] P. Fraigniaud, M. Heinrich, and A. Kosowski. 2016. Local conflict coloring. In *Proceedings 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 625–634.

[16] M. Ghaffari. 2016. An improved distributed algorithm for maximal independent set. In *Proceedings 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 270–277. https://doi.org/10.1137/1.9781611974331.ch20

[17] A. Goldberg, S. Plotkin, and G. Shannon. 1987. Parallel symmetry-breaking in sparse graphs. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. ACM, 315–324.

[18] A. V Goldberg and S. A Plotkin. 1987. Parallel $(\Delta+ 1)$-coloring of constant-degree graphs. *Inform. Process. Lett.* 25, 4 (1987), 241–245.

[19] D. Harris, J. Schneider, and H.-H. Su. 2016. Distributed $(\Delta + 1)$-Coloring in Sublogarithmic Rounds. In *Proceedings 48th ACM Symposium on Theory of Computing (STOC)*. 465–478.

[20] W. Hoeffding. 1963. Probability Inequalities for Sums of Bounded Random Variables. *J. Amer. Statist. Assoc.* 58, 301 (1963), 13–30.

[21] Ö. Johansson. 1999. Simple Distributed $\Delta + 1$-coloring of Graphs. *Info. Proc. Lett.* 70, 5 (1999), 229–232.

[22] F. Kuhn, T. Moscibroda, and R. Wattenhofer. 2016. Local Computation: Lower and Upper Bounds. *J. ACM* 63, 2 (2016), 17:1–17:44.

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

[24] N. Linial. 1992. Locality in Distributed Graph Algorithms. *SIAM J. Comput.* 21, 1 (1992), 193–201.

[25] M. Luby. 1986. A simple parallel algorithm for the maximal independent set problem. *SIAM J. Comput.* 15, 4 (1986), 1036–1053.

[26] M. Naor. 1991. A Lower Bound on Probabilistic Algorithms for Distributive Ring Coloring. *SIAM J. Discrete Mathematics* 4, 3 (1991), 409–412. https://doi.org/10.1137/0404036

[27] A. Panconesi and A. Srinivasan. 1996. On the Complexity of Distributed Network Decomposition. *J. Algor.* 20, 2 (1996), 356–374.

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

[29] J. Schneider and R. Wattenhofer. 2010. A new technique for distributed symmetry breaking. In *Proceedings 29th Annual ACM Symposium on Principles of Distributed Computing (PODC)*. 257–266.