SIAM J. COMPUT.
Vol. 48, No. 1, pp. 122–143

AN EXPONENTIAL SEPARATION BETWEEN RANDOMIZED AND DETERMINISTIC COMPLEXITY IN THE LOCAL MODEL*

YI-JUN CHANG†, TSVI KOPELOWITZ‡, AND SETH PETTIE†

Abstract. Over the past 30 years numerous algorithms have been designed for symmetry breaking problems in the LOCAL model, such as maximal matching, MIS, vertex coloring, and edge coloring. For most problems the best randomized algorithm is at least exponentially faster than the best deterministic algorithm. In this paper we prove that these exponential gaps are *necessary* and establish numerous connections between the deterministic and randomized complexities in the LOCAL model. Each of our results has a very compelling take-away message: *Fast Δ-coloring of trees requires random bits.* Building on a recent randomized lower bound of Brandt et al. [*A lower bound for the distributed Lovász local lemma*, in Proceedings of the 48th ACM Symposium on Theory of Computing (STOC), ACM, New York, 2016, pp. 479–488], we prove that the randomized complexity of Δ-coloring a tree with maximum degree Δ is $O(\log_\Delta \log n + \log^* n)$ for any $\Delta \geq 55$, whereas its deterministic complexity is $\Omega(\log_\Delta n)$ for any $\Delta \geq 3$. This also establishes a large separation between the deterministic complexity of Δ-coloring and $(\Delta+1)$-coloring trees. *There is a gap in the deterministic complexity hierarchy.* We show that any deterministic algorithm for a natural class of problems that runs in $O(1) + o(\log_\Delta n)$ rounds can be transformed to run in $O(\log^* n - \log^* \Delta + 1)$ rounds. If the transformed algorithm violates a lower bound (even allowing randomization), then one can conclude that the problem requires $\Omega(\log_\Delta n)$ time deterministically. This gives an alternate proof that deterministically Δ-coloring a tree with small Δ takes $\Omega(\log_\Delta n)$ rounds. *Graph shattering is necessary.* We prove that the randomized complexity of any natural problem on instances of size n is at least its deterministic complexity on instances of size $\sqrt{\log n}$. This shows that any randomized $O(1)+o(\log_\Delta \log n)$-round algorithm can be *derandomized* to run in deterministically $O(1)+o(\log_\Delta n)$ rounds and hence can be transformed to run in $O(\log^* n - \log^* \Delta + 1)$ rounds. This also shows that a *deterministic* $\Omega(\log_\Delta n)$ lower bound for any problem (Δ-coloring a tree, for example) implies a *randomized* $\Omega(\log_\Delta \log n)$ lower bound. It illustrates that the *graph shattering* technique employed in recent randomized symmetry breaking algorithms is *absolutely essential* to the LOCAL model. For example, it is provably impossible to improve the $2^{O(\sqrt{\log \log n})}$ terms in the complexities of the best MIS and $(\Delta+1)$-coloring algorithms without *also* improving the $2^{O(\sqrt{\log n})}$-round Panconesi–Srinivasan algorithms.

Key words. coloring, distributed algorithm, local model, symmetry breaking

AMS subject classifications. 05C85, 68W15

DOI. 10.1137/17M1117537

1. Introduction.

One of the central problems of theoretical computer science is to determine the value of *random bits*. If the distinction is between computable versus incomputable functions, random bits are provably useless in centralized models (Turing machines) [39]. However, this is not true in the distributed world! The celebrated Fischer–Lynch–Paterson theorem [20] states that asynchronous deterministic agreement is impossible with one unannounced failure, yet it is possible to accomplish with probability 1 using randomization. See Ben-Or [6] and [10, 37, 26]. There are also a number of basic symmetry breaking tasks that are trivially impossible to solve by

*Received by the editors February 21, 2017; accepted for publication (in revised form) November 15, 2018; published electronically January 30, 2019. A preliminary version of this paper was presented at the 57th Annual IEEE Symposium on Foundations of Computer Science, 2016, New Brunswick, NJ.

http://www.siam.org/journals/sicomp/48-1/M111753.html

Funding: This work is supported by NSF grants CCF-1217338, CNS-1318294, and CCF-1514383.

†EECS, University of Michigan, Ann Arbor, MI 48105 (cyijun@umich.edu, pettie@umich.edu).

‡Computer Science, Bar-Ilan University, Ramat Gan, 5290002, Israel (kopelot@gmail.com).

identical, synchronized, deterministic processes—for example, medium access control to an Ethernet-like channel.

In this paper we examine the value of random bits in Linial's [31] LOCAL model. For the sake of clarity, we bifurcate the LOCAL model into two models, RandLOCAL and DetLOCAL. In both models the graph $G = (V, E)$ represents the topology of the communication network. Each vertex hosts a processor, and all vertices run the same algorithm. Each edge supports communication in both directions. The computation proceeds in synchronized *rounds*. In a round, each processor performs some computation and sends a message along each incident edge, which is delivered before the beginning of the next round. Each vertex v is initially aware of its degree $\deg(v)$ and certain global parameters such as $n \stackrel{\text{def}}{=} |V|$, $\Delta = \Delta(G) \stackrel{\text{def}}{=} \max_{v \in V} \deg(v)$, and possibly others.[1] In the LOCAL model the only measure of efficiency is the number of rounds. *All local computation is free, and the size of messages is unbounded.* Henceforth "time" refers to the number of rounds.

DetLOCAL: In order to avoid trivial impossibilities, all vertices are assumed to hold unique $\Theta(\log n)$-bit IDs. Except for the registers holding $\deg(v)$ and $\mathrm{ID}(v)$, the initial state of v is identical to every other vertex. The algorithm executed at each vertex is deterministic.

RandLOCAL: In this model each vertex may locally generate an unbounded number of independent truly random bits. However, there are no globally shared random bits. Except for the register holding $\deg(v)$, the initial state of v is identical to every other vertex. Algorithms in this model operate for a specified number of rounds and have some probability of *failure*, the definition of which is problem specific. We usually only consider algorithms whose global probability of failure is at most $1/\mathrm{poly}(n)$.

Observe that the lack of IDs in RandLOCAL is not a practical limitation. Before the first round each vertex can locally generate a random $\Theta(\log n)$-bit ID, which is unique with probability $1 - 1/\mathrm{poly}(n)$. For technical reasons it is convenient to assume that vertices are not initially differentiated by IDs. Notice that the role of n is different in the two LOCAL models: in DetLOCAL it only affects the ID length, whereas in RandLOCAL it only affects the failure probability.

Early work in the LOCAL models suggested that randomness is of limited help. Naor [32] showed that Linial's $\Omega(\log^* n)$ lower bound [31] for 3-coloring the ring holds even in RandLOCAL. Naor and Stockmeyer [33] proved that the class of problems solvable by $O(1)$-round algorithms is the same in RandLOCAL and DetLOCAL. See also [17] for a generalization of this result. However, in the intervening decades we have seen dozens of examples of symmetry breaking[2] algorithms for RandLOCAL that are substantially faster than their counterparts in DetLOCAL; see [5] for an extensive survey or Table 1.1 for a glimpse at three archetypal problems: Maximal Independent Set (MIS), maximal matching, and $(\Delta + 1)$-coloring.

Graph shattering. The randomized algorithms in Table 1.1 are exponentially faster than their deterministic counterparts *in two ways*. Their dependence on Δ is exponentially faster, and their dependence on n is usually identical to the best deterministic complexity, but for $\mathrm{poly}(\log n)$-size instances. For example, $2^{O(\sqrt{\log n})}$

[1] The assumption that global parameters are common knowledge can sometimes be removed; see Korman, Sereni, and Viennot [28].

[2] Informally, the term symmetry breaking refers to any distributed task whose valid solution satisfies that the output of the vertices in $N(v) \cup \{v\}$ cannot be all the same (e.g., Maximal Independent Set (MIS), maximal matching, and $(\Delta + 1)$-coloring).

TABLE 1.1
The impact of "graph shattering" on three archetypal symmetry breaking problems.

Problem	Model and result	
MIS	DetLOCAL:	$O\Big(\min\Big\{\Delta+\log^* n,\ 2^{O(\sqrt{\log n})}\Big\}\Big)$ [4, 35]
	RandLOCAL:	$O\Big(\log\Delta+2^{O(\sqrt{\log\log n})}\Big)$ [22]
	Lower Bound:	$\Omega\Big(\min\Big\{\sqrt{\log n/\log\log n},\ \log\Delta/\log\log\Delta+\log^* n\Big\}\Big)$ [30, 31, 32]
Maximal matching	DetLOCAL:	$O\big(\min\{\Delta+\log^* n,\ \log^2\Delta\log n\}\big)$ [34, 18]
	RandLOCAL:	$O\big(\log\Delta+\log^3\log n\big)$ [5]
	Lower Bound:	$\Omega\Big(\min\Big\{\sqrt{\log n/\log\log n},\ \log\Delta/\log\log\Delta+\log^* n\Big\}\Big)$ [30, 31, 32]
$(\Delta+1)$-coloring	DetLOCAL:	$O\Big(\min\Big\{\sqrt{\Delta\log\Delta}\log^*\Delta+\log^* n,\ 2^{O(\sqrt{\log n})}\Big\}\Big)$ [3, 21, 35]
	RandLOCAL:	$2^{O(\sqrt{\log\log n})}$ [13]
	Lower Bound:	$\Omega(\log^* n)$ [31, 32]

becomes $2^{O(\sqrt{\log\log n})}$, and $O(\log^2\Delta\log n)$ becomes $O(\log^3\log n)$. This second phenomenon is no coincidence! It is a direct result of the *graph shattering* approach to symmetry breaking used in [5] and further in [7, 14, 16, 22, 24, 25, 29, 36]. The idea is to apply a randomized procedure that fixes some fragment of the output (e.g., part of the MIS is fixed, part of the coloring is fixed, etc.), thereby effectively removing a large fraction of the vertices from further consideration. If it can be shown that the connected components in the subgraph still under consideration have size $\text{poly}(\log n)$, one can revert to the best available *deterministic* algorithm and solve the problem on each component of the "shattered" graph in parallel.

Lower bounds in the LOCAL *model.* Until recently, the main principle used to prove lower bounds in the LOCAL model was *indistinguishability*. The first application of this principle was by Linial [31] himself, who argued that any algorithm for coloring Δ-regular trees either uses $\Omega(\Delta/\log\Delta)$ colors or takes $\Omega(\log_\Delta n)$ time. The proof is as follows: (i) in $o(\log_\Delta n)$ time, a vertex *cannot always distinguish* whether the input graph G is a tree or a graph with girth $\Omega(\log_\Delta n)$; (ii) for all Δ and all n, there exists a Δ-regular graph with girth $\Omega(\log_\Delta n)$ and chromatic number $\chi=\Omega(\Delta/\log\Delta)$; hence[3] (iii) any $o(\log_\Delta n)$-time algorithm for coloring trees could also color such a graph and therefore must use at least χ colors.

A more subtle indistinguishability argument was used by Kuhn, Moscibroda, and Wattenhofer [30], who showed that $O(1)$-approximate vertex cover, maximal matching, MIS, and several other problems have $\Omega(\min\{\log\Delta/\log\log\Delta, \sqrt{\log n/\log\log n}\})$ lower bounds. Recently, Bar-Yehuda, Censor-Hillel, and Schwartzman [1] showed that a $(2+\epsilon)$-approximate vertex cover can be found in $O(\log\Delta/\log\log\Delta)$ time, matching the above lower bound.

By its nature, indistinguishability is not very good at separating randomized and deterministic complexities. Very recently, Brandt et al. [11] developed a lower bound technique that explicitly incorporates error probabilities and proved that several problems on graphs with constant Δ take $\Omega(\log\log n)$ time in RandLOCAL (with error probability $1/\text{poly}(n)$) such as *sinkless orientation*, *sinkless coloring*, and Δ-coloring. Refer to section 2 for definitions of these problems. Since the existence

[3]Linial [31] actually only used the existence of Δ-regular graphs with high girth and chromatic number $\Omega(\sqrt{\Delta})$. See [8] for constructions with chromatic number $\Omega(\Delta/\log\Delta)$.

of a sinkless orientation can be proved with the Lovász local lemma (LLL), this gave $\Omega(\log\log n)$ lower bounds on distributed algorithms for the constructive LLL. See [12, 14, 19, 22, 23, 36] for upper bounds on the distributed LLL.

1.1. New results. In this paper we exhibit an exponential separation between RandLOCAL and DetLOCAL for several *specific* symmetry breaking problems. More generally, we give new connections between the randomized and deterministic complexities of all *locally checkable labeling* (LCL) problems (refer to section 2 for a definition of LCLs), a class that includes essentially any natural symmetry breaking problem.

Separation of RandLOCAL and DetLOCAL. We extend Brandt et al.'s [11] randomized lower bound as follows: Δ-coloring Δ-regular graphs takes $\Omega(\log_\Delta \log n)$ time in RandLOCAL and $\Omega(\log_\Delta n)$ time in DetLOCAL. The hard graphs in this lower bound have girth $\Omega(\log_\Delta n)$, so by the indistinguishability principle, these lower bounds also apply to Δ-coloring trees.

On the upper bound side, Barenboim and Elkin [2] showed that for $\Delta \geq 3$, Δ-coloring trees takes $O(\log_\Delta n + \log^* n)$ time in DetLOCAL. We give an elementary proof that for $\Delta \geq 55$, Δ-coloring trees can be done in $O(\log_\Delta \log n + \log^* n)$ time in RandLOCAL, matching Brandt et al.'s [11] lower bound up to a $\log^* n$ additive term. A more complicated algorithm for Δ-coloring trees could be derived from [36] for $\Delta > \Delta_0$ and some very large constant Δ_0.[4]

There is a gap in the deterministic complexity hierarchy. We prove the existence of a large "gap" in the spectrum of possible complexities in DetLOCAL. The proof shows that any $f(\Delta) + o(\log_\Delta n)$ time algorithm for an LCL problem can be transformed in a black box way to run in $O((1 + f(\Delta))(\log^* n - \log^* \Delta + 1))$ time. Thus, on bounded-degree graphs, there are no "natural" deterministic time bounds between $\omega(\log^* n)$ and $o(\log n)$. Any $\omega(\log^* n)$ lower bound for bounded degree graphs (in either RandLOCAL or DetLOCAL) *immediately* implies an $\Omega(\log n)$ lower bound in DetLOCAL. For small Δ, this gives an alternate proof that Δ-coloring trees takes $\Omega(\log_\Delta n)$ time.

This reduction can be parameterized in many different ways. For example, if one were to develop a deterministic $O(\sqrt{\log n / \log\log n})$-time MIS or maximal matching algorithm—matching one of the Kuhn–Moscibroda–Wattenhofer (KMW) [30] lower bounds—it would immediately imply an $O((\log\Delta/\log\log\Delta)\cdot(\log^* n - \log^* \Delta + 1))$-time MIS/maximal matching algorithm, which almost matches the *other* KMW lower bound. We show that any $O(\log^{1-\frac{1}{k+1}} n)$-time DetLOCAL algorithm for an LCL problem can be transformed to run in $O(\log^k \Delta(\log^* n - \log^* \Delta + 1))$ time. By some strange coincidence, [5] gave an analogous reduction for MIS/maximal matching in bounded arboricity graphs, but for RandLOCAL and in the *reverse direction*. Specifically, any $O(\log^k \Delta + f(n))$-time RandLOCAL MIS/maximal matching algorithm for general graphs can be transformed into an $O(\log^{1-\frac{1}{k+1}} n + f(n))$-time RandLOCAL algorithm for bounded arboricity graphs.

Graph shattering is necessary. We prove that the RandLOCAL complexity for any LCL problem on instances of size n is at least its DetLOCAL complexity on

[4] The reason we are interested in minimizing the $\Delta_0 \leq \Delta$ for which the algorithm works is somewhat technical. It seems as if Δ-coloring trees is a problem whose character makes a *qualitative* transition when Δ is a small enough constant. Using our technique (graph shattering), we may be able to replace 55 with a smaller constant, *but not too small*. Any algorithm that 3-colors 3-regular trees, for example, will need to be qualitatively very different in its design.

instances of size $\sqrt{\log n}$. This *reverses* the implication proved above. For example, if we begin with a proof that Δ-coloring takes $\Omega(\log_\Delta n)$ time in DetLOCAL, then we conclude that it must take $\Omega(\log_\Delta \log n)$ time in RandLOCAL.

This result has a very clear take-away message: the *graph shattering* technique applied by recent randomized symmetry breaking algorithms [5, 14, 16, 22, 25, 7, 29, 36] is *inherent* to the RandLOCAL model, and every optimal RandLOCAL algorithm for instances of size n must, in some way, encode an optimal DetLOCAL algorithm on $\mathrm{poly}(\log n)$-size instances. It is therefore impossible to improve the $2^{O(\sqrt{\log \log n})}$ terms in the RandLOCAL MIS and coloring algorithms of [5, 22, 25, 13, 16] without also improving the $2^{O(\sqrt{\log n})}$-time DetLOCAL algorithms of Panconesi and Srinivasan [35], and it is impossible to improve the $O(\log^3 \log n)$ term in the RandLOCAL maximal matching algorithm of [5, 18] without also improving the $O(\log^2 \Delta \log n)$ DetLOCAL maximal matching algorithm of [18].

Moreover, this result shows that any randomized $O(1) + o(\log_\Delta \log n)$-round algorithm can be *derandomized* to run in deterministically $O(1) + o(\log_\Delta n)$ rounds and hence can be transformed to run in $O(\log^* n - \log^* \Delta + 1)$ rounds. This implies the existence of a gap in the complexities in the RandLOCAL model.

2. Preliminaries.

Notation. For a graph $G = (V, E)$ and for $u, v \in V$, let $\mathrm{dist}_G(v, u)$ be the distance between v and u in G. Let $N(v) = \{u \mid (v, u) \in E\}$ be the neighborhood of v and let $N^r(v) = \{u \mid \mathrm{dist}_G(v, u) \le r\}$ be the set of all vertices within distance r of v, including v. Throughout the paper, $\log n$ is the logarithm to the base 2, i.e., $\log_2 n$.

Locally checkable labeling. The class of *locally checkable labeling* (LCL) [33] problems is intuitively those graph problems whose solutions can be verified in $O(1)$ rounds, given a suitable labeling of the graph. Formally, an LCL problem is defined by a fixed radius r, a fixed set Σ of vertex labels, and a set $\mathcal{C}$ of acceptable labeled subgraphs. For any legal solution I to the problem there is a labeling $\lambda_I : V \to \Sigma$ that encodes I (plus possibly other information) such that for each $v \in V$, the labeled subgraph induced by $N^r(v)$ lies in $\mathcal{C}$. Moreover, for any *nonsolution* I' to the problem, there is no labeling $\lambda_{I'}$ with this property. The following symmetry breaking problems are LCLs for $r = 1$.

- Maximal Independent Set (MIS). Given a graph $G = (V, E)$, find a set $I \subseteq V$ such that for any vertex $v \in V$, we have $N(v) \cap I = \emptyset$ iff $v \in I$.
- k-Coloring. Given a graph $G = (V, E)$, find a labeling $V \to \{1, 2, \ldots, k\}$ such that for each edge $\{u, v\} \in E$, u and v are labeled with different numbers (also called colors).

For MIS it suffices to label vertices with $\Sigma = \{0, 1\}$ indicating whether they are in the MIS. For k-Coloring we use $\Sigma = \{1, \ldots, k\}$. The definition of LCLs is easily generalized to the case where the input graph G is supplemented with some labeling (e.g., an edge coloring) or where λ labels both vertices and edges. Brandt et al. [11] considered the following problems.

- Δ-Sinkless Coloring. Given a Δ-regular graph $G = (V, E)$ and a proper Δ-edge coloring of E using colors in $\{1, 2, \ldots, \Delta\}$, find a Δ-coloring of V using colors in $\{1, 2, \ldots, \Delta\}$ such that there is no edge $\{u, v\} \in E$ for which u, v, and $\{u, v\}$ all have the same color.
- Δ-Sinkless Orientation. Given a Δ-regular graph $G = (V, E)$ and a

proper Δ-edge coloring of E, find an orientation of the edges such that all vertices have out-degree ≥ 1.

Observe that both Δ-Sinkless Coloring and Δ-Sinkless Orientation are LCL graph problems with $r = 1$. For Sinkless Orientation $\Sigma = \{\rightarrow, \leftarrow\}^\Delta$ encodes the directions of all edges incident to a vertex, and the radius $r = 1$ is necessary and sufficient to verify that the orientations declared by both endpoints of an edge are consistent.

In this paper we require that the radius r and the number of labels $|\Sigma|$ are both constants. There exists some task that can be encoded by a constant radius LCL only if nonconstant $|\Sigma|$ is allowed. For instance, any LCL with $r = 1$ for a spanning tree must have $|\Sigma| = \Omega(\text{poly}(n))$ (i.e., the length of label must be $\Omega(\log n)$), and indeed there is a labeling scheme matching the label complexity lower bound [27].

Linial's coloring. In the DetLOCAL model the initial $\Theta(\log n)$-bit IDs can be viewed as an $n^{O(1)}$-coloring of the graph. Our algorithms make frequent use of Linial's [31] coloring algorithm, which recolors the vertices using a smaller palette. Notice that the $\Theta(\log n)$-bit IDs in DetLOCAL can be viewed as a $\text{poly}(n)$-coloring.

Theorem 2.1 ([31]). *Let G be a graph which has been k-colored.*

- *There is a* DetLOCAL *algorithm that computes a $5\Delta^2 \log k$-coloring in one round.*
- *There is a* DetLOCAL *algorithm that computes a $\beta \cdot \Delta^2$-coloring in $O(\log^* k - \log^* \Delta + 1)$ time, where $\beta > 0$ is a universal constant.*

3. The necessity of graph shattering. Theorem 3.1 establishes that the graph shattering technique [5] is optimal and unavoidable in RandLOCAL. In particular, the randomized complexity of any symmetry breaking problem always hinges on its deterministic complexity.

Theorem 3.1. *Let $\mathcal{P}$ be an LCL problem. Define $\mathsf{Det}_{\mathcal{P}}(n, \Delta)$ to be the complexity of the optimal deterministic algorithm for $\mathcal{P}$ in the* DetLOCAL *model, and define $\mathsf{Rand}_{\mathcal{P}}(n, \Delta)$ to be its complexity in the* RandLOCAL *model, with global error probability $1/n$. Then, for sufficiently large n,*

$$\mathsf{Det}_{\mathcal{P}}(n, \Delta) \leq \mathsf{Rand}_{\mathcal{P}}\left(2^{n^2}, \Delta\right).$$

Proof. Let $\mathcal{A}_{\mathsf{Rand}}$ be a randomized algorithm for $\mathcal{P}$. Each vertex running $\mathcal{A}_{\mathsf{Rand}}$ generates a string of $r(n, \Delta)$ random bits and proceeds for $t(n, \Delta)$ rounds, where r and t are two arbitrary functions. The probability that the algorithm fails in any way is at most $1/n$. Our goal is to convert $\mathcal{A}_{\mathsf{Rand}}$ into a deterministic algorithm $\mathcal{A}_{\mathsf{Det}}$ in the DetLOCAL model. Let $G = (V, E)$ be the network on which $\mathcal{A}_{\mathsf{Det}}$ runs. Initially each $v \in V$ knows $n = |V|, \Delta$, and a unique $\text{ID}(v) \in \{0,1\}^{c \log n}$. Let $\mathcal{G}_{n,\Delta}$ be the set of *all* n-vertex graphs with unique vertex IDs in $\{0,1\}^{c \log n}$ and maximum degree at most Δ. Since c is a constant, for sufficiently large n,

$$|\mathcal{G}_{n,\Delta}| \leq 2^{\binom{n}{2} + cn \log n} \ll 2^{n^2} \stackrel{\text{def}}{=} N,$$

regardless of Δ.

Imagine simulating $\mathcal{A}_{\mathsf{Rand}}$ on a graph $G' \in \mathcal{G}_{n,\Delta}$ whose vertices are given input parameters (N, Δ); that is, we imagine G' is disconnected from the remaining $N - n$ vertices. The probability that $\mathcal{A}_{\mathsf{Rand}}$ fails on an N-vertex graph is at most $1/N$, so the probability that any vertex in G' witnesses a failure is also certainly at most $1/N$.

Suppose we select a function $\phi : \{0,1\}^{c \log n} \to \{0,1\}^{r(N,\Delta)}$ uniformly at random from the space of all such functions. Define $\mathcal{A}_{\mathsf{Det}}[\phi]$ to be the *deterministic* algorithm

that simulates $\mathcal{A}_{\mathsf{Rand}}$ for $t(N,\Delta)$ steps, where the string of random bits generated by v is fixed to be $\phi(\mathrm{ID}(v))$. We shall call ϕ a *bad* function if $\mathcal{A}_{\mathsf{Det}}[\phi]$ fails to compute the correct answer on some member of $\mathcal{G}_{n,\Delta}$. By the union bound,

$$\begin{aligned}\Pr_{\phi}(\phi \text{ is bad}) &\le \sum_{G' \in \mathcal{G}_{n,\Delta}} \Pr_{\phi}(\mathcal{A}_{\mathsf{Det}}[\phi] \text{ errs on } G') \\ &= \sum_{G' \in \mathcal{G}_{n,\Delta}} \Pr(\mathcal{A}_{\mathsf{Rand}} \text{ errs on } G', \text{ with input parameters } (N,\Delta)) \\ &\le |\mathcal{G}_{n,\Delta}| / N < 1.\end{aligned}$$

Thus, there exists some good ϕ. Any ϕ can be encoded as a long bit-string $\langle\phi\rangle \stackrel{\text{def}}{=} \phi(0)\phi(1)\cdots\phi(2^{c\log n}-1)$. Define $\phi^\star$ to be the good function for which $\langle\phi^\star\rangle$ is lexicographically first.

The algorithm $\mathcal{A}_{\mathsf{Det}}$ is as follows. Each vertex v, given input parameters (n,Δ), first computes $N = 2^{n^2}, t(N,\Delta), r(N,\Delta)$ and then performs the simulations of $\mathcal{A}_{\mathsf{Rand}}$ necessary to compute $\phi^\star$. Once $\phi^\star$ is computed, it executes $\mathcal{A}_{\mathsf{Det}}[\phi^\star]$ for $t(N,\Delta)$ rounds. By definition, $\mathcal{A}_{\mathsf{Det}}[\phi^\star]$ never errs when run on any member of $\mathcal{G}_{n,\Delta}$.

By the definition of LCL, the problem $\mathcal{P}$ does not depend on n. Therefore, the output resulting from a successful execution of $\mathcal{A}_{\mathsf{Rand}}$ with input parameters (N,Δ) is also a legal solution when the input parameters are (n,Δ). Therefore, the output of $\mathcal{A}_{\mathsf{Det}}$ is guaranteed to be a legal solution. ∎

Remark 1. Theorem 3.1 works equally well when t and r are functions of n, Δ, and possibly other quantitative global graph parameters. For example, the time may depend on measures of local sparsity (as in [16]), arboricity/degeneracy (as in [2, 5]), or neighborhood growth (as in [38]).

Remark 2. The role of the LCL assumption in the proof of Theorem 3.1 is to make sure that $\mathcal{P}$ does not depend on n. This rules out some silly tasks. For example, if $\mathcal{P}$ were the task that asks each vertex v to report the number of vertices in $N^{\sqrt{\log n}}(v)$, then clearly $\mathcal{P}$ is not an LCL, and the RandLOCAL and DetLOCAL complexities of $\mathcal{P}$ are both $\Theta(\sqrt{\log n})$.

Naor and Stockmeyer [33] proved that the class of truly local ($O(1)$-time) problems in RandLOCAL and DetLOCAL is identical for bounded Δ. Theorem 3.1 offers a slight improvement over the Naor–Stockmeyer derandomization, since $\log^* n$ and $\log^*(\sqrt{\log n})$ differ by a constant.

COROLLARY 3.2. *Any* RandLOCAL *algorithm for an LCL taking* $t(n) = 2^{O(\log^* n)}$ *time can be derandomized without asymptotic penalty. The corresponding* DetLOCAL *algorithm runs in* $O(t(n))$ *time.*

4. Lower bounds for Δ-coloring Δ-regular trees. In this section we prove that on Δ-regular graphs with girth $\Omega(\log_\Delta n)$, Δ-coloring takes $\Omega(\log_\Delta \log n)$ time in RandLOCAL and $\Omega(\log_\Delta n)$ time in DetLOCAL. Since the girth of the graphs used to prove these lower bounds is $\Omega(\log_\Delta n)$, by the indistinguishability principle they also apply to the problem of Δ-coloring trees.

Sinkless coloring and sinkless orientations. Brandt et al. [11] proved $\Omega(\log\log n)$ lower bounds on RandLOCAL algorithms that have a $1/\mathrm{poly}(n)$ probability of failure, for sinkless coloring and sinkless orientation of 3-regular graphs. We say that a sinkless coloring algorithm $\mathcal{A}$ has failure probability p if, for *each* individual edge $e = \{u,v\}$, the probability that $\mathrm{color}(u) = \mathrm{color}(v) = \mathrm{color}(\{u,v\})$ is at most p. Thus, by the

union bound, the *global* probability of failure is at most $p|E|$. We say that a sinkless orientation algorithm $\mathcal{A}$ has failure probability p if, for each $v \in V$, the probability that v is a sink is at most p. We say that monochromatic edges and sinks are *forbidden configurations* for sinkless coloring and sinkless orientation, respectively.

The following two lemmas are proven in [11] for $\Delta = 3$. It is straightforward to go through the details of the proof and track the dependence on Δ.

LEMMA 4.1 ([11]). *Let $G = (V, E, \psi)$ be a Δ-regular graph with girth g that is equipped with a proper Δ-edge coloring ψ. Suppose that there is a* RandLOCAL *algorithm $\mathcal{A}$ for Δ-*SINKLESS COLORING *taking $t < \frac{g-1}{2}$ rounds such that for each $e \in E$, $\mathcal{A}$ outputs a forbidden configuration at e with probability at most p. Then there is a* RandLOCAL *algorithm $\mathcal{A}'$ for Δ-*SINKLESS ORIENTATION *taking t rounds such that for each $v \in V$, $\mathcal{A}'$ outputs a forbidden configuration at v with probability at most $2\Delta p^{1/3}$.*

LEMMA 4.2 ([11]). *Let $G = (V, E, \psi)$ be a Δ-regular graph with girth g that is equipped with a proper Δ-edge coloring ψ. Suppose that there is a* RandLOCAL *algorithm $\mathcal{A}'$ for sinkless orientation taking $t < \frac{g-1}{2}$ rounds such that for each $v \in V$, $\mathcal{A}'$ outputs a forbidden configuration at v with probability at most p. Then there is a* RandLOCAL *algorithm $\mathcal{A}$ for Δ-*SINKLESS COLORING *taking $t - 1$ rounds such that for each $e \in E$, $\mathcal{A}$ outputs a forbidden configuration at e with probability at most $4p^{1/(\Delta+1)}$.*

The following theorem generalizes Corollary 25 in [11] to allow nonconstant Δ and arbitrary failure probability p.

THEOREM 4.3. *Any* RandLOCAL *algorithm for Δ-coloring a graph with degree at most Δ and error probability p takes at least $t = \min\{\epsilon \log_{3(\Delta+1)} \ln(1/p),\ \epsilon \log_\Delta n\} - 1$ rounds for a sufficiently small constant $\epsilon > 0$.*

Proof. We assume that $\epsilon \log_{3(\Delta+1)} \ln(1/p) \geq 1$, since otherwise the theorem is trivial as $t < 0$. For any $\Delta \geq 3$ there exists a bipartite Δ-regular graph with girth $\Omega(\log_\Delta n)$; see [15, 9]. Such graphs are trivially Δ-edge colorable. Moreover, any Δ-coloring of such a graph is also a valid Δ-SINKLESS COLORING. Applying Lemmas 4.1 and 4.2, we conclude that any t-round Δ-SINKLESS COLORING algorithm with error probability p can be transformed into a $(t-1)$-round Δ-SINKLESS COLORING algorithm with error probability $4(2\Delta)^{\frac{1}{\Delta+1}} p^{\frac{1}{3(\Delta+1)}} < 7p^{\frac{1}{3(\Delta+1)}}$. Iterating this process t times, it follows that there exists a 0-round Δ-SINKLESS COLORING algorithm with failure probability $O(p^{(\frac{1}{3(\Delta+1)})^t})$. Notice that

$$p^{(\frac{1}{3(\Delta+1)})^t} \leq p^{(\frac{1}{3(\Delta+1)})^{\epsilon \log_{3(\Delta+1)} \ln(1/p)}} = p^{(\ln(1/p))^{-\epsilon}} = \exp(-(\ln(1/p))^{1-\epsilon}).$$

Because the graph is Δ-regular and the vertices are undifferentiated by IDs, any 0-round RandLOCAL algorithm colors each vertex independently according to the same distribution. The probability that any vertex is involved in a forbidden configuration (a monochromatic edge) is therefore at least $1/\Delta^2$. Since $\epsilon \log_{3(\Delta+1)} \ln(1/p) \geq 1$ we have $\Delta < \ln(1/p)$, but

$$\frac{1}{\Delta^2} \geq \exp(-2 \ln \ln(1/p)) \gg \exp\left(-(\ln(1/p))^{1-\epsilon}\right).$$

This is a contradiction since we obtain a 0-round Δ-SINKLESS COLORING algorithm with failure probability less than $1/\Delta^2$. Thus, there is no RandLOCAL Δ-SINKLESS COLORING algorithm that takes t-rounds and errs with probability p. ∎

Corollary 4.4 is an immediate consequence of Theorem 4.3.

COROLLARY 4.4. *Any* RandLOCAL *algorithm for* Δ*-coloring a graph with global error probability* $1/\mathrm{poly}(n)$ *takes* $\Omega(\log_\Delta \log n)$ *time.*

Theorem 4.3 does not immediately extend to DetLOCAL. It is tempting to feel that setting $p = 0$ yields a $\Omega(\log_\Delta n)$ DetLOCAL lower bound. But this is not a correct inference. Recall that in the DetLOCAL model vertices are initially endowed with $O(\log n)$-bit IDs, whereas in RandLOCAL they are undifferentiated, and the naive way of generating such IDs in RandLOCAL has failure probability $1/\mathrm{poly}(n)$.

THEOREM 4.5. *Any* DetLOCAL *algorithm that* Δ*-colors* Δ*-regular graphs with girth* $\Omega(\log_\Delta n)$ *or* Δ*-regular trees requires* $\Omega(\log_\Delta n)$ *time.*

Proof. Let $\mathcal{A}_{\mathsf{Det}}$ be a DetLOCAL algorithm that Δ-colors a graph in $t = t(n, \Delta)$ rounds, and let G be the input graph. We construct a RandLOCAL algorithm $\mathcal{A}_{\mathsf{Rand}}$ taking $O(t)$ rounds as follows. Before the first round each vertex locally generates a random n-bit ID. Assume for the time being that these IDs are unique and therefore constitute a 2^n-coloring of G. Let $G' = (V, \{\{u, v\} \mid \mathrm{dist}_G(u, v) \le 2t + 1\})$.[5] The maximum degree Δ' in G' is clearly less than n. We apply one step of Linial's recoloring algorithm (Theorem 2.1) to G' and obtain a coloring with palette size $O(\Delta'^2 \log(2^n)) = O(n^3)$. A step of Linial's algorithm in G' is simulated in G using $O(t)$ time. Using these colors as $(3 \log n + O(1))$-bit IDs, we simulate $\mathcal{A}_{\mathsf{Det}}$ in G for t steps. Since no vertex can see two vertices with the same ID, this algorithm necessarily behaves as if all IDs are unique. Observe that because $\mathcal{A}_{\mathsf{Det}}$ is deterministic, the only way $\mathcal{A}_{\mathsf{Rand}}$ can err is if the initial n-bit IDs fail to be unique. This occurs with probability $p < n^2/2^n$. By Theorem 4.3, $\mathcal{A}_{\mathsf{Rand}}$ takes $\Omega(\min\{\log_\Delta \log(1/p), \log_\Delta n\}) = \Omega(\log_\Delta n)$ time. ∎

5. Gaps in deterministic and randomized time complexity. The time hierarchy theorem informally says that a Turing machine can solve more problems given more time. A similar question can be asked in the setting of distributed computation. For example, are there natural or contrived problems with DetLOCAL complexity $\Theta((\log^* n)^2)$, $\Theta(\log \log n)$, or $\Theta\left(\sqrt{\log n}\right)$, when $\Delta = O(1)$? In this section, we demonstrate a general technique that allows one to speed up deterministic algorithms in the DetLOCAL model. Based on this technique, we demonstrate the existence of a "gap" in possible DetLOCAL and RandLOCAL complexities, answering the above question in the negative.

A graph class is *hereditary* if it is closed under removing vertices and edges. Examples of hereditary graph classes are general graphs, forests, bounded arboricity graphs, triangle-free graphs, and planar graphs. We prove that, for graphs with constant Δ, the DetLOCAL complexity of *any* LCL problem on a *hereditary* graph class is either $\Omega(\log n)$ or $O(\log^* n)$. Moreover, if the hereditary graph class is also closed under taking disjoint union, then the RandLOCAL complexity of *any* LCL problem is either $\Omega(\log \log n)$ or $O(\log^* n)$.

THEOREM 5.1. *Let* $\mathcal{P}$ *be an LCL graph problem with parameters* r, Σ, *and* $\mathcal{C}$, *and let* $\mathcal{A}$ *be a* DetLOCAL *algorithm for solving* $\mathcal{P}$. *Let* β *be the universal constant*

[5] The parameter $2t + 1$ is explained as follows. Whether an edge $e = \{u, v\}$ is monochromatic depends on the colors of u and v, and this depends on the graph topology and the IDs in the subgraph induced by $N^t(u) \cup N^t(v)$. If all vertices in $N^t(u) \cup N^t(v)$ have distinct IDs, u and v must be colored differently. Notice that the maximum distance within $N^t(u) \cup N^t(v)$ is $2t + 1$. In general, for LCL of radius r, a deterministic algorithm works correctly as long as all vertices within distance $2t + 2r$ have distinct IDs.

from Theorem 2.1. *Suppose that the runtime of $\mathcal{A}$ on an n-vertex graph taken from a hereditary graph class is at most $f(\Delta)+\epsilon\log_\Delta n$, where $f(\Delta)\geq 0$ and $\epsilon=\frac{1}{4+4\log\beta+4r}$ is a constant. Then there exists a* DetLOCAL *algorithm $\mathcal{A}'$ that solves $\mathcal{P}$ on the same instances in $O\left((1+f(\Delta))(\log^* n-\log^*\Delta+1)\right)$ time.*

Proof. Notice that for any instance of $\mathcal{P}$ with n vertices and ID length ℓ, it must be that $\ell\geq\log n$, and so the running time of $\mathcal{A}$ on such instances is bounded by $T(\Delta,\ell)\leq f(\Delta)+\frac{\epsilon\ell}{\log\Delta}$.

Let $G=(V,E)$ be a graph in a hereditary graph class. The algorithm $\mathcal{A}'$ on G works as follows. Let $\tau=1+\log\beta$ be a constant. We use Linial's coloring technique to produce short IDs of length ℓ' that are distinct within distance $4f(\Delta)+2\tau+2r$. Let $G'=(V,E')$ be the graph with

$$E'=\left\{\{u,v\}\in\binom{V}{2}\;\middle|\;\operatorname{dist}_G(u,v)\leq 4f(\Delta)+2\tau+2r\right\}.$$

The maximum degree in G' is clearly at most $\Delta^{4f(\Delta)+2\tau+2r}$. Each vertex $u\in V$ simulates G' by collecting $N^{4f(\Delta)+2\tau+2r}(u)$ in $O(f(\Delta)+\tau+r)$ time.

We simulate the algorithm of Theorem 2.1 on G' by treating each of the ℓ-bit IDs of vertices in V as a color. This produces a $\beta\cdot\Delta^{8f(\Delta)+4\tau+4r}$-coloring, which is equivalent to identifiers of length $\ell'=(8f(\Delta)+4\tau+4r)\log\Delta+\log\beta$. Although these identifiers are not globally unique, they are distinct in $N^{2f(\Delta)+\tau+r}(u)$ for each vertex $u\in V$. The time complexity of this process is

$$(4f(\Delta)+2\tau+2r)\cdot O\left(\log^* n-\log^*\Delta+1\right).$$

Finally, we apply $\mathcal{A}$ on G while implicitly assuming that the graph size is $2^{\ell'}$ and using the shorter IDs. The runtime of this execution of $\mathcal{A}$ is

$$\begin{aligned}
f(\Delta)+\frac{\epsilon\ell'}{\log\Delta}&=f(\Delta)+\frac{\epsilon((8f(\Delta)+4\tau+4r)\log\Delta+\log\beta)}{\log\Delta}\\
&=(1+8\epsilon)f(\Delta)+1+\frac{\epsilon\log\beta}{\log\Delta} && \epsilon(4\tau+4r)=1\\
&\leq(1+8\epsilon)f(\Delta)+\tau && \log\Delta\geq 1,\epsilon<1\\
&\leq 2f(\Delta)+\tau. && 8\epsilon=\frac{2}{\tau+r}\leq 1
\end{aligned}$$

Whether the output labeling of $u\in V$ is legal depends on the labeling of the vertices in $N^r(u)$, which depends on the graph structure and the IDs in $N^{2f(\Delta)+\tau+r}(u)$. Due to the hereditary property of the graph class under consideration, for each $u\in V$, $N^{2f(\Delta)+\tau+r}(u)$ is isomorphic to a subgraph of some $2^{\ell'}$-vertex graph in the same class. Moreover, the shortened IDs in $N^{2f(\Delta)+\tau+r}(u)$ are distinct. Therefore, it is guaranteed that the output of the simulation is a legal labeling.

The total time complexity is

$$\begin{aligned}
&(4f(\Delta)+2\tau+2r)\cdot O(\log^* n-\log^*\Delta+1)+2f(\Delta)+\tau\\
&=O\left((1+f(\Delta))(\log^* n-\log^*\Delta+1)\right).
\end{aligned}$$

∎

Combining Theorem 5.1 with Corollary 4.4 and setting $f(\Delta)=O(1)$ provides a new proof of Theorem 4.5 for small enough Δ. To see this, notice that any lower bound for the RandLOCAL model with error probability $1/\mathrm{poly}(n)$ can be adapted to DetLOCAL since we can randomly pick $O(\log n)$-bit IDs that are distinct with

probability $1 - 1/\text{poly}(n)$. From Corollary 4.4 any DetLOCAL algorithm that Δ-colors a Δ-regular tree requires $\Omega(\log_\Delta \log n)$ time. However, Theorem 5.1 states that any DetLOCAL algorithm running in $O(1) + o(\log_\Delta n)$ time can be sped up to run in $O(\log^* n - \log^* \Delta + 1)$ time. This contradicts the lower bound whenever $\log_\Delta \log n \gg \log^* n - \log^* \Delta + 1$. Hence Δ-coloring a Δ-regular tree takes $\Omega(\log_\Delta n)$ time in DetLOCAL for small enough Δ such that $\log_\Delta \log n \gg \log^* n - \log^* \Delta + 1$.

Theorem 5.1 implies that the deterministic time complexity of any LCL problem is either $O((1 + f(\Delta))(\log^* n - \log^* \Delta + 1))$ or $\Omega(f(\Delta) + \log_\Delta n)$. In particular, when Δ is a constant, Theorem 5.1 implies the following corollary.

COROLLARY 5.2. *The* DetLOCAL *complexity of any LCL problem on any hereditary graph class with* $\Delta = O(1)$ *is either* $\Omega(\log n)$ *or* $O(\log^* n)$.

A simple adaptation of the proof of Theorem 5.1 shows an even stronger dichotomy when $\Delta = 2$.

THEOREM 5.3. *The* DetLOCAL *complexity of any LCL problem on any hereditary graph class with* $\Delta = 2$ *is either* $\Omega(n)$ *or* $O(\log^* n)$.

We remark that an intuitive explanation of the time complexity requirement in Theorems 5.1 and 5.3 is that the diameter of a graph with maximum degree Δ is at least $\Omega(\log_\Delta n)$ for $\Delta \geq 3$ and $\Omega(n)$ when $\Delta = 2$. These theorems imply that any LCL problem is either local (i.e., it can be solved in $O(\log^* n)$ time) or nonlocal (i.e., it needs diameter time on some instances), and there is nothing in between the two extremes.

Combining Theorems 5.1 and 3.1 also yields a gap in the complexities of the RandLOCAL model.

THEOREM 5.4. *Let* $\mathcal{P}$ *be an LCL graph problem with parameters* r, Σ, *and* $\mathcal{C}$, *and let* $\mathcal{A}$ *be a* RandLOCAL *algorithm for solving* $\mathcal{P}$. *Let* $\mathcal{G}$ *be a hereditary graph class that is closed under taking disjoint union. Let* β *be the universal constant from Theorem* 2.1. *Suppose that the runtime of* $\mathcal{A}$ *on an* n*-vertex graph taken from* $\mathcal{G}$ *is at most* $f(\Delta) + (\epsilon/2)\log_\Delta \log n$, *where* $f(\Delta) \geq 0$ *and* $\epsilon = \frac{1}{4+4\log\beta+4r}$ *is a constant. Then there exists a* DetLOCAL *algorithm* $\mathcal{A}'$ *that solves* $\mathcal{P}$ *on the same instances in* $O((1 + f(\Delta))(\log^* n - \log^* \Delta + 1))$ *time.*

Proof. Since $\mathcal{G}$ is closed under taking disjoint union, by Theorem 3.1, for sufficiently large n, the deterministic complexity of $\mathcal{P}$ is at most $f(\Delta) + (\epsilon/2)\log_\Delta \log 2^{n^2} = f(\Delta) + \epsilon \log_\Delta n$. Since $\mathcal{G}$ is a hereditary graph class, by Theorem 5.1, there exists a DetLOCAL algorithm $\mathcal{A}'$ that solves $\mathcal{P}$ in $O((1 + f(\Delta))(\log^* n - \log^* \Delta + 1))$ time. ∎

Setting $\Delta = O(1)$ gives us the following corollary.

COROLLARY 5.5. *Let* $\mathcal{G}$ *be any hereditary graph class with* $\Delta = O(1)$ *that is closed under taking disjoint union. The* RandLOCAL *complexity of any LCL problem on* $\mathcal{G}$ *is either* $\Omega(\log\log n)$ *or* $O(\log^* n)$. *Moreover, any* $O(\log^* n)$ *algorithm can be implemented in* DetLOCAL.

Given a $O(\sqrt{\log n})$-time deterministic algorithm, one may feel that it is possible to use Theorem 5.1 to improve the time complexity to $O(\log^* n)$ since $\sqrt{\log n} = o(\log_\Delta n)$ for the case $\Delta = \exp(o(\sqrt{\log n}))$. However, the class of graphs with $\Delta = \exp(o(\sqrt{\log n}))$ is not hereditary, and so Theorem 5.1 does not apply. Nonetheless, Linial's coloring technique can be used to speed up algorithms with time complexity of the form $f(\Delta) + g(n)$.

THEOREM 5.6. *Let* $\mathcal{P}$ *be an LCL graph problem with parameters* r, Σ, *and* $\mathcal{C}$,

and let $\mathcal{A}$ be a DetLOCAL *algorithm for solving $\mathcal{P}$. Let k be a number such that $0 \leq k \leq O(1)$. Suppose that the runtime of the algorithm $\mathcal{A}$ on an n-vertex graph taken from a hereditary graph class is at most $O(\log^k \Delta + \log^{\frac{k}{k+1}} n)$. Then there exists a deterministic algorithm $\mathcal{A}'$ that solves $\mathcal{P}$ on the same instances in $O(\log^k \Delta(\log^* n - \log^* \Delta + 1))$ time.*

Proof. Let the ID length be ℓ; then it must be that $\ell \geq \log n$, and so the running time of $\mathcal{A}$ on such instances is bounded by $\epsilon_1 \log^k \Delta + \epsilon_2 \ell^{\frac{k}{k+1}}$ for some constants ϵ_1, ϵ_2.

We set $\tau = \epsilon \log^k \Delta$, with the parameter ϵ to be determined. Similar to the proof of Theorem 5.1, the algorithm $\mathcal{A}'$ first produces shortened IDs that are distinct for vertices within distance $2\tau + 2r$ and then simulates $\mathcal{A}$ on the shortened IDs in τ rounds.

Let $G' = (V, E')$ be the graph with

$$E' = \left\{ \{u, v\} \in \binom{V}{2} \;\middle|\; \operatorname{dist}_G(u, v) \leq 2\tau + 2r \right\}.$$

The maximum degree in G' is at most $\Delta^{2\tau+2r}$. Each vertex $u \in V$ simulates G' by collecting $N^{2\tau+2r}(u)$ in $O(\tau + r)$ time.

We simulate the algorithm of Theorem 2.1 on G' by treating each of the ℓ-bit IDs of vertices in V as a color. This produces a $\beta \cdot \Delta^{4\tau+4r}$-coloring, which is equivalent to identifiers of length $\ell' = (4\tau + 4r) \log \Delta + \log \beta$. Although these identifiers are not globally unique, they are distinct in $N^{\tau+r}(u)$ for each vertex $u \in V$. The time complexity of this process is

$$(2\tau + 2r) \cdot O\left(\log^* n - \log^* \Delta + 1\right).$$

Finally, we apply $\mathcal{A}$ on G while implicitly assuming that the graph size is $2^{\ell'}$ and using the shorter IDs. We set ϵ as a large enough number such that $\epsilon_1 + \epsilon_2 \left(4(\epsilon + r + \log \beta)\right)^{\frac{k}{k+1}} \leq \epsilon$. Since $0 \leq k \leq O(1)$, we have $\epsilon = O(1)$. The runtime of this execution of $\mathcal{A}$ is

$$\begin{aligned}
\epsilon_1 \log^k \Delta + \epsilon_2 \left(\ell'\right)^{\frac{k}{k+1}} &= \epsilon_1 \log^k \Delta + \epsilon_2 \left((4\tau + 4r) \log \Delta + \log \beta\right)^{\frac{k}{k+1}} \\
&\leq \epsilon_1 \log^k \Delta + \epsilon_2 \left(4(\epsilon \log^k \Delta + r + \log \beta) \log \Delta\right)^{\frac{k}{k+1}} \\
&\leq \epsilon_1 \log^k \Delta + \epsilon_2 \left(4(\epsilon + r + \log \beta) \log^{k+1} \Delta\right)^{\frac{k}{k+1}} \\
&= \left(\epsilon_1 + \epsilon_2 \left(4(\epsilon + r + \log \beta)\right)^{\frac{k}{k+1}}\right) \log^k \Delta \\
&\leq \epsilon \log^k \Delta \\
&= \tau.
\end{aligned}$$

Whether the output labeling of $u \in V$ is legal depends on the labeling of the vertices in $N^r(u)$, which depends on the graph structure and the IDs in $N^{\tau+r}(u)$. Due to the hereditary property of the graph class under consideration, for each $u \in V$, $N^{\tau+r}(u)$ is isomorphic to a subgraph of some $2^{\ell'}$-vertex graph in the same class. Moreover, the shortened IDs in $N^{\tau+r}(u)$ are distinct. Therefore, it is guaranteed that the output of the simulation is a legal labeling.

The total time complexity is at most

$$(2\tau + 2r) \cdot O(\log^* n - \log^* \Delta + 1) + \tau = O(\log^k \Delta(\log^* n - \log^* \Delta + 1)). \qquad \square$$

A note about MIS lower bounds. Kuhn, Moscibroda, and Wattenhofer demonstrated that for a variety of problems (including MIS) there is a lower bound of $\Omega(\min\{\log\Delta/\log\log\Delta, \sqrt{\log n/\log\log n}\})$ rounds [30]. The lower bound graph they used to prove this result has $\log\Delta/\log\log\Delta = O(\sqrt{\log n/\log\log n})$. The proof framework of Theorem 5.6 can be used to show that if there is a deterministic algorithm $\mathcal{A}$ for MIS that runs in $O(\sqrt{\log n/\log\log n})$ time, then there is another deterministic algorithm $\mathcal{A}'$ running in $O(\log\Delta/\log\log\Delta)\cdot(\log^* n - \log^*\Delta + 1))$ time. Let the runtime of $\mathcal{A}$ be $\epsilon_1\sqrt{\ell/\log\ell}$ for ID length ℓ. Set $\tau = \epsilon\log\Delta/\log\log\Delta$. Then $\ell' = (4\tau+4r)\log\Delta + \log\beta = \Theta(\log^2\Delta/\log\log\Delta)$. The runtime of $\mathcal{A}$ for ID length ℓ' is $\epsilon_1\sqrt{\ell/\log\ell} = \epsilon_1\cdot\sqrt{\epsilon/\log\epsilon}\cdot O(\log\Delta/\log\log\Delta) < \tau$, by choosing a large enough constant ϵ.

Interestingly, Barenboim et al. [5] showed that an MIS algorithm in RandLOCAL running in $O(\log^k\Delta + f(n))$ time implied another RandLOCAL algorithm running in $O(\log^k\lambda + \log^{1-\frac{1}{k+1}} n + f(n))$ time on graphs of arboricity λ. This is analogous to Theorem 5.6 but in the reverse direction.

6. Algorithms for Δ-coloring trees. In section 4, we showed that the problem of Δ-coloring trees has an $\Omega(\log_\Delta n)$ DetLOCAL lower bound and an $\Omega(\log_\Delta\log n)$ RandLOCAL lower bound. These lower bounds have matching upper bounds, up to an additive $\log^* n$ term.

The algorithm of Barenboim and Elkin [2] demonstrates that the deterministic bound is essentially tight. They proved that Δ-coloring unoriented trees, where $\Delta\geq 3$, takes $O(\log_\Delta n + \log^* n)$ time. This is actually a special case of their algorithm, which applies to graphs of bounded arboricity λ.

THEOREM 6.1 ([2]). *For $q\geq 3$, there is a* DetLOCAL *algorithm for q-coloring trees in $O(\log_q n + \log^* n)$ time, independent of Δ.*

Pettie and Su [36] gave randomized algorithms for $(4+o(1))\Delta/\ln\Delta$-coloring triangle-free graphs. Their algorithm makes extensive use of the distributed Lovász local lemma [14] and runs in $\Omega(\log n)$ time. Pettie and Su sketched a proof that Δ-coloring trees takes $O(\log_\Delta\log n + \log^* n)$ time, at least for sufficiently large Δ.

THEOREM 6.2 ([36]). *There exists a large constant Δ_0 such that when $\Delta\geq\Delta_0$, there is a* RandLOCAL *algorithm for Δ-coloring trees in $O(\log_\Delta\log n + \log^* n)$ time.*

The nature of the proof of Theorem 6.2 makes it difficult to calculate a specific Δ_0 for which the theorem applies. Moreover, the proof is only sketched, hidden inside more complicated ideas. We address both of these issues. First, we provide a simple algorithm and elementary proof of Theorem 6.2. Second, we prove Theorem 6.4, which combines Theorem 6.2 with a new technique for constant $\Delta\geq 55$, thereby providing a randomized algorithm for Δ-coloring a tree that runs in $O(\log_\Delta\log n + \log^* n)$ time for any constant $\Delta\geq 55$.

6.1. A simple proof of Theorem 6.2. For a graph $G=(V,E)$ we say that a subset $S\subseteq V$ is a *distance-k* set if the following two conditions are met:

1. The distance between any two distinct vertices $u,v\in S$ is at least k.
2. Define $G^k = (V,E^k)$, where there is an edge $\{u,v\}\in E^k$ if and only if $\text{dist}_G(u,v)=k$. Then S is connected in G^k.

We make use of the following lemma in the proof of Theorem 6.2. While the proof of this lemma is implicit in [5], we reproduce it here for the sake of clarity.

LEMMA 6.3 ([5]). *The number of distinct distance-k sets of size t is less than $4^t\cdot n\cdot\Delta^{k(t-1)}$.*

Proof. A distance-k set is spanned by a tree in G^k. There are fewer than 4^t distinct unlabeled trees of t vertices, and there are fewer than $n\Delta^{k(t-1)}$ ways to embed a t-vertex tree in G^k. The lemma follows since there is an injective mapping from the family of distance-k sets of size t to subtrees of t vertices in G^k. ∎

Proof of Theorem 6.2. Our algorithm has two phases. The first phase, which takes $O(\log^* \Delta)$ rounds, partially colors the graph using colors in $\{1, 2, \ldots, \Delta - \sqrt{\Delta}\}$. The second phase, which takes $O(\log_\Delta \log n + \log^* n)$ rounds, applies a deterministic algorithm to $\sqrt{\Delta}$-color the remaining uncolored vertices using colors in $\{\Delta - \sqrt{\Delta} + 1, \ldots, \Delta\}$. We assume throughout the proof that Δ is at least a large enough constant.

Phase 1. The first phase of the algorithm takes $O(\log^* \Delta)$ rounds. In each round, the algorithm attempts to color some uncolored vertices. We will explain soon how uncolored vertices decide if they participate in a given round. In the beginning of round i, for each vertex $v \in V$, let $\Psi_i(v)$ denote v's available palette (i.e., the set of colors that v can choose in round i), and let $N_i(v)$ denote the set of uncolored vertices adjacent to v that are trying to color themselves in this round. Initially, we set $N_1(v) = N(v)$, and $\Psi_1(v) = \{1, 2, \ldots, \Delta - \sqrt{\Delta}\}$, for all v. That is, in the first round all vertices attempt to color themselves, and they all have the full palette of this phase available for choices of a color.

We maintain the following two properties at each vertex v that is not marked *bad* at round i. Only nonbad vertices attempt to color themselves at round i:

$\mathcal{P}_1(v)$: (*Large Palette Property at* v) $|\Psi_i(v)| \geq \frac{\Delta}{200}$.

$\mathcal{P}_2(v)$: (*Small Degree Property at* v) $|N_i(v)| \leq \frac{\Delta}{c_i}$, where c_i is defined as $c_1 = 1$, $c_2 = 200/199$, and $c_i = \min\left\{\Delta^{0.1},\ c_{i-1} \cdot \exp\left(\frac{c_{i-1}}{3 \cdot 200 \cdot e^{200}}\right)\right\}$ for $i > 2$.

Let t be the smallest number i such that $c_i = \Delta^{0.1}$. Notice that $t = O(\log^* \Delta)$ is the number of the rounds in the first phase.

The intuition behind the two properties $\mathcal{P}_1(v)$ and $\mathcal{P}_2(v)$ is that they ensure that (i) participating vertices always have a large enough palette to use, and (ii) there is a large separation between the number of available colors and the number of uncolored neighbors so that we can color a large fraction of vertices in each round.

For each $1 \leq i \leq t$, the ith round consists of two constant time subroutines ColorBidding(i) and Filtering(i). In ColorBidding(i), each participating vertex v selects a random subset of colors S_v. If there is a color in S_v that does not belong to $\bigcup_{u \in N_i(v)} S_u$, the vertex v *succeeds* and colors itself with any such color. If such a color is chosen, denote it by Color(v). After ColorBidding(i), we execute Filtering(i), which filters out some vertices and thereby prevents $\mathcal{P}_1$ and $\mathcal{P}_2$ from being violated. Such vertices are called *bad* vertices, and they will no longer participate in the remaining rounds of Phase 1.

ColorBidding(i).

Do the following steps in parallel for each uncolored vertex v that is not bad:

1. If $c_i = c_1 = 1$, then S_v contains one color chosen uniformly at random from $\Psi_1(v)$. Otherwise ($c_i > 1$), construct the set S_v by independently including each color of $\Psi_i(v)$ with probability $c_i/|\Psi_i(v)|$.
2. If $S_v \setminus \bigcup_{u \in N_i(v)} S_u \neq \emptyset$, then permanently color v by picking an arbitrary color in $S_v \setminus \bigcup_{u \in N_i(v)} S_u$ for Color(v).
3. $\Psi_{i+1}(v) \leftarrow \Psi_i(v) \setminus \{\text{Color}(u) \mid u \in N_i(v) \text{ is permanently colored}\}$.

We define $N_i'(v)$ as the set of participating vertices *after* ColorBidding($i-1$) and

before Filtering$(i-1)$ that are adjacent to v. In other words,

$$N_i'(v) = N_{i-1}(v) \setminus \{u \mid u \text{ is permanently colored in } \mathsf{ColorBidding}(i-1)\}.$$

Filtering(i).
For each uncolored vertex v that is not bad:
1. If $i = 1$ and $|\Psi_2(v)| - |N_2'(v)| < \frac{\Delta}{200}$, then mark v as a bad vertex.
2. If $1 < i < t$ and $|N_{i+1}'(v)| > \frac{\Delta}{c_{i+1}}$, then mark v as a bad vertex.
3. If $i = t$, then mark v as a bad vertex.

Phase 2. By the filtering rule for $i = t$, all the remaining uncolored vertices after Phase 1 are bad vertices. We color the bad vertices in Phase 2. We will later prove that after Phase 1, with high probability any connected component induced by bad vertices has size at most $\Delta^4 \log n$. We use Theorem 6.1 to $\sqrt{\Delta}$-color such connected components using the $\sqrt{\Delta}$ reserved colors.

Runtime. The runtime of Phase 1 is $t = O(\log^* \Delta)$ rounds. The runtime of Phase 2 is $O\big(\log_{\sqrt{\Delta}}\left(\Delta^4 \log n\right) + \log^*\left(\Delta^4 \log n\right)\big) = O\left(\log_\Delta \log n + \log^* n\right)$. Thus, the total runtime is $O\left(\log_\Delta \log n + \log^* n\right)$ rounds.

Analysis. The analysis of Phase 2 relies only on proving that, with high probability, all connected components induced by bad vertices after Phase 1 are of size at most $\Delta^4 \log n$. Thus, we focus on analyzing Phase 1.

A vertex v that participates in round i may be marked *bad*, depending on the random bits generated by vertices in $N^2(v)$ in this round. Our analysis applies to any partial coloring of $N^2(v)$ that satisfies properties $\mathcal{P}_1$ and $\mathcal{P}_2$ and is entirely independent of the random bits generated by vertices *outside* of $N^2(v)$. The probability that a vertex is marked *bad* is $\exp(-\text{poly}(\Delta))$. This proof is based on the following claim, which is proved by applying standard Chernoff bounds. The proof of the claim is deferred to Appendix A for the sake of presentation.

Claim 1. *The probability that a vertex v is marked as bad in round* 1 *is at most* $\exp(-\Omega(\Delta))$. *For $1 < i \le t$, the probability that a vertex v that participates in round i is marked bad in round i is at most* $\exp(-\Omega(\Delta^{0.1}))$.

By the union bound for all rounds in Phase 1, the probability that any vertex v becomes a bad vertex after Phase 1 is $O(\log^* \Delta) \cdot \exp(-\text{poly}(\Delta)) = \exp(-\text{poly}(\Delta))$, regardless of the choice of random bits for all vertices not in $N^2(v)$. Observe that for any two distinct vertices u and v in any distance-5 set T, we must have $u \notin N^2(v)$. Therefore, just before Phase 2, for any distance-5 set T of size s, the probability that all vertices in T are bad is at most $\exp(-s \cdot \text{poly}(\Delta))$. By Lemma 6.3, there are at most $4^s \cdot n \cdot \Delta^{4(s-1)}$ distinct distance-5 sets T of size s. By the union bound, with probability at least $\left(4^s \cdot n \cdot \Delta^{4(s-1)}\right) \cdot \exp(-s \cdot \text{poly}(\Delta))$, there is no distance-5 set of size s that contains only bad vertices. This probability is upper bounded by n^{-c} for any c when $s = \log n$. Therefore, with high probability all of the connected components induced by bad vertices after Phase 1 are of size at most $\Delta^4 \log n$.

This concludes the proof of Theorem 6.2. ∎

6.2. Algorithm for $\Delta \ge 55$. The proof of Theorem 6.2 in the previous section (and also the one in [36]) is hard to analyze quantitatively without the aid of $O(\cdot)$ notation to hide large, unspecified constants. It seems to require a very large Δ for the proof to go through, since in each round several Chernoff bounds are applied to make sure that key requirements are met. In what follows we present a different algorithm

with a significantly simpler analysis for Δ-coloring trees with small constant Δ. Its dependence on Δ is polynomial, which is fine if $\Delta = O(1)$.

THEOREM 6.4. *For $\Delta \geq 55$, there exists a* RandLOCAL *algorithm that computes a Δ-coloring of a tree G in $O(\log_\Delta \log n + \log^* n)$ time.*

Proof. We assume that $\Delta = O(1)$ is constant, since otherwise we can apply Theorem 6.2. Our algorithm has three phases:

Phase 1. We execute the following procedure to partially color the graph with colors in $\{4, 5, \ldots, \Delta\}$.

> Initially $U \leftarrow V$.
> For i from Δ down to 4, do the following steps in parallel for each vertex $v \in U$:
> 1. Choose a real number $x(v) \in [0, 1]$ uniformly at random.
> 2. Let $K = \{v \mid x(v) < \min_{u \in N(v) \cap U} x(u)\}$ be the set of all vertices holding local minima.
> 3. Find any MIS $I \supseteq K$ of U. All vertices in I are colored i.
> 4. Set $U \leftarrow U \setminus I$ (remove all colored vertices).

The above procedure ensures that the number of uncolored neighbors of a vertex $v \in U$ is at most $i - 1$ after Step 4. Therefore, at the end of Phase 1, we have $|N(v) \cap U| \leq 3$ for any uncolored vertex v.

The MIS required in Step 3 can be computed in $O(\Delta + \log^* n) = O(\log^* n)$ time [4], or in $O(\Delta^2 + \log^* n) = O(\log^* n)$ time via Theorem 2.1.

Phase 2. We will later show that at the end of Phase 1 the set of vertices $S = \{v \in U \mid |N(v) \cap U| = 3\}$ forms connected components of size at most $O(\log n)$ with probability $\geq 1 - n^{-c}$. We apply Theorem 6.1 to 3-color the set S using the colors $1, 2, 3$ in $O(\log \log n)$ time. We then update $U \leftarrow U \setminus S$ after coloring the vertices in S.

Phase 3. For each vertex v that remains uncolored, the number of its available colors (i.e., $\{1, \ldots, \Delta\} \setminus \{\text{color}(u) \mid u \in N(v) \text{ is colored}\}$) is greater than the number of its uncolored neighbors (i.e., $|N(v) \cap U|$). In other words, there must exist two distinct vertices $u, u' \in N(v)$ that are colored by the same color. We apply an $O(\log^* n)$-time MIS algorithm twice to get a 3-coloring of vertices in U with the colors $1', 2', 3'$. For $i' = 1', 2', 3'$, we then recolor each vertex in color class i' using any available color from its palette (which must be nonempty). This finishes a Δ-coloring of the tree G.

In view of the above, to prove the theorem, it suffices to show that the set $S = \{v \in U \text{ s.t. } |N(v) \cap U| = 3\}$ defined in Phase 2 forms connected components of size at most $O(\log n)$ with probability $\geq 1 - n^{-c}$.

With respect to a distance-3 set T of size t, we select any vertex in $V \setminus T$ as a root to make the tree G rooted. For each $v_i \in T$, we write w_i to denote the parent of v_i in G. We also define $D_i = \bigcup_{u \in N(v_i) \setminus \{w_i\}} N(u)$ for each $v_i \in T$. Observe that for all $v_i, v_j \in T, D_i \cap D_j = \emptyset$, as T is a distance-3 set. We have the following claim, whose proof is deferred to Appendix A.

CLAIM 2. *There is some constant ϵ such that the probability for a vertex v_i to be in S conditioned on arbitrary behavior of vertices not in D_i is at most $\frac{1-\epsilon}{4\Delta^3}$.*

Thus, the probability that all vertices in T belong to S is at most $\left(\frac{1-\epsilon}{4\Delta^3}\right)^t$. By Lemma 6.3, the number of distinct distance-3 sets of size t is at most $4^t n \Delta^{3(t-1)}$.

Therefore, as long as

$$t \geq \frac{(c+1)\log n - \log(\Delta^3)}{\log \frac{1}{1-\epsilon}},$$

with probability at most

$$4^t n \Delta^{3(t-1)} \cdot \left(\frac{1-\epsilon}{4\Delta^3}\right)^t = (n/\Delta^3) \cdot (1-\epsilon)^t = (n/\Delta^3) \cdot 2^{-(c+1)\log n + \log(\Delta^3)} \leq n^{-c},$$

there is no distance-3 set of size t whose vertices are all in S.

Since (for $\Delta \geq 2$) any connected subgraph with the number of vertices being at least $\Delta^2 t$ must contain a distance-3 set of size t, we conclude that, with high probability, S forms connected components of size at most $O(\Delta^2 \log n) = O(\log n)$. ∎

Appendix A. Missing proofs.

In this appendix, we present the proofs of Claim 1 and Claim 2. We make use of the standard Chernoff bound in the proof of Claim 1.

LEMMA A.1 (Chernoff bound). *Let X be the sum of n i.i.d. random 0/1 variables. For any $0 < \delta < 1$, we have the following:*

$$\begin{aligned} \textit{For } 0 < \delta < 1, \quad & \Pr[X \leq (1-\delta)\mathbb{E}[X]] < \exp\left(-\delta^2 \mathbb{E}[X]/2\right). \\ \textit{For } 0 < \delta < 1, \quad & \Pr[X \geq (1+\delta)\mathbb{E}[X]] < \exp\left(-\delta^2 \mathbb{E}[X]/3\right). \\ \textit{For } \delta \geq 1, \quad & \Pr[X \geq (1+\delta)\mathbb{E}[X]] < \exp\left(-\delta \mathbb{E}[X]/3\right). \end{aligned}$$

A.1. Proof of Claim 1. The proof relies on the Large Palette Property $\mathcal{P}_1(v)$ and the Small Degree Property $\mathcal{P}_2(v)$. First, observe that our filtering rules imply that these two properties hold after each round:

- The filtering rule for $i = 1$ guarantees that the Large Palette Property $\mathcal{P}_1(v)$ is met for all vertices that remain after the filtering. Notice that $|\Psi_2(v)| - |N_2'(v)| \geq \frac{\Delta}{200}$ implies $|\Psi_i(v)| \geq \frac{\Delta}{200}$ for all i.
- The filtering rule for $i = 1$ and $1 < i < t$ ensures that the Small Degree Property $\mathcal{P}_2(v)$ holds for all i, since $|N_{i+1}'(v)| \leq \frac{\Delta}{c_{i+1}}$ implies that $|N_{i+1}(v)| \leq \frac{\Delta}{c_{i+1}}$. For the case of $i = 1$, $|N_2(v)| \leq |N_2'(v)| < |\Psi_2(v)| - \frac{\Delta}{200} < \frac{199 \cdot \Delta}{200} = \frac{\Delta}{c_2}$.

The proof of Claim 1 is divided into three cases based on the round number.

CLAIM 3. *The probability that a vertex v is marked as bad in round 1 is at most* $\exp(-\Omega(\Delta))$.

Proof. Recall that a vertex v is marked as bad in round 1 only if $|\Psi_2(v)| - |N_2'(v)| < \frac{\Delta}{200}$. We assume $|N(v)| \geq \frac{199 \cdot \Delta}{200}$, since otherwise v is definitely not marked as bad in round 1.

In round 1, each vertex chooses a color uniformly at random, and a vertex v successfully colors itself if its choice of color is different from all its neighbors. For each $u \in N(v)$, let E_u denote the event that u is colored in the first round. Since the available colors for all vertices are identical, by symmetry, we assume that the color choice of v is fixed. Since the graph is a tree, the events $\{E_u\}_{u \in N(v)}$ are independent. Assuming sufficiently large Δ, we have

$$\Pr[E_u] \geq \left(1 - \frac{1}{\Delta - \sqrt{\Delta}}\right)^{|N(u)|} \geq \left(1 - \frac{1}{\Delta - \sqrt{\Delta}}\right)^{\Delta} \geq \frac{1}{3}.$$

By a Chernoff bound (with $\delta = \frac{79}{199}$, and the expected number of colored neighbors being at least $\frac{199}{200} \cdot \frac{\Delta}{3}$), the number of colored neighbors of v in the first round is at least $\frac{\Delta}{5} = (1-\delta) \cdot \frac{199}{200} \cdot \frac{\Delta}{3}$ with probability at least

$$1 - \exp\left(\frac{-\left(\frac{79}{199}\right)^2 \cdot \left(\frac{199}{200} \cdot \frac{\Delta}{3}\right)}{2}\right).$$

Let $S = \{u_1, u_2, \ldots\}$ be the colored subset of $N(v)$. In what follows, we assume that $|S| \geq \frac{\Delta}{5}$. Conditioned on the event that S is the colored subset of $N(v)$, the choice of the color of each u_j is independent and uniform from $\{1, \ldots, \Delta - \sqrt{\Delta}\} \setminus \{c\}$, where c is the color chosen by v. We will argue that, with high probability, many colors appear repeatedly among the vertices in S, thereby creating a separation between the palette size and the number of uncolored neighbors.

If $\frac{\Delta}{10} - |\bigcup_{j=1}^{\Delta/10}\{\text{Color}(u_j)\}| \geq \frac{\Delta}{200}$, then $|\Psi_2(v)| - |N_2'(v)| \geq \frac{\Delta}{200}$, and v is not marked as bad. Otherwise, each u_j, $j > \frac{\Delta}{10}$, chooses a color that is already chosen by some u_k, $k \leq \frac{\Delta}{10}$, with probability at least

$$\frac{\frac{\Delta}{10} - \frac{\Delta}{200}}{\Delta - \sqrt{\Delta} - 1} \geq \frac{1}{11}.$$

As a result, the expected value of $|S| - |\bigcup_{j=1}^{|S|}\{\text{Color}(u_j)\}|$ is at least $\frac{\Delta}{10} \cdot \frac{1}{11} = \frac{\Delta}{110}$. By a Chernoff bound (with $\delta = \frac{9}{20}$), this value is at least $\frac{\Delta}{200} = (1-\delta) \cdot \frac{\Delta}{110}$ with probability at least

$$1 - \exp\left(\frac{-\left(\frac{9}{20}\right)^2 \cdot \frac{\Delta}{110}}{2}\right) = 1 - \exp\left(\frac{-162 \cdot \Delta}{11}\right).$$

By definition, $|\Psi_2(v)| = |\Psi_1(v)| - |\bigcup_{j=1}^{|S|}\{\text{Color}(u_i)\}|$, and $|N_2'(v)| = |N_1(v) \setminus S| = |N_1(v)| - |S|$. In view of the above, with probability at least

$$1 - \exp\left(\frac{-\left(\frac{79}{199}\right)^2 \cdot \left(\frac{199}{200} \cdot \frac{\Delta}{3}\right)}{2}\right) - \exp\left(\frac{-162\Delta}{11}\right) = 1 - \exp(-\Omega(\Delta)),$$

we have $|\Psi_2(v)| - |N_2(v)| \geq \frac{\Delta}{200}$, and v is not marked as bad. ∎

CLAIM 4. *For $1 < i < t$, the probability that a vertex v that participates in round i is marked as bad in round i is at most* $\exp(-\Omega(\Delta^{0.1}))$.

Proof. Our goal is to show that

$$|N_{i+1}'(v)| \leq \max\left\{\frac{|N_i(v)|}{\exp\left(\frac{c_i}{3 \cdot 200 \cdot e^{200}}\right)}, \Delta^{0.9}\right\} \leq \frac{\Delta}{\min\{c_i \cdot \exp\left(\frac{c_i}{3 \cdot 200 \cdot e^{200}}\right), \Delta^{0.1}\}} = \frac{\Delta}{c_{i+1}}$$

holds with high probability.

Each available color for a vertex v is added to S_v independently with probability $\frac{c_i}{|\Psi_i(v)|}$, so the expected value of $|S_v|$ is c_i. By a Chernoff bound, the event of $|S_v| \leq$

$\frac{\Delta}{2\cdot 200} = (1+\delta)c_i$,[6] where $\delta = (\frac{\Delta}{2\cdot 200\cdot c_i} - 1)$, happens with probability at least

$$1 - \exp\left(\frac{-\left(\frac{\Delta}{2\cdot 200\cdot c_i} - 1\right)\cdot c_i}{3}\right) \geq 1 - \exp\left(\frac{-\Delta}{7\cdot 200}\right).$$

For any $u \in N_i(v)$, a color in $\Psi_i(u) \setminus S_v$ belongs to $S_u \setminus \bigcup_{w\in N_i(u)} S_w$ with probability at least

$$\frac{c_i}{|\Psi_i(u)|}\cdot \prod_{w\in N_i(u)\setminus\{v\}}\left(1 - \frac{c_i}{|\Psi_i(w)|}\right) \geq \frac{c_i}{|\Psi_i(u)|}\cdot\left(1 - \frac{c_i\cdot 200}{\Delta}\right)^{\frac{\Delta}{c_i}-1} \geq \frac{c_i}{e^{200}|\Psi_i(u)|}.$$

Notice that the term $\frac{c_i}{|\Psi_i(u)|}$ is the probability that a color in $\Psi_i(u)\setminus S_v$ is chosen to be in S_u, and the term $\prod_{w\in N_i(u)\setminus\{v\}}(1 - \frac{c_i}{|\Psi_i(w)|})$ is the probability that a color in $\Psi_i(u)\setminus S_v$ does not belong to $\bigcup_{w\in N_i(u)} S_w$.

Under the condition that $|S_v| \leq \frac{\Delta}{2\cdot 200}$, we have $|\Psi_i(u)\setminus S_v| \geq \frac{\Delta}{200} - \frac{\Delta}{2\cdot 200} = \frac{\Delta}{2\cdot 200}$. Then the set $S_u \setminus \bigcup_{w\in N_i(u)} S_w$ is empty with probability at most

$$\left(1 - \frac{c_i}{e^{200}|\Psi_i(u)|}\right)^{\frac{\Delta}{2\cdot 200}} \leq \left(1 - \frac{c_i}{e^{200}\Delta}\right)^{\frac{\Delta}{2\cdot 200}} \leq \exp\left(-\frac{c_i}{2\cdot 200\cdot e^{200}}\right).$$

Hence $u \in N_i(v)$ remains uncolored with probability at most $\exp(-\frac{c_i}{2\cdot 200\cdot e^{200}})$.[7]

Case 1. $\exp(-\frac{c_i}{2\cdot 200\cdot e^{200}})\cdot|N_i(v)| \geq \Delta^{0.1}$. We choose δ to be the number such that $(1+\delta)\exp(-\frac{c_i}{2\cdot 200\cdot e^{200}}) = \exp(-\frac{c_i}{3\cdot 200\cdot e^{200}})$. Notice that $\delta = \exp(\frac{c_i}{6\cdot 200\cdot e^{200}}) - 1 \geq \exp(\frac{1}{6\cdot 200\cdot e^{200}}) - 1$, which is at least a positive constant. By a Chernoff bound on all vertices in $N_i(v)$, we have $|N'_{i+1}(v)| \leq \exp(-\frac{c_i}{3\cdot 200\cdot e^{200}})\cdot|N_i(v)|$ with probability at least

$$1 - \exp\left(\frac{-\min\{\delta,\delta^2\}\cdot\exp(-\frac{c_i}{2\cdot 200\cdot e^{200}})\cdot|N_i(v)|}{3}\right) \geq 1 - \exp(-\Omega(\Delta^{0.1})).$$

Case 2. $\exp(-\frac{c_i}{2\cdot 200\cdot e^{200}})\cdot|N_i(v)| < \Delta^{0.1}$. By a Chernoff bound (with $\delta = \Delta^{0.8} - 1$) on all vertices in $N_i(v)$, we conclude that $|N'_{i+1}(v)| \leq \Delta^{0.8}\cdot\exp(-\frac{c_i}{2\cdot 200\cdot e^{200}})\cdot|N_i(v)| \leq \Delta^{0.9}$ with probability at least

$$1 - \exp\left(\frac{-(\Delta^{0.8}-1)\cdot\exp(-\frac{c_i}{2\cdot 200\cdot e^{200}})\cdot|N_i(v)|}{3}\right) \geq 1 - \exp(-\Omega(\Delta^{0.9})).$$

In both cases, we have $|N'_{i+1}(v)| \leq \max\{\frac{|N_i(v)|}{\exp(\frac{c_i}{3\cdot 200\cdot e^{200}})}, \Delta^{0.9}\} = \frac{\Delta}{c_{i+1}}$, and so v is marked as bad with probability at most $\max\{\exp(-\Omega(\Delta^{0.9})), \exp(-\Omega(\Delta^{0.1}))\} = \exp(-\Omega(\Delta^{0.1}))$. ∎

Claim 5. *The probability that a vertex v that participates in round t is marked as bad in round t is at most* $\exp(-\Omega(\Delta^{0.1}))$.

[6] The purpose of bounding $|S_v|$ is to make sure that the set $\Psi_i(u)\setminus S_v$ is large enough for all $u \in N_i(v)$. If v chooses too many colors, then it is more likely that $u \in N_i(v)$ remains uncolored after this round, and this implies an unwanted positive correlation between neighbors of v not being colored.

[7] The calculation of this probability considers colors in $\Psi_i(u)\setminus S_v$ instead of $\Psi_i(u)$. This special treatment of v is to create independence among the vertices in $N_i(v)$, allowing us to use a Chernoff bound on $N_i(v)$ to bound $|N'_{i+1}(v)|$.

Proof. A color in $\Psi_i(v)$ belongs to $S_v \setminus \bigcup_{u \in N_i(v)} S_u$ with probability at least

$$\frac{c_i}{|\Psi_i(v)|} \cdot \prod_{u \in N_i(v)} \left(1 - \frac{c_i}{|\Psi_i(u)|}\right) \geq \frac{c_i}{|\Psi_i(v)|} \cdot \left(1 - \frac{200 \cdot c_i}{\Delta}\right)^{\frac{\Delta}{c_i}} \geq \frac{c_i}{1.1 \cdot e^{200} |\Psi_i(v)|}.$$

Therefore, with probability at most

$$\left(1 - \frac{c_i}{1.1 \cdot e^{200} |\Psi_i(v)|}\right)^{|\Psi_i(v)|} \leq \exp\left(-\frac{\Delta^{0.1}}{1.1 \cdot e^{200}}\right),$$

the vertex v remains uncolored and is marked as bad. ∎

Claim 1 follows from the above three claims.

A.2. Proof of Claim 2. For notational simplicity, we write $v \stackrel{\text{def}}{=} v_i$ and $w \stackrel{\text{def}}{=} w_i$.

Observe that $v \in S$ iff (i) v is not colored in Phase 1, and (ii) for each $i = \Delta$ down to 4, at most one neighbor of v is colored with i in Step 2. In addition, in the beginning of Step 1, we must have $|N(v) \cap U| = i$ for each $i = \Delta$ down to 4.

Now, assume that we are at the beginning of Step 1, the vertex v still has no neighbors of repeated colors, and $|N(v) \cap U| = i$. The probability that a neighbor $u \in N(v) \setminus \{w\}$ is colored i in Step 2 when $x(v) = z \in [0, 1]$ is at least

$$\int_{y=0}^{y=z} \Pr[\forall r \in (N(u) \cap U) \setminus \{v\}, x(r) \leq y] dy \geq \int_{y=0}^{y=z} (1-y)^{i-1} dy = \frac{1 - (1-z)^i}{i}.$$

Notice that the variable y represents the random variable $x(u)$, and u is colored when (i) $y \in [0, z)$, and (ii) $x(r) \in (y, 1]$ for all $r \in (N(u) \cap U) \setminus \{v\}$. Also notice that $|(N(u) \cap U) \setminus \{v\}| \leq i - 1$.

We write $p_i(z) \stackrel{\text{def}}{=} \frac{1-(1-z)^i}{i}$. Then the probability that at most one neighbor of v is colored with i in Step 2 can be upper bounded by

$$\begin{aligned} P_i &= \int_{z=0}^{z=1} \Pr[\text{binom}(|(N(v) \cap U) \setminus \{w\}|, p_i(z)) \leq 1] dz \\ &\leq \int_{z=0}^{z=1} \left[(1 - p_i(z))^{i-1} + (i-1) p_i(z) (1 - p_i(z))^{i-2}\right] dz. \end{aligned}$$

The notation $\text{binom}(n, p)$ denotes the binomial random variable with parameters (n, p). Notice that $|(N(v) \cap U) \setminus \{w\}| \geq i - 1$.

When $i = 4$, the value of P_4 is about 0.88718. As i increases, P_i decreases monotonically, approaching 0.73576. By a numerical calculation, so long as $\Delta \geq 55$, the probability that v is in S conditioned on arbitrary behavior of vertices not in $\bigcup_{u \in N(v) \setminus \{w\}} N(u)$ is at most

$$\prod_{i=4}^{i=\Delta} P_i < \frac{1}{4\Delta^3},$$

as desired.

REFERENCES

[1] R. Bar-Yehuda, K. Censor-Hillel, and G. Schwartzman, *A distributed* $(2+\epsilon)$*-approximation for vertex cover in* $O(\log\Delta/\epsilon\log\log\Delta)$ *rounds*, J. ACM, 64 (2017), 23.

[2] L. Barenboim and M. Elkin, *Sublogarithmic distributed MIS algorithm for sparse graphs using Nash-Williams decomposition*, Distributed Comput., 22 (2010), pp. 363–379.

[3] L. Barenboim, M. Elkin, and U. Goldenberg, *Locally-iterative distributed* $(\Delta+1)$*-coloring below Szegedy-Vishwanathan barrier, and applications to self-stabilization and to restricted-bandwidth models*, in Proceedings of the 38th ACM Symposium on Principles of Distributed Computing (PODC), ACM, New York, 2018, pp. 437–446, https://doi.org/10.1145/3212734.3212769.

[4] L. Barenboim, M. Elkin, and F. Kuhn, *Distributed* $(\Delta+1)$*-coloring in linear (in* Δ*) time*, SIAM J. Comput., 43 (2014), pp. 72–95, https://doi.org/10.1137/12088848X.

[5] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider, *The locality of distributed symmetry breaking*, J. ACM, 63 (2016), 20.

[6] M. Ben-Or, *Another advantage of free choice: Completely asynchronous agreement protocols*, in Proceedings of the 2nd ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC), ACM, New York, 1983, pp. 27–30, https://doi.org/10.1145/800221.806707.

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

[8] B. Bollobás, *Chromatic number, girth and maximal degree*, Discrete Math., 24 (1978), pp. 311–314, https://doi.org/10.1016/0012-365X(78)90102-4.

[9] B. Bollobás, *Extremal Graph Theory*, London Math. Soc. Monogr. 11, Academic Press, London, 1978.

[10] G. Bracha, *An asynchronous* $(n-1)/3$*-resilient consensus protocol*, in Proceedings of the 3rd ACM Symposium on Principles of Distributed Computing (PODC), ACM, New York, 1984, pp. 154–162, https://doi.org/10.1145/800222.806743.

[11] S. Brandt, O. Fischer, J. Hirvonen, B. Keller, T. Lempiäinen, J. Rybicki, J. Suomela, and J. Uitto, *A lower bound for the distributed Lovász local lemma*, in Proceedings of the 48th ACM Symposium on Theory of Computing (STOC), ACM, New York, 2016, pp. 479–488.

[12] Y.-J. Chang, Q. He, W. Li, S. Pettie, and J. Uitto, *The complexity of distributed edge coloring with small palettes*, in Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA), ACM, New York, SIAM, Philadelphia, 2018, pp. 2633–2652, https://doi.org/10.1137/1.9781611975031.168.

[13] Y.-J. Chang, W. Li, and S. Pettie, *An optimal distributed* $(\Delta+1)$*-coloring algorithm?*, in Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC), ACM, New York, 2018, pp. 445–456, https://doi.org/10.1145/3188745.3188964.

[14] K.-M. Chung, S. Pettie, and H.-H. Su, *Distributed algorithms for the Lovász local lemma and graph coloring*, in Proceedings of the 33rd ACM Symposium on Principles of Distributed Computing (PODC), ACM, New York, 2014, pp. 134–143.

[15] X. Dahan, *Regular graphs of large girth and arbitrary degree*, Combinatorica, 34 (2014), pp. 407–426, https://doi.org/10.1007/s00493-014-2897-6.

[16] M. Elkin, S. Pettie, and H. H. Su, $(2\Delta-1)$*-edge coloring is much easier than maximal matching in the distributed setting*, in Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM, New York, SIAM, Philadelphia, 2015, pp. 355–370.

[17] L. Feuilloley and P. Fraigniaud, *Randomized local network computing*, in Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), ACM, New York, 2015, pp. 340–349.

[18] M. Fischer, *Improved deterministic distributed matching via rounding*, in Proceedings of the 31st International Symposium on Distributed Computing (DISC 2017), A. W. Richa, ed., Leibniz Internat. Proc. Informatics (LIPIcs) 91, Schloss Dagstuhl–Leibniz–Zentrum fuer Informatik, Dagstuhl, Germany, 2017, 17, https://doi.org/10.4230/LIPIcs.DISC.2017.17.

[19] M. Fischer and M. Ghaffari, *Sublogarithmic distributed algorithms for Lovász local lemma with implications on complexity hierarchies*, in Proceedings of the 31st International Symposium on Distributed Computing (DISC), Vienna, Austria, 2017, 18.

[20] M. J. Fischer, N. A. Lynch, and M. Paterson, *Impossibility of distributed consensus with one faulty process*, J. ACM, 32 (1985), pp. 374–382, https://doi.org/10.1145/3149.214121.

[21] P. Fraigniaud, M. Heinrich, and A. Kosowski, *Local conflict coloring*, in Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE,

Washington, DC, 2016, pp. 625–634.
[22] M. Ghaffari, *An improved distributed algorithm for maximal independent set*, in Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM, New York, SIAM, Philadelphia, 2016, pp. 270–277, https://doi.org/10.1137/1.9781611974331.ch20.
[23] M. Ghaffari, D. G. Harris, and F. Kuhn, *On derandomizing local distributed algorithms*, in Proceedings of the 59th IEEE Symposium on Foundations of Computer Science (FOCS), IEEE, Washington, DC, 2018, pp. 662–673.
[24] M. Ghaffari and H.-H. Su, *Distributed degree splitting, edge coloring, and orientations*, in Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM, New York, SIAM, Philadelphia, 2017, pp. 2505–2523, https://doi.org/10.1137/1.9781611974782.166.
[25] D. G. Harris, J. Schneider, and H. Su, *Distributed* $(\Delta+1)$*-coloring in sublogarithmic rounds*, J. ACM, 65 (2018), 19, https://doi.org/10.1145/3178120.
[26] V. King and J. Saia, *Byzantine agreement in polynomial expected time*, in Proceedings of the 45th ACM Symposium on Theory of Computing (STOC), ACM, New York, 2013, pp. 401–410, https://doi.org/10.1145/2488608.2488658.
[27] A. Korman, S. Kutten, and D. Peleg, *Proof labeling schemes*, Distributed Comput., 22 (2010), pp. 215–233, https://doi.org/10.1007/s00446-010-0095-3.
[28] A. Korman, J.-S. Sereni, and L. Viennot, *Toward more localized local algorithms: Removing assumptions concerning global knowledge*, Distributed Comput., 26 (2013), pp. 289–308.
[29] K. Kothapalli and S. V. Pemmaraju, *Super-fast* 3*-ruling sets*, in Proceedings of the IARCS Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), LIPIcs 18, Schloss Dagstuhl–Leibniz–Zentrum für Informatik, Dagstuhl, Germany, 2012, pp. 136–147.
[30] F. Kuhn, T. Moscibroda, and R. Wattenhofer, *Local computation: Lower and upper bounds*, J. ACM, 63 (2016), 17.
[31] N. Linial, *Locality in distributed graph algorithms*, SIAM J. Comput., 21 (1992), pp. 193–201, https://doi.org/10.1137/0221015.
[32] M. Naor, *A lower bound on probabilistic algorithms for distributive ring coloring*, SIAM J. Discrete Math., 4 (1991), pp. 409–412, https://doi.org/10.1137/0404036.
[33] M. Naor and L. Stockmeyer, *What can be computed locally?*, SIAM J. Comput., 24 (1995), pp. 1259–1277, https://doi.org/10.1137/S0097539793254571.
[34] A. Panconesi and R. Rizzi, *Some simple distributed algorithms for sparse networks*, Distributed Comput., 14 (2001), pp. 97–100.
[35] A. Panconesi and A. Srinivasan, *On the complexity of distributed network decomposition*, J. Algorithms, 20 (1996), pp. 356–374.
[36] S. Pettie and H.-H. Su, *Distributed algorithms for coloring triangle-free graphs*, Inform. and Comput., 243 (2015), pp. 263–280.
[37] M. O. Rabin, *Randomized Byzantine generals*, in Proceedings of the 24th IEEE Symposium on Foundations of Computer Science (FOCS), IEEE, Washington, DC, 1983, pp. 403–409, https://doi.org/10.1109/SFCS.1983.48.
[38] J. Schneider and R. Wattenhofer, *An optimal maximal independent set algorithm for bounded-independence graphs*, Distributed Comput., 22 (2010), pp. 349–361.
[39] M. Sipser, *Introduction to the Theory of Computation*, 3rd ed., International Thomson Publishing, London, 2012.