2016 IEEE 57th Annual Symposium on Foundations of Computer Science

An Exponential Separation Between Randomized and Deterministic Complexity in the LOCAL Model

Yi-Jun Chang
University of Michigan
Ann Arbor, MI
cyijun@umich.edu

Tsvi Kopelowitz
University of Michigan
Ann Arbor, MI
kopelot@gmail.com

Seth Pettie
University of Michigan
Ann Arbor, MI
pettie@umich.edu

***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. 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:**

1) **Building on the recent randomized lower bounds of Brandt et al. [1], 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.**
2) **We prove 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.**
3) **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 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 also 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 $2^{O(\sqrt{\log \log n})}$ term in the complexities of the best MIS and $(\Delta+1)$-coloring algorithms without *also* improving the $2^{O(\sqrt{\log n})}$-round Panconesi-Srinivasan algorithm.**

***Keywords*-coloring; distributed algorithm; local model; symmetry breaking**

I. Introduction

One of the central problems of theoretical computer science is to determine the value of *random bits*. If the distinction is between computable vs. incomputable functions, random bits are provably useless in centralized models (Turing machines). However, this is not true in the distributed world! The celebrated Fischer-Lynch-Patterson theorem [2] states that asynchronous deterministic agreement is impossible with one unannounced failure, yet it is possible to accomplish with probability 1 using randomization [3].

In this paper we examine the value of random bits in Linial's [4] LOCAL model, which, for the sake of clarity, we bifurcate into two models RandLOCAL and DetLOCAL. In both models the graph $G = (V, E)$ represents the topology of the communications 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. (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

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

[1]The assumption that global parameters are common knowledge can sometimes be removed; see Korman, Sereni, and Viennot [5].

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.72

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 node can locally generate a random $O(\log n)$-bit ID, which is unique with probability $1/\mathrm{poly}(n)$.

Early work in the LOCAL models suggested that randomness is of limited help. Naor [6] showed that Linial's $\Omega(\log^* n)$ lower bound [4] for 3-coloring the ring holds even in RandLOCAL, and Naor and Stockmeyer [7] proved that the class of problems solvable by $O(1)$-round algorithms is the same in RandLOCAL and DetLOCAL. See [8] for a generalization of the Naor-Stockmeyer derandomization. However, in the intervening decades we have seen numerous examples of symmetry breaking algorithms for RandLOCAL that are substantially faster than their counterparts in DetLOCAL. The best DetLOCAL maximal independent set (MIS) algorithms run in $O(\min\{\Delta + \log^* n, 2^{O(\sqrt{\log n})}\})$ time [9], [10], whereas the fastest RandLOCAL MIS algorithm [11] runs in $O(\log \Delta + 2^{O(\sqrt{\log \log n})})$ time. The best DetLOCAL maximal matching algorithm runs in $O(\min\{\Delta+\log^* n, \log^4 n\})$ time [12], [13] whereas the best in the RandLOCAL model is $O(\log \Delta+\log^4 \log n)$ [14]. The best DetLOCAL algorithms for $(\Delta+1)$-coloring and $O(\Delta)$-coloring run in $\min\{\tilde{O}(\sqrt{\Delta})+\log^* n, 2^{O(\sqrt{\log n})}\}$ time [15], [16], [10]. In the RandLOCAL model $O(\Delta)$-coloring can be solved in $2^{O(\sqrt{\log \log n})}$ time and in $O(\log^* n)$ time for sufficiently large $\Delta \gg \log n$. Recently Harris, Schneider, and Su [17] gave a RandLOCAL $(\Delta+1)$-coloring algorithm running in $O(\sqrt{\log \Delta}+2^{O(\sqrt{\log \log n})})$ time. See [14] for an extensive survey.

Graph Shattering: The randomized symmetry breaking algorithms cited above are exponentially faster *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, $2^{O(\sqrt{\log n})}$ becomes $2^{O(\sqrt{\log \log n})}$. This second phenomenon is no coincidence! It is a direct result of the *graph shattering* approach to symmetry breaking used in [14] and further in [18], [19], [20], [11], [21], [17], [22], [23]. 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 $\mathrm{poly}(\log n)$, 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 [4] himself, who argued that any algorithm for coloring degree-Δ 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 degree-Δ graph with girth $\Omega(\log_\Delta n)$ and chromatic number $\chi = \Omega(\Delta/\log \Delta)$, hence[2] (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 [25], 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 [26] 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. [1] 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 $1/\mathrm{poly}(n)$) such as *sinkless orientation*, *sinkless coloring*, and Δ-coloring. Refer to Section II for definitions of these problems. Since the existence 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 [19], [11] for upper bounds on the distributed LLL.

A. 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* problems (refer to Section II for a definition of LCLs), a class that includes essentially any natural symmetry breaking problem.

1) **Separation of** RandLOCAL **and** DetLOCAL**.** We extend Brandt et al.'s [1] randomized lower bound as follows: on degree-Δ graphs, Δ-coloring 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 [27] 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 [1]

[2]Linial [4] actually only used the existence of Δ-regular graphs with high girth and chromatic number $\Omega(\sqrt{\Delta})$. See [24] for constructions with chromatic number $\Omega(\Delta/\log \Delta)$.

lower bound up to a $\log^* n$ additive term. A more complicated algorithm for Δ-coloring trees could be derived from [23], for $\Delta > \Delta_0$ and some very large constant Δ_0.[3]

2) **Randomized lower bounds imply deterministic lower bounds.** We give a second, more generic proof that Δ-coloring trees takes $\Omega(\log_\Delta n)$ time. 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. This reduction can be parameterized in many different ways. Under a different parametrization it shows 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. For example, if one were to develop a deterministic $O(\sqrt{\log n})$-time MIS or maximal matching algorithm—almost matching one of the KMW [25] lower bounds—it would immediately imply an $O(\log \Delta(\log^* n - \log^* \Delta + 1))$-time MIS/maximal matching algorithm, which almost matches the *other* KMW lower bound. By some strange coincidence, [14] 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.

3) **Deterministic lower bounds imply randomized lower bounds.** We prove that for any LCL problem, its RandLOCAL complexity on instances of size n is at least its DetLOCAL complexity on 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 [14], [19], [20], [11], [17], [18], [22], [23] 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 [14], [11], [17], [20] without also improving the $2^{O(\sqrt{\log n})}$-time DetLOCAL algorithms of Panconesi and Srinivasan [10], and it is impossible to improve the $O(\log^4 \log n)$ term in the RandLOCAL maximal matching algorithm of [14] without also improving the $O(\log^4 n)$ DetLOCAL maximal matching algorithm of [13].

[3]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.

II. Preliminaries

Graph 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 $N^r(v) = \{u \mid \mathrm{dist}_G(v, u) \leq r\}$ be the set of all vertices within distance r of v.

Locally Checkable Labeling: The class of *Locally Checkable Labeling* (LCL) [7] problems are 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 finite 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 *non-solution* 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 an assignment $V \to \{1, 2, \ldots, k\}$ such that for each edge $\{u, v\} \in E$, u and v are assigned to 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. [1] 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.

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 [4] coloring algorithm, which recolors the vertices using a smaller palette.

Theorem 1 ([4])**.** *Let G be a graph which has been k-colored. Then it is possible to deterministically re-color G using $5\Delta^2 \log k$ colors in one round.*

Theorem 2 ([4])**.** *There exists a universal constant $\beta > 0$ such that there is a* DetLOCAL *algorithm that computes a $\beta \cdot \Delta^2$-coloring of a graph in $O(\log^* n - \log^* \Delta + 1)$ time.*

III. The Necessity of Graph Shattering

Theorem 3 establishes that the graph shattering technique [14] is optimal and unavoidable in RandLOCAL. In particular, the randomized complexity of any symmetry breaking problem always hinges on its deterministic complexity.

Theorem 3. *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*

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

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 $\mathrm{ID}(v) \in \{0,1\}^{c \log n}$. 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. Regardless of Δ,

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

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,

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

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 $N = 2^{n^2}, t(N, \Delta), r(N, \Delta)$, 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 executed on any member of $\mathcal{G}_{n,\Delta}$. ∎

Remark 1. *Theorem 3 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 [20]), arboricity/degeneracy (as in [27], [14]), or neighborhood growth (as in [28]).*

Naor and Stockmeyer [7] proved that the class of truly local ($O(1)$-time) problems in RandLOCAL and DetLOCAL is identical for bounded Δ. Theorem 3 implies something slightly stronger, since $\log^* n$ and $\log^*(\sqrt{\log n})$ differ by an additive constant.

Corollary 1. *Any* RandLOCAL *algorithm for an LCL running in $t(n) = 2^{O(\log^* n)}$ time can be derandomized without asymptotic penalty. The corresponding* DetLOCAL *algorithm runs in $O(t(n))$ time.*

IV. Lower bounds for Δ-coloring Trees

In this section we prove that on degree-Δ 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. [1] 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 a that 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 [1] for $\Delta = 3$. It is straightforward to go through the details of the proof and track the dependence on Δ.

Lemma 1 ([1])**.** *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 $\forall e \in E$, $\mathcal{A}$ 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 $\forall v \in V$, $\mathcal{A}'$ outputs a forbidden configuration at v with probability at most $2\Delta p^{1/3}$.*

Lemma 2 ([1])**.** *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 $\forall v \in V$, $\mathcal{A}'$ 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 $\forall e \in E$, $\mathcal{A}$ outputs a forbidden configuration at e with probability at most $4p^{1/(\Delta+1)}$.*

The following theorem generalizes Corollary 25 in [1] to allow non-constant Δ and arbitrary failure probability p.

Theorem 4. *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 $\epsilon > 0$.*

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 exist a bipartite Δ-regular graph with girth $\Omega(\log_\Delta n)$; see [29], [30]. Such graphs are trivially Δ-edge colorable. Moreover, any Δ-coloring of such a graph is also a valid Δ-sinkless coloring. Applying Lemmas 1 and 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 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 2 is an immediate consequence of Theorem 4.

Corollary 2. *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 does not immediately extend to DetLOCAL. Recall that in the DetLOCAL model vertices are initially endowed with $O(\log n)$-bit IDs whereas in RandLOCAL they are undifferentiated.

Theorem 5. *Any* DetLOCAL *algorithm that Δ-colors degree-Δ graphs with girth $\Omega(\log_\Delta n)$ or degree-Δ 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 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 $G' = (V, \{(u, v) \mid \mathrm{dist}_G(u, v) \leq 2t + 1\})$. The maximum degree Δ' in G' is clearly less than n. We apply one step of Linial's recoloring algorithm (Theorem 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 $\mathcal{A}_{\mathsf{Rand}}$ takes $\Omega(\min\{\log_\Delta \log(1/p), \log_\Delta n\}) = \Omega(\log_\Delta n)$ time. ∎

V. Gaps in Deterministic 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, does increasing the number of rounds from $\Theta(\log^* n)$ to $\Theta(\log\log n)$ allow one to solve more problems? In this section, we will demonstrate a general technique that allows one to speedup deterministic algorithms in the DetLOCAL model. Based on this technique, we demonstrate the existence of a "gap" in possible DetLOCAL complexities.

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 time complexity of *any* LCL problem on a *hereditary* graph class is either $\Omega(\log n)$ or $O(\log^* n)$.

Theorem 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 β be the universal constant from Theorem 2. Suppose that the cost of $\mathcal{A}$ on instances of $\mathcal{P}$ with n vertices, where the instances are taken from a hereditary graph class, is at most $f(\Delta)+\epsilon\log_\Delta n$ time, 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 an instance of $\mathcal{P}$. 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

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

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 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:

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

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 6 with Corollary 2 and setting $f(\Delta)=O(1)$ provides a new proof of Theorem 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 $1-1/\mathrm{poly}(n)$. From Corollary 2 any DetLOCAL algorithm that Δ-colors a degree-Δ tree requires $\Omega(\log_\Delta\log n)$ time. However, Theorem 6 states that any DetLOCAL algorithm running in $O(1)+o(\log_\Delta n)$ time can be sped up to run in $O\left(\log^* n-\log^*\Delta+1\right)$ time. This contradicts the lower bound whenever $\log_\Delta\log n\gg\log^* n-\log^*\Delta+1$. Hence Δ-coloring a degree-Δ tree takes $\Omega(\log_\Delta n)$ time in DetLOCAL for small enough Δ such that $\log_\Delta\log n\gg\log^* n-\log^*\Delta+1$.

Another consequence of Theorem 2 is that the deterministic time complexity of a problem can either be solved very efficiently (i.e. in $O\left((1+f(\Delta)(\log^* n-\log^*\Delta+1)\right)$ time) or requires $\Omega(f(\Delta)+\log_\Delta n)$ time, which is at least the order of the diameter when the underlying graph is a complete regular tree. Such a consequence is the strongest when Δ is small. For example, if Δ is a constant, Theorem 2 implies the following corollary:

Corollary 3. *The time complexity of any LCL problem on any hereditary graph class that has constant Δ in the* DetLOCAL *model is either $\Omega(\log n)$ or $O(\log^* n)$.*

A simple adaptation of the proof of Theorem 6 shows an even stronger dichotomy when $\Delta=2$.

Theorem 7. *The* DetLOCAL *time 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 interpretation of the time complexity requirement in Theorems 6 and 7 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 $\Delta = 2$. If we allow the possibility for an algorithm to see the entire graph, then the algorithm can solve the problem globally.

Given a $O(\sqrt{\log n})$-time deterministic algorithm, one may feel that it is possible to use Theorem 6 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 6 does not apply. Nonetheless, Linial's coloring technique can be made to speed up algorithms with time complexity of the form $f(\Delta) + g(n)$.

Theorem 8. *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 $\mathcal{P}$. Suppose that the runtime of the algorithm $\mathcal{A}$ on instances of $\mathcal{P}$ from a hereditary graph class is $O(\log^k \Delta + \log^{\frac{k}{k+1}} n)$. 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: 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 $\epsilon_1 \log^k \Delta + \epsilon_2 \ell^{\frac{k}{k+1}}$, 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 6, the algorithm $\mathcal{A}'$ first produces shortened ID 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

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

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 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. By setting ϵ as a large enough number such that $\epsilon_1 + \epsilon_2 \left(4(\epsilon + r + \log\beta)\right)^{\frac{k}{k+1}} \leq \epsilon$, 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 ID 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

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

■

A note about MIS lower bounds: Kuhn, Moscibroda, and Wattenhofer [25] showed that for a variety of problems (including MIS) there is a lower bound of $\min(\log\Delta/\log\log\Delta, \sqrt{\log n/\log\log n})$ rounds. The lower bound graph they used to prove such these result has $\log\Delta/\log\log\Delta = O(\sqrt{\log n/\log\log n})$. By Theorem 8, setting $k = 1$ implies that if there is a deterministic algorithm for MIS that runs in $O(\sqrt{\log n})$ time, then there is another deterministic algorithm running in $O(\log\Delta(\log^* n - \log^*\Delta + 1))$ time. Interestingly, Barenboim, Elkin, Pettie, and Schneider [14] 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.

VI. Algorithms for Δ-coloring Trees

In Section IV, we showed that the problem of Δ-coloring on trees has an $\Omega(\log_\Delta n)$ deterministic lower bound and an $\Omega(\log_\Delta \log n)$ randomized lower bound. These lower bounds have matching upper bounds, up to an additive $\log^* n$ term.

The algorithm of Barenboim and Elkin [27] 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 9 ([27]). *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 [23] 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 [19] 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 10 ([23]). *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 10 makes it difficult to calculate a specific Δ_0 for which the theorem applies. Moreover, the proof is only sketched. We address both of these issues. First, we provide a simple algorithm and elementary proof of Theorem 10. Second, we prove Theorem 11, which combines Theorem 10 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 $\Delta \geq 55$.

A. A simple proof of Theorem 10.

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) For any two distinct vertices $u, v \in S$, we have $u \notin N^{k-1}(v)$.
2) Let $G^k = (V, E^k)$, where there is an edge $(u, v) \in E^k$ if and only if $\mathrm{dist}_G(u, v) = k$. Then S is connected in G^k.

The following lemma is used in the proof of Theorem 10.

Lemma 3 ([14]). *The number of distinct distance-k sets of size t is less than $4^t \cdot n \cdot \Delta^{k(t-1)}$.*

Proof of Theorem 10: Our algorithm has two phases. The first phase, which takes $t = O(\log^* \Delta)$ 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 non-bad 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 $i > 2$.

Notice that c_i is a constant, for all i. 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 palette size and the degree (of the graph induced by uncolored vertices) so that we can color a large fraction of vertices in each round.

For each $1 \leq i \leq t$, the i^{th} round consists of two constant time sub-routines 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 $\mathrm{Color}(v)$. 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 $\mathrm{Color}(v)$.
3) $\Psi_{i+1}(v) \leftarrow \Psi_i(v) \setminus \{\mathrm{Color}(u) \mid u \in N_i(v)$ 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,

$$N'_i(v) = N_{i-1}(v) \setminus \left\{ u \;\middle|\; \begin{array}{l} u \text{ is permanently colored} \\ \text{in ColorBidding}(i-1) \end{array} \right\}.$$

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 a connected component induced by bad vertices has size at most $\Delta^4 \log n$. Hence we use Theorem 9 to $\sqrt{\Delta}$-color such connected components using the $\sqrt{\Delta}$ reserved colors. For simplicity, if this phase lasts for too long (which may happen with low probability) the algorithm just stops and fails.

Runtime: The runtime of Phase 1 is $t = O(\log^* \Delta)$ rounds. The runtime of Phase 2 is $O\big(\log_{\sqrt{\Delta}} \big(\Delta^4 \log n\big) + \log^* \big(\Delta^4 \log n\big)\big) = O\left(\log_\Delta \log n + \log^* n\right)$. 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 claims, which are proved by applying standard Chernoff bounds. See [31] for details.

Claim 1. *The probability that a vertex v is marked as bad in round $i = 1$ is at most $\exp(-\Omega(\Delta))$; this event only depends on the random bits chosen by vertices in $N^2(v)$.*

Claim 2. *The probability that a vertex v that participates in round $1 < i < t$ is marked as bad in round i is at most $\exp(-\Omega(\Delta^{0.1}))$; this event only depends on the random bits chosen by vertices in $N^2(v)$.*

Claim 3. *The probability that a vertex v that participates in round $i = t$ is marked as bad in round i is at most $\exp(-\Omega(\Delta^{0.1}))$; this event only depends on the random bits chosen by vertices in $N^2(v)$.*

By the union bound for all rounds in Phase 1, the probability that any vertex v becomes a bad vertex after Phase 1 is $O(\log^* \Delta)\exp(-\text{poly}(\Delta)) = \exp(-\text{poly}(\Delta))$, regardless of the choice of random bits for all vertices not in $N^2(v)$. 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 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 can be 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 10. ■

B. Algorithm for $\Delta \geq 55$.

The proof of the previous section (and that of [23]) 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 11. *For $\Delta \geq 55$, there exists a* RandLOCAL *algorithm that computes a Δ-coloring of a tree in $O(\log_\Delta \log n + \log^* n)$ time.*

Proof: Due to the antiquated page constraint, we only present the algorithm and omit the analysis. See [31] for a full proof. We assume that $\Delta = O(1)$ is constant, since otherwise we can apply Theorem 10. 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 downto 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 $K = \left\{ v \;\middle|\; x(v) < \min_{u \in N(v) \cap U} x(u) \right\}$ 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 [9], or in $O(\Delta^2 + \log^* n) = O(\log^* n)$ time via Theorem 2.

Phase 2: We can show that the set of vertices $S = \{v \in U \text{ s.t. } |N(v) \cap U| = 3\}$ form connected components of size at most $O(\log n)$ with probability $\geq 1 - n^{-c}$ [31]. Hence we apply Theorem 9 to 3-color the set S (using the colors $1, 2, 3$) in $O(\log \log n)$ time. We then update $U = U \setminus S$ 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 strictly greater than the number of its uncolored neighbors (i.e. $|N(v) \cap U|$). We apply an $O(\log^* n)$-time MIS algorithm twice to get a 3-coloring of vertices in U (with three colors $1', 2', 3'$). For $i' = 1', 2', 3'$, we recolor each vertex in color class i' using any available color from its palette. ■

REFERENCES

[1] 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 48th ACM Symposium on the Theory of Computing (STOC)*, 2016, pp. 479–488.

[2] M. J. Fischer, N. A. Lynch, and M. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, no. 2, pp. 374–382, 1985.

[3] 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)*, 1983, pp. 27–30.

[4] N. Linial, "Locality in distributed graph algorithms," *SIAM J. Comput.*, vol. 21, no. 1, pp. 193–201, 1992.

[5] A. Korman, J.-S. Sereni, and L. Viennot, "Toward more localized local algorithms: removing assumptions concerning global knowledge." *Distributed Computing*, vol. 26, no. 5–6, pp. 289–308, 2013.

[6] M. Naor, "A lower bound on probabilistic algorithms for distributive ring coloring," *SIAM J. Discrete Mathematics*, vol. 4, no. 3, pp. 409–412, 1991.

[7] M. Naor and L. J. Stockmeyer, "What can be computed locally?" *SIAM J. Comput.*, vol. 24, no. 6, pp. 1259–1277, 1995.

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

[9] L. Barenboim, M. Elkin, and F. Kuhn, "Distributed $(\Delta + 1)$-coloring in linear (in Δ) time," *SIAM J. Comput.*, vol. 43, no. 1, pp. 72–95, 2014.

[10] A. Panconesi and A. Srinivasan, "On the complexity of distributed network decomposition," *J. Algor.*, vol. 20, no. 2, pp. 356–374, 1996.

[11] M. Ghaffari, "An improved distributed algorithm for maximal independent set," in *Proceedings 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2016, pp. 270–277.

[12] A. Panconesi and R. Rizzi, "Some simple distributed algorithms for sparse networks," *Distributed Computing*, vol. 14, no. 2, pp. 97–100, 2001.

[13] M. Hańćkowiak, M. Karoński, and A. Panconesi, "On the distributed complexity of computing maximal matchings," *SIAM J. Discrete Mathematics*, vol. 15, no. 1, pp. 41–57 (electronic), 2001.

[14] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider, "The locality of distributed symmetry breaking," *J. ACM*, vol. 63, no. 3, pp. 20:1–20:45, 2016.

[15] L. Barenboim, "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)*, 2015, pp. 345–354.

[16] P. Fraigniaud, M. Heinrich, and A. Kosowski, "Local conflict coloring," in *Proceedings 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2016.

[17] D. Harris, J. Schneider, and H.-H. Su, "Distributed $(\Delta + 1)$-coloring in sublogarithmic rounds," in *Proceedings 48th ACM Symposium on Theory of Computing (STOC)*, 2016, pp. 465–478.

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

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

[20] 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 26th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2015, pp. 355–370.

[21] M. Ghaffari and H.-H. Su, "Distributed degree splitting, edge coloring, and orientations," *CoRR*, vol. abs/1608.03220, 2016.

[22] K. Kothapalli and S. V. Pemmaraju, "Super-fast 3-ruling sets," in *Proceedings IARCS Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, ser. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012, vol. 18, pp. 136–147.

[23] S. Pettie and H.-H. Su, "Distributed algorithms for coloring triangle-free graphs," *Information and Computation*, vol. 243, pp. 263–280, 2015.

[24] B. Bollobás, "Chromatic number, girth and maximal degree," *Discrete Mathematics*, vol. 24, no. 3, pp. 311–314, 1978.

[25] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Local computation: Lower and upper bounds," *J. ACM*, vol. 63, no. 2, pp. 17:1–17:44, 2016.

[26] 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," in *Proceedings 35th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 2016, pp. 3–8.

[27] L. Barenboim and M. Elkin, "Sublogarithmic distributed MIS algorithm for sparse graphs using Nash-Williams decomposition," *Distributed Computing*, vol. 22, no. 5-6, pp. 363–379, 2010.

[28] J. Schneider and R. Wattenhofer, "An optimal maximal independent set algorithm for bounded-independence graphs," *Distributed Computing*, vol. 22, no. 5-6, pp. 349–361, 2010.

[29] X. Dahan, "Regular graphs of large girth and arbitrary degree," *Combinatorica*, vol. 34, no. 4, pp. 407–426, 2014.

[30] B. Bollobás, *Extremal graph theory*, ser. London Mathematical Society Monographs. London: Academic Press Inc. [Harcourt Brace Jovanovich Publishers], 1978, vol. 11.

[31] Y.-J. Chang, T. Kopelowitz, and S. Pettie, "An exponential separation between randomized and deterministic complexity in the local model," *CoRR*, vol. abs/1602.08166, 2016.