The Locality of Distributed Symmetry Breaking

LEONID BARENBOIM and MICHAEL ELKIN, Ben-Gurion University
SETH PETTIE, University of Michigan
JOHANNES SCHNEIDER, ETH Zürich

Symmetry-breaking problems are among the most well studied in the field of distributed computing and yet the most fundamental questions about their complexity remain open. In this article we work in the LOCAL model (where the input graph and underlying distributed network are identical) and study the *randomized* complexity of four fundamental symmetry-breaking problems on graphs: computing MISs (maximal independent sets), maximal matchings, vertex colorings, and ruling sets. A small sample of our results includes the following:

—An MIS algorithm running in $O(\log^2 \Delta + 2^{O(\sqrt{\log\log n})})$ time, where Δ is the maximum degree. This is the first MIS algorithm to improve on the 1986 algorithms of Luby and Alon, Babai, and Itai, when $\log n \ll \Delta \ll 2^{\sqrt{\log n}}$, and comes close to the $\Omega(\frac{\log \Delta}{\log\log \Delta})$ lower bound of Kuhn, Moscibroda, and Wattenhofer.
—A maximal matching algorithm running in $O(\log \Delta + \log^4 \log n)$ time. This is the first significant improvement to the 1986 algorithm of Israeli and Itai. Moreover, its dependence on Δ is *nearly optimal*.
—A $(\Delta+1)$-coloring algorithm requiring $O(\log \Delta + 2^{O(\sqrt{\log\log n})})$ time, improving on an $O(\log \Delta + \sqrt{\log n})$-time algorithm of Schneider and Wattenhofer.
—A method for reducing symmetry-breaking problems in low arboricity/degeneracy graphs to low-degree graphs. (Roughly speaking, the arboricity or degeneracy of a graph bounds the density of any subgraph.) Corollaries of this reduction include an $O(\sqrt{\log n})$-time maximal matching algorithm for graphs with arboricity up to $2^{\sqrt{\log n}}$ and an $O(\log^{2/3} n)$-time MIS algorithm for graphs with arboricity up to $2^{(\log n)^{1/3}}$.

Each of our algorithms is based on a simple but powerful technique for reducing a *randomized* symmetry-breaking task to a corresponding *deterministic* one on a poly$(\log n)$-size graph.

Categories and Subject Descriptors: G.2.2 [**Graph Theory**]: Graph Algorithms; F.2.2 [**Nonnumerical Algorithms and Problems**]: Computations on Discrete Structures

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Distributed networks, matching, MIS, vertex coloring

This work was supported by NSF Grants No. CCF-0746673, No. CCF-1217338, No. CNS-1318294, and No. CCF-1514383; US-Israel Binational Science Foundation Grant No. 2008390; Israeli Academy of Science Grant No 593/11; and Israel Science Foundation Grant No. 724/15. This research was partly performed while S. Pettie was on sabbatical at the Center for Massive Data Algorithmics (MADALGO), Aarhus University, which is supported by Danish National Research Foundation Grant No. DNRF84.
A preliminary version of this article appeared in the Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science (FOCS), 2012.
Authors' addresses: L. Barenboim, Department of Mathematics and Computer Science, The Open University of Israel, 1 University Road, P.O.B. 808, Raanana 43107, Israel; email: leonidb@openu.ac.il; M. Elkin, Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel; email: elkinm@cs.bgu.ac.il; S. Pettie (corresponding author), Department of Electrical Engineering and Computer Science, University of Michigan, 2260 Hayward Street, Ann Arbor, MI 48103, USA; email: pettie@umich.edu; J. Schneider, ABB Corporate Research, Switzerland; email: johannes.schneider@alumni.ethz.ch.

DOI: http://dx.doi.org/10.1145/2903137

ACM Reference Format:
Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. 2016. The locality of distributed symmetry breaking. J. ACM 63, 3, Article 20 (June 2016), 45 pages.
DOI: http://dx.doi.org/10.1145/2903137

1. INTRODUCTION

Breaking symmetry is one of the central themes in the theory of distributed computing. At initialization the nodes of a distributed system are assumed to be in the same state, possibly with distinct node IDs, yet to perform any computation the nodes frequently must take different roles. That is, they must somehow break their initial symmetry. In this article we study several of the most fundamental symmetry-breaking tasks in the LOCAL model [Linial 1992]: computing *maximal independent sets* (MIS), *maximal matchings*, *ruling sets*, and *vertex colorings*. These problems are defined below. In the LOCAL model, each node of the input graph G hosts a processor, which is only aware of its neighbors and upper bounds on various graph parameters such as n and Δ, which are the number of nodes and maximum degree, respectively.[1] The computation proceeds in synchronized rounds in which each processor sends one unbounded message along each edge. *Time* is measured by the number of rounds; local computation is free. At the end of the computation, each node must report its portion of the output, that is, whether it is in the MIS or ruling set, which incident edge is part of the matching, or its assigned color. This model should be contrasted with CONGEST, which is identical to LOCAL except messages consist of $O(1)$ words, that is, $O(\log n)$ bits. Refer to Peleg [2000, Ch. 1-2] for a discussion of distributed models. None of our algorithms seriously abuse the power of the LOCAL model. Our message size and local computation are always $O(\text{poly}(\Delta)\log n)$, usually $O(\text{poly}(\log n))$, and in several cases $O(1)$.

Let us define the four problems formally.

- MAXIMAL INDEPENDENT SET. Given $G = (V, E)$, find any set $I \subseteq V$ such that no two nodes in I are adjacent and I is maximal with respect to inclusion. (That is, every $v \notin I$ is adjacent to some member of I.)
- (α, β)-RULING SET. Given $G(V, E)$, find any $R \subset V$ such that for every $u \in V$, $\text{dist}(u, R) \leq \beta$ and for every $u \in R$, $\text{dist}(u, R \backslash \{u\}) \geq \alpha$. Note that $(2, 1)$-ruling sets are maximal independent sets. (Here $\text{dist}(u, X)$ is the length of a shortest path from u to any member of X.)
- MAXIMAL MATCHING. Given $G = (V, E)$, find any matching $M \subseteq E$ (consisting of node-disjoint edges) that is maximal with respect to inclusion.
- K-COLORING. Given $G = (V, E)$, find a proper coloring $\text{Color} : V \to \{1, \ldots, K\}$, that is, one for which $(u, v) \in E$ implies $\text{Color}(u) \neq \text{Color}(v)$. We are mainly interested in $(\Delta + 1)$-colorings, whose existence is trivially guaranteed.

We study the complexities of these problems on general graphs, as well as graphs with a specified *arboricity* λ. By definition $\lambda(G)$ is the minimum number of edge-disjoint forests that cover E, which is roughly the maximum density of any subgraph. We believe arboricity is an important graph parameter as it robustly captures the notion of *sparsity* without imposing any strict structural constraints, such as planarity or the like. We always have $\lambda \leq \Delta$, but, in general, λ could be significantly smaller than Δ. Most sparse graph classes, for example, have $\lambda = O(1)$ though their maximum degree is unbounded. These include planar graphs ($\lambda = 3$), graphs avoiding a fixed minor,

[1]This assumption can sometimes be removed. Korman, Sereni, and Viennot [2013] presented a method to convert non-uniform distributed algorithms (which know n, Δ, and possibly other parameters) into uniform distributed algorithms. The problems susceptible to this method must satisfy a couple properties, the most important of which is that any partial solution may be extended to a complete solution.

bounded genus graphs, and graphs of bounded treewidth or pathwidth. However, none of our algorithms actually depend on having $\lambda = O(1)$.

1.1. The State of the Art in Distributed Symmetry Breaking

The reader will soon notice two striking features of prior research on distributed symmetry breaking: the wide gulf between the efficiency of deterministic and randomized algorithms and the paltry number of algorithms that are *provably* optimal. It is typical to see randomized algorithms that are *exponentially* faster (in terms of n or Δ) than their deterministic counterparts, and they are usually simpler to analyze and simpler to implement. Very few problems can be solved in $O(1)$ time, independent of Δ and n. The $\omega(1)$ lower bounds of Linial [1992] and Kuhn, Moscibroda, and Wattenhofer [2004] are known to be tight in only a few cases, typically on very special classes of graphs.

We survey lower bounds and algorithms for each of the symmetry-breaking problems below. Tables I–IV provide an at-a-glance history of the problems. In the tables, deterministic algorithms are indicated by DET. All other algorithms are randomized, which return a correct answer with high probability.[2]

Lower Bounds. Linial [1992] proved that $\log^{(k)} n$-coloring the n-cycle takes $\Omega(k)$ time and, therefore, that $O(1)$-coloring the n-cycle takes $\Omega(\log^* n)$ time. On the n-cycle, MIS, maximal matching, and ruling sets are equivalent to $O(1)$-coloring, so Linial's lower bound applies to these problems as well. Kuhn, Moscibroda, and Wattenhofer [2004] (henceforth, KMW) proved that $O(1)$-approximate minimum vertex cover (MVC) takes $\Omega(\min\{\sqrt{\frac{\log n}{\log\log n}}, \frac{\log\Delta}{\log\log\Delta}\})$ time.[3] Since 2-approximate MVC is reducible to maximal matching and maximal matching is reducible to MIS (on the line graph of the original graph), the KMW lower bound implies $\Omega(\min\{\sqrt{\frac{\log n}{\log\log n}}, \frac{\log\Delta}{\log\log\Delta}\})$ lower bounds on these problems as well. It does *not* apply to coloring problems, nor the (α, β)-ruling set problem except when $(\alpha, \beta) = (2, 1)$.

Deterministic MIS. The fastest deterministic MIS algorithms for general graphs run in $2^{O(\sqrt{\log n})}$ time [Panconesi and Srinivasan 1996] and $O(\Delta + \log^* n)$ time [Barenboim et al. 2014]. The Panconesi-Srinivasan [1996] result is actually a *network decomposition* algorithm, which can be used to solve many symmetry-breaking problems in $2^{O(\sqrt{\log n})}$ time. It improved on an earlier algorithm of Awerbuch et al. [1989] running in $2^{O(\sqrt{\log n \log\log n})}$ time. Recent work on deterministic MIS algorithms has focused on restricted graph classes. Schneider and Wattenhofer [2010b] gave an optimal $O(\log^* n)$-time MIS algorithm for growth-bounded graphs.[4] Barenboim and Elkin [2010, 2013] gave an $O(\lambda\sqrt{\log n} + \log n)$-time MIS algorithm, and another that runs in $O(\frac{\log n}{\delta \log\log n})$ when the arboricity is $\lambda = (\log n)^{1/2-\delta}$. The subsequent vertex coloring algorithms of Barenboim and Elkin [2011] give, as corollaries, MIS algorithms running in $O(\lambda + \min\{\lambda^\epsilon \log n, \log^{1+\epsilon} n\})$ time and $O(\lambda^{1+\epsilon} + \log\lambda \log n)$ time, where $\epsilon > 0$ influences the leading constants.

[2]An event occurs *with high probability* if its probability is at least $1 - n^{-c}$ for an arbitrarily large c, where c may influence other constants, for example, those hidden in asymptotic running times.

[3]The same authors later claimed a stronger lower bound of $\Omega(\min\{\sqrt{\log n}, \log\Delta\})$ [Kuhn et al. 2010]. Very recently Bar-Yehuda, Censor-Hillel, and Schwartzman [2016] pointed out an error in their proof.

[4]A graph class has bounded growth if for each $v \in V$ and radius r, the maximum size of an independent set in v's r-neighborhood is a constant depending on r. For example, unit-disc graphs have bounded growth.

Table I. Maximal Independent Set

Citation	Running Time		Graphs
Linial [1992]	$\Omega(\log^* n)$		n-cycle
Kuhn et al. [2004]	$\Omega\left(\min\left\{\sqrt{\frac{\log n}{\log\log n}}, \frac{\log\Delta}{\log\log\Delta}\right\}\right)$		General
Luby [1986] and Alon et al. [1986]	$\log n$		General
Panconesi and Srinivasan [1996]	$2^{O(\sqrt{\log n})}$	Det.	General
Barenboim et al. [2014]	$\Delta + \log^* n$	Det.	General
Barenboim and Elkin [2010, 2011]	$\frac{\log n}{\delta \log\log n}$	Det.	$\lambda = \log^{1/2-\delta} n$
	$\lambda\sqrt{\log n} + \log n$	Det.	All λ, Fixed $\epsilon > 0$
	$\lambda + \min\{\lambda^\epsilon \log n, \log^{1+\epsilon} n\}$	Det.	
	$\lambda^{1+\epsilon} + \log\lambda \log n$	Det.	
Schneider and Wattenhofer [2010b]	$\log^* n$	Det.	Bounded growth
Lenzen and Wattenhofer [2011]	$\sqrt{\log n}\log\log n$		Trees ($\lambda = 1$)
This article	$\log^2\Delta + 2^{O(\sqrt{\log\log n})}$		General
	$\log^2\Delta + \frac{\log\log n}{\delta\log\log\log n}$		$\lambda = \log^{1/2-\delta}\log n$
	$\log^2\lambda + \log^{2/3} n$		All λ
	$\log^2\Delta + \lambda^{1+\epsilon} + \log\lambda\log\log n$		All λ, Fixed $\epsilon > 0$
	$\log^2\Delta + \lambda + \lambda^\epsilon \log\log n$		
	$\log^2\Delta + \lambda + (\log\log n)^{1+\epsilon}$		
	$\sqrt{\log n\log\log n}$		Trees ($\lambda = 1$)
	$\log\Delta\log\log\Delta + \frac{\log\log n}{\log\log\log n}$		
	$\log\Delta\log\log n + 2^{O(\sqrt{\log\log n})}$		Girth > 6

Randomized MIS. Nearly 30 years ago Luby [1986] and Alon, Babai, and Itai [1986] presented very simple randomized MIS algorithms running in $O(\log n)$ time. These algorithms are faster than the best deterministic algorithms when $\Delta = \omega(\log n)$ and remain the fastest MIS algorithms for general graphs when running time is expressed solely as a function of n. Lenzen and Wattenhofer [2011] showed that in the special case of trees ($\lambda = 1$), an MIS can be computed in $O(\sqrt{\log n}\log\log n)$ time with high probability.[5]

[5]See footnote 9.

Table II. Maximal Matching

Citation	Running Time		Graphs
Linial [1992]	$\Omega(\log^* n)$		n-cycle
Kuhn et al. [2004]	$\Omega\left(\min\left\{\sqrt{\frac{\log n}{\log\log n}}, \frac{\log\Delta}{\log\log\Delta}\right\}\right)$		General
Israeli and Itai [1986]	$\log n$		General
Hanckowiak et al. [2001]	$\log^4 n$	Det.	General
	$\log^3 n$	Det.	Bipartite
Panconesi and Rizzi [2001]	$\Delta + \log^* n$	Det.	General
Barenboim and Elkin [2010]	$\frac{\log n}{\delta \log\log n}$	Det.	$\lambda = \log^{1-\delta} n$
	$\lambda + \log n$	Det.	All λ
This article	$\log\Delta + \log^4 \log n$		General
	$\log\Delta + \log^3 \log n$		Bipartite
	$\log\Delta + \frac{\log\log n}{\delta \log\log\log n}$		$\lambda = \log^{1-\delta} \log n$
	$\log\lambda + \sqrt{\log n}$		All λ
	$\log\Delta + \lambda + \log\log n$		

Deterministic Maximal Matching. Panconesi and Srinivasan's [1996] network decomposition algorithm implies a deterministic $2^{O(\sqrt{\log n})}$-time maximal matching algorithm. This bound was dramatically improved by Hańćkowiak et al. [2001] to $O(\log^4 n)$. When $\Delta = o(\log^4 n)$, maximal matchings can be computed faster, in $O(\Delta + \log^* n)$ time, using the algorithm of Panconesi and Rizzi [2001]. Barenboim and Elkin [2010, 2013] gave improved algorithms for low arboricity graphs. Their algorithms run in $O(\lambda + \log n)$ time, for any λ, and in $O(\frac{\log n}{\delta \log\log n})$ time when $\lambda = \log^{1-\delta} n$.

Randomized Maximal Matching. Since a maximal matching in G is simply an MIS in the line graph of G, the randomized MIS algorithms of Luby [1986] and Alon et al. [1986] can be used to solve maximal matching in $O(\log n)$ time as well.[6] Israeli and Itai [1986] presented a direct randomized algorithm for computing maximal matchings in $O(\log n)$ time. This algorithm is faster than the deterministic algorithms when $\Delta = \omega(\log n)$ and remains the fastest maximal matching algorithm whose running time is expressed solely as a function of n.

Deterministic Vertex Coloring. The vertex coloring problem allows for a tradeoff between the palette size (number of colors) and running time. Linial [1992] proved that $O(\Delta^2)$-coloring can be computed deterministically in $O(\log^* n)$ time, independent of Δ. Szegedy and Vishwanathan [1993] later improved the running time of this algorithm to $\frac{1}{2}\log^* n + O(1)$. The best deterministic $(\Delta + 1)$-coloring algorithms run in $2^{O(\sqrt{\log n})}$

[6]These simulations increase the local computation at each node.

Table III. Vertex Coloring

Citation	Colors	Running Time	
Linial [1992]	3 (on the n-cycle)	$\Omega(\log^* n)$	
Cole and Vishkin [1986]		$\log^* n + O(1)$	Det.
Luby [1986]	$\Delta+1$	$\log n$	
Johansson [1999]			
Panconesi and Srinivasan [1996]		$2^{O(\sqrt{\log n})}$	Det.
Barenboim et al. [2014]		$\Delta + \log^* n$	Det.
Schneider and Watten [2010a]		$\log\Delta + \sqrt{\log n}$	
This article		$\log\Delta + 2^{O(\sqrt{\log\log n})}$	
		$\log\Delta + \lambda^{1+\epsilon} + \log\lambda\log\log n$	
		$\log\Delta + \lambda + \lambda^{\epsilon}\log\log n$	
		$\log\Delta + \lambda + (\log\log n)^{1+\epsilon}$	
	$\Delta + O(\lambda)$	$\log\Delta + \lambda^{\epsilon}\log\log n$	
		$\log\Delta + \lambda^{\epsilon} + (\log\log n)^{1+\epsilon}$	
	$\Delta + \lambda^{1+\epsilon}$	$\log\Delta + \log\lambda\log\log n$	
	$O(\Delta)$	$2^{O(\sqrt{\log\log n})}$	
Kothapalli et al. [2006]		$\sqrt{\log n}$	
Barenboim and Elkin [2011]		$\min\{\Delta^{\epsilon}\log n, \Delta^{\epsilon} + \log^{1+\epsilon} n\}$	Det.
	$O(\lambda)$	$\min\{\lambda^{\epsilon}\log n, \lambda^{\epsilon} + \log^{1+\epsilon} n\}$	Det.
	$\Delta^{1+\epsilon}$	$\log\Delta\log n$	Det.
	$\lambda^{1+\epsilon}$	$\log\lambda\log n$	Det.
Schneider and Wattenhofer [2010a]	$O(\Delta + \log n)$	$\log\log n$	
	$\Delta\log^{(k)} n + \log^{1+1/k} n$	k (for $k < \log^* n$)	
Kuhn and Wattenhofer [2006]	$\Delta\log n\log^{(k)} n$	k (for $k < \log^* n$)	
Linial [1992]	$O(\Delta^2)$	$\log^* n + O(1)$	Det.
Szegedy and Vishwanath [1993]		$\frac{1}{2}\log^* n + O(1)$	Det.
Barenboim and Elkin [2010]	$\lambda \cdot n^{1/k}$	$\Omega(k)$	
Kothapalli and Pemmaraju [2011]		k (for $\log\log n < k < \sqrt{\log n}$)	

Table IV. Ruling Sets

Citation	(α, β)	Running Time
trivial	$(2, 1)$	MIS time
	$(\alpha, (\alpha-1)\beta)$	$\alpha \cdot (2, \beta)$-ruling set time (See text, Section 1.1)
Awerbuch et al. [1989]	$(2, \log n)$	$\log n$ Det.
Gfeller and Vicari [2007]	$(1, O(\log\log\Delta))$	$\log\log\Delta$ (See text, Section 1.1)
Schneider and Wattenhofer [2010a]	$(2, \beta)$	$2^{\beta/2}\log^{\frac{2}{\beta-1}} n$
Kothapalli and Pemmaraju [2012]	$(2, 2)$	$(\log^{1/2}\Delta)(\log^{1/4} n)$
	$(2, 3)$ $(\lambda = 1)$	$(\log\log n)^2 \log\log\log n$
	$(2, 3)$ $(\lambda = O(1))$	$(\log\log n)^3$
Schneider et al. [2013]	$(2, \beta\Delta^{1/\beta})$	$\beta + \log^* n$ Det.
	$(2, \beta)$	$\beta\Delta^{2/\beta} + \log^* n$ Det.
Schneider et al. [2013] + Gfeller and Vicari [2007]	$(2, O(\log\log n))$	$\log\log n$
Barenboim and Elkin [2010] + Awerbuch et al. [1989]	$(2, \log\lambda + \sqrt{\log n})$	$\log\lambda + \sqrt{\log n}$ Det.
Bisht et al. [2014]	$(2, \beta)$	$\beta\log^{\frac{1}{\beta-1}}\Delta + 2^{O(\sqrt{\log\log n})}$
This article	$(2, \beta)$	$\beta\log^{\frac{1}{\beta-1/2}}\Delta + 2^{O(\sqrt{\log\log n})}$

time [Panconesi and Srinivasan 1996] or $O(\Delta + \log^* n)$ time [Barenboim et al. 2014]. Even if the palette size is enlarged to $O(\Delta)$, the Panconesi-Srinivasan [1996] algorithm remains the fastest, when time is expressed as a function of n. However, Barenboim and Elkin [2011] gave an $O(\min\{\lambda^\epsilon \log n, \lambda^\epsilon + \log^{1+\epsilon} n\})$-time algorithm for $O(\lambda)$-coloring and an $O(\log\lambda\log n)$-time algorithm for $\lambda^{1+\epsilon}$-coloring. (The hidden constants are exponential in $1/\epsilon$.) Since the arboricity λ is at most Δ, one can substitute Δ for λ in the bounds cited above.

Randomized Vertex Coloring. As usual, significantly faster coloring algorithms can be obtained using randomization. Luby [1986] gave a reduction from $(\Delta+1)$-coloring to MIS, which implies an $O(\log n)$ time randomized algorithm. A direct $O(\log n)$-time $(\Delta+1)$-coloring algorithm was analyzed by Johansson [1999]. By enlarging the palette, vertex coloring can be solved dramatically faster. Kothapalli et al. [2006] showed that $O(\sqrt{\log n})$ time suffices for computing an $O(\Delta)$-coloring for any Δ. Schneider and Wattenhofer [2010a] gave an $O(\log\Delta + \sqrt{\log n})$-time $(\Delta+1)$-coloring algorithm, for any Δ, and several faster $O(\Delta)$-coloring algorithms when Δ is sufficiently large. For example, when $\Delta = \Omega(\log n)$, $O(\Delta)$-coloring can be computed in $O(\log\log n)$ time and, when $\Delta = \Omega(\log^{1+1/\log^* n} n)$, $O(\Delta)$-coloring can be computed in $O(\log^* n)$ time. Kuhn and Wattenhofer [2006] showed that $O(\Delta\log n\log^{(k)} n)$-coloring is computable in $O(k)$ time and, in particular, an $O(\Delta\log^2 n)$-coloring could be computed in a single round.

Ruling Sets. As noted earlier, an MIS is a $(2, 1)$-ruling set. More generally, an $(\alpha, (\alpha - 1)\beta)$-ruling set can be found by computing a $(2, \beta)$-ruling set in the graph $G^{[1,\alpha-1]}$, whose edge set consists of pairs (u, v) for which $\text{dist}_G(u, v) \in [1, \alpha - 1]$. (See Section 2 for details of graph notation.) A distributed algorithm in $G^{[1,\alpha-1]}$ can be simulated in G with an $(\alpha - 1)$-factor slowdown. This reduction changes various graph parameters so it is not always applicable. For example, $\Delta(G^{[1,\alpha-1]})$ is roughly $(\Delta(G))^{\alpha-1}$ and $\lambda(G^{[1,\alpha-1]})$ cannot be bounded as a function of $\lambda(G)$.

Awerbuch et al. [1989] gave a deterministic $(2, \log n)$-ruling set algorithm running in $O(\log n)$ time. Schneider, Elkin, and Wattenhofer [2013] recently discovered a $(2, \beta)$-ruling set algorithm running in $O(\beta\Delta^{2/\beta} + \log^* n)$ time, for any integer parameter β, and another $(2, \beta\Delta^{1/\beta})$ ruling set algorithm running in $O(\beta + \log^* n)$ time.

These are the only deterministic ruling set algorithms. Using randomization, Gfeller and Vicari [2007] showed that a $(1, O(\log\log\Delta))$-ruling set could be computed such that the maximum degree in the graph induced by the ruling set is $O(\log^5 n)$. Schneider and Wattenhofer [2010a] gave a randomized algorithm for computing a $(2, \beta)$-ruling set in $O(2^{\beta/2}\log^{2/(\beta-1)} n)$ time. This bound was improved by Bisht et al. [2014] to $O(\beta\log^{1/(\beta-1)}\Delta + 2^{O(\sqrt{\log\log n})})$ time. In earlier work, Kothapalli and Pemmaraju [2012] gave a randomized $(2, 2)$-ruling set algorithm running in $O(\log^{1/2}\Delta \cdot \log^{1/4} n)$ time and a randomized $(2, 3)$-ruling set algorithm running in poly$(\log\log n)$ time for graphs with arboricity $\lambda = O(1)$.

1.2. The Union Bound Barrier

Our algorithms confront a fundamental barrier in randomized distributed algorithms we call the *union bound barrier*, which, to our knowledge, has never been explicitly discussed.

Consider a generic symmetry-breaking algorithm that works as follows. The nodes execute some number of iterations of an $O(1)$-time randomized experiment, the purpose of which is to commit to some fragment of the output. That is, some nodes are committed to the MIS or ruling set, some edges are committed to the matching, some nodes commit to a color, and so on.

The experiment *fails* at each node v with probability $1 - \Omega(1)$. For example, failure may be defined as the event that no edge incident to v joins the matching. The failure events are not independent in general but are independent for sufficiently distant nodes. If the random experiment takes t time steps, then nodes at distance at least $2t + 1$ are influenced by disjoint sets of nodes. Although each node succeeds after $\Theta(1)$ time in expectation, the union bound only lets us claim that a *global* solution is reached with probability $1 - n^{-\Omega(1)}$ if the failure probability *at each node* is $n^{-\Omega(1)}$, necessitating $\Theta(\log n)$ time. Symmetry-breaking algorithms based on a random experiment with failure probability p seem intrinsically incapable of running in $o(\log_{1/p} n)$ time.[7] However, there are several conceivable strategies one could use to escape this conclusion. Among them,

Use no randomness. Deterministic algorithms have no probability of failure.

Redefine failure. If the experiment is kept the same but the notion of failure is relaxed such that it only occurs with probability $n^{-\Omega(1)}$, then the union bound *can* be applied.

[7]Moreover, existing randomized algorithms [Luby 1986; Alon et al. 1986; Israeli and Itai 1986] do not even fit in this framework. They do *not* guarantee each node succeeds with probability $\Omega(1)$, only that an $\Omega(1)$-fraction of the edges are incident to nodes that succeed with probability $\Omega(1)$.

We borrow an idea used in early constructive algorithms for the Lovász Local Lemma [Beck 1991; Alon 1991] and more recently by Rubinfeld et al. [2011], which combines elements from both of the strategies above.

All of our algorithms consists of two discrete phases. In Phase I we execute $O(\log \Delta)$ or $\text{poly}(\log \Delta)$ iterations (rather than $\Theta(\log n)$) of an experiment whose *local* probability of failure is $1 - \Omega(1)$. Using the fact that failure events are independent for sufficiently distant nodes, we show that *every* connected component in the remaining graph[8] has size $s = \text{poly}(\log n)$ or, in one case, $s = \text{poly}(\Delta) \log n$, with probability $1 - n^{-\Omega(1)}$. In other words, rather than apply the union bound to n events (that each individual node survives), we apply the union bound to a much larger set of events corresponding to the survival of components with more than s nodes.

In Phase II we revert to the best available *deterministic* algorithm and apply it to each connected component, letting it run for time sufficient to solve any instance on s nodes. (If there is a component with more than s nodes, then this is a *global* failure, which occurs with probability $n^{-\Omega(1)}$.) This two-phase structure explains some conspicuous features of our results listed in Tables I–IV. The running times are always expressed as two (or more) terms, one that usually depends on $\log \Delta$ and another that exactly matches the time bound of one of the deterministic algorithms, except that it is *scaled down exponentially*. In other words, $2^{\sqrt{\log n}}$ becomes $2^{\sqrt{\log \log n}}$, $\frac{\log n}{\log \log n}$ becomes $\frac{\log \log n}{\log \log \log n}$, and so on.

The *union bound barrier* refers to the limitations attendant to any analysis that employs the union bound to upper bound the global probability of failure. A natural question is whether the union bound barrier is truly a barrier for distributed symmetry-breaking algorithms in the LOCAL model. To be very specific: Is it true that any optimal randomized symmetry-breaking algorithm must take something like our two-phase approach? Must every optimal randomized algorithm for n-node graphs contain within it an optimal deterministic algorithm for $\text{poly}(\log n)$-size inputs?

1.3. New Results

We introduce numerous symmetry-breaking algorithms using the two-phase strategy outlined in Section 1.2. For Phase I we design new iterated randomized experiments and analyze their local probability of failure. After Phase I the connected components in the surviving subgraph have size $\text{poly}(\log n)$ or $\text{poly}(\Delta) \log n$ with high probability. For Phase II we invoke the best available deterministic algorithm, usually applied in a black-box fashion. For general graphs there always happens to be one best deterministic algorithm. However, for low arboricity graphs, we have access to several algorithms, each of which is asymptotically superior for different values of λ, Δ, and n.

For graphs with a large disparity between λ and Δ, the method described above does not get optimal results. We give a general randomized reduction showing that MIS and maximal matching are reducible in $O(\log^{1-\gamma} n)$ time to instances with maximum degree $\lambda \cdot 2^{\log^{\gamma} n}$ for any $\gamma \in (0, 1)$. This reduction allows us to obtain algorithms whose running time is *sub*logarithmic in n, given algorithms that run in time polylogarithmic in Δ.

We shall now discuss the results claimed in Tables I–IV.

MIS and Ruling Sets. Our primary result is a new MIS algorithm running in $O(\log^2 \Delta + 2^{O(\sqrt{\log \log n})})$ time, which is within a $\log \Delta$ factor of the KMW lower bound. Moreover, this is the *first* improvement to the 1986 algorithms of Luby [1986] and Alon

[8]That is, the portion not dominated by the independent set (in the case of MIS), not adjacent to a matched edge (in the case of maximal matching), and so on.

et al. [1986] for such a broad range of degrees: from $\Delta = \Omega(\log n)$ to $2^{O(\sqrt{\log n})}$. The Phase II portion of this algorithm is rather complicated since we cannot afford to apply an existing MIS algorithm in a black box fashion. After Phase I the surviving components are shown to have size $\text{poly}(\Delta)\log n$. By invoking the Panconesi-Srinivasan [1996] algorithm on each component, Phase II would run in $2^{O(\sqrt{\log(\text{poly}(\Delta)\log n)})}$ time, which is fine if $\Delta = \text{poly}(\log n)$ but not if Δ is just slightly super-logarithmic. We prove that by a certain deterministic clustering procedure, each component can be decomposed into $\log n$ clusters with diameter $O(\log \Delta)$. A version of the Panconesi-Srinivasan [1996] algorithm can then be simulated on the cluster graph formed by virtually contracting each cluster to a single node.

Using our degree-reduction routine, we can solve MIS on graphs with arboricity λ in $O(\log^{1-\gamma} n + \log^2(\lambda \cdot 2^{\log^\gamma n}) + 2^{O(\sqrt{\log\log n})})$ time, which simplifies to $O(\log^2 \lambda + \log^{2/3} n)$ when $\gamma = 1/3$. Other MIS algorithms that depend at least linearly on λ can be generated by invoking one of the MIS algorithms of Barenboim and Elkin [2011].

Finally, we give an $O(\log \Delta \log\log \Delta + \frac{\log\log n}{\log\log\log n})$-time algorithm for MIS on trees ($\lambda = 1$), which, using the degree-reduction routine with $\gamma = 1/2 - o(1)$, implies a time bound of $O(\sqrt{\log n \log\log n})$, independent of Δ.[9] With minor modifications, this algorithm can be made to work on general graphs with girth greater than 6, not just trees. The *girth* of a graph is the length of its shortest cycle.

Bisht et al. [2014] showed how to reduce the $(2, \beta)$-ruling set problem on degree-Δ graphs to an MIS problem on graphs with degree much smaller than Δ. Using their reduction and our new MIS algorithm, we get a $(2, \beta)$-ruling set algorithm running in $O(\beta \log^{\frac{1}{\beta-1/2}} \Delta + 2^{O(\sqrt{\log\log n})})$ time. This result is notable because it establishes a provable gap between the complexity of computing an MIS (a $(2, 1)$-ruling set) and a $(2, 2)$-ruling set. By the KMW bound, an MIS cannot be computed in $o(\log \Delta)$ time, whereas $(2, 2)$-ruling sets can be computed in $O(\log^{2/3} \Delta + 2^{O(\sqrt{\log\log n})})$ time.[10]

Maximal Matching. We give a new maximal matching algorithm running in $O(\log \Delta + \log^4 \log n)$ time using $O(1)$-size messages, that is, it works in the CONGEST model. In some ways this is our strongest result. Its dependence on Δ nearly matches the $\Omega(\frac{\log \Delta}{\log\log \Delta})$ KMW [2004] lower bound. Using the degree-reduction routine with $\gamma = 1/2$, we obtain a maximal matching algorithm running in $O(\log \lambda + \sqrt{\log n})$ time. Since the KMW graphs have arboricity $\lambda = 2^{\Theta(\sqrt{\log n \log\log n})}$, this algorithm is nearly optimal for that particular arboricity. Generalizing the KMW lower bound, we prove that *even on trees*, maximal matching requires $\Omega(\sqrt{\frac{\log n}{\log\log n}})$ time. Thus, our algorithm is nearly optimal for all λ from 1 to $2^{O(\sqrt{\log n})}$. Using the Barenboim-Elkin [2010, 2013] maximal matching algorithm, we obtain more results that are superior when λ is small and $\log \Delta = o(\sqrt{\log n})$. For example, when $\lambda = O(1)$, a maximal matching can also be computed in $O(\log \Delta + \frac{\log\log n}{\log\log\log n})$ time.

[9] Lenzen and Wattenhofer [2011] claimed an MIS algorithm running in $O(\sqrt{\log n \log\log n})$ time on trees, but there is a flaw in their analysis. We repair this flaw in Section 8. By incorporating Lemma 8.3 into the proof of Lenzen and Wattenhofer [2011, Lemma 4.8], the resulting algorithm would only run in $O(\sqrt{\log n}\log\log n)$ time.

[10] When time bounds are expressed in terms of n (rather than Δ), our result only demonstrates that $(2, 3)$-ruling sets are easier to compute than MISs. They can be computed in $O(\log^{2/5} \Delta + 2^{O(\sqrt{\log\log n})}) = O(\log^{2/5} n)$ time, whereas MISs need $\Omega(\sqrt{\frac{\log n}{\log\log n}})$ time [Kuhn et al. 2004].

Vertex Coloring. The vertex coloring problem, in one respect, qualitatively differs from maximal matching and MIS. In Phase II of the MIS and matching algorithms, each connected component forms a (small) instance of MIS or maximal matching. However, in our vertex coloring algorithms, at the beginning of Phase II some nodes have been permanently colored, which affects the palettes of their as-yet uncolored neighbors. Thus, the connected components of uncolored nodes form instances of the *list-coloring* problem—each vertex may hold a palette of an arbitrary set of allowable colors. This distinction sometimes makes no difference.

Our main coloring result is a $(\Delta + 1)$-coloring algorithm running in $O(\log \Delta + 2^{O(\sqrt{\log \log n})})$ time,[11] which improves the $O(\log \Delta + \sqrt{\log n})$ bound of Schneider and Wattenhofer [2010a] and implies that $O(\Delta)$-coloring can be computed in $2^{O(\sqrt{\log \log n})}$ time, independent of Δ. The KMW lower bounds do not apply to vertex coloring. So long as the Panconesi-Srinivasan algorithm goes unimproved, it will be difficult or impossible to improve the dependence on n.

By invoking the Barenboim-Elkin [2010, 2011, 2013] coloring algorithms we obtain numerous results for graphs with small arboricity. Since the Barenboim-Elkin algorithms do *not* solve the general list-coloring problem, we have to start Phase II with a "fresh" palette of unused colors. This fact leads to $(\Delta + \Omega(\lambda))$-coloring algorithms whose running time is sublinear in λ and $(\Delta + 1)$-coloring algorithms whose running time is at least linear in λ.

1.4. Recent Developments

Our two-phase approach to randomized symmetry breaking has influenced a diverse set of recent results.[12] It has been applied to computing ruling sets [Bisht et al. 2014; Kothapalli and Pemmaraju 2012]; see Table IV. Chung et al. [2014] gave distributed algorithms for the constructive Lovász local lemma [Moser and Tardos 2010], which involves the computation of certain approximate MISs. Pettie and Su [2015] gave fast $O(\Delta / \ln \Delta)$-coloring algorithms for triangle-free graphs and other natural graph classes. Elkin et al. [2015] gave various edge-coloring algorithms for general graphs and vertex coloring algorithms for locally sparse graphs. One consequence of their results is that $(\Delta + 1)$-coloring can be computed in $O(\log \lambda) + 2^{O(\sqrt{\log \log n})}$ time for all λ, Δ, n, and in $O(\log^* n)$ time for certain ranges of the parameters. Very recently Ghaffari [2016] exhibited a new MIS algorithm running in $O(\log \Delta + 2^{O(\sqrt{\log \log n})})$ time, which has nearly optimal dependence on Δ. Harris et al. [2016] discovered a $(\Delta + 1)$-coloring algorithm running in $O(\sqrt{\log \Delta} + 2^{O(\sqrt{\log \log n})})$ time, demonstrating a separation between the complexity of $(\Delta + 1)$-coloring and MIS. In a recent breakthrough in *deterministic* complexity, Barenboim [2015] discovered a $(\Delta + 1)$-coloring algorithm running in $O(\Delta^{3/4} \log \Delta + \log^* n)$ time and a $(1 + o(1))\Delta$-coloring algorithm running in $\tilde{O}(\sqrt{\Delta} + \log^* n)$ time.

1.5. Organization

In Section 2 we review some notation for graphs and their parameters, as well as some useful symmetry-breaking primitives due to Awerbuch et al. [1989] and Panconesi and Srinivasan [1996]. Sections 3–6 are devoted to algorithms for the four symmetry-breaking problems on *general* graphs. In Section 7 we present a new degree-reduction method (parameterized by the arboricity) and derive numerous results for small

[11] The algorithm actually solves the list-coloring problem, where a vertex v's palette contains $\deg(v)+1$ colors.
[12] Somewhere along the way this technique has become known as *graph shattering* [Su 2015; Ghaffari 2016], referring to the pieces of the graph output by the Phase I part of the algorithm.

arboricity graphs. Section 8 presents a faster algorithm for MIS on trees and graphs of girth greater than 6. We conclude and discuss some open problems in Section 9.

In our analyses we use several standard concentration inequalities due to Chernoff, Janson, and Azuma-Hoeffding. The statements of these theorems can be found in Appendix A. Refer to Dubhashi and Panconesi [2009] for derivations of these and other concentration bounds.

2. PRELIMINARIES

2.1. Graph Notation

Let $G = (V, E)$ be the undirected input graph and underlying distributed network. Define $\Gamma_H(v)$, $\hat{\Gamma}_H(v)$, and $\deg_H(v)$ to be the neighborhood, inclusive neighborhood, and degree of v with respect to a graph H. Typically H is an induced subgraph of G. Formally,

$$\Gamma_H(v) \stackrel{\text{def}}{=} \{u \mid (v,u) \in E(H)\},$$
$$\hat{\Gamma}_H(v) \stackrel{\text{def}}{=} \{v\} \cup \Gamma_H(v),$$
$$\text{and} \quad \deg_H(v) \stackrel{\text{def}}{=} |\Gamma_H(v)|.$$

For succinctness we sometimes put $U \subseteq V(G)$ or $U \subseteq E(G)$ in the subscript to refer to the subgraph of G induced by U. The subscript may be omitted altogether if $H = G$.

We assume the nodes know global graph parameters[13] such as $n \stackrel{\text{def}}{=} |V(G)|$, $\Delta \stackrel{\text{def}}{=} \max_{v \in V} \deg_G(v)$, and, if applicable, the arboricity $\lambda(G)$. To simplify calculations we often assume n, Δ, and λ are at least some sufficiently large constant. The arboricity of a graph H is the minimum number of forests that cover $E(H)$. By the Nash-Williams [1964] theorem, $\lambda(H)$ can also be defined as

$$\lambda(H) \stackrel{\text{def}}{=} \max \left\{ \left\lceil \frac{\left|E(H) \cap \binom{U}{2}\right|}{|U|-1} \right\rceil \;\middle|\; U \subseteq V(H) \text{ and } |U| \geq 2 \right\},$$

that is, roughly the edge density of any subgraph of H with at least two nodes. Other measures of graph sparsity are, for our purposes, equivalent to λ. For example, the *degeneracy* of a graph H is defined to be

$$d(H) \stackrel{\text{def}}{=} \max_{U \subseteq V(H)} \min_{v \in U} \deg_U(v).$$

It is known that $\lambda(H) \leq d(H) \leq 2\lambda(H) - 1$.

Our matching algorithms internally generate *directed* graphs. In a directed graph H, the *indegree* and *outdegree* of v (written $\text{indeg}_H(v)$ and $\text{outdeg}_H(v)$) are the number of edges oriented towards v and away from v, respectively, and $\deg_H(v) \stackrel{\text{def}}{=} \text{indeg}_H(v) + \text{outdeg}_H(v)$. A *pseudoforest* is a directed graph in which all nodes have outdegree at most 1.

Let $\text{dist}_H(u, v)$ be the distance (length of the shortest path) between u and v in H. For any integers $1 \leq a \leq b$, define

$$H^{[a,b]} \stackrel{\text{def}}{=} \left(V(H), \{(u,v) \mid \text{dist}_H(u,v) \in [a,b]\}\right)$$
$$\text{and} \quad H^a \stackrel{\text{def}}{=} H^{[a,a]}.$$

In other words, we put edges between pairs whose distance is in the interval $[a, b]$.

[13]This assumption can be removed for many of our algorithms. See Korman et al. [2013].

2.2. Decompositions and Ruling Sets

A network decomposition is a powerful tool used in symmetry-breaking algorithms. The fastest known deterministic decomposition algorithm is due to Panconesi and Srinivasan [1996]. See Awerbuch et al. [1989] and Linial and Saks [1993] for earlier decomposition algorithms.

Definition 2.1 (*Network Decompositions*). Let H be an n-vertex graph. A $(d(n), c(n))$-network decomposition is a pair $(\mathscr{D}, \mathscr{C})$ such that $\mathscr{D}$ is a partition of $V(H)$ into clusters, each with diameter at most $d(n)$, and $\mathscr{C} : \mathscr{D} \to \{1, \ldots, c(n)\}$ is a proper $c(n)$-coloring of the graph derived by contracting the clusters. More formally, we have $\mathscr{D} = \{D_i\}$, where $\bigcup_i D_i = V(H)$, $D_i \cap D_{i'} = \emptyset$ for $i \neq i'$, and if $v, v' \in D_i$ then $\text{dist}_{D_i}(v, v') \leq d(n)$. If there exists $(v, v') \in E(H)$ with $v \in D_i$ and $v' \in D_{i'}$, then $\mathscr{C}(D_i) \neq \mathscr{C}(D_{i'})$.

Theorem 2.2 (Panconesi and Srinivasan [1996]). *A* $(2^{O(\sqrt{\log n})}, 2^{O(\sqrt{\log n})})$*-network decomposition can be computed deterministically in* $2^{O(\sqrt{\log n})}$ *time.*

Definition 2.3 and Theorem 2.4 generalize, slightly, Awerbuch et al.'s [1989] original definition of a ruling set.

Definition 2.3 (*Ruling Sets*). Let H be a graph and $U \subseteq V(H)$. An (α, β)-ruling set for U (w.r.t. H) is a node set $R \subseteq U$ such that for each $v \in U$, $\text{dist}_H(v, R) \leq \beta$ and, if $v \in R$, then $\text{dist}_H(v, R \backslash \{v\}) \geq \alpha$. For example, maximal independent sets are $(2, 1)$-ruling sets for $V(H)$ with respect to H.

Theorem 2.4 (Awerbuch et al. [1989]). *Let* H *be a graph and* $U \subseteq V(H)$. *Given a proper* K*-coloring of* $H^{[1,\alpha-1]}$, *an* $(\alpha, (\alpha-1)\lceil \log K \rceil)$*-ruling set* R *for* U *can be computed in* $(\alpha-1)\lceil \log K \rceil$ *time, together with a partition of* U *into a set* $\{\text{Cluster}(u) \subseteq U \mid u \in R\}$ *of disjoint radius-*$(\alpha-1)\lceil \log K \rceil$ *clusters, each of which is a connected set in* $H^{[1,\alpha-1]}$.

Proof. Let $\chi : V \to \{1, \ldots, K\}$ be the coloring. Initially set $\text{Cluster}(u) \leftarrow \{u\}$ for all $u \in U$. Recursively, and in parallel, compute two $(\alpha, (\alpha-1)(\lceil \log K \rceil - 1))$-ruling sets R_0 and R_1 for, respectively,

$$U_0 = \{v \in U \mid \chi(v) \in \{1, \ldots, \lfloor K/2 \rfloor\}\}$$
$$\text{and } U_1 = \{v \in U \mid \chi(v) \in \{\lfloor K/2 \rfloor + 1, \ldots, K\}\}.$$

After these recursive calls $\{\text{Cluster}(u) \mid u \in R_i\}$ is a radius-$(\alpha-1)(\lceil \log K \rceil - 1)$ clustering of U_i, for $i \in \{0, 1\}$. We calculate the final ruling set R as follows.

$$R \leftarrow R_0 \cup \{v \in R_1 \mid \text{dist}_H(v, R_0) \geq \alpha\}.$$

For each $u \in R$, let $L_u \subset R_1 \backslash R$ be the set of all v "knocked out" in this round for which $(\text{dist}_H(v, u), \text{ID}(u))$ was lexicographically minimum; that is, we assign v to the closest R-node, breaking ties arbitrarily. The cluster for each $u \in R$ is defined as follows:

$$\text{Cluster}(u) \leftarrow \text{Cluster}(u) \cup \bigcup_{v \in L_u} \text{Cluster}(v).$$

In other words, a $v \in R_1$ that is knocked out at this stage merges $\text{Cluster}(v)$ into $\text{Cluster}(u)$. Because $\text{dist}_H(v, u) \leq \alpha - 1$, it is guaranteed that $\text{Cluster}(u)$ is connected in $H^{[1,\alpha-1]}$. Once R_0 and R_1 are computed, in $(\alpha-1)(\lceil \log K \rceil - 1)$ time, R and the final clustering can be computed in $\alpha - 1$ additional time. □

If the nodes of H are endowed with distinct β-bit IDs, then we can use them as a proper 2^β-coloring and compute an $(\alpha, (\alpha-1)\beta)$-ruling set in $O((\alpha-1)\beta)$ time. (This was Awerbuch et al.'s [1989] original algorithm.) However, a better bound can be obtained by first computing a good coloring.

COROLLARY 2.5. *Let H be a graph with maximum degree Δ whose nodes are assigned distinct β-bit IDs. For any $\alpha \geq 2$ and $U \subseteq V(H)$, an $(\alpha, 2(\alpha-1)^2(\log\Delta + O(1)))$-ruling set for U with respect to H can be computed in $O(\alpha \log^* \beta + \alpha^2 \log\Delta)$ time, together with a radius-$2(\alpha-1)^2(\log\Delta + O(1))$-clustering of U, each cluster of which is connected in $H^{[1,\alpha-1]}$.*

PROOF. The graph $H^{[1,\alpha-1]}$ has maximum degree less than $\hat{\Delta} \stackrel{\text{def}}{=} \Delta^{\alpha-1}$. The first step is to $O(\hat{\Delta}^2)$-color $H^{[1,\alpha-1]}$ in $O(\alpha \log^* \beta)$ time. The coloring algorithms of Linial [1992] and Szegedy and Vishwanathan [1993] take $O(\log^* \beta)$ time steps in $H^{[1,\alpha-1]}$, each of which can be simulated with $\alpha - 1$ time steps in H. By Theorem 2.4, an $(\alpha, (\alpha-1)\log(O(\hat{\Delta}^2)))$-ruling set can be computed for U in $O(\alpha \log(\hat{\Delta}^2))$ time. Note that $(\alpha-1)\log(O(\hat{\Delta}^2)) = 2(\alpha-1)^2(\log\Delta + O(1))$. □

Remark 2.6. Clearly, any (α, β)-ruling set R gives a natural radius-β clustering of U. (Simply put each $u \in U$ into the cluster of its closest R-node with respect to dist_H, breaking ties arbitrarily.) However, this clustering will have the undesirable property that clusters may intersect multiple connected components of $H^{[1,\alpha-1]}$. The analysis of the MIS algorithm of Section 3 uses the property that the clusters of Corollary 2.5 are connected in $H^{[1,\alpha-1]}$.

2.3. Miscellany

In each of our algorithms there is some arbitrary (constant) parameter c that controls the failure probability, which is always of the form $n^{-\Omega(c)}$. All logarithms are base 2 unless specified otherwise. We make repeated use of the inequality $(1+x) \leq e^x$, which holds for all x.

3. A MAXIMAL INDEPENDENT SET ALGORITHM

In Section 3.1 we give an $O(\log^2\Delta)$-time randomized algorithm called IndependentSet that computes a large, but not necessary maximal, independent set. A new two-phase MIS algorithm is presented in Section 3.2. In Phase I it invokes IndependentSet to find a set I with two properties, (i) all surviving vertices in $V(G)\backslash\hat{\Gamma}(I)$ form components with size $\text{poly}(\Delta)\log n$[14] and (ii) all $(5, O(\log\Delta))$-ruling sets in each component have size less than $\log n$. As a consequence of property (i) we can bound the message size by $\text{poly}(\Delta)\log n$. (In the worst case, a message encodes the topology of the entire component.) Using property (ii) we can extend I to an MIS in $O(\log\Delta \cdot \exp(O(\sqrt{\log\log n})))$ time, deterministically. Phase I succeeds with probability $1 - n^{-\Omega(1)}$ and if it does succeed, Phase II succeeds with probability 1.

Refer to Figures 1 and 3 for the pseudocode of IndependentSet and MIS.

3.1. Computing an Almost Maximal Independent Set

The IndependentSet algorithm uses a generalization of Luby's [1986] randomized experiment. It consists of $\log\Delta$ *scales*, each composed of $O(\log\Delta)$ Luby steps. The purpose of the kth scale is to reduce the maximum degree in the surviving graph to $\Delta/2^k$. At some nodes this invariant will fail to hold with some non-negligible probability. We call such nodes *bad* and remove them from consideration. The components induced by bad nodes are reconsidered in Phase II of the MIS algorithm.

LEMMA 3.1. *Consider a single iteration of Step 2a (a "Luby step") in* IndependentSet. *If $v \in V_{IB}$ and $\deg_{IB}(v) > \Delta/2^k$ before the iteration, then the probability that $v \in \hat{\Gamma}(I)$ after the iteration is at least $(1 - e^{-1/2})e^{-1}$.*

[14]Recall that $\hat{\Gamma}(I) \stackrel{\text{def}}{=} I \cup \Gamma(I)$ contains all vertices in or adjacent to I.

IndependentSet(Graph G)

(1) Initialize sets $I, B \subset V(G)$:
  $I \leftarrow \emptyset$ {an independent set}
  $B \leftarrow \emptyset$ {a set of 'bad' nodes}
  Throughout, let $V_{IB} \stackrel{\text{def}}{=} V(G)\backslash(\hat{\Gamma}(I) \cup B)$ be the nodes still under consideration: those not marked bad and not in or adjacent to the independent set. Let G_{IB} be the graph induced by V_{IB} and let Γ_{IB} and $\deg_{IB}$ be the neighborhood and degree functions w.r.t. G_{IB}.
(2) For each *scale* k from 1 to $\log\Delta + 1$,
  (a) Execute $c\log\Delta$ iterations of steps i and ii.
    i. Each node $v \in V_{IB}$ chooses a random bit $b(v)$:

$$b(v) \leftarrow \begin{cases} 1 & \text{with probability } 1/(\deg_{IB}(v)+1) \\ 0 & \text{with probability } 1 - 1/(\deg_{IB}(v)+1) \end{cases}$$

    ii. $I \leftarrow I \cup \{v \in V_{IB} \mid b(v) = 1 \text{ and } b(u) = 0 \text{ for all } u \in \Gamma_{IB}(v)\}$.
    (Add nodes to the independent set.)
  (b) $B \leftarrow B \cup \{v \in V_{IB} \mid \deg_{IB}(v) > \Delta/2^k\}$.
    (Mark high-degree nodes as bad.)
(3) $B \leftarrow B\backslash\hat{\Gamma}(I)$.
  (Bad nodes adjacent to I no longer need to be considered bad.)
(4) Return (I, B).

Fig. 1. An algorithm for computing an almost maximal independent set.

PROOF. Let $\hat{\Gamma}_{IB}(v) = \{v = v_0, v_1, v_2, \ldots, v_{\deg_{IB}(v)}\}$ be the inclusive neighborhood of v. By assumption $\deg_{IB}(v) > \Delta/2^k$ and since $v_1, \ldots, v_{\deg_{IB}(v)}$ were not marked bad (placed in B) in the last execution of Step 2b, $\deg_{IB}(v_i) \le \Delta/2^{k-1}$ for each $i \le \deg_{IB}(v)$. Let $i^\star \in \{0, \ldots, \deg_{IB}(v)\}$ be the first index for which $b(v_{i^\star}) = 1$. The probability that $i^\star$ exists is

$$1 - \prod_{i=0}^{\deg_{IB}(v)} \left(1 - \frac{1}{\deg_{IB}(v_i)+1}\right) \ge 1 - \left(1 - \frac{1}{\Delta/2^{k-1}+1}\right)^{\Delta/2^k+1} > 1 - e^{-1/2}.$$

If $i^\star$ does exist, then $v_{i^\star}$ is included in the independent set I if all its neighbors set their b-values to zero. This occurs with probability

$$\prod_{u \in \Gamma_{IB}(v_{i^\star})\backslash\{v_0,\ldots,v_{i^\star-1}\}} \left(1 - \frac{1}{\deg(u)+1}\right) \ge \left(1 - \frac{1}{\Delta/2^{k-1}+1}\right)^{\Delta/2^{k-1}} > e^{-1}.$$

Nodes $v_0, \ldots, v_{i^\star-1}$ are excluded from consideration since, by definition of $i^\star$, they have already set their b-values to zero. Thus, after one iteration of Step 2a, v is in $\hat{\Gamma}(I)$ with probability $(1 - e^{-1/2})e^{-1} \approx 0.145$. See Figure 2 for an illustration. □

LEMMA 3.2. *Let $U \subset V(G)$ be a node set such that $\text{dist}_G(u, U\backslash\{u\}) \ge 5$ for each $u \in U$. The probability that $U \subseteq B$ after a call to* IndependentSet(G) *is less than $\Delta^{-c|U|/5}$.*

PROOF. The event that a node $v \in V_{IB}$ appears in $\hat{\Gamma}(I)$ after one iteration of Step 2a depends only on the random bits chosen by v's neighbors and neighbors' neighbors. Since all nodes in U are mutually at distance at least five, in each iteration the events that they appear in $\hat{\Gamma}(I)$ are independent. Call a node $v \in V_{IB}$ *vulnerable* in a particular

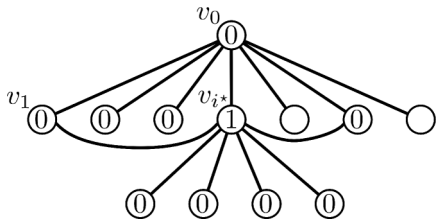

Fig. 2. The node v_0 is eliminated if some node in its inclusive neighborhood joins the independent set. This occurs if some $v_{i^\star}$ chooses $b(v_{i^\star}) = 1$ and $1 \notin b(\Gamma(v_{i^\star}))$.

iteration of Step 2a if $\deg_{IB}(v) > \Delta/2^k$. We cannot say for certain when a node will be vulnerable, but eventually each must, for some k, be vulnerable throughout scale k, until it appears in $\hat{\Gamma}(I)$ or is placed in B at the end of the scale. By Lemma 3.1 the probability that an individual node ends up in B is at most $p^{c \log \Delta}$, where $p = 1 - (1 - e^{-1/2})e^{-1} \approx 0.855$. Since $\log p < -0.22$, $p^{c \log \Delta} = \Delta^{c \log p} < \Delta^{-c/5}$. Since outcomes for U-nodes are independent in any iteration of Step 2a, the probability that all nodes in U end up in B is at most $\Delta^{-c|U|/5}$. □

LEMMA 3.3. *Let* (I, B) *be the pair returned by* IndependentSet(G). *For* $t = \log_\Delta n$, (I, B) *satisfies the following properties with probability* $1 - n^{-c/5+14}$.

(1) *There does not exist any* $U \subset V_{IB}$ *with* $|U| = t$ *such that for every* $U' \subset U$, $\text{dist}_G(U', U \backslash U') \in [5, 12]$. *(Here* $\text{dist}_G(A, B) \stackrel{\text{def}}{=} \min_{a \in A, b \in B} \text{dist}_G(a, b)$.*)*
(2) *All components in the graph induced by* V_{IB} *have fewer than* $t\Delta^4$ *nodes.*

PROOF. A set $U \subset V$ satisfying the criteria of Part (1) forms a t-node tree in the graph $G^{[5,12]}$. (This tree is not necessarily unique.) The number of rooted unlabeled t-node trees is less than 4^t since the Euler tour of such a tree can be encoded as a bit-vector with length $2t$. The number of ways to embed such a tree in $G^{[5,12]}$ is less than $n \cdot \Delta^{12(t-1)}$: There are n choices for the root and less than Δ^{12} choices for each subsequent node. By Lemma 3.2 the probability that $U \subseteq B$ is less than $\Delta^{-ct/5}$. By a union bound, the probability that any such U is contained in B is less than

$$4^t \cdot n \cdot \Delta^{12(t-1)} \cdot \Delta^{-ct/5} < n^{\log_\Delta 4 + 13 - c/5} < n^{-c/5+14}.$$

Turning to Part (2), suppose there is such a connected component C with $t\Delta^4$ nodes. We can find a subset U of the nodes satisfying the criteria of Part (1) by the following greedy procedure. Choose an arbitrary initial node $v_1 \in C$ and set $U \leftarrow \{v_1\}$. Iteratively select a $v_i \in C \backslash U$ for which $\text{dist}_G(v_i, U) = 5$, set $U \leftarrow U \cup \{v_i\}$, and then remove from consideration all nodes within distance 4 of v_i. The number removed is less than Δ^4, and hence U has size at least $(t\Delta^4)/\Delta^4 = t$. □

3.2. The MIS Algorithm

The pseudocode for MIS appears in Figure 3. We walk through each step of the algorithm below. Recall that IndependentSet(G) returns an independent set I and set of "bad" nodes B.

Step 1. After Step 1 we have an independent set I and a set of bad nodes $B = V_{IB} = V(G) \backslash \hat{\Gamma}(I)$. By Lemma 3.3(2), with high probability each connected component in G_{IB} has at most $t \cdot \Delta^4$ nodes and therefore at most $t \cdot \Delta^5/2$ edges, where $t = \log_\Delta n$. Step 1 (and Step 2) require only one-bit messages since each node only has to notify its neighbors about its status (whether in I or not, whether in V_{IB} or not) and the b-values it selects in each round. The purpose of Step 1 is *merely* to break G into components of

MIS(Graph G)

Phase I:

(1) $(I, B) \leftarrow \mathsf{IndependentSet}(G)$.

The following steps focus on a single connected component C in G_{IB}. They are executed in parallel for each such C.

(2) $(I_C, B_C) \leftarrow \mathsf{IndependentSet}(C)$.

Phase II:

(3) $R_C \leftarrow$ a $(5, 32\log\Delta + O(1))$-ruling set for $B_C = V(C)\backslash\hat{\Gamma}(I_C)$ w.r.t. C.

Let $\{\mathrm{Cluster}(x) \subseteq B_C \mid x \in R_C\}$ be the corresponding clustering of B_C.

(4) Form the cluster graph $C^\star$.

$$C^\star \leftarrow \left(R_C, \left\{(x, x') \;\middle|\; \begin{array}{l}\text{there exists } (v, v') \in E(C) \text{ such that}\\ v \in \mathrm{Cluster}(x) \text{ and } v' \in \mathrm{Cluster}(x')\end{array}\right\}\right)$$

(5) $(\mathscr{D}, \mathscr{C}) \leftarrow$ a $\left(2^{O(\sqrt{\log\log n})}, 2^{O(\sqrt{\log\log n})}\right)$-network decomposition of $C^\star$.

(6) Compute the clustering of $V(C)$ defined by $\mathscr{D}$. For each $D \in \mathscr{D}$,

$$\mathrm{Cluster}^\star(D) \leftarrow \bigcup_{x \in D} \mathrm{Cluster}(x).$$

(7) For each color $k \in \left\{1, \ldots, 2^{O(\sqrt{\log\log n})}\right\}$,

(a) For each cluster $D \in \mathscr{D}$ with $\mathscr{C}(D) = k$, in parallel,

$J_D \leftarrow$ an MIS of the graph induced by $\mathrm{Cluster}^\star(D)\backslash\hat{\Gamma}(I_C)$.

(b) $I_C \leftarrow I_C \cup \bigcup_{\substack{D \in \mathscr{D}:\\ \mathscr{C}(D)=k}} J_D$.

(8) Return $I \cup \bigcup_{C \text{ in } G_{IB}} I_C$.

Fig. 3. A maximal independent set algorithm.

size $t \cdot \Delta^4$. Since the remaining steps operate on each component in G_{IB} independently, the message size required for Steps 2–8 is $O(\Delta^5 \log_\Delta n)$.

Step 2. At this point, we could simply run Panconesi and Srinivasan's [1996] deterministic MIS algorithm on each component. This would take time $2^{O(\sqrt{\log(t\Delta^4)})}$, which is not the desired bound, unless Δ happens to be polylogarithmic in n. In order to make this approach work for all Δ, we need to reduce the "effective" size of each component C to at most $\log n$, independent of Δ. After Step 2 we have partitioned $V(C) \subseteq V_{IB}$ into $\hat{\Gamma}(I_C)$ and B_C. As we argue below, Lemma 3.3(1) implies that each connected component of $B_C^{[1,4]}$ (the distance interval $[1, 4]$ being with respect to dist_C) is partitioned into $\log n$ low-radius clusters. This is the property of (I_C, R_C) that we use in subsequent steps.

Steps 3 and 4. Recall that nodes are assigned distinct $O(\log n)$-bit IDs. Using Corollary 2.5 with $\alpha = 5$, we can compute a $(5, 32\log\Delta + O(1))$-ruling set R_C for B_C in $O(\log\Delta + \log^* n)$ time, together with an $O(\log\Delta)$-radius clustering $\{\mathrm{Cluster}(u) \mid u \in R_C\}$

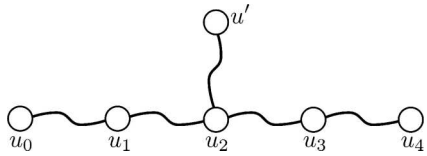

Fig. 4. The edges in $B_C^{[1,4]}$ between nodes in $\{u_0, \ldots, u_4, u'\}$ represent paths in C with length at most 4. If u_2 did not join the ruling set R'_C, then it must have been within distance 4 of some $u' \in R'_C$.

such that each cluster Cluster(u) is connected in $B_C^{[1,4]}$. Without loss of generality, assume henceforth that $B_C^{[1,4]}$ has one connected component. If not, then we apply these arguments to each connected component separately. Unfortunately, Lemma 3.3(1) cannot be applied directly to upper bound $|R_C|$ since for some $U' \subset R_C$, $\text{dist}_C(U', R_C \setminus U')$ is only guaranteed to be in the interval $[5, O(\log \Delta)]$, not $[5, 12]$. The solution is to consider a *superset* of R_C that does satisfy the criteria of Lemma 3.3(1).

LEMMA 3.4. *There exists an $R'_C \supset R_C$ such that for every $U' \subset R'_C$,* $\text{dist}_C(U', R'_C \setminus U') \in [5, 12]$.

PROOF. Initialize $R'_C \leftarrow R_C$ and consider each $v \in B_C \setminus R_C$ in turn, setting $R'_C \leftarrow R'_C \cup \{v\}$ if $\text{dist}_C(v, R'_C) \geq 5$. Suppose, for the sake of obtaining a contradiction, that after this process completes there is a set $U' \subset R'_C$ such that $\text{dist}_C(U', \overline{U'}) \geq 13$, where $\overline{U'} = R'_C \setminus U'$. Consider the paths between U'-nodes and $\overline{U'}$-nodes in $B_C^{[1,4]}$. Since each edge in these paths reflects at most four edges in C, the shortest path in $B_C^{[1,4]}$ from a U'-node to a $\overline{U'}$-node must have length at least $4 = \lceil 13/4 \rceil$. For the sake of specificity, suppose the path is $(u_0, u_1, u_2, u_3, u_4)$ and has length exactly 4, where $u_0 \in R'_C \cap U'$, $u_4 \in R'_C \cap \overline{U'}$, and $u_1, u_2, u_3 \notin R'_C$. See Figure 4. Clearly u_1, u_3 were excluded from R'_C because $\text{dist}_C(u_1, u_0) \leq 4$ and $\text{dist}_C(u_3, u_4) \leq 4$. However, u_2 must have been excluded because $\text{dist}_C(u_2, u') \leq 4$ for some other $u' \in R'_C$. Observe that both (u', u_2, u_1, u_0) and (u', u_2, u_3, u_4) are both strictly shorter than $(u_0, \ldots, u_4)$. Thus, regardless of whether $u' \in U'$ or $u' \in \overline{U'}$, the path $(u_0, \ldots, u_4)$ is *not* the shortest path from U' to $\overline{U'}$ in $B_C^{[1,4]}$, a contradiction. □

Lemma 3.4 states that R_C is contained within a set $R'_C \subseteq B_C$ to which Lemma 3.3(1) can be applied, proving that $|R_C| \leq |R'_C| \leq t$. Thus, the cluster graph $C^\star$ obtained by contracting each cluster Cluster(x) to a single node consists of connected components having at most t nodes.

Steps 5 and 6. We run Panconesi and Srinivasan's [1996] decomposition algorithm on each connected component of $C^\star$. (See Remark 3.6 for a discussion of the subtle difficulties in implementing this algorithm.) Since $|R_C| \leq t = \log_\Delta n < \log n$, we can compute a $(2^{O(\sqrt{\log\log n})}, 2^{O(\sqrt{\log\log n})})$-network decomposition $(\mathscr{D}, \mathscr{C})$ in $2^{O(\sqrt{\log\log n})}$ time. Since the underlying network is C, not $C^\star$, each step of this algorithm requires $64 \log \Delta + O(1)$ steps to simulate in C. The total time is therefore $\log \Delta \cdot 2^{O(\sqrt{\log\log n})}$. Since Cluster$^\star(D)$ is the union of disjoint clusters in $\{\text{Cluster}(x) \mid x \in D\}$, the diameter of Cluster$^\star(D)$ with respect to dist_C is at most $(64 \log \Delta + O(1)) \cdot 2^{O(\sqrt{\log\log n})}$.

Step 7. We extend I_C to an MIS on C using the network decomposition. For each color class, for each cluster D, supplement I_C with an MIS J_D on Cluster$^\star(D)/\hat{\Gamma}(I_C)$. These MISs are computed by the trivial algorithm and in parallel: A representative node in

D retrieves the status of all nodes in Cluster$^\star(D)$, in $O(\log\Delta \cdot 2^{O(\sqrt{\log\log n})})$ time, and then computes an MIS J_D and announces it to all nodes in Cluster$^\star(D)$. At the end of this process I_C is a maximal independent set on C.

Step 8 and Correctness. The set returned in Step 8, $I \cup \bigcup_C I_C$, is usually an MIS of G. However, poor random choices in Steps 1 and 2 can cause the algorithm to fail during Step 5. With high probability, the ruling sets $\bigcup_C R_C$ form connected components in $B_C^{[1,4]}$ of size at most t. If any is larger than t, then Steps 3 and 4 will be executed without error, but Step 5 may fail to produce a $(2^{O(\sqrt{\log\log n})}, 2^{O(\sqrt{\log\log n})})$-network decomposition in the time allotted. If this occurs, then Steps 6 and 7 cannot be executed.

Running Time. The time for Steps 1 and 2 is $O(\log^2\Delta)$ and the time for Steps 3 and 4 is $O(\log\Delta + \log^* n)$. Steps 5–7 take $O(\log\Delta)\cdot\exp(O(\sqrt{\log\log n}))$ time. In total, the time is $O(\log^2\Delta + \log\Delta\cdot\exp(O(\sqrt{\log\log n})))$, which is $O(\log^2\Delta + \exp(O(\sqrt{\log\log n})))$.

THEOREM 3.5. *In a graph with maximum degree Δ, an MIS can be computed in $O(\log^2\Delta + \exp(O(\sqrt{\log\log n})))$ time, with high probability, using messages with size $O(\Delta^5\log_\Delta n)$.*

Remark 3.6. One must be careful in applying deterministic algorithms in Phase II in a black box fashion. In the proof of Theorem 3.5 we reduced the number of clusters per component to t and deduced that the Panconesi-Srinivasan [1996] algorithm runs in $\log\Delta\cdot 2^{O(\sqrt{\log t})}$ time on each component. This is not a correct inference. The stated running time of the Panconesi-Srinivasan algorithm depends on nodes being endowed with $O(\log t)$-bit IDs (if the number of nodes is t), whereas in Step 5 nodes still have their original $O(\log n)$-bit IDs. There is a simple generic fix for this problem. Suppose a deterministic Phase II algorithm $\mathcal{A}$ runs in time $T = T(t)$ on any instance C with size t whose nodes are assigned distinct $O(\log t)$-bit labels. Let k be minimal such that $t \geq \log^{(k)} n$. Just before executing $\mathcal{A}$, first compute an $O(t^2\log^{(k)} n) = O(t^3)$-coloring in the graph $C^{[1,2T]}$ with Linial's [1992] algorithm and use these colors as $(3\log t + O(1))$-bit node IDs. This takes $O(Tk)$ time, that is, $O(T)$ time whenever $t = \log^{(O(1))} n$. As far as $\mathcal{A}$ can tell, all nodes have distinct IDs since no node can "see" two nodes with the same ID.

4. AN ALGORITHM FOR MAXIMAL MATCHING

The Match procedure given in Figure 5 is a generalized version of one iteration of the Israeli-Itai [1986] matching algorithm. It is given not-necessarily-disjoint node sets U_1, U_2 and a matching M and returns a matching on $U_1 \times U_2$ that is node-disjoint from M. It works as follows. Each unmatched node in U_1 *proposes* to an unmatched neighbor in U_2, selected uniformly at random. Each node in U_2 receiving a proposal *accepts* one, breaking ties by node ID. The accepted proposals form a set of *directed* paths and cycles. At this point each node v generates a bit $b(v)$: 0 if v is at the beginning of a path, 1 if at the end of a path, and uniformly at random otherwise. A directed edge (u, v) enters the matching if and only if $b(u) = 0$ and $b(v) = 1$. Refer to Figure 6 for an execution of Match on a small graph.

The procedure MaximalMatching has a two-phase structure. Phase I consists of $O(\log\Delta)$ *stages* in which the matching, M, is supplemented using two calls to Match. After Phase I all components of unmatched vertices have fewer than $s = (c\ln n)^9$ nodes, with probability $1 - n^{-\Omega(c)}$. We apply the deterministic $O(\log^4 s) = O(\log^4\log n)$ time

Match(U_1, U_2, M)

(1) Each $u \in U_1 \backslash V(M)$ *proposes* to $\text{prop}(u)$:
$\text{prop}(u) \leftarrow$ a random neighbor of u in $U_2 \backslash V(M)$.

(2) Each $v \in U_2 \backslash V(M)$ with a proposal *accepts* the best one:
$\text{prop}^\star(v) \leftarrow \underset{u\,:\,\text{prop}(u)=v}{\arg\max} \{\text{ID}(u)\}$.

(3) $F \leftarrow \{(\text{prop}^\star(v), v) \mid v \in U_2$ for which $\text{prop}^\star(v)$ exists$\}$
(F is a set of directed edges. It consists of directed paths and cycles.)

(4) Each $v \in U_1 \cup U_2$ with $\deg_F(v) > 0$ chooses a $b(v) \in \{0, 1\}$:

$$b(v) \leftarrow \begin{cases} 0 & \text{if } \text{indeg}_F(v) = 0, \\ 1 & \text{if } \text{outdeg}_F(v) = 0, \\ \text{a random value in } \{0,1\} & \text{otherwise.} \end{cases}$$

(5) Return the matching $\{(u, v) \in F \mid b(u) = 0 \text{ and } b(v) = 1\}$.

Fig. 5. An algorithm for computing a matching on $U_1 \times U_2$ disjoint from M.

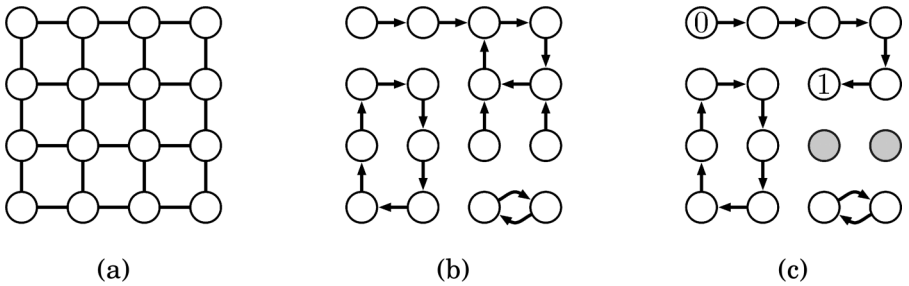

Fig. 6. One possible execution of Match$(V, V, \emptyset)$. Left: The undirected input graph $G = (V, E)$. Middle: The directed pseudoforest $(V, \{(u, \text{prop}(u))\})$ induced by the proposals. Right: F consists of directed paths and cycles. The beginning and end of each path are labeled 0 and 1, respectively. Grayed, isolated nodes receive no label. All other nodes are assigned random labels in $\{0, 1\}$.

maximal matching algorithm of Hańćkowiak et al. [2001] on each component, in parallel. In total the running time is $O(\log \Delta + \log^4 \log n)$.

Let $V_i \overset{\text{def}}{=} V(G) \backslash V(M)$ be the set of unmatched nodes just *before* stage i. For brevity, we let $\deg_i$ and Γ_i be the degree and neighborhood functions for the graph induced by V_i. The parameters for stage i are given below. Roughly speaking, δ_i is the maximum degree at stage i, $\tau_i = 2\delta_i/(c \ln n)$ is a certain "low-degree" threshold, and $\nu_i = \delta_i \tau_i / 2$ is a bound on the sum of degrees of nodes in $\Gamma_i(v)$, for any v. Define

$$\delta_i \overset{\text{def}}{=} \frac{\Delta \sqrt{c \ln n}}{\rho^i},$$

$$\tau_i \overset{\text{def}}{=} \frac{2\Delta}{\rho^i \sqrt{c \ln n}},$$

$$\text{and } \nu_i \overset{\text{def}}{=} \frac{\Delta^2}{\rho^{2i}} = \frac{\delta_i \tau_i}{2}, \quad \text{where } \rho \overset{\text{def}}{=} \sqrt{16/15} < 1.033.$$

MaximalMatching(Graph G)

Phase I:

(1) Initialize $M_0 \leftarrow \emptyset$

(2) For each *stage* i from 0 to $z \stackrel{\text{def}}{=} \log_\rho \Delta + \log_{4/3}(c \ln n) - 1$.

$M \leftarrow M \cup \mathsf{Match}(V_i^{\text{lo}}, V_i^{\text{hi}}, M)$

$M \leftarrow M \cup \mathsf{Match}(V_i, V_i, M)$

Phase II:

Let $\mathscr{C}$ be the connected components in the graph induced by V_z containing less than $(c \ln n)^9$ nodes.

(3) For each $C \in \mathscr{C}$,

$M_C \leftarrow$ a maximal matching on C

(4) Return $M \cup \bigcup_{C \in \mathscr{C}} M_C$.

Fig. 7. A maximal matching algorithm.

Define the low-degree and high-degree nodes before stage i to be

$$V_i^{\text{lo}} \stackrel{\text{def}}{=} \{v \in V_i \mid \deg_i(v) \leq \tau_{i+1}\}$$
$$\text{and } V_i^{\text{hi}} \stackrel{\text{def}}{=} \{v \in V_i \mid \deg_i(v) > \delta_{i+1}\}.$$

Note that nodes with degree between τ_{i+1} and δ_{i+1} are in neither set. In stage i we supplement the current matching, first, with a matching on $V_i^{\text{lo}} \times V_i^{\text{hi}}$, and then with a matching on V_i. As we soon show, certain invariants will hold after stage i with probability $1 - \exp(-\Omega(\tau_i))$. Thus, in order to obtain high probability bounds, we must switch to a different analysis when $\tau_i = \Theta(\log n)$, that is, when the maximum degree is $\delta_i = \Theta(\log^2 n)$.

The algorithm always returns a matching. According to Phase II of Figure 7, $\mathscr{C}$ is the set of all connected components leftover after Phase I that have size at most $(c \ln n)^9$. Thus, if $\mathscr{C}$ does not exclude any connected components, then the matching returned after Phase II will be maximal.[15] Our goal is therefore to show that, with high probability, after Phase I there is no connected component of unmatched nodes with size greater than $(c \ln n)^9$. In the lemma below $\deg(S)$ is short for $\sum_{u \in S} \deg(u)$, where $S \subset V$.

LEMMA 4.1. *Define $i^\star$ to be the last stage for which $\tau_{i^\star} \geq 2c \ln n$. With probability $1 - 2n^{-c/660+1}$, the following bounds hold for all $v \in V(G)$ after each stage $i < i^\star$:*

$$\deg_{i+1}(v) \leq \delta_{i+1}$$
$$\textit{and } \deg_{i+1}^{(2)}(v) \leq \nu_{i+1},$$

where $\deg_{i+1}^{(2)}(v) \stackrel{\text{def}}{=} \deg_{i+1}(\Gamma_{i+1}(v))$.

[15]Note that individual nodes generally do not know whether they are in a component of $\mathcal{C}$. In Phase II they will execute a deterministic maximal matching algorithm for enough steps to complete on any graph with $(c \ln n)^9$ nodes. It is only if a node *fails* to terminate in time that it deduces that it was not in $\mathcal{C}$ after all and that this execution of MaximalMatching has failed.

PROOF. The inequalities hold trivially when $i = 0$. We analyze the probability that they hold after stage i, assuming they hold just before stage i. For the sake of minimizing notation we use $\deg_i, \Gamma_i$, and so on, to refer to the degree and neighborhood functions just before *each* call to Match in stage i. This should not cause confusion.

Consider a node $v \in V_i$ at the beginning of stage i. By assumption $\deg_i(v) \leq \delta_i$ and $\deg_i^{(2)}(v) \leq \nu_i$. Since, by definition, nodes in V_i^{lo} have degree at most τ_{i+1}, v has less than $\nu_i/\tau_{i+1} = \delta_{i+1} \cdot (\rho^2/2)$ neighbors that are *not* in V_i^{lo}. We argue that if $v \in V_i^{\text{hi}}$ (that is, $\deg_i(v) > \delta_{i+1}$), then v will be matched in the *first* call to Match in stage i with probability $1 - \exp((1 - \rho^2/2)c \ln n/2)$. Note that the forest induced by the proposals consists solely of stars (all edges being directed from V_i^{lo} to V_i^{hi}) which implies that F, the graph consisting of accepted proposals, consists solely of single-edge paths. Single-edge paths in F are always committed to the matching since their endpoints' b-values are chosen deterministically in Step 4 of Match to satisfy the criterion of Step 5. Thus, $v \in V_i^{\text{hi}}$ will be matched if any neighbor $u \in V_i^{\text{lo}}$ chooses (u, v) in Step 2. The probability that this does *not* occur is at most

$$\begin{aligned}\left(1 - \frac{1}{\tau_i}\right)^{|\Gamma_i(v) \cap V_i^{\text{lo}}|} &\leq \left(1 - \frac{1}{\tau_i}\right)^{\left(1 - \frac{\rho^2}{2}\right)\delta_{i+1}} \\ &\leq \exp\left(-\left(1 - \frac{\rho^2}{2}\right)\frac{\delta_{i+1}}{\tau_i}\right) \\ &= \exp\left(-\left(1 - \frac{\rho^2}{2}\right)\frac{c \ln n}{2\rho}\right) < n^{-0.22c} \qquad \{\rho < 1.033\}\end{aligned}$$

By a union bound, every $v \in V_i^{\text{hi}}$ will be matched with probability more than $1 - n^{-c/5+1}$. Therefore, we proceed under the assumption that after the first call to Match in stage i, all unmatched nodes have degree less than δ_{i+1}. It remains to show that after the second call to Match, $\deg_{i+1}^{(2)}(v) \leq \nu_{i+1}$ for all $v \in V(G)$.

A node v will be guaranteed to have positive degree in F under two circumstances: (i) Some node offers v a proposal or (ii) among those nodes proposing to prop(v), v has the highest ID. Once v is in a path or cycle in F it becomes matched with probability at least 1/2. (It is actually exactly 1/2, except if v is in a single-edge path, in which case it is 1.)

In the following analysis, we first expose the proposals made by all nodes in $V_i \setminus \hat{\Gamma}_i(v)$ then expose the proposals of $\hat{\Gamma}_i(v)$ in *descending* order of node ID. Consider the moment just before a neighbor $u \in \Gamma_i(v)$ makes a proposal. If at least $\deg_i(u)/2$ neighbors of u have yet to receive a proposal (by nodes already evaluated) then place u in set A, otherwise place u in set B. If u is put in set A and u *does* offer prop(u) its first proposal thus far—implying that u will have positive degree in F—then also place u in set A'. See Figure 8 for an illustration.

We split the rest of the analysis into two cases depending on whether A-nodes or B-nodes account for the larger share of edges in v's 2-neighborhood. In both cases we show that $\deg_{i+1}^{(2)}(v) \leq \nu_{i+1}$ with high probability.

4.1. Case I: The *A*-Nodes

We first analyze the case that $\deg_i(A) \geq \deg_i^{(2)}(v)/2 \geq \nu_{i+1}/2$. (If $\deg_i^{(2)}(v)$ is already less than ν_{i+1}, then there is nothing to prove.) Observe that each node u, once in A, is moved

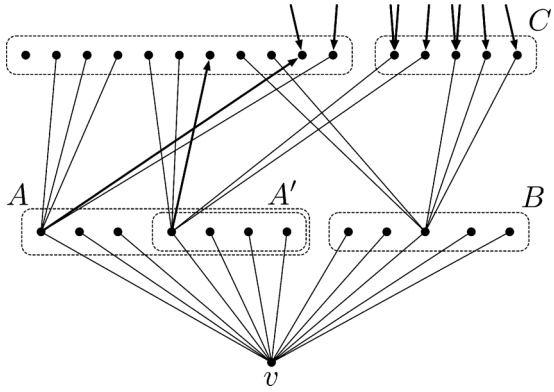

Fig. 8. The neighborhood of v is partitioned into A and B, and A is partitioned into A' and $A \backslash A'$. Proposals are indicated by directed edges. A node is in A if a majority of its neighbors do not already have a proposal and in B otherwise. An A-node is in A' if it makes the first proposal to a node. A node is in C if it is adjacent to B and has a proposal. Note: Nodes with a proposal that are adjacent to A but not B are not in C. Contrary to the depiction, A-nodes and B-nodes may be adjacent and C may intersect both A and B.

to A' with probability at least 1/2, and, if so, contributes $\deg_i(u) \leq \delta_{i+1}$ to $\deg_i(A')$.[16] The probability that, after evaluating each $u \in \Gamma_i(v)$, $\deg_i(A')$ is less than a $\frac{1}{\sqrt{2}}$-fraction of its expectation is

$$
\begin{aligned}
&\Pr(\deg_i(A') < \tfrac{1}{\sqrt{2}} \cdot \mathrm{E}[\deg_i(A')]) \\
&\leq \exp\left(-\frac{\left((1-\frac{1}{\sqrt{2}})\,\mathrm{E}[\deg_i(A')]\right)^2}{2\sum_{u\in A}(\deg_i(u))^2}\right) && \{\text{Corollary A.5}\} \\
&\leq \exp\left(-\frac{\left((1-\frac{1}{\sqrt{2}})\frac{1}{2}\deg_i(A)\right)^2}{2(\deg_i(A)/\delta_{i+1})\delta_{i+1}^2}\right) && \{\text{linearity of expectation}\} \\
&\leq \exp\left(-\left(\frac{(1-\frac{1}{\sqrt{2}})^2}{8}\right)\left(\frac{\deg_i(A)}{\delta_{i+1}}\right)\right) \\
&\leq \exp\left(-\left(\frac{(1-\frac{1}{\sqrt{2}})^2}{32}\right)\tau_{i+1}\right) && \left\{\deg_i(A) \geq \frac{\nu_{i+1}}{2} = \frac{\delta_{i+1}\tau_{i+1}}{4}\right\} \\
&< n^{-c/187} && \{\tau_{i+1} \geq \tau_{i^\star} \geq 2c\ln n\}.
\end{aligned}
$$

We proceed under the assumption that this unlikely event does not hold, so $\deg_i(A') \geq \frac{1}{\sqrt{2}} \cdot \mathrm{E}[\deg_i(A')] \geq \frac{1}{2\sqrt{2}} \cdot \deg_i(A) \geq \frac{1}{4\sqrt{2}} \cdot \nu_{i+1}$. Since each node with positive degree in F is matched with probability at least 1/2, by linearity of expectation $\mathrm{E}[\deg_i(A') - \deg_{i+1}(A')] \geq \frac{1}{2}\deg_i(A')$. Moreover, whether $v \in A'$ is matched depends only on the b-values of neighboring nodes in F. The dependency graph of these events has chromatic number $\chi = 5$ since the nodes of a cycle can be 5-colored such that any two nodes within distance 2 receive different colors. The probability that $\deg_i(A') - \deg_{i+1}(A')$ is

[16]Note that this process fits in the martingale framework of Corollary A.5. Here X_j is the state of the system after evaluating the jth neighbor u of v and Z_j is $\deg_i(u)$ if u joins A' and 0 otherwise, which is a function of X_j. Thus, each Z_j has a range of at most δ_{i+1}.

less than a $\frac{1}{\sqrt{2}}$-fraction of its expectation is therefore

$$\begin{aligned}
&\Pr\Big(\deg_i(A') - \deg_{i+1}(A') < \tfrac{1}{\sqrt{2}} \cdot \mathrm{E}[\deg_i(A') - \deg_{i+1}(A')]\Big) \\
&\leq \exp\left(-\frac{2\Big(\Big(1-\frac{1}{\sqrt{2}}\Big)\mathrm{E}[\deg_i(A') - \deg_{i+1}(A')]\Big)^2}{\chi \cdot \sum_{u\in A'}(\deg_i(u))^2}\right) && \{\text{Theorem A.3, } \chi = 5\} \\
&\leq \exp\left(-\frac{2\Big(\Big(1-\frac{1}{\sqrt{2}}\Big)\frac{1}{2}\deg_i(A')\Big)^2}{\chi \cdot (\deg_i(A')/\delta_{i+1})\delta_{i+1}^2}\right) \\
&\leq \exp\left(-\left(\frac{(1-\frac{1}{\sqrt{2}})^2}{10}\right)\left(\frac{\deg_i(A')}{\delta_{i+1}}\right)\right) \\
&\leq \exp\left(-\left(\frac{(1-\frac{1}{\sqrt{2}})^2}{80\sqrt{2}}\right)\tau_{i+1}\right) && \left\{\deg_i(A') \geq \frac{\nu_{i+1}}{4\sqrt{2}} = \frac{\delta_{i+1}\tau_{i+1}}{8\sqrt{2}}\right\} \\
&< n^{-c/660} && \{\tau_{i+1} \geq \tau_{i^\star} \geq 2c\ln n\}.
\end{aligned}$$

To sum up, if this unlikely event does not occur,

$$\begin{aligned}
\deg_i^{(2)}(v) - \deg_{i+1}^{(2)}(v) &\geq \deg_i(A') - \deg_{i+1}(A') && \{\text{because } A' \subseteq \Gamma_i(v)\} \\
&\geq \frac{1}{\sqrt{2}} \cdot \mathrm{E}[\deg_i(A') - \deg_{i+1}(A')] \\
&\geq \frac{1}{2\sqrt{2}} \cdot \deg_i(A') \;\geq\; \left(\frac{1}{2\sqrt{2}}\right)^2 \cdot \deg_i(A) \;\geq\; \frac{1}{16}\deg_i^{(2)}(v).
\end{aligned}$$

Thus, with high probability, $\deg_{i+1}^{(2)}(v) \leq \frac{15}{16} \cdot \deg_i^{(2)}(v)$.

4.2. Case II: The *B*-Nodes

We now turn to the case when $\deg_i(B) \geq \frac{1}{2} \cdot \deg_i^{(2)}(v) \geq \frac{1}{2} \cdot \nu_{i+1}$. By definition, just before any $u \in B$ makes its proposal, at least $\frac{1}{2} \cdot \deg_i(u)$ of its neighbors have already received a proposal. We do not care who u proposes to. Let $C \subseteq \Gamma_i(B)$ be the set of nodes in B's neighborhood that receive at least one proposal. For $x \in C$, let $\deg_B(x) \leq \delta_{i+1}$ be the number of its neighbors in B. Thus, if x is matched, then $\deg^{(2)}(v)$ is reduced by at least $\deg_B(x)$. It follows that

$$\begin{aligned}
\deg_B(C) = \sum_{x\in C}\deg_B(x) = \sum_{u\in B}\deg_C(u) &\geq \sum_{u\in B}\frac{1}{2}\cdot\deg_i(u) && \{\text{by defn. of } u \in B\} \\
= \frac{1}{2}\cdot\deg_i(B) \geq \frac{1}{4}\cdot\deg_i^{(2)}(v) &> \frac{1}{4}\cdot\nu_{i+1}.
\end{aligned}$$

Since C-nodes are matched with probability 1/2, by linearity of expectation, $\mathrm{E}[\deg_{i+1}(B)] \leq \deg_i(B) - \frac{1}{2}\cdot\deg_B(C) \leq \frac{3}{4}\deg_i(B)$. We bound the probability that $\deg_{i+1}(B)$ deviates from its expectation using Janson's inequality in exactly the same way as we

bounded $\deg_{i+1}(A')$. It follows that

$$\begin{aligned}
&\Pr\left(\deg_{i+1}(B) \geq \deg_i(B) - \frac{1}{4}\cdot\deg_B(C)\right)\\
&\leq \exp\left(-\frac{2(\frac{1}{4}\deg_B(C))^2}{\chi\cdot\sum_{x\in C}(\deg_B(x))^2}\right) && \{\text{Theorem A.3}\}\\
&\leq \exp\left(-\frac{1}{40}\cdot\frac{(\deg_B(C))^2}{(\deg_B(C)/\delta_{i+1})\delta_{i+1}^2}\right) && \{\chi = 5, \deg_B(x)\leq\delta_{i+1}\}\\
&\leq \exp\left(-\frac{1}{320}\tau_{i+1}\right) && \{\deg_B(C)\geq \nu_{i+1}/4 = \delta_{i+1}\tau_{i+1}/8\}\\
&\leq n^{-c/160} && \{\tau_{i+1}\geq\tau_{i^\star}\geq 2c\ln n\}.
\end{aligned}$$

Thus, with high probability

$$\deg^{(2)}_{i+1}(v) \leq \deg^{(2)}_i(v) - \frac{1}{4}\cdot\deg_B(C) \leq \frac{15}{16}\cdot\deg^{(2)}_i(v),$$

since $\deg_B(C)\geq\frac{1}{4}\cdot\deg^{(2)}_i(v)$. Whether we are in Case I or Case II, $\deg^{(2)}_{i+1}(v)\leq\frac{15}{16}\cdot\deg^{(2)}_i(v)\leq\frac{15}{16}\cdot\nu_i$ with high probability. Since $\nu_{i+1}=\nu_i/\rho^2$, we set $\rho=\sqrt{16/15}$.

By a union bound, the probability of error at *any* node is at most $2n^{-c/660+1}$. This covers the probability that the first call to Match fails to match all V_i^{hi}-nodes or the second call fails to make $\deg^{(2)}_{i+1}(v)\leq\nu_{i+1}$ for all $v\in V_i$. □

4.3. The Emergence of Small Components

Lemma 4.1 implies that before stage $i^\star < \log_\rho\Delta$, the maximum degree is at most $\delta_{i^\star} = \tau_{i^\star}(c/2)\ln n \leq (c\ln n)^2$. In Lemmas 4.2 and 4.3, we prove that after another $O(\log\log n)$ iterations of the Match procedure, all components of unmatched vertices have size at most $(c\ln n)^9$, with high probability. Thus, Phase II of MaximalMatching correctly extends the matching after Phase I to a maximal matching.

Lemma 4.2. *For any node v and any stage i,* $\Pr(\deg_{i+1}(v)\leq\frac{3}{4}\cdot\deg_i(v))\geq\frac{1}{4}$.

Proof. We analyze the expected drop in v's degree during the *second* call to Match (the one in which all nodes participate) and then apply Markov's inequality. Expose the proposals in descending order of node ID and consider the moment just before v makes its proposal. Let $P\subseteq\Gamma_i(v)$ be those neighbors already holding a proposal and $Q\subseteq\Gamma_i(v)$ be the neighbors with no proposal. All nodes in P will be matched with 1/2 probability and v will be matched with 1/2 probability *if* it proposes to a member of Q. The probability v is matched is at least $\frac{\epsilon}{2}$, where $\epsilon = |Q|/\deg_i(v)$. The probability that $u\in P$ is still a neighbor of v after this call to Match is therefore at most $\frac{1}{2}(1-\frac{\epsilon}{2})$. The probability that $u\in Q$ is still a neighbor is at most $1-\frac{\epsilon}{2}$. By linearity of expectation,

$$\begin{aligned}
\mathrm{E}[\deg_{i+1}(v)] &\leq \left(\epsilon\left(1-\tfrac{\epsilon}{2}\right)+\tfrac{1}{2}(1-\epsilon)\left(1-\tfrac{\epsilon}{2}\right)\right)\cdot\deg_i(v)\\
&= (1-\tfrac{\epsilon}{2})(\tfrac{1}{2}+\tfrac{\epsilon}{2})\cdot\deg_i(v)\\
&\leq \left(\tfrac{3}{4}\right)^2\cdot\deg_i(v) && \{\text{maximized at } \epsilon = 1/2\}.
\end{aligned}$$

That is, we lose at least a $\frac{7}{16}$-fraction of v's neighbors in expectation. By Markov's inequality, $\Pr(\deg_{i+1}(v)\leq\frac{3}{4}\cdot\deg_i(v))\geq\frac{1}{4}$. □

LEMMA 4.3. *Let $\hat{G}$ be the subgraph induced by unmatched nodes at some point in Phase I, whose maximum degree is at most $\hat{\Delta}$. After $12\log_{4/3}\hat{\Delta}$ more stages in Phase I, all components of unmatched nodes have size at most $t\hat{\Delta}^4$ with probability $1-n^{-c}$, where $t \stackrel{\text{def}}{=} c\ln n$.*

PROOF. The proof follows the same lines at that of Lemmas 3.2 and 3.3 but has some added complications. We say v is *successful* in stage i if $\deg_{i+1}(v) \le \frac{3}{4}\cdot\deg_i(v)$. If v experiences $\log_{4/3}\hat{\Delta}$ successes, then either v has been matched or all neighbors of v are matched.

The events that u and v are successful in a particular stage i are independent if $\text{dist}_{\hat{G}}(u,v) \ge 5$ since the success of u and v only depend on the random choices of nodes within distance 2. Any subgraph of size $t\hat{\Delta}^4$ must contain a subset T of t nodes such that (i) each pair of nodes in T is at distance at least 5 and (ii) T forms a t-node tree in $\hat{G}^5$. Call T a *distance-5 set* if $|T| = t$ and it satisfies (i) and (ii). There are less than $4^t\cdot n\cdot\hat{\Delta}^{5(t-1)}$ distance-5 sets in $\hat{G}$. (There are less than 4^t topologically distinct trees with t nodes and less than $n\hat{\Delta}^{5(t-1)}$ ways to embed one such tree in $\hat{G}^5$.)

Consider any distance-5 set T. Over $12\log_{4/3}\hat{\Delta}$ consecutive stages, $v\in T$ experiences some number of successful stages. Call this random variable X_v and define $X \stackrel{\text{def}}{=} \sum_{v\in T} X_v$. By Lemma 4.2 and linearity of expectation,

$$\text{E}[X] = \sum_{v\in T} E[X_v] \ge t\cdot\tfrac{1}{4}(12\log_{4/3}\hat{\Delta}) = 3t\log_{4/3}\hat{\Delta}.$$

If $X \ge t\log_{4/3}\hat{\Delta}$, then some $X_v \ge \log_{4/3}\hat{\Delta}$, implying that v becomes isolated and therefore that no component contains all T-nodes. We will call T *successful* if any member of T becomes isolated. By a Chernoff bound (Theorem A.2), the probability that T is unsuccessful is at most

$$\begin{aligned}
\Pr\big(X < t\log_{4/3}\hat{\Delta}\big) &\le \Pr\left(X < \frac{1}{3}\cdot\text{E}[X]\right)\\
&\le \exp\left(-\frac{2\big(\frac{2}{3}\,\text{E}[X]\big)^2}{4t\log_{4/3}\hat{\Delta}}\right)\\
&\le \exp\big(-2t\log_{4/3}\hat{\Delta}\big) \qquad \{\text{E}[X] \ge 3t\log_{4/3}\hat{\Delta}\}.\\
&= \hat{\Delta}^{-(2\log_{4/3}e)t}
\end{aligned}$$

After $12\log_{4/3}\hat{\Delta}$ stages, if there exists a component with size $t\hat{\Delta}^4$, then it must contain an unsuccessful subset T. By the union bound, this occurs with probability less than

$$\begin{aligned}
&4^t\cdot n\cdot\hat{\Delta}^{5(t-1)}\cdot\hat{\Delta}^{-(2\log_{4/3}e)t}\\
&< 4^{c\ln n}\cdot n\cdot\hat{\Delta}^{(5-2\log_{4/3}e)\cdot c\ln n}\\
&< n^{-c} \qquad \{\text{for } \hat{\Delta} \text{ sufficiently large. Note: } 5-2\log_{4/3}e < 0\}. \quad \square
\end{aligned}$$

THEOREM 4.4. *In a graph with maximum degree Δ, a maximal matching can be computed in $O(\log\Delta + \log^4\log n)$ time with high probability using $O(1)$-size messages. When the graph is bipartite and 2-colored, the time bound becomes $O(\log\Delta + \log^3\log n)$.*

PROOF. After $i^\star = \log_\rho(\Delta/(c\ln n)^{3/2})$ stages in Phase I the maximum degree is $\hat{\Delta} = (c\ln n)^2$, with high probability. After another $4\log_{4/3}\hat{\Delta}$ stages in Phase I all connected

components have at most $s \stackrel{\text{def}}{=} \hat{\Delta}^4 \cdot c \ln n = (c \ln n)^9$ nodes, with high probability. We execute the deterministic maximal matching algorithm of Hańćkowiak et al. [2001] for time sufficient to solve any instance on s nodes: $O(\log^4 s)$ time for general graphs and $O(\log^3 s)$ time for bipartite, 2-colored graphs. Both Phase I and Phase II can be implemented with $O(1)$-size messages; that is, this algorithm works in the CONGEST model. □

5. VERTEX COLORING

We consider a slightly more stringent version of $(\Delta+1)$-coloring called $(\deg+1)$-coloring, where each node v must adopt a color from the palette $\{1, \ldots, \deg(v)+1\}$, or, more generally, an arbitrary set with size $\deg(v)+1$.[17] Although the palette of a node does not depend on Δ, our algorithm still requires that nodes know Δ and n.[18]

In Section 5.1, we define and analyze a natural $O(1)$-time algorithm called OneShotColoring that colors a subset of the nodes. Johannson [1999] showed that $O(\log n)$ applications of a variant of OneShotColoring suffice to $(\Delta+1)$-color a graph, with high probability. Our goal is to show something stronger. We show that after $O(\log \Delta)$ applications of OneShotColoring, all nodes have at most $O(\log n)$ uncolored neighbors that each have $\Omega(\log n)$ uncolored neighbors. This property allows us to reduce the resulting $(\deg+1)$-coloring problem to *two* $(\deg+1)$-coloring problems on subgraphs with maximum degree $O(\log n)$. It is shown that, on these instances, $O(\log\log n)$ further applications of OneShotColoring suffice to reduce the size of all uncolored components to $\text{poly}(\log n)$. In Phase II we apply the deterministic $(\deg+1)$-coloring algorithm of Panconesi and Srinivasan [1996] to the $\text{poly}(\log n)$-size uncolored components. The remainder of this section constitutes a proof of Theorem 5.1.

THEOREM 5.1. *In a graph with maximum degree Δ, a $(\deg+1)$-coloring can be computed in $O(\log \Delta + \exp(O(\sqrt{\log\log n})))$ time using $\text{poly}(\log n)$-length messages.*

5.1. Analysis of OneShotColoring

The algorithm maintains a proper partial coloring $\text{Color} : V(G) \to \{\perp, 1, \ldots, \Delta+1\}$, where $\perp$ denotes no color and $\text{Color}(v) \in \{\perp, 1, \ldots, \deg(v)+1\}$. Initially $\text{Color}(v) \leftarrow \perp$ for all $v \in V(G)$. Before a call to OneShotColoring some nodes have already committed to their final colors. Each remaining uncolored node v chooses $\text{Color}^\star(v)$, a color selected uniformly at random from its remaining palette. It may be that neighbors of v also choose $\text{Color}^\star(v)$. If v holds the highest ID among all such nodes contending for $\text{Color}^\star(v)$, then it permanently commits to that color. The pseudocode for OneShotColoring appears in Figure 9.

We analyze the properties of OneShotColoring from the point of view of some arbitrary uncolored node $v \in U$. Note that whether v is colored depends only on its behavior and the behavior of neighbors with larger IDs, denoted $\Gamma_U^>(v) \stackrel{\text{def}}{=} \{u \in \Gamma_U(v) \mid \text{ID}(u) > \text{ID}(v)\}$. Define $\Psi^{-1}(q) \stackrel{\text{def}}{=} \{u \in \Gamma_U^>(v) \mid q \in \Psi(u)\}$ to be the set of v's uncolored neighbors that are contending for color q and have higher IDs. Define $w(q) = \sum_{u \in \Psi^{-1}(q)} 1/|\Psi(u)|$ to be the *weight* of color q. In other words, each neighbor u distributes $1/|\Psi(u)|$ units of weight to each color in its palette. Note that $1/|\Psi(u)| \le 1/(\deg_U(u)+1) \le 1/2$. The probability

[17] Some applications [Amir et al. 2014] demand $(\deg+1)$-colorings, not $(\Delta+1)$-colorings.

[18] Again, this assumption is for convenience. It can be removed using the method of Korman et al. [2013]. However, the Korman et al. technique does *not* apply to any "black box" $(\Delta+1)$-coloring algorithm since the problem of completing a partial $(\Delta+1)$-coloring is not itself an instance of $(\Delta+1)$-coloring.

OneShotColoring(G, Color)

Define $U \subseteq V(G)$ and $\Psi : V(G) \to 2^{\{1,\ldots,\Delta+1\}}$ as follows.

$$U \stackrel{\text{def}}{=} \{u \in V(G) \mid \mathrm{Color}(u) = \perp\}, \qquad \text{the uncolored vertices,}$$

$$\text{and} \quad \Psi(v) \stackrel{\text{def}}{=} \{1, \ldots, \deg(v)+1\} \setminus \mathrm{Color}(\Gamma(v)), \qquad v\text{'s available palette.}$$

The following steps are executed for all $v \in U$, in parallel.

(1) Select a $\mathrm{Color}^\star(v) \in \Psi(v)$ uniformly at random.

(2) If $\mathrm{ID}(v) > \max\{\mathrm{ID}(u) \mid u \in \Gamma_U(v)$ and $\mathrm{Color}^\star(u) = \mathrm{Color}^\star(v)\}$, Permanently assign $\mathrm{Color}(v) \leftarrow \mathrm{Color}^\star(v)$.

Fig. 9. An $O(1)$-round algorithm for extending a partial coloring.

that $q \in \Psi(v)$ is *available* to v after exposing $\mathrm{Color}^\star(\Gamma^>_U(v))$ is

$$\begin{aligned}\Pr(q \notin \mathrm{Color}^\star(\Gamma^>_U(v))) &= \prod_{u \in \Psi^{-1}(q)} \left(1 - \tfrac{1}{|\Psi(u)|}\right) \\ &\geq \prod_{u \in \Psi^{-1}(q)} \left(\tfrac{1}{4}\right)^{1/|\Psi(u)|} \qquad (1) \\ &= \left(\tfrac{1}{4}\right)^{w(q)}.\end{aligned}$$

Inequality (1) follows from the fact that $(1-x) \geq (1/4)^x$ when $x \in [0, 1/2]$. Let $X_q \in \{0, 1\}$ be the indicator variable for the event that q is available and $X = \sum_q X_q$. By linearity of expectation, $\mathrm{E}[X] \geq \sum_{q \in \Psi(v)} \left(\frac{1}{4}\right)^{w(q)}$. By the convexity of the exponential function, this quantity is minimized when all color weights are equal. Hence,

$$\begin{aligned}\mathrm{E}[X] \geq \sum_{q \in \Psi(v)} \left(\tfrac{1}{4}\right)^{w(q)} &\geq |\Psi(v)| \cdot \left(\tfrac{1}{4}\right)^{\sum_q w(q)/|\Psi(v)|} \\ &\geq |\Psi(v)| \cdot \left(\tfrac{1}{4}\right)^{\deg_U(v)/|\Psi(v)|} \qquad (2) \\ &|\Psi(v)|/4. \qquad (3)\end{aligned}$$

Inequalities (2) and (3) follow from the fact that each neighbor in $\Gamma^>_U(v)$ can contribute at most one unit of weight and that $|\Psi(v)| \geq \deg_U(v) + 1 \geq \deg^>_U(v) + 1$. We will call v *happy* if $X \geq |\Psi(v)|/8$, that is, if the number of available colors is at least half its expectation. Let $\mathscr{H}_v$ be the event that v is happy. The variables $\{X_q\}$ are not independent. However, Dubhashi and Ranjan [1998] showed that $\{X_q\}$ are negatively correlated and, more generally, that all balls and bins experiments of this form give rise to negatively correlated variables.[19] By Theorem A.2,

$$\Pr(\overline{\mathscr{H}_v}) \stackrel{\text{def}}{=} \Pr\left(X < \frac{|\Psi(v)|}{8}\right) < \exp\left(-\frac{2 \cdot (|\Psi(v)|/8)^2}{|\Psi(v)|}\right) = \exp\left(-\frac{|\Psi(v)|}{32}\right).$$

Lemma 5.2 summarizes the relevant properties of OneShotColoring used in the next section.

[19] In this situation the colors are bins and the neighbors' choices are balls.

$(\deg+1)$-Coloring(Graph G)

Phase I:

(1) Initialize $\mathrm{Color}(v) \leftarrow \perp$, for all $v \in V(G)$.

(2) Repeat $\log_{16/15}\Delta$ times:
OneShotColoring(G, Color)

(3) Form high-degree and low-degree graphs.

$$U \leftarrow \{v \in V(G) \mid \mathrm{Color}(v) = \perp\} \qquad \text{uncolored nodes}$$
$$U^{\mathrm{hi}} \leftarrow \{v \in U \mid \deg_U(v) > \hat{\Delta} \stackrel{\mathrm{def}}{=} c \ln n\} \qquad \text{high-degree nodes}$$
$$G^{\mathrm{hi}} \leftarrow \text{ the graph induced by } U^{\mathrm{hi}}$$
$$G^{\mathrm{lo}} \leftarrow \text{ the graph induced by } U \backslash U^{\mathrm{hi}}$$

(4) Repeat $5\log_{4/3}\hat{\Delta}$ times:
OneShotColoring$(G^{\mathrm{hi}}, \mathrm{Color})$

(5) Repeat $5\log_{4/3}\hat{\Delta}$ times:
OneShotColoring$(G^{\mathrm{lo}}, \mathrm{Color})$

Phase II:

(6) Color all remaining uncolored components of G^{hi} with size at most $\hat{\Delta}^3$.

(7) Color all remaining uncolored components of G^{lo} with size at most $\hat{\Delta}^3$.

Fig. 10. A $(\Delta+1)$-coloring algorithm.

LEMMA 5.2. *Let U be the uncolored nodes before a call to* OneShotColoring *and $v \in U$ be arbitrary.*

(*1*) *(Johansson [1999])* $\Pr(v \text{ is colored}) > 1/4$.

(*2*) $\Pr(\mathscr{H}_v) > 1 - \exp(-\frac{\deg_U(v)+1}{32})$.

5.2. A (deg +1)-Coloring Algorithm

It goes without saying that our $(\deg+1)$-Coloring algorithm (Figure 10) has a two-phase structure. The ultimate goal of Phase I is to reduce the global problem to some number of independent $(\deg+1)$-coloring subproblems, each on $\mathrm{poly}(\log n)$-size components, which can be colored deterministically in Phase II. We first prove that this is possible with $O(\log\log n)$ applications of OneShotColoring, *if* the uncolored subgraph already has maximum degree $\mathrm{poly}(\log n)$.

LEMMA 5.3. *Apply an arbitrary proper partial coloring to G, and let $\hat{\Delta}$ be the maximum degree in the subgraph induced by uncolored nodes. After $5\log_{4/3}\hat{\Delta}$ iterations of* OneShotColoring*, all uncolored components have less than $t\hat{\Delta}^2$ nodes with probability $1 - n^{-c}$, where $t \stackrel{\mathrm{def}}{=} c\log_{\hat{\Delta}} n$.*

PROOF. The proof is similar to that of Lemmas 3.3 and 4.3. Whether a node is colored depends only on the color choices of nodes in its inclusive neighborhood. Thus, if two nodes are at distance at least 3, then their coloring events are independent. Let $T \subset U$ be a distance-3 set, that is, one for which (i) $|T| = t = c\log_{\hat{\Delta}} n$, (ii) the distance between each pair of nodes is at least 3, and (iii) T forms a tree in the uncolored part of G^3. There are less than $4^t \cdot n \cdot \hat{\Delta}^{3(t-1)} < n^{4c}$ distance-3 sets and the probability that one is entirely uncolored after $5\log_{4/3}\hat{\Delta}$ iterations of OneShotColoring is, by Lemma 5.2, less

than

$$\left(\tfrac{3}{4}\right)^{5t\log_{4/3}\hat{\Delta}} = \left(\tfrac{3}{4}\right)^{5(c\log_{\hat{\Delta}} n)\log_{4/3}\hat{\Delta}} = n^{-5c}.$$

By a union bound, no distance-3 set exists with probability $n^{4c-5c} = n^{-c}$. Moreover, if there were an uncolored component with size $t\hat{\Delta}^2$ after $5\log_{4/3}\hat{\Delta}$ iterations of OneShotColoring, it would have to contain such a distance-3 set. □

Lemma 5.3 implies a (deg +1)-coloring algorithm running in $O(\log\Delta + \exp(O(\sqrt{\log(\Delta^2\log n)})))$ time. Once the component size is less than $\Delta^2\log n$ we can apply the deterministic (deg +1)-coloring algorithm of Panconesi and Srinivasan [1996] to each uncolored component. The exponential dependence on $\sqrt{\log\Delta}$ is undesirable. Using Lemma 5.2 we show that, roughly speaking, degrees decay geometrically with each call to OneShotColoring, with high probability. This will allow us to reduce the dependence on n to $\exp(O(\sqrt{\log\log n}))$.

LEMMA 5.4. *Define* $U^{\text{hi}} = \{u \in U \mid \deg_U(u) > \hat{\Delta}\}$ *to be those high-degree uncolored nodes where* $\hat{\Delta} \stackrel{\text{def}}{=} c\ln n$. *Let* U_0 *and* U_1 *be the uncolored nodes before and after a particular call to* OneShotColoring. *Let* $\mathscr{H} \stackrel{\text{def}}{=} \bigcap_{v\in U_0^{\text{hi}}} \mathscr{H}_v$ *be the event that all* U_0^{hi} *nodes are happy.*

(*1*) $\Pr(\overline{\mathscr{H}}) < n^{-c/32+1}$.
(*2*) $\Pr(\deg_{U_1^{\text{hi}}}(v) \le \frac{15}{16}\cdot\deg_{U_0^{\text{hi}}}(v)) > 1 - n^{-c/512} - n^{-c/32+1}$.

PROOF. By Lemma 5.2(2), the definition of $\hat{\Delta} = c\ln n$, and the union bound,

$$\Pr(\overline{\mathscr{H}}) \;<\; |U_0^{\text{hi}}|\cdot\exp\left(-\frac{\hat{\Delta}+1}{32}\right) \;<\; n^{-c/32+1}.$$

In other words, with high probability, every vertex in U_0^{hi} has a 1/8 fraction of its palette available to it.

Turning to Part 2, fix any vertex $v \in U_0^{\text{hi}}$. There are two ways a neighbor of v in U_0^{hi} can fail to be a neighbor in U_1^{hi} after this call to OneShotColoring. It can either be colored (in which case it is not in U_1) or a sufficient number of its neighbors can be colored so that it is no longer in U_1^{hi}. We ignore the second possibility and analyze the number of neighbors of v in U_0^{hi} that are colored. List the nodes of $\Gamma_{U_0^{\text{hi}}}(v)$ in decreasing order of ID as $u_1,\ldots,u_{\deg_{U_0^{\text{hi}}}(v)}$. At step 0 we expose $\text{Color}^\star(u)$ for all $u \notin \Gamma_{U_0^{\text{hi}}}(v)$ and at step i we expose $\text{Color}^\star(u_i)$. Let $\mathbf{Y}_i$ be the information exposed after step i. Whether u_i is successfully colored is a function of $\mathbf{Y}_i$. Moreover, the probability that u_i is colored, given $\mathbf{Y}_{i-1}$, is precisely the fraction of its palette that is still available, according to $\mathbf{Y}_{i-1}$. Let $X_i \in \{0,1\}$ be the indicator variable for the event that u_i is colored and $X = \sum_i X_i$. Unless the unlikely event $\overline{\mathscr{H}}$ occurs,

$$\Pr(X_i = 1 \mid \mathbf{Y}_{i-1}) = \Pr(u_i \text{ is colored} \mid \mathbf{Y}_{i-1}) \ge 1/8,$$

and by Corollary A.5,

$$\Pr(X < \tfrac{1}{16}\deg_{U_0^{\text{hi}}}(v) \mid \mathscr{H}) < \exp\left(-\frac{(\frac{1}{16}\deg_{U_0^{\text{hi}}}(v))^2}{2\deg_{U_0^{\text{hi}}}(v)}\right) = \exp\left(-\frac{1}{512}\deg_{U_0^{\text{hi}}}(v)\right) \le n^{-c/512}.$$

Thus, by a union bound, $\deg_{U_1^{\text{hi}}}(v) \le \frac{15}{16}\deg_{U_0^{\text{hi}}}(v)$ holds for *all* $v \in U_0^{\text{hi}}$, with probability $1 - n^{-c/512+1} - n^{-c/32+1}$. □

Lemma 5.4 implies that after $\log_{16/15}\Delta$ iterations of OneShotColoring, with high probability no node has $\hat{\Delta} = c\ln n$ uncolored neighbors, each having $\hat{\Delta}$ uncolored neighbors. At this point we break the remaining $(\deg+1)$-coloring problem into two subproblems with maximum degree $\hat{\Delta}$. The first subproblem is on the graph induced by U^{hi}, the second is on $U\backslash U^{\text{hi}}$. The maximum degree in U^{hi} is $\hat{\Delta}$, by the observation above, and the maximum degree in $U\backslash U^{\text{hi}}$ is $\hat{\Delta}$ by definition. According to Lemma 5.3, after $O(\log\hat{\Delta}) = O(\log\log n)$ more iterations of OneShotColoring, the size of all uncolored components is less than $s = \hat{\Delta}^2 \cdot c\log_{\hat{\Delta}} n < \hat{\Delta}^3$. Each can be $(\deg+1)$-colored deterministically in $\exp(O(\sqrt{\log s})) = \exp(O(\sqrt{\log\log n}))$ time using the algorithm of Panconesi and Srinivasan [1996]. The failure probability of the $(\deg+1)$-Coloring algorithm (see Figure 10 for pseudocode) is therefore $O(n^{-c/515+2})$.

6. RULING SETS

The $(2,\beta)$ ruling set algorithm of Bisht et al. [2014] works as follows. Given a graph $G = (V, E)$ with maximum degree Δ, the algorithm generates a series of node sets $V(G) = R_0 \supseteq R_1 \supseteq \cdots \supseteq R_{\beta-1} \supseteq R_\beta$ with three properties, namely

(i) R_i dominates R_{i-1}, that is, $\hat{\Gamma}(R_i) \supseteq R_{i-1}$,
(ii) the maximum degree in the graph induced by R_i is $\Delta_i \approx 2^{(\log\Delta)^{1-i\epsilon}}$, and
(iii) R_β is an MIS in the graph induced by $R_{\beta-1}$.

Property (i) implies that for all $v \in V(G)$, $\text{dist}(v, R_\beta) \le \beta$. Together with Property (iii), this implies that R_β is a $(2,\beta)$-ruling set.

Using our MIS algorithm, the time to compute R_β from $R_{\beta-1}$ is $O(\log^2\Delta_{\beta-1} + \exp(O(\sqrt{\log\log n}))) = O(\log^{2(1-(\beta-1)\epsilon)}\Delta + \exp(O(\sqrt{\log\log n}))$, so we want to make ϵ as large as possible. On the other hand, the time to compute R_i from R_{i-1} is $O(\log_{\Delta_i}\Delta_{i-1}) = O(\log^\epsilon\Delta)$. Balancing these costs we get a time bound of $O(\beta\log^{\frac{1}{\beta-1/2}}\Delta + \exp(O(\sqrt{\log\log n})))$ using messages with length $\text{poly}(\Delta_{\beta-1})\log n$. The improvement over Bisht et al.'s [2014] time bound (namely, $O(\log^{\frac{1}{\beta-1}}\Delta + \exp(O(\sqrt{\log\log n}))))$ comes solely from a better MIS algorithm.

The algorithm for computing R_i from R_{i-1} (which satisfies Properties (i) and (ii)) was first described by Kothapalli and Pemmaraju [2012]. For the sake of completeness, we reproduce this sparsification algorithm and its analysis. Refer to Figure 11 for the pseudocode of Sparsify and $(2,\beta)$-RulingSet.

LEMMA 6.1 (KOTHAPALLI AND PEMMARAJU [2012]). *Given $G = (V, E)$ and a threshold f, a subset $U \subseteq V$ can be computed in $O(\log_f\Delta)$ time such that for every $v \in V(G)$, $\text{dist}_G(v, U) \le 1$, and for every $v \in U$, $\deg_U(v) \le 2cf\ln n$, with probability n^{-c+2}.*

PROOF. Consider an execution of Sparsify(G, f). Let U_i be U after the ith iteration of the loop and $V_i \stackrel{\text{def}}{=} V\backslash\hat{\Gamma}(U_i)$. Assume, inductively, that just before the ith iteration the maximum degrees in the graphs induced by V_{i-1} and U_{i-1} are at most Δ/f^{i-1} and $f\cdot 2c\ln n$. These bounds hold trivially when $i = 1$. Each $v \in V_{i-1}$ is included in U_i independently with probability $c\ln nf^i/\Delta$, so the probability that a $v \in V_{i-1}$ with $\deg_{V_{i-1}}(v) > \Delta/f^i$ is not in $\hat{\Gamma}(U_i)$ is less than $(1 - c\ln nf^i/\Delta)^{\Delta/f^i} < n^{-c}$. Furthermore, if $v \in U_i$, then

$$\text{E}[\deg_{U_i}(v)] = \deg_{V_{i-1}}(v)\cdot c\ln nf^i/\Delta \le cf\ln n.$$

By Theorem A.1, the probability that $\deg_{U_i}(v) \ge 2cf\ln n$ is at most $\exp(-fc\ln n/3) < n^{-c}$. Note that since v and its neighborhood are permanently removed from

Sparsify(Graph G, Integer f)

(1) Initialize $U \leftarrow \emptyset$.
(2) For i from 1 to $\log_f \Delta$,
    (a) For each $v \in V(G) \setminus \hat{\Gamma}(U)$, independently, and in parallel:
        Set $U \leftarrow U \cup \{v\}$ with probability $(c \ln n) f^i / \Delta$.
(3) Return U.

$(2, \beta)$-RulingSet(Graph G)

(1) $R_0 \leftarrow V(G)$
(2) For i from 1 to $\beta - 1$
    $R_i \leftarrow \mathsf{Sparsify}(G_{i-1}, f_i)$, where G_{i-1} is the graph induced by R_{i-1}.
(3) $R_\beta \leftarrow \mathsf{MIS}(G_{\beta-1})$
(4) Return(R_β)

Fig. 11. Kothapalli and Pemmaraju's [2012] sparsification and ruling set algorithm.

consideration, it never acquires new neighbors in U, so $\deg_{U_i}(v) = \deg_U(v)$. Thus, with high probability, the induction hypothesis holds for the next iteration. □

THEOREM 6.2. *A $(2, \beta)$-ruling set can be computed in $O(\beta \log^{\frac{1}{\beta-1/2}} \Delta + \exp(O(\sqrt{\log \log n})))$ time with high probability.*

PROOF. The algorithm simply consists of $\beta - 1$ calls to Sparsify followed by a call to MIS. Every node in R_{i-1} is in or adjacent to R_i, for $1 \leq i < \beta$, which implies that $\text{dist}(v, R_\beta) \leq \beta$ for all $v \in V$. Since R_β is an independent set it is also a $(2, \beta)$-ruling set. The time to compute R_β is on the order of

$$\frac{\log \Delta}{\log f_1} + \frac{\log(f_1 \log n)}{\log f_2} + \cdots + \frac{\log(f_{\beta-2} \log n)}{\log f_{\beta-1}} + \log^2(f_{\beta-1} \log n) + \exp(O(\sqrt{\log \log n})).$$

Setting $\log f_i = (\log \Delta)^{1-i(\frac{2}{2\beta-1})}$, the time for each call to Sparsify is $O((\log \Delta)^{\frac{2}{2\beta-1}})$ and the time for the final MIS is $\exp(O(\sqrt{\log \log n}))$ plus

$$\log^2 f_{\beta-1} = (\log \Delta)^{2\left(1-(\beta-1)\frac{2}{2\beta-1}\right)} = (\log \Delta)^{\frac{2}{2\beta-1}}. \quad \square$$

Theorem 6.2 highlights an intriguing open problem. Together with the KMW lower bound, it shows that $(2, 2)$-ruling sets are provably easier to compute than $(2, 1)$-ruling sets, the upper bound for the former being $O(\log^{2/3} \Delta + \exp(O(\sqrt{\log \log n})))$ and the lower bound on the latter being $\Omega(\frac{\log \Delta}{\log \log \Delta})$. Is it possible to obtain any non-trivial lower bound on the complexity of computing $(2, \beta)$-ruling sets for some $\beta > 1$? In order to apply [Kuhn et al. 2004] one would need to invent a reduction from $O(1)$-approximate minimum vertex cover to $(2, \beta)$-ruling sets.

7. BOUNDED ARBORICITY GRAPHS

Recall that a graph has arboricity λ if its edge set is the union of λ forests. In the proofs of Lemma 7.1 and Theorem 7.2, $\deg_{E'}(u)$ is the number of edges incident to u in $E' \subseteq E$ and $\deg_{V'}(u)$ is the number of neighbors of u in $V' \subseteq V$.

LEMMA 7.1. *Let G be a graph of m edges, n nodes, and arboricity λ.*

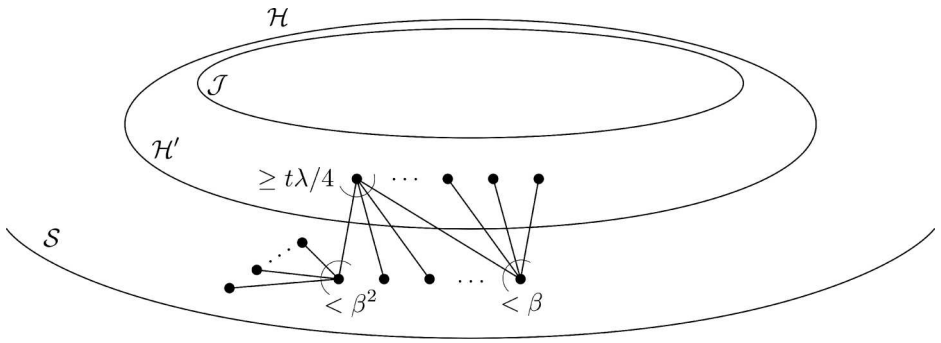

Fig. 12. Good $\mathcal{S}$-nodes have fewer than β neighbors in $\mathcal{H}'$ and fewer than β^2 neighbors in $\mathcal{S}$. Good $\mathcal{H}'$-nodes have at least $t\lambda/4$ good neighbors in $\mathcal{S}$.

(1) $m < \lambda n$.
(2) *The number of nodes with degree at least* $t \geq \lambda + 1$ *is less than* $\lambda n/(t - \lambda)$.
(3) *The number of edges whose endpoints both have degree at least* $t \geq \lambda + 1$ *is less than* $\lambda m/(t - \lambda)$.

Proof. Part 1 follows from the definition of arboricity. For Parts 2 and 3, let $U = \{v \mid \deg_G(v) \geq t\}$ be the set of high-degree nodes. We have that

$$\begin{aligned}\lambda n > m &\geq |\{(u, v) \in E(G) \mid u \in U \text{ or } v \in U \text{ or both}\}| \\ &\geq \sum_{u \in U} (t - \deg_U(u)) + \frac{1}{2} \sum_{u \in U} \deg_U(u) \\ &\geq t \cdot |U| - |E(U)| > (t - \lambda) \cdot |U|.\end{aligned}$$

Thus $|U| < \lambda n/(t - \lambda)$, proving Part 2. Part 3 follows since the number of such edges is less than $\lambda |U| \leq \lambda m/(t - \lambda)$. □

Theorem 7.2. *Let* G *be a graph of arboricity* λ *and maximum degree* Δ*, and let* $t \geq \max\{(5\lambda)^8, (4(c+1)\ln n)^7\}$ *be a parameter. In* $O(\log_t \Delta)$ *time, we can find an independent set* $I \subseteq V(G)$ *(or a matching* $M \subseteq E(G)$*) such that with probability at least* $1 - n^{-c}$*, the maximum degree in the graph induced by* $V \backslash \hat{\Gamma}(I)$ *(or the graph induced by* $V \backslash V(M)$*) is at most* $t\lambda$.

Proof. In $O(\log_t \Delta)$ rounds we commit nodes to I (or edges to M) and remove all incident nodes (or incident edges). Let G be the graph still under consideration before some round and let $\mathcal{H} = \{v \in V \mid \deg_G(v) \geq t\lambda\}$ be the remaining high-degree nodes. Our goal is to reduce the size of $\mathcal{H}$ by roughly a $t^{1/7}$ factor in $O(1)$ rounds. Let $\mathcal{J} = \{v \in \mathcal{H} \mid \deg_{\mathcal{H}}(v) \geq t\lambda/2\}$. It follows that any node $v \in \mathcal{H}' \stackrel{\text{def}}{=} \mathcal{H} \backslash \mathcal{J}$ has $\deg_{V \backslash \mathcal{H}}(v) \geq t\lambda/2$ since at most $t\lambda/2$ of its neighbors can be in $\mathcal{H}$. Let $\tilde{E}$ be any set of edges crossing the cut $(\mathcal{H}, V \backslash \mathcal{H})$ such that for $v \in \mathcal{H}'$, $\deg_{\tilde{E}}(v) = t\lambda/2$. In other words, discard all but $t\lambda/2$ edges incident to each $\mathcal{H}'$ node arbitrarily. Let $\mathcal{S} = \{u \mid v \in \mathcal{H}' \text{ and } (v, u) \in \tilde{E}\}$ be the neighborhood of $\mathcal{H}'$ with respect to $\tilde{E}$. Note that $|\mathcal{S}| \leq t\lambda |\mathcal{H}'|/2$. See Figure 12.

We define bad $\mathcal{S}$-nodes, bad $\tilde{E}$-edges, and bad $\mathcal{H}'$-nodes as follows, where $\beta = t^{1/7}$:

$$\begin{aligned}B_{\mathcal{S}} &= \left\{u \in \mathcal{S} \mid \deg_{\tilde{E}}(u) \geq \beta \text{ or } \deg_{\mathcal{S}}(u) \geq \beta^2 \text{ or both}\right\}, \\ B_{\tilde{E}} &= \left\{(u, v) \in \tilde{E} \;\middle|\; u \in B_{\mathcal{S}}\right\}, \\ \text{and } B_{\mathcal{H}'} &= \left\{v \in \mathcal{H}' \;\middle|\; \deg_{B_{\tilde{E}}}(v) > \lambda t/4\right\}.\end{aligned}$$

Note that nodes can determine in $O(1)$ time whether they are in $B_{\mathcal{S}}$ or $B_{\mathcal{H}'}$ and which incident edges are in $B_{\tilde{E}}$. By Lemma 7.1(3) the number of edges $(u, v) \in B_{\tilde{E}}$ designated bad because $\deg_{\tilde{E}}(u) \geq \beta$ is less than $\lambda|\tilde{E}|/(\beta - \lambda)$. By Lemma 7.1(2) the number of *additional* edges $(u, v) \in B_{\tilde{E}}$ designated bad because $\deg_{\mathcal{S}}(u) \geq \beta^2$ is at most $(\beta - 1)\lambda|\mathcal{S}|/(\beta^2 - \lambda)$ since there are less than $\lambda|\mathcal{S}|/(\beta^2 - \lambda)$ such nodes and each contributes fewer than β edges to $\tilde{E}$. In total we have

$$\begin{aligned}
|B_{\tilde{E}}| &< \frac{\lambda|\tilde{E}|}{\beta - \lambda} + \frac{(\beta - 1)\lambda|\mathcal{S}|}{\beta^2 - \lambda} \\
&\leq \frac{\lambda(t\lambda|\mathcal{H}'|/2)}{\beta - \lambda} + \frac{(\beta - 1)\lambda(t\lambda|\mathcal{H}'|/2)}{\beta^2 - \lambda} && \{|\mathcal{S}| \leq |\tilde{E}| = t\lambda|\mathcal{H}'|/2\} \\
&= \lambda^2 t|\mathcal{H}'|\left(\frac{1}{2(\beta - \lambda)} + \frac{\beta - 1}{2(\beta^2 - \lambda)}\right) \\
&< \frac{\lambda^2 t|\mathcal{H}'|}{\beta - \lambda}.
\end{aligned}$$

A bad $v \in \mathcal{H}'$ is incident to more than $t\lambda/4$ edges in $B_{\tilde{E}}$, so

$$|B_{\mathcal{H}'}| < \frac{|B_{\tilde{E}}|}{t\lambda/4} < \frac{4\lambda|\mathcal{H}'|}{\beta - \lambda}. \tag{4}$$

Our goal now is to select some nodes for the MIS (or edges for the maximal matching) that eliminate all good $\mathcal{H}'$ nodes, with high probability. Each node $u \in \mathcal{S}\backslash B_{\mathcal{S}}$ selects a random real in $(0, 1)$ and joins the MIS if it holds a local maximum. The probability that u joins the MIS is $1/(\deg_{\mathcal{S}\backslash B_{\mathcal{S}}}(u) + 1) \geq 1/\beta^2$, and this event is clearly independent of all $u' \in \mathcal{S}\backslash B_{\mathcal{S}}$ at distance (in $\mathcal{S}\backslash B_{\mathcal{S}}$) at least 3. Note that since the maximum degree in the graph induced by $\mathcal{S}\backslash B_{\mathcal{S}}$ is less than β^2, the neighborhood of any good $v \in \mathcal{H}'\backslash B_{\mathcal{H}'}$ contains a subset of at least $(t\lambda/4)/\beta^4$ nodes, each pair of which is at distance at least 3 with respect to $\mathcal{S}\backslash B_{\mathcal{S}}$. (Such a set could be selected greedily.) Thus, the probability that no neighbor of $v \in \mathcal{H}'\backslash B_{\mathcal{H}'}$ joins the MIS is at most

$$\left(1 - \frac{1}{\beta^2}\right)^{t\lambda/(4\beta^4)} < e^{-t\lambda/(4\beta^6)} = e^{-t^{1/7}\lambda/4} \leq 1/n^{c+1}.$$

Thus, with high probability, all good nodes $\mathcal{H}'\backslash B_{\mathcal{H}'}$ are eliminated. Any remaining high-degree nodes must be in either $\mathcal{J}$ or $B_{\mathcal{H}'}$. By Lemma 7.1 and (4),

$$|\mathcal{J}| + |B_{\mathcal{H}'}| < \frac{\lambda|\mathcal{H}|}{t/2 - \lambda} + \frac{4\lambda|\mathcal{H}'|}{\beta - \lambda} < \frac{5\lambda|\mathcal{H}|}{\beta - \lambda}.$$

Since $\beta = t^{1/7} \geq (5\lambda)^{8/7}$, the number of high-degree nodes is reduced by a $t^{\Omega(1)}$ factor. Thus, after $O(\log_t \Delta)$ time all high-degree nodes have been eliminated with probability $1 - 1/n^c$.

In the case of maximal matching, we want to select a matching that matches all $\mathcal{H}'$ nodes. Each $u \in \mathcal{S}\backslash B_{\mathcal{S}}$ chooses an edge $(u, v) \in \tilde{E}\backslash B_{\tilde{E}}$ uniformly at random and proposes to v that (u, v) be included in the matching. Any $v \in \mathcal{H}'\backslash B_{\mathcal{H}'}$ receiving a proposal accepts one arbitrarily and becomes matched. A good $v \in \mathcal{H}'\backslash B_{\mathcal{H}'}$ has at least $\deg_{\tilde{E}\backslash B_{\tilde{E}}}(v) \geq t\lambda/4$ neighbors $u \in \mathcal{S}\backslash B_{\mathcal{S}}$ with $\deg_{\tilde{E}\backslash B_{\tilde{E}}}(u) < \beta$, so the probability that v receives no proposal (and remains unmatched) is less than $(1 - 1/\beta)^{t\lambda/4} < e^{-t^{6/7}\lambda/4} < o(1/n^{c+1})$. As in the case of MIS, the number of high-degree nodes is reduced by a $t^{\Omega(1)}$ factor in $O(1)$ time. (For the maximal matching problem our proof could be simplified somewhat since edges inside $\mathcal{S}$ play no part in the algorithm and need not be classified as good or bad.) □

7.1. Consequences of Theorem 7.2

Theorems 7.3, 7.4, 7.5, and 7.6 give new bounds on the complexity of maximal matching, MIS, vertex coloring, and ruling sets in terms of λ. Some results are a consequence of Theorem 7.2. Others are obtained by combining the Phase I portion of our algorithms from Sections 3–5 with one of the Barenboim-Elkin [2010, 2011, 2013] algorithms for Phase II.

THEOREM 7.3. *In a graph with maximum degree Δ and arboricity λ, a maximal matching can be computed in time on the order of*

$$\min\left\{\log\lambda + \sqrt{\log n},\ \ \log\Delta + \lambda + \log\log n\right\}$$

for all λ, and in time $O(\log\Delta + \frac{\log\log n}{\delta\log\log\log n})$ when $\lambda = (\log\log n)^{1-\delta}$.

PROOF. The first maximal matching bound is a consequence of Theorem 7.2 and Theorem 4.4. We reduce the maximum degree to $\lambda t \stackrel{\text{def}}{=} \lambda\cdot\max\{2^{\sqrt{\log n}}, (5\lambda)^8\}$ in $O(\log_t n) = O(\sqrt{\log n})$ time and find a maximal matching of the resulting graph in $O(\log(\lambda t) + \log^4\log n) = O(\log\lambda + \sqrt{\log n})$ time. To obtain the second and third bounds we use the same algorithm from Theorem 4.4, but rather than invoke [Hańćkowiak et al. 2001] on each component of $s \le (c\ln n)^9$ nodes, we use the deterministic maximal matching algorithms of Barenboim and Elkin [2010, 2013]. Their algorithms run in $O(\frac{\log s}{\delta\log\log s})$ time on graphs with size s and arboricity $\lambda = \log^{1-\delta} s$ and in time $O(\lambda + \log s)$ in general. □

THEOREM 7.4. *In a graph with maximum degree Δ and arboricity λ, a maximal independent set (MIS) can be computed in time on the order of*

$$\min\left\{\begin{array}{l}\log^2\lambda + \log^{2/3} n,\\ \log^2\Delta + \lambda + \lambda^{\epsilon}\log\log n,\\ \log^2\Delta + \lambda + (\log\log n)^{1+\epsilon},\\ \log^2\Delta + \lambda^{1+\epsilon} + \log\lambda\log\log n\end{array}\right\}$$

for all λ and any constant $\epsilon > 0$. When $\lambda = (\log\log n)^{1/2-\delta}$, an MIS can be computed in $O(\log^2\Delta + \frac{\log\log n}{\delta\log\log\log n})$ time.

PROOF. The first bound is a consequence of Theorem 7.2 and Theorem 3.5. We can reduce the maximum degree to $\lambda t \stackrel{\text{def}}{=} \lambda\cdot\max\{2^{(\log n)^{1/3}}, (5\lambda)^8\}$ in $O(\log_t n) = O(\log^{2/3} n)$ time, then find an MIS in the resulting graph in $O(\log^2(\lambda t) + \exp(O(\sqrt{\log\log n}))) = O(\log^2\lambda + \log^{2/3} n)$ time.

To obtain the remaining bounds we execute IndependentSet on the input graph, which, with high probability, returns an independent set I such that the components induced by $B \stackrel{\text{def}}{=} V(G)\backslash\hat{\Gamma}(I)$ have size at most $s = \Delta^4\log_\Delta n$. On each component we invoke one of the deterministic coloring algorithms of Barenboim and Elkin [2010, 2011, 2013] for small arboricity graphs, then construct an MIS in time linear in the number of color classes. For any fixed $\epsilon > 0$, a $\lambda^{1+\epsilon}$-coloring can be computed in $O(\log\lambda\log s)$ time, which gives an MIS algorithm running in time

$$\begin{aligned} &O\big(\log^2\Delta + \lambda^{1+\epsilon} + \log\lambda\log(\Delta^4\log n)\big)\\ &\quad = O\big(\log^2\Delta + \lambda^{1+\epsilon} + \log\lambda\log\log n\big),\end{aligned}$$

since $\lambda \leq \Delta$. Alternatively, we could use a slower $O(\lambda)$-coloring algorithm running in time $O(\min\{\lambda^\epsilon \log s, \lambda^\epsilon + (\log s)^{1+\epsilon}\})$,[20] leading to an MIS algorithm running in time

$$O\big(\log^2 \Delta + \lambda + \min\{\lambda^\epsilon \log(\Delta^4 \log n),\ (\log(\Delta^4 \log n))^{1+\epsilon}\}\big)$$
$$= O\big(\log^2 \Delta + \lambda + \min\{\lambda^\epsilon \log\log n,\ (\log\log n)^{1+\epsilon}\}\big). \quad \square$$

THEOREM 7.5. *Fix a constant $\epsilon > 0$. A graph of maximum degree Δ and arboricity λ can, with high probability, be $(\Delta + \lambda^{1+\epsilon})$-colored in $O(\log \Delta + \log \lambda \log\log n)$ time or $(\Delta + O(\lambda))$-colored in $O(\log \Delta + \min\{\lambda^\epsilon \log\log n,\ \lambda^\epsilon + (\log\log n)^{1+\epsilon}\})$ time. Furthermore, a $(\deg+1)$-coloring can, with high probability, be computed in time on the order of*

$$\min\left\{\begin{array}{l} \log \Delta + \lambda + \lambda^\epsilon \log\log n, \\ \log \Delta + \lambda + (\log\log n)^{1+\epsilon}, \\ \log \Delta + \lambda^{1+\epsilon} + \log \lambda \log\log n \end{array}\right\}.$$

PROOF. Following the algorithm from Section 5, we first execute $O(\log \Delta)$ iterations of OneShotColoring then decompose the problem into two subproblems on a graph with maximum degree $\hat{\Delta} \stackrel{\text{def}}{=} \Theta(\log n)$. On each subproblem we perform another $O(\log \hat{\Delta})$ iterations of OneShotColoring, after which the subgraph induced by uncolored nodes consists, with high probability, of components with size at most $s = \hat{\Delta}^2 \log_{\hat{\Delta}} n = o(\log^3 n)$. At this point we apply one of the deterministic Barenboim and Elkin [2011] coloring algorithms to each such component using a fresh palette of p previously unused colors, say $\{-1, \ldots, -p\}$. We can find a p-coloring with $p = \lambda^{1+\epsilon}$ in $O(\log \lambda \log s) = O(\log \lambda \log\log n)$ time or with $p = O(\lambda)$ in $O(\min\{\lambda^\epsilon \log s, \lambda^\epsilon + (\log s)^{1+\epsilon}\}) = O(\min\{\lambda^\epsilon \log\log n, \lambda^\epsilon + (\log\log n)^{1+\epsilon}\})$ time. Every $v \in V(G)$ has been assigned a color $\text{Color}(v) \in \{1, \ldots, \deg(v)+1\} \cup \{-1, \ldots, -p\}$. To obtain a $(\deg+1)$-coloring we examine each color $\kappa \in \{-1, \ldots, -p\}$ in turn, letting every node v with $\text{Color}(v) = \kappa$ recolor itself using an available color from $\{1, \ldots, \deg(v)+1\}$. $\square$

THEOREM 7.6 ([BARENBOIM AND ELKIN 2010] + [AWERBUCH ET AL. 1989]). *A $(2, O(\log \lambda + \sqrt{\log n}))$-ruling set can be computed deterministically in $O(\log \lambda + \sqrt{\log n})$ time.*

PROOF. Begin by computing a decomposition of the edge set into $\lambda \cdot 2^{\sqrt{\log n}}$ oriented forests, in $O(\sqrt{\log n})$ time [Barenboim and Elkin 2010, Section 3]. Given this decomposition, compute an $O(\lambda^2 \cdot 2^{2\sqrt{\log n}})$-coloring, in $O(\log^* n)$ time [Barenboim and Elkin 2010, Section 5.1.2]. Finally, using this coloring, compute a $(2, O(\log \lambda + \sqrt{\log n}))$-ruling set in $O(\log \lambda + \sqrt{\log n})$ time [Awerbuch et al. 1989]. $\square$

7.2. Maximal Matching in Trees

Our maximal matching algorithm from Theorem 7.3 runs in $O(\sqrt{\log n})$ time for every arboricity λ from 1 (trees) to $2^{O(\sqrt{\log n})}$. We argue that this bound is nearly optimal even for $\lambda = 1$ by appealing to the KMW lower bound [Kuhn et al. 2004]. In Kuhn et al. [2004] it is shown that there exist constants $c' > c$ such that any (possibly randomized) algorithm for computing an approximate minimum vertex cover (MVC) in graphs with girth at least $c' \cdot \sqrt{\frac{\log n}{\log\log n}}$ either (i) runs in $c\sqrt{\frac{\log n}{\log\log n}}$ time, or (ii) has approximation ratio $\omega(1)$. We review below a well known reduction from 2-approximate MVC to maximal

[20]The leading constant in the palette size is exponential in $1/\epsilon$.

matching, which implies an $\Omega(\sqrt{\frac{\log n}{\log\log n}})$ lower bound for maximal matching algorithms that succeed with high probability. The graphs used in the KMW bound have arboricity $2^{O(\sqrt{\log n \log\log n})}$, so it does not directly imply an $\Omega(\sqrt{\frac{\log n}{\log\log n}})$ lower bound on trees.

THEOREM 7.7. *For some absolute constant $c > 0$, no algorithm can, with probability $1 - n^{-2}$, compute a maximal matching on a tree in $c\sqrt{\frac{\log n}{\log\log n}}$ time, nor in $c\frac{\log \Delta}{\log\log \Delta} + o(\sqrt{\frac{\log n}{\log\log n}})$ time for every Δ.*

PROOF. We first recount the lower bound for maximal matching on general graphs. Suppose, for the purpose of obtaining a contradiction, that there exists a maximal matching algorithm running in time $c\sqrt{\frac{\log n}{\log\log n}}$ on the KWM graph that fails with probability at most $1/n$. To obtain an approximate MVC algorithm, run the maximal matching algorithm for $c\sqrt{\frac{\log n}{\log\log n}}$ time. Any matched node joins the approximate MVC, as well as any node that detects a local violation, namely a node incident to another unmatched node. As the MVC is at least the size of any maximal matching, the expected approximation ratio of this algorithm is at most $2 \cdot \Pr(\text{no failure occurs}) + n \cdot \Pr(\text{some failure occurs}) \leq 2 + n \cdot \frac{1}{n} = 3$, a contradiction. Hence there is no algorithm that runs for $c \cdot \sqrt{\frac{\log n}{\log\log n}}$ time in graphs with girth at least $c' \cdot \sqrt{\frac{\log n}{\log\log n}}$ that computes a maximal matching with probability at least $1 - 1/n$.

We use an indistinguishability argument to show that the $\Omega(\sqrt{\frac{\log n}{\log\log n}})$ lower bound also holds for trees, and therefore any class of graphs that includes trees. Observe that to show a lower bound for a randomized algorithm, it is enough to prove the same lower bound under the assumption that the identities of graph nodes were selected independently and uniformly at random, from, say, $[1, n^{10}]$. Suppose there is, in fact, an algorithm that given a tree with a random (in the above sense) assignment of identities, constructs a maximal matching within $c \cdot \sqrt{\frac{\log n}{\log\log n}}$ time with success probability at least $1 - n^{-2}$. Run this algorithm for $c \cdot \sqrt{\frac{\log n}{\log\log n}}$ time on the KMW graph G with girth $c' \cdot \sqrt{\frac{\log n}{\log\log n}}$, assuming random assignment of identities in G. Due to the girth bound, the view of every node in G is identical to its view in a tree, and thus from its perspective a correct maximal matching must be computed with probability at least $1 - n^{-2}$. By a union bound, a correct maximal matching for the entire graph G will be computed with probability at least $1 - n^{-1}$, a contradiction.

The KMW graph has maximum degree $\Delta = 2^{\Theta(\sqrt{\log n \log\log n})}$ and girth $\Theta(\frac{\log \Delta}{\log\log \Delta})$. All the KMW-based $\Omega(\sqrt{\frac{\log n}{\log\log n}})$ lower bounds can be scaled down to $\Omega(\frac{\log \Delta}{\log\log \Delta})$ lower bounds (for any $\Delta < 2^{O(\sqrt{\log n \log\log n})}$) simply by applying the lower bound argument to the union of numerous identical KMW graphs. □

Remark 7.8. Theorem 7.7 posited the existence of a maximal matching algorithm for trees whose *global* probability of failure is n^{-2}. When we run this algorithm on the KMW graph we can no longer use n^{-2} as the global failure probability. It may be that, when run in an actual tree, nodes within distance $c\sqrt{\frac{\log n}{\log\log n}}$ of a leaf node fail with probability zero: all the failure probability is concentrated at the small set of nodes

TreeIndependentSet(Graph G)

(1) Initialize sets $I, B \subset V(G)$:
$I \leftarrow \emptyset$ {an independent set}
$B \leftarrow \emptyset$ {a set of 'bad' nodes}
Throughout, let $V_{IB} \stackrel{\text{def}}{=} V(G) \setminus (\hat{\Gamma}(I) \cup B)$ be the nodes still under consideration. Let G_{IB} be the graph induced by V_{IB} and let Γ_{IB} and $\deg_{IB}$ be the neighborhood and degree functions w.r.t. G_{IB}.
(2) For each scale k from 1 to $\log(\frac{\Delta}{48 \ln \Delta})$,
(a) Execute $\log_{5/4}(33 \ln \Delta)$ iterations of steps i and ii.
i. Each node $v \in V_{IB}$ chooses a priority $r(v)$.

$$r(v) \leftarrow \begin{cases} 0, & \text{if } \left|\{w \in \Gamma_{IB}(v) \mid \deg_{IB}(w) > \Delta/2^k\}\right| \\ & \quad > \Delta(8 \ln \Delta + 1)/2^{k+1}, \\ \text{a random real in } (0,1), & \text{otherwise.} \end{cases}$$

ii. $I \leftarrow I \cup \{v \in V_{IB} \mid r(v) > \max\{r(w) \mid w \in \Gamma_{IB}(v)\}$
(Add nodes to the independent set.)
(b) $B \leftarrow B \cup \left\{v \in V_{IB} \;\middle|\; |\{w \in \Gamma_{IB}(v) \mid \deg_{IB}(w) > \Delta/2^k\}| > \Delta/2^{k+2}\right\}$.
(Mark nodes that violate Invariant 8.1 as bad.)
(3) Return (I, B).

Fig. 13. An algorithm for computing an almost maximal independent set in a tree.

that cannot "see" the leaves. In the KMW graph *all* nodes think they are in this small set. We must assume, pessimistically, that failure occurs at *every* node in the KMW graph with probability n^{-2}.

Remark 7.9. Theorem 7.7, strangely, does not imply any lower bound for the MIS problem on trees, even though MIS appears to be just as hard as maximal matching on any class of graphs. The $\Omega(\sqrt{\frac{\log n}{\log \log n}})$ lower bound for MIS [Kuhn et al. 2004] is obtained by considering the *line graph* of the KWM graph, which has girth 3, not $\Theta(\sqrt{\frac{\log n}{\log \log n}})$. Thus, our indistinguishability argument does not apply.

8. MIS IN TREES AND HIGH GIRTH GRAPHS

One of the MIS algorithms of Luby [1986] works as follows. In each round each remaining node v chooses a random real $r(v) \in (0,1)$ and includes itself in the MIS if $r(v)$ is greater than $\max_{w \in \Gamma(v)} r(w)$, thereby *eliminating* v and its neighborhood from further consideration.[21] Observe that the probability that v joins the MIS in a round is $1/(\deg(v)+1)$, irrespective of the degrees of its neighbors.

We would like to say that degrees decay geometrically, that is, after $O(k)$ iterations of Luby's algorithm the maximum degree is $\Delta/2^k$, with high probability. Invariant 8.1 is not quite this strong but just as useful, algorithmically. It states that after $O(k \log \log \Delta)$ iterations, no node has $\Delta/2^{k+2}$ neighbors with degree at least $\Delta/2^k$, provided that $\Delta/2^k$ is not too small.

INVARIANT 8.1. *At the end of scale k, for all* $v \in V_{IB}$,

$$\left|\left\{w \in \Gamma_{IB}(v) \mid \deg_{IB}(w) > \Delta/2^k\right\}\right| \;\le\; \max\{\Delta/2^{k+2}, 12 \ln \Delta\}.$$

[21] In practice it suffices to generate only the $O(\log n)$ most significant bits. That is, nodes choose an integer from, say, $\{1, \ldots, n^{10}\}$ uniformly at random.

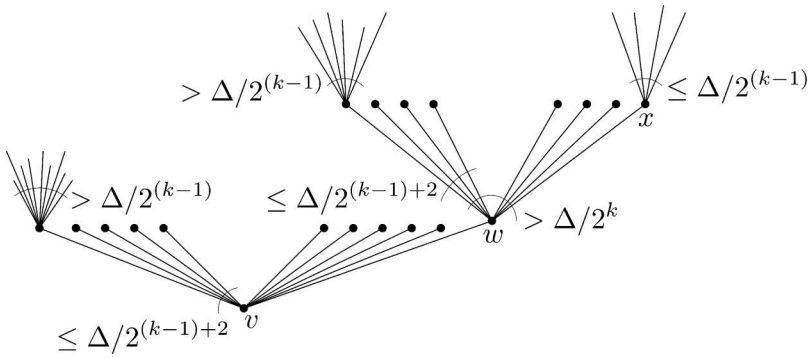

Fig. 14. The kth scale of TreeIndependentSet, from the perspective of v. Only v's neighbors with degree greater than $\Delta/2^k$ are shown; w is one such neighbor. They are partitioned into those with degrees in $(\Delta/2^{k-1}, \infty)$ and $(\Delta/2^k, \Delta/2^{k-1}]$. The first category numbers at most $\Delta/2^{(k-1)+2}$; the second category is unbounded. At most $\Delta/2^{(k-1)+2}$ of w's neighbors have degree more than $\Delta/2^{(k-1)}$, leaving at least half with degree at most $\Delta/2^{(k-1)}$. If any neighbor x joins the MIS, then w will be eliminated.

Randomness plays no role in Invariant 8.1: It holds with probability 1. Any node that violates the invariant is marked *bad* (placed in B) and temporarily excluded from consideration. As we will soon prove, the probability a node is marked bad is $1/\text{poly}(\Delta)$. We will only make use of Invariant 8.1 when $\Delta/2^{k+2}$ is, in fact, greater than $12\ln\Delta$, so the $12\ln\Delta$ term will not be mentioned until we need to have a lower bound on $\Delta/2^{k+2}$.

LEMMA 8.2. *In one iteration of scale k, a node w with* $\deg_{IB}(w) > \Delta/2^k$ *is eliminated (appears in* $\hat{\Gamma}(I)$*) with probability at least* $(1-o(1))(1-e^{-1/4}) > 0.22$*. Moreover, this probability holds even if we condition on arbitrary behavior at a single neighbor of w.*

PROOF. By Invariant 8.1, $|\{x \in \Gamma_{IB}(w) \mid \deg_{IB}(x) > \Delta/2^{k-1}\}| \leq \Delta/2^{k+1}$. Let M be the neighbors of w with degree at most $\Delta/2^{k-1}$, so $|M| \geq \deg_{IB}(w) - \Delta/2^{k+1} > \Delta/2^{k+1}$. Refer to the portion of Figure 14 depicting w and its neighborhood. The probability that w is eliminated is minimized when M-nodes attain their maximum degree $\Delta/2^{k-1}$, so in the calculations below we shall assume this is the case. Let $x^\star \in M$ be the first neighbor for which $r(x^\star) > \max\{r(y) \mid y \in \Gamma_{IB}(x^\star)\backslash\{w\}\}$. The probability $x^\star$ exists is at least

$$\Pr(x^\star \text{ exists}) = 1 - \prod_{x\in M}\left(1 - \frac{1}{\deg_{IB}(x)}\right) \geq 1 - \left(1 - \frac{1}{\Delta/2^{k-1}}\right)^{\Delta/2^{k+1}} > 1 - e^{-1/4}.$$

Since, in the most extreme case, $\deg_{IB}(x) = \Delta/2^{k-1}$, $\Pr(x^\star \text{ joins } I \mid x^\star \text{ exists}) = \Pr(r(x^\star) > r(w) \mid x^\star \text{ exists}) \geq 1 - \frac{1}{\Delta/2^{k-1}+1}$. The probability that w is eliminated is therefore at least $(1 - \frac{1}{\Delta/2^{k-1}+1})(1-e^{-1/4}) > (1 - \frac{1}{96\ln\Delta})(1-e^{1/4}) > 0.22 > 1/5$. Moreover, this probability is perturbed by a negligible $(1 - \Theta(1/\Delta/2^k)) = (1 - o(1))$ factor if one conditions on arbitrary behavior by a single neighbor of w. □

LEMMA 8.3. *In any scale, a node v is included in B with probability at most* $1/\Delta^2$*, independent of the behavior of any one neighbor of v.*

PROOF. Fix a node v and let $N = \{w \in \Gamma_{IB}(v) \mid \deg_{IB}(w) > \Delta/2^k\}$ at the beginning of scale k. See Figure 14. In the figure, only N-node neighbors of v are depicted. If $|N| \leq \Delta/2^{k+2}$, then the invariant is already satisfied at v, so assume otherwise. There are two cases, depending on the size of N.

Case 1: $|N|$ is large. We argue that if $|N| > \Delta(8\ln\Delta+1)/2^{k+1}$, then v is eliminated with probability at least $1-\Delta^{-2}$ in a single iteration and can therefore be bad with probability at most Δ^{-2}. According to the algorithm, $r(v) = 0$, so v has no chance to hold a locally maximum r-value. Since, by Invariant 8.1, v has at least $|N| - \Delta/2^{k+1} > 8\Delta\ln\Delta/2^{k+1}$ neighbors with degree at most $\Delta/2^{k-1}$, the probability that v is not eliminated is at most the probability that no N-node joins I. This occurs with probability at most

$$\left(1-\frac{1}{\Delta/2^{k-1}}\right)^{|N|-\Delta/2^{k+1}} \le \exp\left(\frac{8\Delta\ln\Delta}{2^{k+1}}\cdot\frac{2^{k-1}}{\Delta}\right) = \Delta^{-2}.$$

Case 2: $|N|$ is small. In this case $|N| \le \Delta(8\ln\Delta+1)/2^{k+1}$, that is, $|N|$ is within a $O(\log\Delta)$ factor of satisfying Invariant 8.1. By Lemma 8.2 each N-node, so long as it has degree at least $\Delta/2^k$, is eliminated with probability at least $1/5$. Moreover, these events are independent, conditioned on some arbitrary behavior at v, the only common neighbor of N-nodes. Thus, each node will survive $\log_{5/4}(4(8\ln\Delta+1)) = O(\log\log\Delta)$ iterations with probability $1/[4(8\ln\Delta+1)]$. The expected number of surviving N-nodes is therefore less than $\Delta/2^{k+3}$. By a Chernoff bound (Theorem A.1), the probability that this quantity exceeds twice its expectation, thereby putting v into B, is $\exp(-(\Delta/2^{k+3})/3)$, which is at most Δ^{-2} since $\Delta/2^k \ge 48\ln\Delta$. □

LEMMA 8.4. *All connected components in the subgraph induced by B have at most $t = c\log_\Delta n$ nodes with probability $1-n^{-c/2}$.*

PROOF. There are less than 4^t topologically distinct rooted t-node trees and at most $n\Delta^{t-1}$ ways to embed such a tree, say, T, in the graph. There are $(\log\Delta)^t$ schedules for when (in which scale) the T-nodes were added to B. Since the probability that each T-node becomes bad in a scale is at most Δ^{-2}, *independent of the behavior of its parent in T*, the probability that B contains a t-node tree is at most

$$\begin{aligned}&4^t\cdot n\Delta^{t-1}\cdot(\log\Delta)^t\cdot\Delta^{-2t}\\ &\quad< (4\log\Delta)^{c\log_\Delta n}\cdot n^{c+1}\cdot n^{-2c}\\ &\quad< n^{-c/2}.\end{aligned}$$

The last inequality holds when Δ is at least some sufficiently large constant. □

8.1. The TreeMIS Algorithm

Let us review the situation. TreeIndependentSet(G) takes $O(\log\Delta\log\log\Delta)$ time and returns a pair (I, B) satisfying two properties, the second of which holds with probability $1-n^{-c/2}$.

—Although the degree of nodes in the graph induced by $V_{IB} = V(G)\setminus(\hat{\Gamma}(I)\cup B)$ is not bounded, no node has $12\ln\Delta$ neighbors with degree at least $48\ln\Delta$.
—The graph induced by B is composed of small connected components, each with size at most $t \le c\log_\Delta n$.

The TreeMIS algorithm (Figure 15) starts by obtaining a pair (I, B) satisfying these properties and then extends I to a maximal independent set in three stages. It partitions V_{IB} into low- and high-degree sets V_{lo} and V_{hi}. By definition, the graph induced by V_{lo} has maximum degree $48\ln\Delta$ and by the first property above the graph induced by V_{hi} has maximum degree $12\ln\Delta$. An MIS I_{lo} for V_{lo} can be computed deterministically in $O(\log\Delta+\log^* n)$ time [Barenboim et al. 2014], that is, in time linear in

TreeMIS(Graph G)

Phase I:

(1) $(I, B) \leftarrow$ TreeIndependentSet(G)

Phase II:

(2) Partition $V_{IB} = V(G)\backslash(\hat{\Gamma}(I) \cup B)$ into low- and high-degree sets.
$V_{\mathrm{lo}} \leftarrow \{v \in V_{IB} \mid \deg_{IB}(v) \leq 48 \ln \Delta\}$
$V_{\mathrm{hi}} \leftarrow \{v \in V_{IB} \mid \deg_{IB}(v) > 48 \ln \Delta\}$

(3) Compute maximal independent sets on V_{lo} and V_{hi}.
$I_{\mathrm{lo}} \leftarrow$ an MIS of the graph induced by V_{lo}.
$I_{\mathrm{hi}} \leftarrow$ an MIS of the graph induced by $V_{\mathrm{hi}}\backslash\hat{\Gamma}(I_{\mathrm{lo}})$.

Let $\mathscr{C}$ be the set of connected components with size at most $c \ln n$ in the graph induced by $B\backslash\hat{\Gamma}(I \cup I_{\mathrm{lo}} \cup I_{\mathrm{hi}})$.

(4) For each $C \in \mathscr{C}$,
$I_C \leftarrow$ an MIS of C

(5) Return $I \cup I_{\mathrm{lo}} \cup I_{\mathrm{hi}} \cup \bigcup_{C \in \mathscr{C}} I_C$

Fig. 15. A maximal independent set algorithm for unoriented trees.

the degree.[22] An MIS I_{hi} for $V_{\mathrm{hi}}\backslash\hat{\Gamma}(I_{\mathrm{lo}})$ can then be computed, also in $O(\log \Delta + \log^* n)$ time. At this point only B-nodes may not be adjacent to some node in $I \cup I_{\mathrm{lo}} \cup I_{\mathrm{hi}}$. For each component C in the graph induced by $B\backslash\hat{\Gamma}(I \cup I_{\mathrm{lo}} \cup I_{\mathrm{hi}})$ we compute an MIS I_C in $O(\log t / \log\log t) = O(\frac{\log\log n}{\log\log\log n})$ time using the Barenboim-Elkin [2010] algorithm.

In total, the running time of TreeMIS is $O(\log \Delta \log\log \Delta + \frac{\log\log n}{\log\log\log n})$ and its failure probability is less than $n^{-c/2}$.

THEOREM 8.5. *In an unoriented tree with maximum degree Δ, a maximal independent set can, with high probability, be computed in time on the order of*

$$\min\left\{\log \Delta \log\log \Delta + \frac{\log\log n}{\log\log\log n},\quad \sqrt{\log n \log\log n}\right\}.$$

PROOF. The $O(\log \Delta \log\log \Delta + \frac{\log\log n}{\log\log\log n})$ bound was shown above. If $\Delta > \hat{\Delta} \stackrel{\text{def}}{=} 2^{\sqrt{\log n / \log\log n}}$, use Theorem 7.2 to reduce the maximum degree to $\hat{\Delta}$ in $O(\log_{\hat{\Delta}} n) = O(\sqrt{\log n \log\log n})$ time and then compute an MIS in $O(\log \hat{\Delta} \log\log \hat{\Delta} + \frac{\log\log n}{\log\log\log n}) = O(\sqrt{\log n \log\log n})$ time. □

8.2. MIS on High Girth Graphs

Our analysis of TreeIndependentSet and TreeMIS requires that certain events are independent and this independence is guaranteed if the radius-3 neighborhood around

[22] Since we are already spending $O(\log \Delta \log\log \Delta)$ time in TreeIndependentSet, we can afford to use a simpler MIS algorithm [Kuhn and Wattenhofer 2006] running in $O(\log \Delta \log\log \Delta + \log^* n)$ time.

each node looks like a tree. In other words, parts of the analysis do not distinguish between actual trees and graphs with girth greater than 6.[23]

In order to make the analysis work on graphs with girth greater than 6, we need to make a number of small modifications to TreeIndependentSet and TreeMIS.

—We substitute $\log n$ for $\log \Delta$ in Invariant 8.1. It now states that at the end of scale k, for all $v \in V_{IB}$,

$$\left|\left\{w \in \Gamma_{IB}(v) \mid \deg_{IB}(w) > \Delta/2^k\right\}\right| \leq \max\{\Delta/2^{k+2}, c \ln n\}$$

for some sufficiently large c.

—We change the critical threshold in TreeIndependentSet from $\Delta(8 \ln \Delta + 1)/2^{k+1}$ to $\Delta(8 \ln n + 1)/2^{k+1}$ and change the number of iterations per scale from $O(\log \log \Delta)$ to $O(\log \log n)$.

—Lemmas 8.3 and 8.4 now claim that after $\log(\Delta/(4c \ln n))$ scales,

—In G_{IB}, each node has no more than $c \ln n$ neighbors with degree greater than $4c \ln n$.

—With high probability, namely $1 - n^{-\Omega(c)}$, all nodes satisfy Invariant 8.1. That is, $B = \emptyset$.

—Provided that $B = \emptyset$, in order to extend I to an MIS we only need to find an MIS I_{lo} of V_{lo} and I_{hi} of $V_{\text{lo}} \setminus \hat{\Gamma}(I_{\text{hi}})$. Since the graphs induced by V_{lo} and V_{hi} have maximum degree $4c \ln n$, this takes $\exp(O(\sqrt{\log \log n}))$ time using the MIS algorithm of Section 3.

THEOREM 8.6. *In a graph of girth greater than 6 (in which no cycle has length at most 6), an MIS can be computed in* $O(\log \Delta \log \log n + \exp(O(\sqrt{\log \log n})))$ *time with high probability.*

9. CONCLUSIONS

In this work, we have advanced the state of the art in randomized symmetry breaking using a powerful new set of algorithmic tools. Our MIS and maximal matching algorithms represent the first significant improvements (for general graphs) to the classic algorithms of the 1980s [Luby 1986; Alon et al. 1986; Israeli and Itai 1986]. Our maximal matching algorithms (for general graphs, trees, and low-arboricity graphs) are among a small group of nearly optimal symmetry-breaking algorithms for a wide range of parameters. However, we feel the most important contribution of this work is the identification of the *union bound barrier*.

All of our algorithms reduce an n-node instance of the problem to a disjoint set of $\text{poly}(\log n)$-node components,[24] which is the threshold beyond which known randomized symmetry-breaking strategies fail to achieve a $(\log n)^{o(1)}$ running time. Even if the probability of failure on one component is small, by the union bound the probability of failure on *some* component is nearly certain. Unless, of course, the probability of failure is *zero*, meaning we forswear random bits altogether and opt to use the best available deterministic algorithm. We conjecture that the union bound barrier is "real" and, in particular, that $(\log n)^{o(1)}$-time randomized algorithms must revert to a deterministic algorithm. If true, then this means that the randomized complexities of many symmetry-breaking problems are tethered to their deterministic counterparts. For example, we could not hope to get rid of the $2^{O(\sqrt{\log \log n})}$ terms in our MIS and coloring algorithms

[23]The analysis could probably be made to work for graphs with girth 6 or 5, but it does *not* work for graphs of girth 4. If the graph is formed by grafting together a sequence of bipartite $\Delta/2 \times \Delta/2$ cliques, then the probability a node becomes bad after one scale of TreeIndependentSet is not $1/\text{poly}(\Delta)$ but $\exp(-\Omega((\log \log \Delta)^2/\log \log \log \Delta))$.

[24]Or, in the case of the MIS algorithm, $(\text{poly}(\Delta) \log n)$-size components.

without *first* improving the $2^{O(\sqrt{\log n})}$-time Panconesi-Srinivasan [1996] algorithms. We also could not hope to achieve an (optimal) $O(\min\{\frac{\log \Delta}{\log\log \Delta} + \log^* n, \sqrt{\frac{\log n}{\log\log n}}\})$-time algorithm for MIS or maximal matching unless that algorithm were already deterministic.

After the initial publication of this work, there have been several breakthroughs in randomized symmetry breaking wthat have built on the two-phase approach developed here. Ghaffari [2016] gave an MIS algorithm running in $O(\log \Delta + 2^{O(\sqrt{\log\log n})})$ time, and Harris, Schneider, and Su [2016] developed a $(\Delta+1)$-coloring algorithm running in $O(\sqrt{\log \Delta} + 2^{O(\sqrt{\log\log n})})$ time. In terms of Δ, both improve on our algorithms quadratically. However, the dependence on n is unchanged, which highlights the difficulty of circumventing the union bound barrier.

APPENDIX

A. CONCENTRATION INEQUALITIES

See Dubhashi and Panconesi [2009] for proofs of these and related concentration bounds.

THEOREM A.1 (CHERNOFF). *Let X be the sum of n independent, identically distributed 0/1 random variables. For any $\delta \in (0, 1)$,*

$$\Pr(X < (1-\delta)\,\mathrm{E}[X]) < \exp(-\delta^2\,\mathrm{E}[X]/2)$$
$$\textit{and}\quad \Pr(X > (1+\delta)\,\mathrm{E}[X]) < \exp(-\delta^2\,\mathrm{E}[X]/3).$$

THEOREM A.2 (NEGATIVE CORRELATION). *Let $X = X_1 + \cdots + X_n$ be the sum of n random variables, where the $\{X_i\}$ are independent or negatively correlated. Then for any $t > 0$:*

$$\Pr(X \geq \mathrm{E}[X] + t), \Pr(X \leq \mathrm{E}[X] - t) \;\leq\; \exp\left(-\frac{2t^2}{\sum_i (a_i' - a_i)^2}\right),$$

where $a_i \leq X_i \leq a_i'$.

THEOREM A.3 (JANSON). *For $X = X_1 + \cdots + X_n$ the sum of n random variables and $t > 0$,*

$$\Pr(X \geq \mathrm{E}[X] + t), \Pr(X \leq \mathrm{E}[X] - t) \;\leq\; \exp\left(-\frac{2t^2}{\chi \cdot \sum_i (a_i' - a_i)^2}\right),$$

where $a_i \leq X_i \leq a_i'$ and χ is the fractional chromatic number of the dependency graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$. By definition $\mathscr{V} = \{X_1, \ldots, X_n\}$, and the edge set $\mathscr{E}$ satisfies the property that X_i is independent of $\mathscr{V} \backslash \Gamma(X_i)$, for all i.

THEOREM A.4 (AZUMA-HOEFFDING). *A sequence $Y_0, \ldots, Y_n$ is a martingale with respect to $X_0, \ldots, X_n$ if Y_i is a function of $X_0, \ldots, X_i$ and $\mathrm{E}[Y_i \mid X_0, \ldots, X_{i-1}] = Y_{i-1}$. For such a martingale with bounded differences $a_i \leq Y_i - Y_{i-1} \leq a_i'$,*

$$\Pr(Y_n > Y_0 + t),\ \Pr(Y_n < Y_0 - t) \leq \exp\left(-\frac{t^2}{2\sum_i (a_i' - a_i)^2}\right).$$

COROLLARY A.5. *Let $Z = Z_1 + \cdots + Z_n$ be the sum of n random variables and $X_0, \ldots, X_n$ be a sequence, where Z_i is uniquely determined by $X_0, \ldots, X_i$, $\mu_i = \mathrm{E}[Z_i \mid X_0, \ldots, X_{i-1}]$, $\mu = \sum_i \mu_i$, and $a_i \leq Z_i \leq a_i'$. Then*

$$\Pr(Z > \mu + t),\ \Pr(Z < \mu - t) \leq \exp\left(-\frac{t^2}{2\sum_i (a_i' - a_i)^2}\right).$$

PROOF. Define the martingale $Y_0, \ldots, Y_n$ w.r.t. $X_0, \ldots, X_n$ by $Y_0 = 0$ and $Y_i = Y_{i-1} + Z_i - \mu_i$, and then apply Theorem A.4. Note $Y_n - Y_0 = Z - \mu$ and the range of $Y_i - Y_{i-1}$ still has size $a_i' - a_i$. □

Note that Corollary A.5 says that one random variable, Z, is well concentrated around another random variable, namely μ.

ACKNOWLEDGMENTS

We thank the two anonymous reviewers for their thoughtful comments, as well as James Hegeman, Sriram Pemmaraju, and Mohsen Ghaffari for pointing out flaws in earlier proofs of Lemmas 5.4 and 3.4.

REFERENCES

N. Alon. 1991. A parallel algorithmic version of the local lemma. *Rand. Struct. Algor.* 2, 4 (1991), 367–378.

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

A. Amir, O. Kapah, T. Kopelowitz, M. Naor, and E. Porat. 2014. The family holiday gathering problem or fair and periodic scheduling of independent sets. *CoRR* abs/1408.2279 (2014).

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

R. Bar-Yehuda, K. Censor-Hillel, and G. Schwartzman. 2016. A distributed $(2+\epsilon)$-approximation for vertex cover in $O(\log \Delta/\epsilon \log\log \Delta)$ rounds. *CoRR* abs/1602.03713 (2016).

L. Barenboim. 2015. Deterministic $(\Delta+1)$-coloring in sublinear (in Δ) time in static, dynamic and faulty networks. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing (PODC)*. 345–354. DOI:http://dx.doi.org/10.1145/2767386.2767410

L. Barenboim and M. Elkin. 2010. Sublogarithmic distributed MIS algorithm for sparse graphs using Nash-Williams decomposition. *Distrib. Comput.* 22, 5–6 (2010), 363–379.

L. Barenboim and M. Elkin. 2011. Deterministic distributed vertex coloring in polylogarithmic time. *J. ACM* 58, 5 (2011), 23.

L. Barenboim and M. Elkin. 2013. *Distributed Graph Coloring: Fundamentals and Recent Developments*. Morgan & Claypool Publishers, San Francisco, CA.

L. Barenboim, M. Elkin, and F. Kuhn. 2014. Distributed $(\Delta+1)$-coloring in linear (in Δ) time. *SIAM J. Comput.* 43, 1 (2014), 72–95.

J. Beck. 1991. An algorithmic approach to the Lovász local lemma. I. *Rand. Struct. Algor.* 2, 4 (1991), 343–366.

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

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

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

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

D. P. Dubhashi and D. Ranjan. 1998. Balls and bins: A study in negative dependence. *J. Rand. Struct. Algs.* 13, 2 (1998), 99–124.

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

B. Gfeller and E. Vicari. 2007. A randomized distributed algorithm for the maximal independent set problem in growth-bounded graphs. In *Proceedings 26th Annual ACM Symposium on Principles of Distributed Computing (PODC)*. 53–60.

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

M. Hańćkowiak, M. Karoński, and A. Panconesi. 2001. On the distributed complexity of computing maximal matchings. *SIAM J. Discr. Math.* 15, 1 (2001), 41–57 (electronic).

D. Harris, J. Schneider, and H.-H. Su. 2016. Distributed $(\Delta + 1)$-coloring in sublogarithmic rounds. In *Proceedings 48th ACM Symposium on Theory of Computing (STOC)*.

A. Israeli and A. Itai. 1986. A fast and simple randomized parallel algorithm for maximal matching. *Info. Proc. Lett.* 22, 2 (1986), 77–80.

Ö. Johansson. 1999. Simple distributed $\Delta + 1$-coloring of graphs. *Info. Proc. Lett.* 70, 5 (1999), 229–232.

A. Korman, J.-S. Sereni, and L. Viennot. 2013. Toward more localized local algorithms: Removing assumptions concerning global knowledge. *Distrib. Comput.* 26, 5–6 (2013), 289–308.

K. Kothapalli and S. V. Pemmaraju. 2011. Distributed graph coloring in a few rounds. In *Proceedings 30th Annual ACM Symposium on Principles of Distributed Computing (PODC)*. 31–40.

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

K. Kothapalli, C. Scheideler, M. Onus, and C. Schindelhauer. 2006. Distributed coloring in $\tilde{O}(\sqrt{\log n})$ bit rounds. In *Proceedings 20th International Parallel and Distributed Processing Symposium (IPDPS)*.

F. Kuhn, T. Moscibroda, and R. Wattenhofer. 2004. What cannot be computed locally! In *Proceedings 23rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*. 300–309.

F. Kuhn, T. Moscibroda, and R. Wattenhofer. 2010. Local computation: Lower and upper bounds. *CoRR* abs/1011.5470 (2010).

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

C. Lenzen and R. Wattenhofer. 2011. MIS on trees. In *Proceedings 30th Annual ACM Symposium on Principles of Distributed Computing (PODC)*. 41–48.

N. Linial. 1992. Locality in distributed graph algorithms. *SIAM J. Comput.* 21, 1 (1992), 193–201.

N. Linial and M. E. Saks. 1993. Low diameter graph decompositions. *Combinatorica* 13, 4 (1993), 441–454.

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

R. A. Moser and G. Tardos. 2010. A constructive proof of the general Lovász local lemma. *J. ACM* 57, 2 (2010). DOI: http://dx.doi.org/10.1145/1667053.1667060

C. St. J. A. Nash-Williams. 1964. Decomposition of finite graphs into forests. *J. London Math. Soc.* 39 (1964), 12.

A. Panconesi and R. Rizzi. 2001. Some simple distributed algorithms for sparse networks. *Distrib. Comput.* 14, 2 (2001), 97–100.

A. Panconesi and A. Srinivasan. 1996. On the complexity of distributed network decomposition. *J. Algor.* 20, 2 (1996), 356–374.

D. Peleg. 2000. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, Philadelphia, PA.

S. Pettie and H.-H. Su. 2015. Distributed algorithms for coloring triangle-free graphs. *Inform. Comput.* 243 (2015), 263–280.

R. Rubinfeld, G. Tamir, S. Vardi, and N. Xie. 2011. Fast local computation algorithms. In *Proceedings of the 1st Symposium on Innovations in Computer Science (ICS)*. 223–238. See also *CoRR* abs/1104.1377.

J. Schneider, M. Elkin, and R. Wattenhofer. 2013. Symmetry breaking depending on the chromatic number or the neighborhood growth. *Theor. Comput. Sci.* 509 (2013), 40–50.

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

J. Schneider and R. Wattenhofer. 2010b. An optimal maximal independent set algorithm for bounded-independence graphs. *Distrib. Comput.* 22, 5–6 (2010), 349–361.

H.-H. Su. 2015. *Algorithms for Fundamental Problems in Computer Networks*. Ph.D. Dissertation. University of Michigan.

M. Szegedy and S. Vishwanathan. 1993. Locality based graph coloring. In *Proceedings 25th ACM Symposium on Theory of Computing (STOC)*. 201–207.

Received February 2015; revised February 2016; accepted March 2016