RANDOMIZED DISTRIBUTED EDGE COLORING VIA AN EXTENSION OF THE CHERNOFF-HOEFFDING BOUNDS*

ALESSANDRO PANCONESI† AND ARAVIND SRINIVASAN‡

Abstract. Certain types of routing, scheduling and resource allocation problems in a distributed setting can be modeled as edge coloring problems. We present fast and simple randomized algorithms for edge coloring a graph, in the synchronous distributed point–to–point model of computation. Our algorithms compute an edge–coloring of a graph G with n nodes and maximum degree Δ with at most $1.6\Delta + O(\log^{1+\delta} n)$ colors with high probability (arbitrarily close to 1), for any fixed $\delta > 0$; they run in polylogarithmic time. The upper bound on the number of colors improves upon the $(2\Delta - 1)$–coloring achievable by a simple reduction to vertex–coloring.

To analyze the performance of our algorithms, we introduce new techniques for proving upper bounds on the tail probabilities of certain random variables. The *Chernoff–Hoeffding bounds* are fundamental tools that are used very frequently in estimating tail probabilities. However, they assume stochastic independence among certain random variables, which may not always hold. Our results extend the Chernoff–Hoeffding bounds to certain types of random variables which are not stochastically independent. We believe that these results are of independent interest, and merit further study.

Key words. edge coloring, distributed algorithms, parallel algorithms, probabilistic algorithms, Chernoff-Hoeffding bounds, stochastic dependence, λ-correlation, correlation inequalities, large deviations

AMS subject classifications. 05C85, 60C05, 60F10, 60G50, 68Q22, 68Q25, 68R10

1. Introduction.

An important limitation for a distributed network without global memory is *locality of computation*: sending messages to faraway nodes being expensive, communication should only take place between nearby nodes. Models of parallel computation like the PRAM abstract this problem of locality away by assuming the existence of a global shared memory with fast concurrent access. We are interested in studying how fast individual processors can compute their portion of the output in a message-passing distributed system, with such "local" information alone. The model we consider is the synchronous distributed point-to-point model, in which the processors are arranged as the vertices of an n-vertex graph $G = (V, E)$, and where all communication is via the edges of G alone. In this model, we study the edge coloring problem, a basic combinatorial problem with many applications to distributed computing. Edge colorings can be used to model certain types of jobshop scheduling, packet routing, and resource allocation problems in a distributed setting. For example, the problem of scheduling I/O operations in a certain parallel architecture can be modeled as follows (Jain, Somalwar, Werth & Browne [9]). We are given a set of processes $\mathcal{P}$ and a set of resources $\mathcal{R}$ such that each process $p \in \mathcal{P}$ needs a subset $f(p) \subseteq \mathcal{R}$ of the resources where: (i) each process p needs every resource in $f(p)$ for a unit of time each, and (ii) p can use the resources in $f(p)$ in any order. From this, we can construct a bipartite graph $G_{\mathcal{P},\mathcal{R}} = (\mathcal{P}, \mathcal{R}, E_{\mathcal{P},\mathcal{R}})$ where

* A preliminary version of this work appears as "Fast Randomized Algorithms for Distributed Edge Coloring", in the *ACM Symposium on Principles of Distributed Computing*, pages 251–262, August 1992. This research was conducted while the authors were with the Department of Computer Science, Cornell University, Ithaca, NY 14853, USA, and was supported in part by NSF PYI award CCR–89–96272 with matching support from UPS and Sun Microsystems.

† Informatik, Freie Universität Berlin, Takustr. 9, 14195 Berlin, Deutschland. E-mail: `ale@inf.fu-berlin.de`. Supported by an Alexander von Humboldt research fellowship.

‡ Dept. of Information Systems & Computer Science, National University of Singapore, Singapore 119260, Republic of Singapore. E-mail: `aravind@iscs.nus.sg`

$E_{\mathcal{P},\mathcal{R}} = \{(p,r)|\ p \in \mathcal{P} \wedge r \in f(p)\}$. An edge coloring of $G_{\mathcal{P},\mathcal{R}}$ with c colors yields a schedule for the processes to use the resources within c time units. Optimal colorings correspond to optimal schedules.

Edge coloring can also be used in distributed models in situations where broadcasts are infeasible or undesirable: an edge coloring of the network results in a schedule for each processor to communicate with at most one neighbor at every step; at time step i, processors communicate via the edges colored i only. Using a "small" number of colors reduces the wastage of time in this schedule.

Related Work. Note that Δ colors are necessary to edge color a graph with maximum degree Δ. Vizing showed that it is always possible to edge color a graph with $\Delta + 1$ colors and gave a polynomial time algorithm to compute such a coloring ([22]; see, for instance, Bollobás [3]). Efforts to parallelize Vizing's theorem have failed so far. The best known algorithm is an RNC algorithm of Karloff & Shmoys using $\Delta + O(\Delta^{1/2+\epsilon})$ colors for any fixed $\epsilon > 0$; this algorithm has been derandomized in NC (Berger & Rompel [2], Motwani, Naor & Naor [15]). In the distributed model, the best edge coloring algorithm known prior to this work was to apply a vertex coloring algorithm to the line graph $L(G)$ of G. There are fast (polylogarithmic) randomized vertex coloring algorithms that use $(\Delta + 1)$ and Δ colors, which translate to $(2\Delta - 1)$– and $(2\Delta - 2)$– edge coloring algorithms respectively (Luby [13], Panconesi & Srinivasan [17]). In the deterministic case, there are no known $(2\Delta - 1)$–edge coloring algorithms of polylogarithmic running time; the best running time is $2^{O(\sqrt{\log n})}$, which is asymptotically better than any fixed root of n but which grows faster than any polylogarithmic function of n [17]. Interestingly, distributed Δ–edge coloring for bipartite graphs requires $\Omega(diameter(G))$ time even with randomization [17], whereas this can be done in $O(\log n)$ time deterministically in the PRAM model [11].

Our contributions. In this paper, we present fast and simple randomized algorithms to edge color G with at most $1.6\Delta + O(\log^{1+\delta} n)$ colors with high probability for any fixed $\delta > 0$, where Δ is the maximum degree of the vertices of G. At the heart of our analysis is an extension of the *Chernoff–Hoeffding bounds*, which are key tools in bounding the tail probabilities of the sums of independent random variables (Chernoff [4], Hoeffding [8], Raghavan [18]).

Our edge coloring algorithm is based on a very simple randomized algorithm to color bipartite graphs, which can be explained in a few lines. Given a bipartite graph $G = (A, B, E)$ with maximum degree Δ, each vertex in B picks distinct colors from $\{1, 2, \ldots, \Delta\}$ at random for its edges without replacement, *i.e.*, edges incident to the same vertex in B get different colors. Then, each vertex $v \in A$ checks, for each color α, if more than one of its incident edges has color α and if so, chooses one of them at random as the *winner*, and all the other edges of color α which are incident to v are decolored. The key claim is that for every vertex, the number of decolored edges incident to it is at most $\Delta(1+\epsilon)/e$ with high probability for any fixed $\epsilon > 0$, where e is the base of natural logarithms. Assuming that this holds, we can repeat the above iteration with a set of $\Delta(1+\epsilon)/e$ *fresh* colors, and so on. In spite of its simplicity the algorithm requires an interesting probabilistic analysis; this is based upon an extension of the Chernoff-Hoeffding bounds to a certain case of dependence among the random variables, which we call λ–correlation. We believe that these results have the potential for further applications, and merit further study.

A preliminary version of this work appeared in [16], where we showed how to edge color using at most $1.6\Delta + O(\log^{2+\delta} n)$ colors. By presenting a tighter analysis of the tail probabilities, we improve this to $1.6\Delta + O(\log^{1+\delta} n)$ colors here.

In §2, we define the basic notation used, and §3 describes our main analysis tool– the extension of the Chernoff-Hoeffding bounds. Section 4 presents our algorithm, whose performance is analyzed in §5. Some extensions and applications of this work are described in §6.

2. Notation. A *message-passing distributed network* is an undirected graph $G = (V, E)$ where vertices, or nodes, correspond to processors and edges correspond to bi-directional communication links. Each node has its unique ID. There is no shared memory and processors can communicate only by sending messages through the network. The network is *synchronous*, *i.e.*, computation takes places in a sequence of *rounds*; in each round, each node does any amount of local computation, sends messages to its neighbors in the graph, and reads messages sent to it by its neighbors. The time complexity of a distributed algorithm, or *protocol*, is given by the number of rounds needed to compute a given function.

Though each node has no knowledge about the topology of the entire network, it knows upper bounds n and Δ on the total number of nodes and maximum degree of the network, respectively. We also sketch an alternative algorithm if Δ and n are unknown, but the constant factor in the $O(\log^{1+\delta} n)$ term in the number of colors used, is higher in this case.

Notice that in this model the cost of sending a message from one vertex to another is proportional to the length of a shortest path between the two vertices. Hence, if we want a protocol to run for t rounds, then each vertex can communicate only with vertices at distance at most t from it. This is not so in the PRAM model, where the shared memory allows any two processors to communicate in one unit of time. Lower bounds for distributed computation imposed by this locality, have been presented by Linial [12]. Also as mentioned before, distributed Δ–edge coloring for bipartite graphs requires $\Omega(diameter(G))$ time, even with randomization [17]. In particular, we cannot two-color the *vertices* of a bipartite graph G distributively in $o(diameter(G))$ time.

Given an undirected graph $G = (V, E)$ we denote by Δ its maximum degree, *i.e.*, the maximum number of edges incident with any node; by d_u we denote the degree of vertex u, by $N(u)$ we denote the set of neighbors of u, and by $\delta(u)$ we denote the set of edges incident with u.

Given a positive integer n, $[n]$ denotes the set $\{1, 2, \ldots, n\}$. The permanent of a (possibly non–square) matrix M with c columns and r rows where $c \le r$, is defined as the natural extension of the permanent of square matrices. Let $\mathcal{P} = \{\pi \mid \pi : [c] \to [r],\ \pi \text{ is one–to–one}\}$. Then,

$$perm(M) \doteq \sum_{\pi \in \mathcal{P}} \prod_{i=1}^{n} M_{\pi(i),i}.$$

An event $\mathcal{A}$ is said to happen *with high probability* (w.h.p.) if $Pr(\mathcal{A}) \ge 1 - 1/f(n)$ for some superpolynomial function $f(n)$ (*i.e.*, $n^c = o(f(n))$ for all fixed $c > 0$).

In our algorithms we will use Luby's vertex coloring algorithm [13] as a subroutine. When applied to the line graph of G, the algorithm computes a $(2\Delta(G) - 1)$–edge coloring of G, with its running time being $O(\log n)$ *w.h.p.* The algorithm only needs local information– a vertex only needs to know its own degree. Another property of the algorithm that we will use is that vertices can be added dynamically to the graph, each vertex u with its own palette of $\deg(u) + 1$ colors, and the algorithm still works as claimed (the running time is $O(\log n)$ from the time of the last insertion). Other algorithms could be used as well but we refer to this algorithm of Luby for conciseness.

3. The Chernoff-Hoeffding Bounds Extension. In this section we introduce our extension of the Chernoff-Hoeffding bounds which are important tools used in estimating the tail probabilities of random variables. Given n *independent* random variables $X_1, X_2, \ldots, X_n$, these bounds are used in deriving an upper bound on the upper tail probability $Pr(X \geq (1+\epsilon)\mu)$, where $X = \sum_{i=1}^n X_i$, $\mu = E[X]$, and $\epsilon > 0$. We extend these bounds to a certain case of dependency among the X_i's, which we call λ–correlation.

Let us review Chernoff's approach to upper bound the upper tail probability of a random variable X, when X is the sum of *independent* binary random variables $X_1, X_2, \ldots, X_n$ [4] (this idea is apparently originally due to S. N. Bernstein [8]). The idea is to use Markov's inequality on the random variable e^{tX} for an arbitrary $t > 0$, and minimize with respect to t. That is, to use the fact that

$$\begin{aligned} Pr(X > (1+\epsilon)\mu) &= Pr(e^{tX} > e^{t(1+\epsilon)\mu}) \\ &\leq \frac{E[e^{tX}]}{e^{t(1+\epsilon)\mu}}, \end{aligned}$$

and minimize the last ratio for $t > 0$. This is achieved by finding a good upper bound for the numerator $E[e^{tX}]$ by using the fact that X is the sum of *independent* random variables. It is standard (see, *e.g.*, Raghavan [18]) to use this for showing that in this case if $X_i \in \{0,1\}$ for each i, then

$$\min_{t>0} \frac{E[e^{tX}]}{e^{t(1+\epsilon)\mu}} \leq F(\mu,\epsilon) \doteq \left[\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right]^{\mu}. \tag{1}$$

Hoeffding [8] considers a more general case where X is the sum of n independent and *bounded* random variables $X_i \in [a_i, b_i]$, and uses the above approach to show that if $E[X] = \mu$, then for $\epsilon > 0$,

$$\min_{t>0} \frac{E[e^{tX}]}{e^{t(1+\epsilon)\mu}} \leq G(\mu,\epsilon,\vec{a},\vec{b}) \doteq \exp\left[-\frac{2\,\mu^2\epsilon^2}{\sum_{i\in[n]}(b_i-a_i)^2}\right]. \tag{2}$$

The bounds (1) and (2) will be used in our proofs. Henceforth, we refer to these bounds of Chernoff and Hoeffding as the CH-bounds. If ϵ is a fixed positive quantity no greater than 1 (which will be true in all our applications), then $F(\mu,\epsilon) \leq e^{-\epsilon^2\mu/3}$. Hence, if $\mu = \Omega(\log^{1+\delta} n)$, then $F(\mu,\epsilon)$ is the inverse of a superpolynomial function of n, for any fixed $\delta > 0$ (similar considerations apply to $G(\mu,\epsilon,\vec{a},\vec{b})$). This fact makes the CH-bounds a powerful tool for deriving strong performance guarantees for randomized algorithms and will be used repeatedly in this paper.

3.1. The General Case. We now introduce λ-correlation and prove the general extension of the CH-bounds. Section 3.2 will then discuss an important special case of the results of this section.

Our proof is based on the observation that if we can upper bound each term $E[X^k]$ of the Maclaurin expansion of $E[e^{tX}]$ by $\lambda\, E[\hat{X}^k]$ where $\hat{X}$ is the sum of independent random variables, and if $E[e^{t\hat{X}}] \leq B$, then $E[e^{tX}] \leq \lambda B$. We start with the following definition.

DEFINITION 3.1. Let $X_1, X_2, \ldots, X_n$ be bounded random variables such that $X_i \in [a_i, b_i]$ and let $X = \sum_{i\in[n]} X_i$. The X_i's are λ-*correlated* if there exists a collection of independent twin random variables $\hat{X}_i \in [a_i, b_i]$ such that:

(i) $E[X] \leq E[\hat{X}]$, where $\hat{X} = \sum_{i\in[n]} \hat{X}_i$, and

(ii) for all $I \subseteq [n]$ and positive integers $s_i,\ \ i \in I$,

$$E[\prod_{i\in I} X_i^{s_i}] \leq \lambda \prod_{i\in I} E[\hat{X}_i^{s_i}].$$

The main theorem now is

THEOREM 3.2. *Let X be the sum of λ–correlated random variables $X_1, X_2, \ldots, X_n$ where $X_i \in [a_i, b_i]$, and let $\hat{X}$ be the sum of the n twin variables $\hat{X}_i$. Then,*

$$Pr(X > (1+\epsilon)\ E[\hat{X}]) \leq \lambda\ G(E[\hat{X}], \epsilon, \vec{a}, \vec{b}).$$

Proof. Let $\mu = E[\hat{X}]$. As in the classical proof, we start by introducing a positive parameter t and by applying Markov's inequality to the variable e^{tX}:

$$\begin{aligned} Pr(X > (1+\epsilon)\mu) &= Pr(e^{tX} > e^{t(1+\epsilon)\mu}) \\ &\leq \frac{E[e^{tX}]}{e^{t(1+\epsilon)\mu}}. \end{aligned}$$

By the hypotheses of the boundedness of X, we may apply linearity of expectation to an infinite series:

$$E[e^{tX}] = E\left[\sum_{k=0}^{\infty} \frac{t^k X^k}{k!}\right] = \sum_{k=0}^{\infty} \frac{t^k E[X^k]}{k!}.$$

Now, $X^k = (\sum_{i=1}^n X_i)^k$ is a sum of terms of the form $\prod_{i\in I} X_i^{s_i}$ for some $I \subseteq [n]$ and positive integers s_i. Hence by linearity of expectation and the assumption of λ–correlation,

$$E[X^k] \leq \lambda E[\hat{X}^k].$$

Thus,

$$E[e^{tX}] \leq \lambda \sum_{k=0}^{\infty} \frac{t^k E[\hat{X}^k]}{k!} = \lambda\ E[e^{t\hat{X}}].$$

By the already discussed result of Hoeffding [8], when $\hat{X}$ is the sum of n independent bounded random variables $\hat{X}_i \in [a_i, b_i]$,

$$\min_{t>0} \frac{E[e^{t\hat{X}}]}{e^{t(1+\epsilon)\mu}} \leq G(\mu, \epsilon, \vec{a}, \vec{b}). \quad \square$$

In this paper we will use the special case of Theorem 3.2 where $X_i \in [0,1]$, $i \in [n]$. In this case, $F(\mu, \epsilon)$ is also an upper bound for the upper tail of X.

COROLLARY 3.3. *Let X be the sum of n λ–correlated random variables $X_i \in [0,1]$. Then,*

$$Pr(X > (1+\epsilon)E[\hat{X}]) \le \lambda\, F(\mu,\epsilon).$$

Proof. Let $E[\hat{X}] = \mu$. When $\hat{X}$ is the sum of n independent random variables $\hat{X}_i \in [0,1]$, Hoeffding (cf. Theorem 1 of [8]) shows that if $t \in (0, 1-\mu/n)$ and $\epsilon = nt/\mu$, then

$$Pr\left(\frac{\hat{X}-\mu}{n} \ge t\right) \le \min_{s>0} \frac{E[e^{s\hat{X}}]}{e^{s(1+\epsilon)\mu}} \le \left(\frac{\mu}{\mu+nt}\right)^{\mu+nt} \left(1 + \frac{n\,t}{n-\mu-nt}\right)^{n-\mu-nt}.$$

By the proof of Theorem 3.2, we see that $E[e^{sX}] \le \lambda E[e^{s\hat{X}}]$, for any $s > 0$. Thus by applying the standard approximation $1 + x \le e^x$ for $x = nt/(n-\mu-nt)$, we get

$$Pr(X \ge (1+\epsilon)\mu) \le \lambda \left(\frac{e^{\epsilon}}{(1+\epsilon)^{(1+\epsilon)}}\right)^{\mu} = \lambda F(\mu,\epsilon). \qquad \square$$

3.2. Binary Random Variables. An important special case of Definition 3.1 is when $X_i \in \{0,1\}$. In this case, the condition on the expectations simplifies considerably to become

$$Pr\left(\bigwedge_{i\in I} X_i = 1\right) \le \lambda \prod_{i\in I} Pr(\hat{X}_i = 1)$$

for all $I \subseteq [n]$[1]. This special case is interesting in its own right and hence we record it as

THEOREM 3.4. *Let $X_1, X_2, \ldots, X_n$ be given 0-1 random variables with $X = \sum_i X_i$. If there exist* independent *random variables $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n$ with $\hat{X} = \sum_i \hat{X}_i$ and $E[X] \le E[\hat{X}]$ such that for all $I \subseteq [n]$,*

$$Pr\left(\bigwedge_{i\in I} X_i = 1\right) \le \lambda \prod_{i\in I} Pr(\hat{X}_i = 1),$$

then

$$Pr(X > (1+\epsilon)E[\hat{X}]) \le \lambda\, F(E[\hat{X}],\epsilon).$$

The statement follows immediately from Corollary 3.3. Notice that λ-correlation follows if the X_i's are "negatively correlated" in the following sense:

$$Pr\left(\bigwedge_{i\in I} X_i = 1\right) \le \prod_{i\in I} Pr(X_i = 1)$$

for all $I \subseteq [n]$. We now present an example where precisely this kind of situation arises, and which will also be used later in this paper.

[1] This was defined as "self-weakening with parameter λ" in [16].

Suppose we have n balls that are thrown uniformly and independently at random into n bins, and that we want to estimate the number B of empty (missed) bins. Let B_i be an indicator random variable that is 1 if bin i is empty and 0 otherwise. For any $i \in [n]$,

$$Pr(B_i = 1) = \left(1 - \frac{1}{n}\right)^n \leq \frac{1}{e}.$$

It follows that $E[B] = E[\sum_i B_i] \leq n/e$. The B_i's are 1–correlated. To see this, consider a subset $J \subseteq [n]$ and any $i \in [n] - J$. Then,

$$\begin{aligned} Pr(B_i = 1 \mid \bigwedge_{j \in J} B_j = 1) &= \left(1 - \frac{1}{n - |J|}\right)^n \\ &\leq \left(1 - \frac{1}{n}\right)^n \\ &= Pr(B_i = 1). \end{aligned}$$

By straightforward induction, this implies that for all $I \subseteq [n]$,

$$Pr\left(\bigwedge_{i \in I} B_i = 1\right) \leq \prod_{i \in I} Pr(B_i = 1).$$

Thus the B_i are 1-correlated, where we may take the twin variables $\hat{B}_i$ to be i.i.d. 0-1 random variables with $Pr(B_i = 1) = 1/e$ for each i. Hence, not only is $E[B] \leq n/e$ but, by Theorem 3.4,

$$Pr(B > (1 + \epsilon)n/e) \leq F(n/e, \epsilon).$$

Remarks. The above fact can also be given a completely different (and simple) proof via the theory of martingales using Azuma's inequality (see, for example, Alon, Spencer & Erdős [1] or McDiarmid [14].) We have presented this proof to illustrate our techniques. Also, S. Jain has proved the following lemma [19]: *Let $a_1, a_2, \ldots, a_n$ be n random trials (not necessarily independent) such that the probability that trial a_i 'succeeds' is bounded above by p regardless of the outcomes of the other trials. Then if X is the random variable that represents the number of 'successes' in these n trials, and Y is a binomial variable with parameters (n, p), then:* $Pr[X \geq k] \leq Pr[Y \geq k]$, $0 \leq k \leq n$.

The assumptions of Jain's lemma are strictly stronger than those of 1–correlation. For instance, in the balls and bins example,

$$Pr(B_n = 1 | \bigwedge_{i \in [n-1]} B_1 = 0) = \frac{n-1}{n+1},$$

which, for $n \geq 3$, is greater than $Pr(B_n = 1)$ ($\approx 1/e$). Note, however, that our result does not subsume Jain's lemma, since his result upper bounds $Pr(X \geq k)$ by the true binomial upper bound, while we only upper bound it by the CH bound.

4. The Edge Coloring Algorithm. We now present our randomized distributed edge coloring algorithm. The algorithm uses an idea of Karloff & Shmoys to reduce the problem of edge coloring general graphs to that of edge coloring a special class of bipartite graphs [10]. The Karloff & Shmoys scheme uses the fact that bipartite graphs can be edge colored optimally in the PRAM model of computation, which is provably impossible in our distributed model [17]. Instead, we use a distributed subroutine that computes a "good" coloring. Also different is the handling of the "leftover" graphs at the end of the recursion which we color by making use of Luby's vertex coloring algorithm.

The input to the algorithm is a distributed network $G = (V, E)$, and some fixed $\epsilon, \delta > 0$. In addition, each node knows upper bounds n and Δ on $|V|$ and the maximum degree $\Delta(G)$ of G, respectively. This information is not necessary but yields better multiplicative constants. The case where Δ and n are unknown is sketched towards the end of §5.1.

The algorithm is recursive and computes an edge coloring of G using at most $1.6\Delta + O(\log^{1+\delta} n)$ colors and runs in $O(\log n)$ time; both these bounds hold *w.h.p.* Let THRESHOLD $= \log^{1+\delta} n$, and $new(\Delta) = \Delta/2 + \sqrt{\Delta \log^{1+\delta/2} n}$; the algorithm is as follows.

> **If** $\Delta \leq 16$ THRESHOLD **then** edge-color G with $2\Delta - 1$ colors using Luby's algorithm and exit[2]; **else** execute the following.
>
> 1. Compute a random partition of $V(G)$ into black and white vertices (all vertices flip a fair coin independently and in parallel). Let $G[B]$ be the subgraph induced by the black vertices, $G[W]$ be the subgraph induced by the white vertices, and $G[B, W]$ be the bipartite subgraph formed by the edges having endpoints of different colors.
> 2. Edge color $G[B, W]$ using our bipartite edge-coloring algorithm described below, with the parameters $new(\Delta)$, ϵ and δ.
> 3. Set $\Delta := new(\Delta)$ and recurse on $G[B]$ and $G[W]$ using the *same* set of fresh new colors on both graphs, with the same parameters ϵ and δ as before. (REMARK: though the bipartite algorithm modifies its first parameter $new(\Delta)$ in the course of its execution, we assume that it is passed "by value", *i.e.*, that the value of $new(\Delta)$ referred to here and in step 2 above is the same.)

REMARK: $new(\Delta)$ is meant to be an upper bound on $G[B], G[W]$ and $G[B, W]$. It is easily seen via the standard CH bounds that it is indeed so *w.h.p.*, if $\Delta \geq$ THRESHOLD and hence if $\Delta \geq 16$ THRESHOLD [10].

We now present our main algorithm–a distributed algorithm to color the bipartite graphs produced above.

4.1. Distributed Edge Coloring of Bipartite Graphs. Given a bipartite graph $G = (A, B, E)$ we assume that each vertex knows whether it belongs to A or B. This is an important assumption because such information cannot be computed fast distributively as mentioned in §2, but it is verified for the bipartite graphs generated by the Karloff & Shmoys scheme. From now on, we will refer to vertices in A as the

[2] When $\Delta = O(polylog(n))$ we can compute a $2\Delta - 1$ coloring *deterministically* in $O(polylog(n))$ time using an algorithm based on the idea of removing maximal matchings. We prefer to use Luby's algorithm here for conciseness.

top vertices and to the vertices in B as the *bottom* vertices.

Given parameters Δ_C, ϵ and δ such that Δ_C is an upper bound on the degree of G, the algorithm takes $O(\log n)$ time and colors the bipartite graph G with at most $1.6\Delta_C + O(\log^{1+\delta} n)$ colors *w.h.p.*, as long as $\delta > 0$ is any constant (ϵ is used in the algorithm). During any iteration of the algorithm, Δ_C is meant to be an upper bound on the degree of the current graph; we will prove later that *this holds w.h.p. as long as* $\Delta_C \geq \log^{1+\delta} n = \text{THRESHOLD}$. From the remark in §4, we can assume that this is valid when the bipartite algorithm is called first. As we will briefly discuss in §5.1 this is not needed, but gives better constants. The algorithm is as follows:

1. PART I: While $\Delta_C \geq \text{THRESHOLD}$ do:

Let G_C be the current graph. Pick a set χ of Δ_C *fresh new* colors.

(i) *(Random proposal of bottom vertices)* In parallel and independently of the other vertices in B, each vertex $v \in B$ assigns a temporary color to each edge in $\delta(v)$ with uniform probability *without replacement, i.e.* edge e_1 is assigned color $\alpha \in \chi$ with probability $1/\Delta_C$, e_2 is assigned $\beta \in \chi - \{\alpha\}$ with probability $1/(\Delta_C - 1)$ and so on.

(ii) *(Lottery of top vertices)* (REMARK: The coloring so far is consistent around any vertex $v \in B$ but can be inconsistent around a vertex $u \in A$.) For each $u \in A$, let $C_u(\alpha)$ be the set of edges in $\delta(u)$ with temporary color α. Each vertex $u \in A$ selects a *winner* uniformly at random in $C_u(\alpha)$, for each nonempty $C_u(\alpha)$. All other edges, the *losers*, are decolored and assigned $\perp$.

(iii) Set $\Delta_C := \Delta_C(1+\epsilon)/e$. $G_\perp$, the subgraph of G_C induced by the losers (*i.e.*, by the $\perp$-edges), becomes the new current graph.

2. PART II: Let G_r be the remaining graph. Edge color G_r with at most $2\Delta(G_r) - 1$ colors by executing Luby's vertex coloring algorithm on the line graph of G_r.

Since we use new colors in each iteration, it is clear that when the algorithm terminates, G has been edge-colored legally. It is also apparent that the algorithm works based on local information alone. We now turn to placing bounds on the number of colors used, and the running time.

5. Analysis of the Algorithm. Since the analysis is fairly involved, we first present a higher-level description of it.

5.1. The Basic Structure of the Analysis. Our key claim will be that in every iteration of part (I) of the bipartite edge coloring algorithm, the maximum degree of the graph shrinks by a factor of at least $(1+\epsilon)/e$ *w.h.p.*, as long as $\Delta_C \geq \text{THRESHOLD}$. That is,

$$\Delta(G_\perp) \leq (1+\epsilon)\,\frac{\Delta(G_C)}{e}$$

with high probability, for any fixed $\epsilon > 0$. The condition $\Delta_C \geq \text{THRESHOLD}$ ensures that the failure probability given by the extension of the CH-bounds is the inverse of a superpolynomial function. Hence, w.h.p., no vertex will violate the degree condition. The reason for setting $\text{THRESHOLD} = \log^{1+\delta} n$ will be apparent from the probabilistic analysis.

Once the key claim is established, we can bound both the total number of colors used and the running time of the algorithm. To bound the number of colors used observe that if the degree of the graph shrinks at every iteration by at least a $(1+\epsilon)/e$ factor *w.h.p.* then the maximum degree of G_r is at most $\log^{1+\delta} n$ *w.h.p.*

Hence, if $\Delta_C \geq$ THRESHOLD, then *w.h.p.*, the number of colors used by the bipartite algorithm is at most

$$BC(\Delta_C) \leq \Delta_C + \frac{\Delta_C}{e}(1+\epsilon) + \ldots + \frac{\Delta_C}{e^k}(1+\epsilon)^k + 2\log^{1+\delta} n,$$

where k is the smallest integer such that $\Delta_C(1+\epsilon)^k/e^k \leq \log^{1+\delta} n$. Thus, for a suitable $\epsilon' > 0$ which depends on ϵ and which can be made arbitrarily small, $BC(\Delta_C)$ is at most

$$\begin{aligned} BC(\Delta_C) &\leq \left(\frac{e}{e-1} + \epsilon'\right)\Delta_C + \left(2 - \frac{e}{e-1} - \epsilon'\right)\log^{1+\delta} n \\ &< 1.585\,\Delta_C + 0.4\,\log^{1+\delta} n \\ &< 1.59\,\Delta_C \end{aligned}$$

when $\Delta_C > 8\,\log^{1+\delta} n$. The running time of the algorithm is $O(\log n)$ because part (I) takes $O(\log \Delta_C)$ time, and part (II), *i.e.*, Luby's algorithm, takes $O(\log n)$ time.

Note that if $\Delta \geq 16$ THRESHOLD in the main algorithm, then $\Delta_C > 8\,\log^{1+\delta} n$ is true for the bipartite algorithm and hence the above analysis is valid; also note that if $\Delta \leq 16\,\log^{1+\delta} n$, then we use Luby's subroutine directly in our main algorithm.

Thus if $\Delta \geq 16$ THRESHOLD in the main algorithm, then the recurrence for the total number of colors used is

$$\begin{aligned} TC(\Delta) &\leq BC\left(\frac{\Delta}{2} + \sqrt{\Delta \log^{1+\delta/2} n}\right) + TC\left(\frac{\Delta}{2} + \sqrt{\Delta \log^{1+\delta/2} n}\right) \\ &< 1.59\,\Delta + o(\Delta) \\ &< 1.6\,\Delta\,. \end{aligned}$$

If $\Delta \leq 16\,\log^{1+\delta} n$, then the main algorithm uses Luby's subroutine directly to get a $2\Delta - 1 \leq 32\,\log^{1+\delta} n$ edge coloring. Hence, the total number of colors to color *any* graph is at most $1.6\,\Delta + 32\,\log^{1+\delta} n$ for any fixed $\delta > 0$, *w.h.p.*

A Truly Distributed Algorithm. We now sketch the modifications needed to handle the case when both Δ and n are unknown. Each node u initially computes the value $\Delta_u = \max_{v \in N(u)} \deg(v)$. The recursion of the Karloff & Shmoys scheme and the loop of part (I) of the bipartite subroutine are then repeated for $c \log \Delta_u$ times, for a constant $c > 0$ chosen large enough. A vertex u is said to be *active* as long as no more than $c \log \Delta_u$ rounds have elapsed; it is *inactive* otherwise. An edge incident on an inactive node is inactive. It is convenient to think of Luby's algorithm as run directly by the edges. An inactive and yet uncolored edge f will wait until all its neighboring edges are either colored or inactive, at which point it starts running Luby's algorithm with a palette of $deg(f) + 1$ fresh colors, where $\deg(f)$ denotes the number of inactive edges incident upon f. There are two main observations to prove the correctness and the bounds on the number of colors used by this modified algorithm. First, all the neighbors of a vertex u will stay active for at least $c \log \deg(u)$ rounds. Hence, all vertices such that $\deg(u) = \Omega(\log^{1+\delta} n)$ will be able (*w.h.p.*) to color enough edges to reduce their degree until it drops to $O(\log^{1+\delta} n)$. Second, as discussed in §2, Luby's algorithm still works correctly when vertices (in our case, edges) are added dynamically.

The high probability analysis carries through with these modifications. Simple calculations show that with these modifications the total number of colors used increases to at most $1.6\Delta + 160 \log^{1+\delta} n$. We omit the calculations for this modified algorithm, which are similar to those presented here for the case where n and Δ are known.

We now return to the case where n and Δ are known, and turn to the task of proving the key claim. We wish to show that given a graph G and Δ such that $\Delta \geq \Delta(G)$ and $\Delta \geq$ THRESHOLD then, after one iteration of Part (I) of the bipartite algorithm, the maximum degree of the new graph, $\Delta(G_\perp)$, is at most $(1+\epsilon)\Delta/e$ *w.h.p.*, for any fixed $\epsilon > 0$. It turns out that the analysis is considerably easier for the top vertices than for the bottom vertices. We begin with the easy part.

5.2. Analysis of Top Vertices. Let u be a generic top vertex with incident edges $i = (u, v_i)$. Recall that a *loser* is an edge which, after having got a tentative color in the random proposal, lost the lottery and got decolored. So, the new degree of u is given by the number of losers incident with it.

From the point of view of a top vertex, the random proposal and the lottery are equivalent to the following random experiment. For each edge i incident on u we introduce a ball i, and for each color k we introduce a bin k; the assignment of a tentative color to an edge by the algorithm is equivalent to throwing each ball into one of the Δ bins independently and uniformly at random, since the bottom vertices assign tentative colors with uniform probability and independently of the other bottom vertices. Recalling that we have at most Δ balls and exactly Δ bins:

$$\begin{aligned} \sharp losers &= \sharp balls - \sharp winners \\ &\leq \sharp bins - \sharp nonempty\ bins \\ &= \sharp empty\ bins. \end{aligned}$$

Let X be a random variable denoting the number of losers. To estimate X and its tail distribution we will study the random variable $B = \sharp empty\ bins$. For this purpose, we introduce Δ many indicator random variables B_i, where $B_i = 1$ if bin i is empty, and 0 otherwise; hence, $B = \sum_{i\in[\Delta]} B_i$. Notice that $X \leq B$ always. The variable B was studied in §3 where it was shown that $E[B] \leq \Delta/e$ and that the B_i's are 1-correlated, which implies that $Pr(B > (1+\epsilon)\Delta/e) \leq F(\Delta/e, \epsilon)$. Since $E[X] \leq E[B]$ and $Pr(X > (1+\epsilon)\Delta/e) \leq Pr(B > (1+\epsilon)\Delta/e)$ we get

THEOREM 5.1. *Let u be a top vertex and X be the random variable denoting the number of losers incident on it. Then, $E[X] \leq \Delta/e$ and*

$$Pr(X > (1+\epsilon)\Delta/e) \leq F(\Delta/e, \epsilon).$$

5.3. Analysis of Bottom Vertices. In this section we analyze what happens to the new degree of a generic bottom vertex v_b. This case is considerably harder to handle than the previous one, because of the way in which the random variables describing the process are correlated. For a top vertex, the dependency among the variables was playing for us; given that some edges incident on a top vertex are losers, the probability of having another loser decreases. For a bottom vertex the situation is reversed: having some edges lose the lottery might even make the probability of having another loser increase. The problem can be seen in the following situation.

Let $x_1 = v_b$ and x_2 be bottom vertices, and y_1 and y_2 be top vertices which induce a four-cycle, *i.e.,* there is an edge $e_{i,j} = (x_i, y_j)$ for $i, j = 1, 2$. Suppose we are given that $e_{1,1}$ got tentative color α and lost the lottery, and that $e_{1,2}$ got tentative color β; we will argue intuitively that given this, the probability of $e_{1,2}$ losing the lottery has increased. Since $e_{1,1}$ lost, the probability of $e_{2,1}$ getting tentative color α increases, which implies that the probability of $e_{2,2}$ getting tentative color β also increases, and this increases the probability of $e_{1,2}$ losing the lottery.

For the sake of the analysis we modify the algorithm as follows: instead of performing all random proposals in parallel, suppose that the bottom vertices perform their random proposals sequentially, in some order. This does not modify the probability distributions because the choices are still done independently. We want to focus our attention on the last vertex v_b performing the random proposal. We will use the fact that when v_b performs its random proposal, all edges not incident on v_b already have a tentative color. By symmetry, any upper bound on the probabilities we can find for v_b will hold for all bottom vertices.

Let $i \in [d_{v_b}]$ denote an edge incident with the bottom vertex v_b, with the other endpoint of i being u_i. We introduce the indicator random variables

$$X_i = \begin{cases} 1 & i \text{ loses the lottery} \\ 0 & \text{otherwise} \end{cases}$$

and want to study the expectation and tail probability distribution of $X = \sum_{i \in [d_{v_b}]} X_i$. Computing the expectation is easy.

LEMMA 5.2. *$E[X] \leq \Delta/e$.*

Proof. Let v_b be the bottom vertex. It is sufficient to show that $Pr(X_i = 1) \leq 1/e$, for all $i \in [d_{v_b}]$. From the analysis of the top vertices, we know that the expected number of losers incident with u_i is at most Δ/e and hence that $\sum_{j \in \delta(u_i)} Pr(j \text{ loses}) \leq \Delta/e$. By symmetry, $Pr(j \text{ loses}) \leq 1/e$ for all $j \in \delta(u_i)$, and hence $Pr(X_i = 1) \leq 1/e$. □

We now study the tail probability distribution of X. Our goal is to show that $X \leq (1+\epsilon)\Delta/e$ *w.h.p.*, for any fixed $\epsilon > 0$. Establishing this claim will take several lemmas.

We use a method different from the preliminary version of this work [16], to present stronger results. We first invoke a result of Schmidt, Siegel & Srinivasan [20], which in fact was motivated in part by [16] and in particular, by the notion of λ-correlation.

Define, for $z = (z_1, z_2, \ldots, z_n) \in \Re^n$, a family of symmetric polynomials $q_j(z)$, $0 \leq j \leq n$, where $q_0(z) \equiv 1$, and for $1 \leq j \leq n$,

$$q_j(z) \doteq \sum_{1 \leq i_1 < i_2 \cdots < i_j \leq n} z_{i_1} z_{i_2} \cdots z_{i_j}.$$

THEOREM 5.3 ([20]). *Let $Y_1, Y_2, \ldots, Y_n$ be arbitrary (not necessarily independent) 0-1 random variables with $Y = \sum_{i=1}^n Y_i$. Then for any $a > 0$ and any positive probability event Z,*

$$Pr(Y \geq a|Z) \leq \min_{\ell=1,\ldots,a} \frac{E[q_\ell(Y_1, Y_2, \ldots, Y_n)|Z]}{\binom{a}{\ell}}.$$

Proof. The actual theorem of [20] is stated unconditionally without reference to Z, but the above conditional extension is easily derivable from its proof, as follows.

Since the Y_is are binary, it is easily seen that for any $\ell \le a$, given that Z occured, $(Y \ge a)$ implies $(q_\ell(Y_1, Y_2, \ldots, Y_n) \ge \binom{a}{\ell})$. Thus by Markov's inequality,

$$Pr(Y \ge a|Z) \le \frac{E[q_\ell(Y_1, Y_2, \ldots, Y_n)|Z]}{\binom{a}{\ell}}. \quad \square$$

To bound the upper tail of X we will define an event $\mathcal{A}$ such that $\mathcal{A}$ happens *w.h.p.*, and such that for a suitably chosen k,

$$\frac{E[q_k(X_1, \ldots, X_{d_{v_b}})|\mathcal{A}]}{\binom{\Delta(1+\epsilon)e^{-1}}{k}} = e^{-\Omega(\Delta\epsilon^2)}. \tag{3}$$

In combination with Theorem 5.3, this will show that $Pr(X \le (1+\epsilon)\Delta/e)$ almost surely because,

$$\begin{aligned} Pr(X > (1+\epsilon)\Delta/e) &= Pr(X > (1+\epsilon)\Delta/e \mid \mathcal{A})\ Pr(\mathcal{A})\ + Pr(X > (1+\epsilon)\Delta/e \mid \mathcal{A}^c)\ Pr(\mathcal{A}^c) \\ &\le Pr(X > (1+\epsilon)\Delta/e \mid \mathcal{A})\ +\ Pr(\mathcal{A}^c) \qquad (4) \\ &\le \min_{i \in [d_{v_b}]} \frac{E[q_i(X_1, \ldots, X_{d_{v_b}})|\mathcal{A}]}{\binom{\Delta(1+\epsilon)e^{-1}}{i}} + Pr(\mathcal{A}^c) \\ &\le \frac{E[q_k(X_1, \ldots, X_{d_{v_b}})|\mathcal{A}]}{\binom{\Delta(1+\epsilon)e^{-1}}{k}} + Pr(\mathcal{A}^c), \end{aligned}$$

which is small by assumption (note that $\mathcal{A}$ happens *w.h.p.*). To prove the upper bound (3), we will focus on a generic term $Pr\left(\bigwedge_{i\in I} X_i = 1|\mathcal{A}\right)$ in

$$E[q_k(X_1, \ldots, X_{d_{v_b}})|\mathcal{A}] = \sum_{I \subseteq [d_{v_b}], |I|=k} E[\prod_{i\in I} X_i|\mathcal{A}] = \sum_{I \subseteq [d_{v_b}], |I|=k} Pr\left(\bigwedge_{i\in I} X_i = 1|\mathcal{A}\right),$$

which will also suggest to us a suitable choice for the event $\mathcal{A}$.

Consider then a generic subset $I = \{w_1, w_2, \ldots, w_k\} \subseteq [d_{v_b}]$ of edges incident on the bottom vertex v_b (corresponding to the neighbors $\{u_{w_i}\}$ of v_b), and let us see how to compute $Pr\left(\bigwedge_{i\in I} X_i = 1\right)$. Without loss of generality we assume $I = [k]$. Recall that we are analyzing the situation where v_b is the last vertex to perform its random proposal. This means that prior to the assignment of a tentative color to edge $i = (v_b, u_i)$, all other edges incident on u_i already have their tentative color. Using the balls-and-bins language, we can say that prior to throwing ball i at random into one of the bins at vertex u_i, all balls coming from the other neighbors of u_i have been thrown. We will think of i as a red ball and of the other edges at u_i as white balls. Once the red ball is thrown in, say, bin $\ell \in [\Delta]$, a winner is selected uniformly at random among all (*i.e.*, red *and* white) balls in bin ℓ. All other balls, the losers, are discarded. Notice that the probability of discarding the red ball is itself a random variable which depends on the particular placement of the white balls prior to throwing the red ball. (Hence, we will study the conditional probability that the red ball loses the lottery, given a placement of white balls.)

Given any placement of white balls at u_i, we construct a vector of probabilities C_i as follows. Let $a_{\ell,i}$ denote the number of white balls in bin $\ell \in [\Delta]$ of vertex u_i, and let $p_{\ell,i} = a_{\ell,i}/(1 + a_{\ell,i})$ denote the probability that the red ball loses the lottery given that it was thrown in bin ℓ (equivalently, $p_{\ell,i}$ is the probability that edge i loses, given that it got tentative color ℓ). For each neighbor u_i of our bottom vertex v_b, we construct the corresponding vector $C_i = (p_{1,i}, p_{2,i}, \ldots, p_{\Delta,i})$. We then construct a $\Delta \times k$ matrix M_I whose i-th column is the vector C_i. The next lemma explains why this matrix is relevant to us. From now on, let $p(m,\ell) \doteq m(m-1)\cdots(m-\ell+1)$.

LEMMA 5.4.

$$Pr\left(\bigwedge_{i\in I} X_i = 1\right) = \frac{perm(M_I)}{p(\Delta,k)}.$$

Proof. The random proposal of v restricted to I, is equivalent to choosing a one-to-one function $\pi : I \to [\Delta]$ uniformly at random among the set $\mathcal{P}$ of all such functions. Recall that the entry $M_{i,j}$ of M_I is the probability $p_{i,j}$ that edge w_j loses given that it is given color i. Hence,

$$\begin{aligned} Pr(\wedge_{i\in I} X_i = 1) &= \sum_{\pi\in\mathcal{P}} Pr(\wedge_{i\in I} X_i = 1 \mid \pi \textit{ is selected})\, Pr(\pi \textit{ is selected}) \\ &= \sum_{\pi\in\mathcal{P}} \left(M_{\pi(1),1}\, M_{\pi(2),2}\, \ldots\, M_{\pi(k),k}\right) \prod_{i=0}^{k-1} \frac{1}{\Delta - i} \\ &= \frac{perm(M_I)}{p(\Delta,k)}. \quad \square \end{aligned}$$

We now want to find a good upper bound for $perm(M_I)$. The following lemma gives a simple upper bound that is sufficient for our purposes.

LEMMA 5.5. *Let M be a matrix with c columns and r rows ($c \le r$) and non-negative entries. Let S_i denote the sum of the entries of the i-th column of M. Then,* $perm(M) \le \prod_{i\in[c]} S_i$.

Proof. Let $\mathcal{P} = \{\pi \mid \pi : [c] \to [r],\ \pi \text{ is one-to-one}\}$. Then,

$$\begin{aligned} perm(M) &= \sum_{\pi\in\mathcal{P}} M_{\pi(1),1}\, M_{\pi(2),2}\, \ldots\, M_{\pi(c),c} \\ &\le (M_{1,1} + \ldots + M_{r,1})(M_{1,2} + \ldots + M_{r,2}) \ldots (M_{1,c} + \ldots + M_{r,c}) \\ &= \prod_{i\in[c]} S_i. \quad \square \end{aligned}$$

The next lemma relates the value of S_i to that of $1/e \ge Pr(i \text{ loses})$, $i \in \delta(v_b)$. It is an application of the general definition of λ–correlation. Before the proof of the lemma, we establish

PROPOSITION 5.6. *If $0 \le p \le 1$, $q = 1 - p$ and m is a positive integer, then*

$$\sum_{r=1}^{m} \binom{m}{r} p^r q^{m-r} \frac{r}{r+1} = 1 - \frac{(1-q^{m+1})}{p(m+1)}.$$

Proof. Let

$$\begin{aligned} f(p) &= \sum_{r=1}^{m} \binom{m}{r} p^r q^{m-r} \frac{k}{k+1} \\ &= 1 - q^m - \sum_{r=1}^{m} \binom{m}{r} p^r q^{m-r} \frac{1}{k+1}. \end{aligned}$$

Integrating both sides of the binomial expansion

$$(x+q)^m = \sum_{r=0}^{m} \binom{m}{r} x^r q^{m-r}$$

between 0 and p, we get

$$\frac{1-q^{m+1}}{m+1} = p\,(1-f(p)),$$

from which the proposition follows. □

We now return to our scenario where v_b is the last bottom vertex to pick tentative colors for its edges. Recall that we are focusing on a set $I = \{w_1, w_2, \ldots, w_k\}$ of edges incident on v_b and that we want a good upper bound on $Pr\left(\bigwedge_{i \in I} X_i = 1\right)$; we had also assumed that $I = [k]$ without loss of generality. Combining Lemma 5.4 and Lemma 5.5 we get

$$Pr\left(\bigwedge_{i \in I} X_i = 1\right) \leq \frac{perm(M_I)}{p(\Delta, k)} \leq \frac{\Pi_{i=1}^{k} S_i}{p(\Delta, k)}$$

where for each $i \in [d_{v_b}]$, S_i is defined to be the sum of the entries in C_i. So, a good upper bound on S_i for each i, will hopefully translate into a good upper bound for $Pr\left(\bigwedge_{i \in I} X_i = 1\right)$. The next lemma says that $S_i \leq \Delta(1+\epsilon_1)/e$ *w.h.p.* for any fixed $\epsilon_1 > 0$, and for each i. Thus, a good choice for $\mathcal{A}$ is

$$\mathcal{A} : \text{“}S_i \leq \Delta(1+\epsilon_1)/e \text{ for each } i \in [d_{v_b}]\text{”},$$

where ϵ_1 will be fixed later. The next lemma is an application of the general definition of λ–correlation.

LEMMA 5.7. *Let $i = (v_b, u_i)$ be any edge in $[d_{v_b}]$, and S_i be the sum of the entries of C_i. Then, $E[S_i] \leq \Delta/e = \mu$ and*

$$\forall \epsilon_1 > 0, \quad Pr(S_i > (1+\epsilon_1)\mu) \leq F(\mu, \epsilon_1).$$

Proof. Let Z_ℓ be the random variable denoting the number of white balls in bin ℓ of u_i, and $Y_\ell = Z_\ell/(Z_\ell + 1)$ be the random variable denoting the probability that the red ball loses the lottery given that it lands in bin ℓ. Then, $S_i = Y \doteq \sum_\ell Y_\ell$. Note that the Y_ℓ's are bounded random variables with values in $[0,1]$. We will show that $E[Y] \leq \Delta/e$ and that the Y_ℓ's are 1–correlated (under the general definition of λ–correlation), which will give our claim.

We may assume that the total number d of white balls equals $\Delta - 1$ (*i.e.*, that the degree of u_i is Δ): $Pr(Y > (1+\epsilon_1)\Delta/e)$ is maximized at $d = \Delta - 1$, as d varies from 1 to $\Delta - 1$. (To see this, assume $d = \Delta - 1 - \ell < \Delta - 1$. Add ℓ *yellow* balls to the white balls, and run two experiments. In one experiment throw the white and red balls and compute the probability that the red ball loses the lottery. In the other experiment, throw white, yellow and red balls and again compute the probability that the red ball loses. In both experiments, let us look at the bin where the red ball fell. The probability that the red ball loses is $b/(b+1)$ for the first experiment, and $(b+y)/(b+y+1)$ for the second, where b and y are, respectively, the number of white and yellow balls in the bin. Since $y \geq 0$, $b/(b+1) \leq (b+y)/(b+y+1)$. If $Y_i(d)$ indicates the variable Y_i when u_i has degree d, then $Y_i(d) \leq Y_i(\Delta)$ for all $i \in [\Delta]$ and $d \in [\Delta]$.)

First, we will show that, for all i, $E[Y_i] \leq 1/e$ and then we will show that, for any set of ℓ indices $J \subseteq [\Delta]$ and strictly positive integers s_i,

$$E[\prod_{i \in J} Y_i^{s_i}] \leq 1/e^{\ell}. \tag{5}$$

Given this we can apply Corollary 3.3 by introducing n independent twin 0-1 random variables $\hat{Y}_i$ such that $E[\hat{Y}_i] = Pr(\hat{Y}_i = 1) = 1/e$. Since the $\hat{Y}_i$'s are binary, inequality (5) is the same as

$$E[\prod_{i \in J} Y_i^{s_i}] \leq \prod_{i \in J} E[\hat{Y}_i] = \prod_{i \in J} E[\hat{Y}_i^{s_i}],$$

which is to say that the Y_i's are 1–correlated. Noting that $0 \leq Y_i \leq 1$, it suffices to show that

$$E[\prod_{i \in J} Y_i] \leq 1/e^{\ell}. \tag{6}$$

Without loss of generality we can assume $J = [\ell]$. We will prove inequality (6) by induction on $\ell \geq 1$; when $\ell = 1$,

$$\begin{aligned} E[Y_1] &= \sum_{r=0}^{\Delta-1} \binom{\Delta-1}{r} \left(\frac{1}{\Delta}\right)^r \left(1 - \frac{1}{\Delta}\right)^{\Delta-1-r} \frac{r}{r+1} \\ &= (1 - 1/\Delta)^{\Delta} \leq 1/e, \end{aligned}$$

where the second equality follows from Proposition 5.6. Notice that for all $j \in [\Delta]$, $E[Y_j] = E[Y_1] \leq 1/e$. When $\ell > 1$, the law of conditional probabilities gives

$$E[\prod_{i \in [\ell]} Y_i] = E[Y_1 Y_2 \cdots Y_{\ell-1}\ E[Y_\ell \mid Y_1 Y_2 \cdots Y_{\ell-1}]] \tag{7}$$

Suppose we show that for all *non-zero* $c_i \in [0,1]$ with $i \in [\ell - 1]$,

$$E[Y_\ell \mid \bigwedge_{i=1}^{\ell-1} Y_i = c_i] \leq \frac{1}{e}; \tag{8}$$

then, since the product $Y_1 Y_2 \cdots Y_{\ell-1}$ in equation (7) is zero when any c_i is zero, we see by induction on ℓ that

$$\begin{aligned} E[\prod_{i=1}^{\ell} Y_i] &= E[Y_1 Y_2 \cdots Y_{\ell-1}\ E[Y_\ell \mid Y_1 Y_2 \cdots Y_{\ell-1}]] \\ &\leq \frac{1}{e} E[\prod_{i=1}^{\ell-1} Y_i] \\ &\leq \frac{1}{e^{\ell}}. \end{aligned}$$

Hence, the claim follows if we can show that inequality (8) holds.

If a_i denotes the number of white balls that fell into bin i, then $c_i = a_i/(a_i+1)$. Let $a = \sum_{i=1}^{\ell-1} a_i \geq \ell - 1$, $p = 1/(\Delta - \ell + 1)$, and $q = 1 - p$. Then

$$\begin{aligned} E[Y_\ell | \bigwedge_{i=1}^{\ell-1} Y_i = c_i] &= E[Y_\ell | \bigwedge_{i=1}^{\ell-1} Z_i = a_i] \\ &= \sum_{r=1}^{\Delta-1-a} t(r,a), \end{aligned}$$

where

$$t(r,a) \doteq \binom{\Delta - 1 - a}{r} p^r q^{\Delta-1-a-r} \frac{r}{r+1}.$$

It is easy to check that $t(r,a) \geq t(r,a+1)$. As a consequence, the maximum value of $E[Y_\ell | \bigwedge_{i=1}^{\ell-1} Y_i = c_i]$ is attained at $a = \ell - 1$, in which case we have

$$\begin{aligned} \sum_{r=1}^{\Delta-1-a} t(r,a) &= \sum_{r=1}^{\Delta-\ell} t(r,\ell-1) \\ &= \sum_{r=1}^{\Delta-\ell} \binom{\Delta-\ell}{r} p^r q^{\Delta-\ell-r} \frac{r}{r+1} \\ &= q^{\Delta-\ell+1} \leq 1/e, \end{aligned}$$

by Proposition 5.6. □

We remark that a short proof of Lemma 5.7 can be derived, using the elegant work of [7].

Define $\epsilon_1 \doteq \epsilon/10$. Thus, defining the event $\mathcal{A}$ as "$S_i \leq \Delta(1+\epsilon_1)/e$ for each $i \in [d_{v_b}]$", Lemma 5.7 gives the bound

$$Pr(\mathcal{A}^c) \leq \Delta F(\Delta/e, \epsilon_1). \tag{9}$$

Now given that $\mathcal{A}$ holds, Lemma 5.5 shows that $perm(M_I) \leq (\Delta(1+\epsilon_1)/e)^k$ and thus from Lemma 5.4,

$$Pr\left(\bigwedge_{i\in I} X_i = 1 | \mathcal{A}\right) \leq \frac{(\Delta(1+\epsilon_1)/e)^k}{p(\Delta,k)}. \tag{10}$$

We now turn to defining k suitably, to get a good tail bound. Invoking Theorem 5.3 for

$$X = \sum_{i\in[d_{v_b}]} X_i$$

in conjunction with (10), we see that if $a = \Delta(1+\epsilon)/e$, then

$$Pr(X > a | \mathcal{A}) \quad \leq \quad \frac{E[q_i(X_1,\ldots,X_{d_{v_b}})|\mathcal{A}]}{\binom{a}{k}}$$

$$\begin{aligned}
&\leq \frac{\binom{d_{v_b}}{k}(\Delta(1+\epsilon_1)/e)^k}{p(\Delta,k)\binom{a}{k}} \\
&\leq \frac{\binom{\Delta}{k}(\Delta(1+\epsilon_1)/e)^k}{p(\Delta,k)\binom{a}{k}} \\
&= \frac{(\Delta(1+\epsilon_1)/e)^k}{p(\Delta(1+\epsilon)/e,k)}. \qquad (11)
\end{aligned}$$

To lower bound $p(\Delta(1+\epsilon)/e, k)$, we need

LEMMA 5.8. *For positive integers t and ℓ, $t^\ell/p(t,\ell) \leq e^{\ell^2/t}$, if $\ell \leq t/2$.*

Proof. We first note that $\ln(1-x) \geq -2x$, for $0 \leq x \leq 1/2$. This is true, since if we define $f(x) \doteq \ln(1-x) + 2x$, then $f(0) = 0$ and $f'(x) = (1-2x)/(1-x)$, which is non-negative for $0 \leq x \leq 1/2$. Now,

$$\begin{aligned}
\frac{p(t,\ell)}{t^\ell} &= \prod_{i=1}^{\ell-1} \frac{(t-i)}{t} \\
&= \exp\left(\sum_{i=1}^{\ell-1} \ln(1-\frac{i}{t})\right) \\
&\geq \exp\left(-\sum_{i=1}^{\ell-1} \frac{2i}{t}\right) \qquad \text{(since } \ell \leq t/2) \\
&= \exp\left(-\frac{(\ell-1)\ell}{t}\right) \\
&\geq e^{-\ell^2/t}. \quad \square
\end{aligned}$$

We now set $k = \lfloor \Delta\epsilon/3e \rfloor$. Using Lemma 5.8 and the facts $\epsilon_1 = \epsilon/10$, $1+\epsilon \geq e^{\epsilon/2}$ and $1+\epsilon_1 \leq e^{\epsilon_1}$, we see, from (11), that

$$Pr(X \geq \Delta(1+\epsilon)/e | \mathcal{A}) \leq e^{-\Omega(\Delta\epsilon^2)}. \qquad (12)$$

Applying bounds (9) and (12) to (4), we finally arrive at

$$Pr(X \geq \Delta(1+\epsilon)/e) \leq e^{-\Omega(\Delta\epsilon^2)}. \qquad (13)$$

We can now see why the parameter THRESHOLD must be $\Omega(\log^{1+\delta} n)$: the failure probability (13) goes to zero superpolynomially fast if $\Delta = \Omega(\log^{1+\delta} n)$, for any fixed $\delta > 0$. Using (13), we conclude our analysis with

THEOREM 5.9. *The new degree of the graph after one iteration of Part (I) of the bipartite algorithm is at most $(1+\epsilon)\Delta/e$ w.h.p., for any fixed $\epsilon > 0$.*

6. Extensions and Applications of the Algorithm. Recently, Panconesi and Dubhashi have improved our bounds by presenting a randomized distributed edge-coloring algorithm that runs in polylogarithmic time and uses at most $\Delta(1+o(1)) + O(\log n)$ colors *w.h.p.*[6]. However, we feel that this work has independent interest owing to the tools developed to analyze the algorithm. We now describe some recent applications of this work.

Our results on λ–correlation have been used to prove the performance of a randomized rounding technique for multicommodity flow (Young [23]), and to provide an elementary method to bound the upper tail of the number of prime factors of random integers (Srinivasan [21]). As mentioned in §5.3, the work of [20] which expands the applicability of CH-type bounds to more non-independent scenarios, was inspired in part by this work. Our results on upper tail bounds for sums of bounded λ–correlated random variables, have been generalized in [20].

Our algorithm has also been used and extended in the context of emulating PRAM algorithms using a limited number of processors [5].

Acknowledgments. Our sincere thanks go to David Shmoys, for his continued guidance and support. We are grateful to Éva Tardos for an important idea about analyzing the algorithm, and to Suresh Chari, Devdatt Dubhashi, David Pearson, Desh Ranjan, Pankaj Rohatgi, and Stephen Vavasis for useful discussions. We also thank the referees for their several valuable comments.

REFERENCES

[1] N. Alon, J. Spencer, and P. Erdős, *The Probabilistic Method*, Wiley–Interscience Series, John Wiley & Sons, Inc., New York, 1992.

[2] B. Berger and J. Rompel, *Simulating ($\log^c n$)-wise independence in NC*, Journal of the ACM, 38 (1991), pp. 1026–1046.

[3] B. Bollobás, *Graph Theory*, Springer Verlag, New York, 1979.

[4] H. Chernoff, *A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations*, Annals of Mathematical Statistics, 23 (1952), pp. 493–509.

[5] X. Deng and P. Dymond, *Minimizing communication phases in optimal parallel algorithms*, Technical Report 94-04, Department of Computer Science, York University, 1994.

[6] D. Dubhashi and A. Panconesi, *Near-optimal distributed edge coloring*, 3rd Annual European Symposium on Algorithms (ESA 95), September 1995, Springer-Verlag LNCS 979, pp. 448–459.

[7] D. Dubhashi and D. Ranjan, *Balls and Bins: A Study in Correlations*, submitted.

[8] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, American Statistical Association Journal, 58 (1963), pp. 13–30.

[9] R. Jain, K. Somalwar, J. Werth, and J. C. Browne, *Scheduling parallel I/O operations in multiple bus systems*, Journal of Parallel and Distributed Computing, 16 (1992), pp. 352–362.

[10] H. J. Karloff and D. B. Shmoys, *Efficient parallel algorithms for edge coloring problems*, Journal of Algorithms, 8 (1987), pp. 39–52.

[11] G. F. Lev, N. Pippenger, and L. G. Valiant, *A fast parallel algorithm for routing in permutation networks*, IEEE Transactions on Computers, 30 (1981), pp. 93–100.

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

[13] M. Luby, *Removing randomness in parallel computation without a processor penalty*, Journal of Computer and System Sciences, 47 (1993), pp. 250–286.

[14] C. McDiarmid, *On the method of bounded differences*, in *Surveys in Combinatorics*, number 141 in London Math. Soc. Lecture Notes Series, Cambridge University Press, 1989, pp. 148–188.

[15] R. Motwani, J. Naor, and M. Naor, *The probabilistic method yields deterministic parallel algorithms*, Journal of Computer and System Sciences, 49 (1994), pp. 478–516.

[16] A. Panconesi and A. Srinivasan, *Fast randomized algorithms for distributed edge coloring*, in Proc. ACM Symposium on Principles of Distributed Computing, 1992, pp. 251–262.

[17] A. Panconesi and A. Srinivasan, *Improved distributed algorithms for coloring and network decomposition problems*, in Proc. ACM Symposium on Theory of Computing, 1992, pp. 581–592.

[18] P. Raghavan, *Lecture notes on randomized algorithms*, Technical Report RC 15340 (#68237), IBM T.J.Watson Research Center, January 1990.

[19] R. Raman, *The power of Collision: Randomized parallel algorithms for chaining and integer sorting*, in Proc. 10th Annual FST & TCS Conference, Lecture Notes in Computer Science # 472, Springer-Verlag, Berlin, December 1990, pp. 161–175. Also available as University of Rochester CS Dept. TR 336, March 1990 (Revised January 1991).

[20] J. P. Schmidt, A. Siegel, and A. Srinivasan, *Chernoff-Hoeffding bounds for applications with limited independence*, SIAM Journal on Discrete Mathematics, 8 (1995), pp. 223–250.

[21] A. Srinivasan, *On the distribution of the number of prime factors of integers*, Manuscript, 1993.
[22] V. G. Vizing, *On an estimate of the chromatic class of a p–graph*, Diskret. Anal., 3 (1964), pp. 25–30. In Russian.
[23] N. Young, *Randomized rounding without solving the linear program*, in Proc. ACM/SIAM Symposium on Discrete Algorithms, 1995, pp. 170–178.