Fast Distributed Algorithms for Brooks-Vizing Colourings

David A. Grable*
Institut für Informatik
Humboldt-Universität zu Berlin
D-10099 Berlin Germany
grable@informatik.hu-berlin.de

Alessandro Panconesi†
BRICS
University of Århus
DK-8000 Århus C, Denmark
ale@brics.dk

Abstract

Let G be a Δ–regular graph with n vertices and girth at least 4 such that $\Delta \gg \log n$. We give very simple, randomized, distributed algorithms for vertex colouring G with Δ/k colours in $O(k + \log n/\log\Delta)$ communication rounds, for $k = O(\log\Delta)$. The algorithm may fail or exceed the above running time, but the probability that this happens is $o(1)$, a quantity that goes to zero as n grows. The probabilistic analysis relies on a powerful generalization of Azuma's martingale inequality that we dub the Method of Bounded Variances.

1 Introduction

Vertex colouring is a much studied problem in combinatorics and computer science for its theoretical as well as its practical aspects. In this paper we are concerned with the "distributed" version of a question stated by Vizing, concerning a Brooks-like theorem for sparse graphs. Roughly, the question asks whether there exist colourings using many fewer than Δ colours, where Δ denotes the maximum degree of the graph, provided that some sparsity conditions are satisfied. In this paper we show that such colourings not only exist, but that they can be quickly computed by extremely simple distributed, randomized algorithms. Before stating our results precisely, we review the relevant facts.

For any graph G of maximum degree Δ with n vertices, the following *trivial algorithm* computes a $\Delta + 1$ (list) colouring. Each vertex u initially has a list, or *palette*, of $\deg(u) + 1$ colours. The computation proceeds in rounds. During each round, each uncoloured vertex, in parallel, first performs a *trivial attempt*: it picks a tentative colour at random from its palette and if no neighbour picks the same colour, the colour becomes final and the algorithm stops for that vertex. Otherwise, the vertex's palette undergoes a *trivial update*—the colours succesfully used by the neigbours are removed—and a new attempt is performed in the next round.

Henceforth we shall refer to this as "the" trivial algorithm. The trivial algorithm always computes a valid colouring regardless of the composition of the initial lists, and does so in $O(\log n)$ rounds with high probability—that is, with probability approaching 1 as the number of vertices increases [13].

It is apparent that the trivial algorithm is distributed, since each vertex only relies on information from the neighbouring vertices. The well-known distributed algorithm for the same problem given by Luby [18] amends the trivial algorithm in the following way: at the beginning of each round every uncoloured vertex is *asleep*. Each such vertex *awakes* with probability p and executes a trivial attempt (in Luby's paper $p = \frac{1}{2}$). Then, whether or not the vertex awoke, the palette undergoes a trivial update. At the end of the round the vertex goes back to sleep.

*Supported by Deutsche Forschungsgemeinschaft project number Pr 296/4-2.

†This research was partly done when visiting the Humboldt University in Berlin thanks to financial support of the Alexander von Humboldt foundation.

We shall refer to this variant of the trivial algorithm as the *dozing-off* algorithm. The dozing-off algorithm has the same asymptotic performance as the trivial algorithm, but its analysis just needs pairwise independence. Luby used this fact to carry out a derandomization procedure in the PRAM model.

Can better colourings—i.e. colourings using fewer colours—be computed efficiently in a distributed setting? In 1948 Brooks gave a theorem that characterizes the graphs which are not Δ–colourable: a graph is Δ–colourable if and only if it is neither an odd cycle nor a $\Delta+1$ clique (see, for instance, [3]).

The proof of Brooks' theorem is actually a polynomial time sequential algorithm. Δ–colourings can also be quickly (i.e. in polylogarithmic time) computed in the PRAM model [11, 14, 15]. In fact, a distributed version of Brooks' theorem can be derived from a certain locality property of Δ-colourings, yielding the following: There is no $o(n)$ randomized, synchronous protocol to Δ-colour paths, cycles or cliques. For all other graphs, there is a randomized protocol which, with high probability, computes a Δ-colouring in polylogarithmically many rounds [21]. (The property in question, holding for graphs which are neither cliques, paths nor cycles, is this: If G is Δ–coloured except for one last vertex, it is possible to complete the colouring by a simple recolouring operation along an "augmenting" path of length $O(\log_\Delta n)$ starting from the uncoloured vertex [21].)

It is an open problem whether randomization is necessary in all of the above algorithmic results; the asymptotically best deterministic protocols known to date need $O(n^{\epsilon(n)})$ rounds, where $\epsilon(n)$ tends (very slowly) to zero as the number of vertices grows [1, 22].

In a 1968 paper Vizing asked whether upper bounds for the chromatic number better than those given by Brooks' Theorem existed, provided some sparsity conditions were satisfied. In particular, he asked what happens for triangle-free graphs. We shall refer to colourings of triangle-free graphs using significantly fewer than Δ colours as *Brooks-Vizing colourings*.

This existential problem was settled about two decades later. A. Johansson [12] showed that every triangle-free graph has chromatic number $O(\Delta/\log\Delta)$. This is best-possible up to a constant factor, since Bollobás had shown the existence of graphs with arbitrarily high girth such that $\chi(G) = \Omega(\Delta/\log\Delta)$ (the girth of a graph G is the size of the smallest cycle therein) [4].

Johansson's result, as well as an earlier result of Kim [17], which shows that graphs of girth at least 5 have chromatic number $(1+o(1))\Delta/\log\Delta$, make use of certain distributed colouring algorithms, but their results are only existential in the following sense. They show only that the probability that the algorithm produces a valid colouring is positive. Their analyses do not rule out the possibility of their algorithms failing with a high probability. (But then, this was not their main concern.)

In this paper, we show that Brooks-Vizing colourings can be computed efficiently even in a distributed setting. We present very simple randomized, distributed algorithms, which are also easily implementable on a PRAM and in the sequential setting, and demonstrate that they produce the desired colourings with high probability.

Our algorithms are variants of the dozing-off algorithm. In fact, when the input graph has no 4-cycles (girth 5 or greater) the algorithm is the dozing-off algorithm modified so that the probability that vertices awake is not constant but varies with the round. This probability, initially very low, quickly rises to one, causing the algorithm to be behave as the trivial one from that point on. For the general triangle-free (girth 4) case, the algorithm adds a mechanism which forces the vertex degrees of the uncoloured portion of the graph to remain roughly regular. This regularity is extremely useful in the analysis, but unfortunately gives the algorithm a somewhat high message and space complexity. It may be, however, that this mechanism is unnecessary. The simplicity of the algorithms is an appealing feature and we expect them to work quite well in practice.

Although these algorithms display similarities to those in Kim's work [17], we have striven for speed and simplicity. Moreover, our analyses are much simpler than those given there. It is perhaps worth remarking that our analyses demonstrate that Brooks-Vizing colourings are not rare, for the randomized colour assignments computed by the algorithm are almost always such colourings. Furthermore, the analyses highlight the role played by the girth assumption.

Our result may be conveniently stated as follows: We give an algorithm which, for any triangle-free, Δ-regular input graph G such that $\Delta \geq \log^{1+\delta} n$, where $\delta > 0$ is any fixed constant, computes with probability $1-o(1)$ a vertex colouring of G using Δ/k colours, for any $k \leq \alpha \log \Delta$, where α is a constant which depends on δ. Moreover, with probability $1 - o(1)$, the colouring will be completed within $O\left(k + \frac{\log n}{\log \Delta}\right)$ rounds in the synchronous, message-passing distributed model of computation (with no shared memory). Both of the above $o(1)$ terms are functions going to 0 with n, the number of vertices in the network.

The statement of the theorem allows some flexibility in the choice of k and Δ. For instance, by choosing $\Delta \geq n^{c/\log\log n}$, where $c > 0$ is any constant, and $k = \log\log n$, the algorithm will compute a $(\Delta/\log\log n)$-colouring in just $O(\log\log n)$ rounds. Or, by choosing $\Delta \geq n^{c/\sqrt{\log n}}$ the algorithm will compute a $(\Delta/\sqrt{\log n})$-colouring in $O(\sqrt{\log n})$ rounds. Notice also that the algorithm works for $k = \Omega(\log \Delta)$, thereby matching the lower bounds of Bollobás and the existential statements of Johansson and Kim. It should be pointed out however that our statement is weaker than their existential statement insofar as it needs the additional assumption $\Delta = \Omega(\log n)$. This, as well as the regularity assumption (also assumed in [12, 17]), might in fact be an artifact of our analysis which relies on large deviation inequalities that cease to give strong enough bounds for lower values of Δ. The probabilistic analysis makes use of a powerful generalization of Azuma's inequality, described and commented upon in Section 3, that is very well suited for the analysis of algorithms [8, 10, 6, 7].

Although we stated our result in its most general form, for brevity's sake, we shall actually present a slightly weaker version. Namely, we shall show that the above statement holds with the running time replaced by $O(k + \log n)$.

2 Preliminaries

We shall make use of the following definitions, notation and conventions:

- $B(A, p_a)$ denotes a random variable which is binomially distributed over the set A and where p_a denotes the probability that for $a \in A$ the outcome is `heads`;
- We shall make use of the asymptotic notation: $a \sim b$ means $a = (1 + o(1))b$.
- $f(n) \gg g(n)$ means that $g(n)/f(n)$ goes to zero as n grows;
- A graph G is *almost D-regular* if $\deg(u) \sim D$ for all vertices u;
- N_u^k denotes the set of vertices at distance exactly k from u. N_u stands for N_u^1;
- All logarithms are natural logarithms; e will always denote the base of the natural exponential.

The following fact will be used repeatedly (see, for instance, [20]).

Fact 1 *Let $A(n)$ and $B(n)$ be such that $A(n)^2 B(n) = o(1)$ (n tending to infinity, as always). Then,*

$$(1 - A(n))^{B(n)} = (1 + o(1))e^{-A(n)B(n)}.$$

Of the many versions of the Chernoff-Hoeffding bounds, we shall use the following.

Fact 2 *Let $X = B(A,p)$ and $\mu := \mathrm{Ex}[X]$, then, for all $0 \le t \le \mu$,*

$$\Pr[|X-\mu| \ge t] \le 2\exp\left\{-\frac{t^2}{3\mu}\right\}.$$

3 The Method of Bounded Variances

Recent years have seen the increased utilization of sophisticated tools from probability theory to assist in the analysis of probabilistic algorithms. Of particular importance are tools which allow one to establish that the randomized algorithm under study obeys the concentration of measure phenomenon. Roughly speaking, this means that the random variables describing the behaviour of the algorithm are sharply concentrated around their expectations. For an introduction to this important and fascinating subject see, for instance, [7, 20, 23]. In this paper we make use of a powerful generalization of Azuma's martingale inequality developed by the first author. This method was originally called "query games" but since we now feel that "The Method of Bounded Variances" (henceforth MOBV) is a better terminology, we shall adopt it. For the genesis of the method and the proofs, see [8]. For an introduction to the area of concentration of measure and a comparative analysis of the various existing methods, including recent ones such as Talagrand's inequalities, see [7, 23]. For applications of the MOBV to the analysis of algorithms see [6, 8, 10] and, obviosuly, the present paper!

In order to make the paper as self-contained as possible we now describe the MOBV. Assume we have a probability space generated by independent random variables X_i (choices), where choice X_i is from the finite set A_i, and a function $Y = f(X_1, \ldots, X_n)$ on that probability space. We are interested in proving a sharp concentration result on Y, i.e. to bound $\Pr[|Y - \mathrm{Ex}[Y]| > a]$, for any a, as well as we can. The well-known Chernoff-Hoeffding bounds give essentially best possible estimates when $Y = \sum_i X_i$. The Method of Bounded Differences (MOBD) [19], a nicely packaged generalization of a martingale inequality known as Azuma's inequality, allows one to consider any function $Y = f(X_1, \ldots, X_n)$ which satisfies the "bounded difference" requirement that changing the choice of one of the variables does not affect the final value of Y by too much. More precisely, the result states that if, for all vectors A and B differing only in the i-th coordinate,

$$|f(A) - f(B)| \le c_i \tag{1}$$

then

$$\Pr\left[|Y - \mathrm{Ex}[Y]| > \sqrt{\varphi \sum_i c_i^2/2}\right] \le 2e^{-\varphi}. \tag{2}$$

When using this inequality, the crux of the matter is to find values c_i which are as small as possible. In an extension of this result known as the MOBD in expected form, Condition 1 is replaced by

$$|\,\mathrm{Ex}[f|X_1, \ldots, X_i] - \mathrm{Ex}[f|X_1, \ldots, X_{i-1}]| \le c_i. \tag{3}$$

This might translate in much stronger bounds, but at the expense of involved computations. The MOBV offers a "quick-and-dirty" solution, by allowing to find quite good estimates of the c_i essentially without computations (for a more comprehensive discussion of these issues, see [7]).

We now describe the method precisely. Consider the following procedure, the aim of which is to determine the value of Y. We can ask queries of the form "what was the i-th choice?"—i.e. "what was the choice of X_i?"—in any order we want. The answer to the query is the random choice of X_i.

A *querying strategy* for Y is a decision tree whose internal nodes designate queries to be made. Each node of the tree represents a query of the type "what was the random choice of X_i?". A node has as many children as there are random choices for X_i. It might be helpful to think of the edges as labeled with the particular $a \in A_i$ corresponding to that random choice. In this fashion, every path from the root to a node which goes through vertices corresponding to $X_{i_1}, \ldots, X_{i_k}$ defines an assignment $a_1, \ldots, a_k$ to these random variables. We can think of each node as storing the value $\mathrm{Ex}[Y \mid X_{i_1} = a_1, \ldots, X_{i_k} = a_k]$. In particular, the leaves store the possible values of Y, since by then all relevant random choices have been determined.

Define the *variance of a query* (internal node) q concerning choice X_i to be

$$v_q = \sum_{a \in A_i} \Pr[X_i = a]\; \mu_{q,a}^2,$$

where

$$\mu_{q,a} = \mathrm{Ex}[Y \mid X_i = a \text{ and all previous queries}] - \mathrm{Ex}[Y \mid \text{all previous queries}].$$

By "all previous queries," we mean the condition imposed by the queried choices and exposed values determined by the path from the root of the strategy down to the node q. In words, $\mu_{q,a}$ measures the amount which our expectation changes when the answer to query q is revealed to be a.

Also define the *maximum effect* of query q as

$$c_q = \max_{a,b \in A_i} |\mu_{q,a} - \mu_{q,b}|.$$

As we shall see, in practice good upper bounds on c_q are very easy to obtain.

A *line of questioning* ℓ is a path in the decision tree from the root to a leaf and the *variance of a line of questioning* is the sum of the variances of the queries along it. Finally, the *variance of a strategy* $\mathcal{S}$ is the maximum variance over all lines of questioning

$$V(\mathcal{S}) = \max_{\ell} \sum_{q \in \ell} v_q.$$

The use of the term variance is meant to be suggestive (but hopefully not confusing): $V(\mathcal{S})$ is an upper bound on the variance of Y. The variance plays essentially the same role as the term $\sum_i c_i^2$ in the MOBD, but it is a much better upper bound to the variance of Y.

Proposition 1 *Let $\mathcal{S}$ be a strategy for determining Y and let the variance of $\mathcal{S}$ be at most V. Then for every $0 \leq \varphi \leq V/\max c_q^2$,*

$$\Pr\left[|Y - \mathrm{Ex}[Y]| > 2\sqrt{\varphi V}\right] \leq 2e^{-\varphi}.$$

One reason why the term v_q is often much smaller than the c_i^2 term found in the MOBD is that the probabilities that X_i takes on value a for the various $a \in A_i$ can be factored into the computation. This is especially apparent when the $\mu_{q,a}$'s take on only two values (when holding q fixed and varying $a \in A_i$). This partitions the space A_i into two regions, which we call the YES region and the NO region. These regions correspond to two mutually exclusive events, the

YES and NO events. In this paper, for instance, the A_i will be a set of colours and the YES event will often be of the form "was choice X_i colour γ?".

Denote the probabilities of these two events by $p_{q,\text{YES}}$ and $p_{q,\text{NO}} = 1 - p_{q,\text{YES}}$ and the values of $\mu_{q,a}$ taken on within each of these region by $\mu_{q,\text{YES}}$ and $\mu_{q,\text{NO}}$. In [6] the following fact is proven.

Fact 3

$$v_q = \sum_{a \in A_i} \Pr[X_i = a]\ \mu_{q,a}^2 \leq p_{q,\text{YES}} c_q^2.$$

We shall use this fact repeatedly.

A simpler bound, which applies regardless of the values of the $\mu_{q,a}$'s, is

$$v_q \leq c_q^2/4. \tag{4}$$

The proof appears in [6].

A further improvement that makes the method particularly versatile is its ability to dispense with subspaces of very low probability where the c_i's are very large. In other words, there might be just a few, very low probability vectors of choices (outcomes) which lead to lines of questioning with very large variance. This of course drives up the variance of the whole strategy, possibly ruining the usefulness of the result. We'd like to be able to ignore these exceptional outcomes and the corresponding lines of questioning. The following corollary, also proven in [8], says that we can.

Corollary 1 *Let M and m be, respectively, the minimum and maximum values taken by Y over all possible outcomes. Let $\mathcal{C}$ be a set of exceptional outcomes. Consider a strategy for determining $(Y|\overline{\mathcal{C}})$, i.e. assuming that the actual outcome is not in $\mathcal{C}$. If the variance of this strategy is at most V then*

$$\Pr[|Y - \text{Ex}[Y]| > 2\sqrt{\varphi V} + (M - m)\Pr(\mathcal{C})] \leq 2e^{-\varphi} + \Pr[\mathcal{C}],$$

for every $0 \leq \varphi \leq V / \max c_q^2$.

In this paper we shall apply this corollary systematically.

A rule of thumb to decide whether the bounds on the variance are going to be good enough is that if the variance is of the same order as the expectation then the bounds given by the MOBV are comparable in strength with those of the Chernoff-Hoeffding bounds.

4 Girth 5 or more

In this section we assume the distributed network to be a Δ-regular graph with girth at least 5. We'll give a colouring algorithm for this situation and analyse it. Then later, we will add a mechanism for dealing with the presence of 4-cycles.

Each node of the network knows the value of Δ and of a positive real k. The goal is a colouring with Δ/k colours. The protocol is as follows.

Algorithm Dozing-off
Input: a graph G of maximum degree Δ such that $\Delta \gg \log|V(G)|$ and a parameter $k = O(\log \Delta)$.
Output: a Δ/k vertex colouring of G.
begin
  for each vertex u **in parallel**

```
/* initialize palette and wake-up parameter */
A := {1, ..., Δ/k}; p := 1/k.
p := 1/k.
repeat until u is coloured:
    /* attempt to colour */
    Awake with probability p.
    if awake then choose a tentative colour t uniformly from A.
        if no neighbour picked t during this round then
            make colour t permanent and stop.
    /* update palette and wake-up parameter */
    A := A − {c | a neighbour of u coloured itself with c}.
    if p < 1 then    p := min{1, 1/(1/p − 1/e)}.
end
```

The algorithm is exactly the dozing-off algorithm where the probability of awakening increases from round to round, until, after ek rounds, $p = 1$. From this point on, the algorithm is just the trivial algorithm. Our goal is to establish the following theorem.

Theorem 1 *Algorithm Dozing-off computes a D/k–colouring of the input graph G in $O(k + \log n/\log \Delta)$ rounds with high probability. "High probability" means that the probability that the algorithm fails or that it exceeds the above running time is $o(1)$, a quantity that goes to zero as the number of vertices of the graph grows.*

In the analysis of the algorithm we keep track of the following random variables:

- For each vertex u and round i, the size of u's palette at round i; denoted by $s_i(u)$;
- For each vertex u, colour c and round i, the number of uncoloured neighbours of u which have c in their palettes; denoted by $d_i(u,c)$.
- For each vertex u and round i, the number of uncoloured neighbours of u; denoted by $\deg_i(u)$.

Our goal here is to analyse the algorithm sufficiently to show that with high probability there exists a round $i = O(k)$ such that, for every vertex u,

$$s_i(u) > \deg_i(u) + 1. \tag{5}$$

We will refer to this as the *termination condition*. This will occur after the algorithm has switched to its trivial phase. It is known that if the graph satisfies condition 5 the trivial algorithm will *with certainty* complete the colouring and it will do so within $O(\log n)$ rounds with high probability [13]. Therefore, if we prove that the termination condition is attained in $O(k)$ rounds with high probability, we can conclude that the dozing-off algorithm succeeds in $O(k + \log n)$ rounds with high probability– a slightly weaker version of Theorem 1. As stated in the introduction, with a little bit more work it is possible to show that the running time is actually $O(k + \log n/\log \Delta)$. For the sake of brevity, we shall content ourselves to carry the analysis only up to the point where the termination condition holds. We remark that this is in any case the most difficult and interesting part of the analysis.

In order to establish the termination condition, we shall show that the random variables $s_i(u)$, $d_i(u,c)$, and $\deg_i(u)$ are approximated very well by "ideal" values s_i, d_i and D_i, which will be defined by means of suitable recurrences. After that is done, it is a simple matter to determine at what point $s_i \gg D_i$, which implies the termination condition. Indeed, we do this first.

Not surprisingly, the random variables evolve differently during the dozing-off and trivial phases of the algorithm, prompting two sets of recurrences for the ideal values. The first set applies for round $i < ek$, during the dozing-off phase. Here,

$$\begin{aligned} s_{i+1} &= s_i \exp(-1/e), \\ d_{i+1} &= d_i \left(1 - \frac{s_i}{d_i}\frac{1}{e}\right) \exp(-1/e), \text{ and} \\ D_{i+1} &= D_i \left(1 - \frac{s_i}{d_i}\frac{1}{e}\right). \end{aligned} \tag{6}$$

with initial conditions $D_0 = d_0 = \Delta$ and $s_0 = \Delta/k$.

The recurrences depend on the ratio d_i/s_i, which, during the dozing-off phase, satisfies the recurrence

$$\frac{d_{i+1}}{s_{i+1}} = \frac{d_i}{s_i}\left(1 - \frac{s_i}{d_i}\frac{1}{e}\right) = \frac{d_i}{s_i} - \frac{1}{e}.$$

Thus we immediately see that the ratio decreases in each step by the constant $1/e$. After $ek = O(k)$ steps, we'll no longer be in the situation that $s_i < d_i$. Observe that this number of steps is constant if k is. Also notice that the wake-up probability p is in every round i defined to be exactly equal to the inverse ratio s_i/d_i, implying that the algorithm switches from the dozing-off to the trivial phase in $ek = O(k)$ steps. This value of p was chosen in order to maximize the probability that a vertex colours itself during the dozing-off phase.

After ek steps,

$$d_{ek} = s_{ek} = s_0 \exp(-ek/e) = \frac{\Delta}{ke^k}. \tag{7}$$

Since, at all times, the vertex palettes must be nonempty, $\Delta/ke^k \geq 1$ is a necessary condition (on k in terms of Δ). Asymptotically, this condition is satisfied for all $k = (\log \Delta)/\beta$ with $\beta > 1$ and not satisfied for $k = \log \Delta$. Thus we see that $k = O(\log \Delta)$ is a necessary condition for the algorithm to work. This is in accordance with the lower bound given by Bollobás for the existence of Brooks-Vizing colourings [4].

The ratio D_i/d_i is initially 1, but increases by a factor of $\exp(1/e)$ in each step. Therefore, $D_{ek}/d_{ek} = e^k$ and we see that

$$D_{ek} = \frac{\Delta}{k}. \tag{8}$$

Thus at the end of the dozing-off phase, we are still quite far from attaining the termination condition, so we must continue our analysis into the trivial phase.

During the trivial phase, the ideal values obey the following recurrences:

$$\begin{aligned} s_{i+1} &= s_i \exp\left\{-\frac{d_i}{s_i} e^{-d_i/s_i}\right\}, \\ d_{i+1} &= d_i \left(1 - e^{-d_i/s_i}\right) \exp\left\{-\frac{d_i}{s_i} e^{-d_i/s_i}\right\}, \text{ and} \\ D_{i+1} &= D_i \left(1 - e^{-d_i/s_i}\right). \end{aligned} \tag{9}$$

Once again, these recurrences depend on the ratio d_i/s_i. Since

$$\frac{d_{i+1}}{s_{i+1}} = \frac{d_i}{s_i}\left(1 - e^{-d_i/s_i}\right) < \left(\frac{d_i}{s_i}\right)^2, \tag{10}$$

we see that this ratio goes to zero at a doubly exponential rate!

This implies that D_i decreases very much faster than s_i—exactly what we need for condition (5). To see this, bound s_i and D_i as follows (for the second one, use the fact that $e^{-x} \geq 1 - x$):

$$\begin{aligned} s_{i+1} &> s_i \exp\left\{-\frac{d_i}{s_i}\right\} \\ D_{i+1} &\leq D_i\left(1 - \left(1 - \frac{d_i}{s_i}\right)\right) = D_i\frac{d_i}{s_i}. \end{aligned} \tag{11}$$

From these recurrences we see that while D_i also decreases doubly exponentially, the rate at which s_i decreases is much slower and ever decreasing.

Since the ratio d_i/s_i may be identically one at the beginning of the trivial phase ($i = ek$), the inequality (10) would not be very useful if we started there. Instead, take the values after one round of the trivial phase ($i = ek + 1$). At that point we have $s_{ek+1} > \Delta/ke^{k+1}$, $D_{ek+1} \leq \Delta/k$ and

$$\rho := \frac{d_{ek+1}}{s_{ek+1}} \leq 1 - \frac{1}{e} < 1.$$

Now, using Equation (10), we get

$$\frac{d_{ek+i}}{s_{ek+i}} \leq \rho^{2^{i-1}}.$$

Therefore, using (11) repeatedly, we see that

$$D_{ek+i} \leq D_{ek}\ \rho^{2^{i+1}-1} \leq \frac{\Delta}{k}\ \rho^{2^{i+1}-1},$$

while

$$s_{ek+i} \geq s_{ek+1}\exp\left\{-\sum_{j=0}^{i-2}\rho^{2^j}\right\} \geq s_{ek+1}\exp\left\{-\sum_{j=0}^{i-2}\rho^{j}\right\} \geq \frac{\Delta}{ke^{k+1}}\exp\left\{-\frac{\rho}{1-\rho}\right\}. \tag{12}$$

Thus within $O(\log k)$ additional rounds,

$$D_i < cs_i \tag{13}$$

for all constants $c < 1$. Also, if $k = O(\log \Delta)$ then $s_i > 1$ for every i, which intuitively means that the algorithm never runs out of colours.

To summarize, if we could show that the random variables $\deg_i(u)$, $s_i(u)$ and $d_i(u, c)$ were arbitrarily close to the ideal values D_i, s_i and d_i then Equation (13) would imply that the termination condition holds. Our aim for the rest of the section is to show that this is indeed the case. That is, we will prove the following theorem.

Theorem 2 *With high probability*

$$s_i(u) = (1 \pm e_i)s_i, \qquad d_i(u,c) = (1 \pm e_i)d_i, \text{ and } \qquad \deg_i(u) = (1 \pm e_i)D_i \tag{14}$$

for all rounds i, colours c and vertices u, where the error factors e_i are defined by

$$e_i = C^i\sqrt{\frac{ke^{2k}\log n}{\Delta}},$$

for a constant $C > 1$, and $e_0 = 0$. "High probability" means that the probability that Equation (14) does not hold for any round, vertex or colour is $o(1)$, a quantity that goes to zero as the number of vertices grows.

The proof of the theorem is rather involved and will take several pages. Before plunging into the proof, we show that the assumptions $\Delta = \Omega(\log^{1+\delta} n)$ and $k = O(\log \Delta)$ imply that e_i's are $o(1)$'s, thereby ensuring correct termination. Since we will need the statement of the Theorem to be true with a small e_i for all rounds $i \leq ek + O(\log k)$, we would like

$$e_{ek+O(\log k)} = C_1^k \sqrt{\frac{\log n}{\Delta}} \ll 1,$$

which is just the condition $C_2^k \log n \ll \Delta$, for suitable constants C_1 and C_2. Assuming that $\Delta \geq \log^{1+\delta} n$, for any $\delta > 0$, this condition becomes $C_2^k \ll \Delta^{1+1/(1+\delta)}$, which is just that

$$k \leq \frac{\log \Delta}{\beta}$$

for an appropriately large, but constant β.

So, if the e_i's behave as advertised in Theorem 2, the statements concerning the running time and the correctness of the algorithm (i.e. Theorem 1) follow.

Although the actual computations differ, the proof of Theorem 2 for the dozing-off phase and for the trivial phase is conceptually the same. The proof is by induction on the round number i. The base case $i = 0$ holds with equality. The induction step consists of two parts. First, it is shown that the approximate equality holds in expectation. That is, we first show that the expectations are as promised:

$$\begin{aligned} \mathrm{Ex}[s_{i+1}(u)] &\sim s_{i+1}, & (15)\\ \mathrm{Ex}[d_{i+1}(u,c)] &\sim d_{i+1}, \text{ and} & (16)\\ \mathrm{Ex}[\deg_{i+1}(u)] &\sim D_{i+1} & (17) \end{aligned}$$

for all rounds i, vertices u, and colours c. Then it is shown that that the random variables $s_{i+1}(u)$, $d_{i+1}(u,c)$ and $\deg_{i+1}(u)$ are sharply concentrated around their expectations, i.e.

$$\begin{aligned} \mathrm{Ex}[s_{i+1}(u)] &\sim s_{i+1}(u), & (18)\\ \mathrm{Ex}[d_{i+1}(u,c)] &\sim d_{i+1}(u,c), \text{ and} & (19)\\ \mathrm{Ex}[\deg_{i+1}(u)] &\sim \deg_{i+1}(u). & (20) \end{aligned}$$

for all rounds i, vertices u, and colours c. The use of the asymptotic equality in place of explicit error factors is justified provided that the e_i's behave as stated in Theorem 2.

We now sketch the road map that will bring us to prove Theorem 2. In the next subsection, we shall prove that, assuming Equations (15)—(20), then Equations (15) through (17) still hold after the execution of one additional round of the algorithm. Then, in a separate subsection, again assuming that Equation (15)—(20) hold by induction, we shall show that after an additional round of the algorithm Equations (18) through (20) are also still valid. This does not quite prove the theorem because, round after round, the compounded error might grow too large. Therefore, in a final subsection we perfom the error analysis showing that the error can be kept under control. We feel that presenting the proof in this modular fashion offers the best compromise between the opposing needs of precision and intuition. Hopefully, the main thrust of the argument and the main ideas will emerge clearly, in spite of the unavoidable menagerie of cluttering technical details.

4.1 Deriving the recurrences

Next in our proof is to show that Equations (15)—(17) hold. The discussion will also clarify the genesis of the recurrences governing the behaviour of the algorithm. We argue separately for the dozing-off and trivial phase.

In what follows, we focus on an arbitrary round during the dozing-off phase and assume by induction that the ith version of Equation (14) holds. To simplify notation, s, d, and D will denote the values s_i, d_i, and D_i and s', d', and D' will denote the values s_{i+1}, d_{i+1}, and D_{i+1}. We will also omit explicit error factors like $(1 \pm e_i)$ and just write asymptotic equality.

By "u c–colours" we will refer to the event that vertex u colours itself succesfully with colour c during the current round, and by "c decays at u" we will refer to the event that colour c is deleted from u's palette at the end of the current round. The current palette of a vertex u is denoted by A_u.

In the rest of the subsection, we focus on an arbitrary uncoloured vertex u. During the dozing-off phase, $s < d$ and the vertices awake with probability $p = \frac{s}{d}$.

Lemma 1 *During the dozing-off phase, for each uncoloured vertex u and each colour c in u's palette,*

$$\Pr[u \ c\text{-}colours] \sim \frac{p}{s}\left(1 - \frac{p}{s}\right)^d \sim \frac{1}{d}\frac{1}{e}.$$

Proof Let N_{uc} be the set of u-neighbours which have colour c in their palettes. Recalling that, by definition, $p = \frac{s}{d}$ and the induction hypotheses $|A_u| \sim s$ and $\deg(u, c) = |N_{uc}| \sim d$, for all u and c,

$$\begin{aligned}
\Pr[u \ c\text{-colours}] &= \frac{p}{|A_u|} \prod_{v \in N_{uc}} \left(1 - \frac{p}{|A_v|}\right) \\
&= \frac{1 + o(1)}{d}\left(1 - \frac{(1 + o(1))}{d}\right)^{d(1+o(1))} \\
&= (1 + o(1))\frac{1}{d}\frac{1}{e}.
\end{aligned}$$

The last equality follows from Fact 1 by taking $A(n) := (1 + o(1))/d$ and $B(n) := (1 + o(1))d$. $\triangledown$

From the lemma, again using Fact 1 and the induction hypothesis, we see that,

$$\Pr[u \text{ colours}] = \sum_{c \in A_u} \Pr[v \ c\text{-colours}] \sim \frac{s}{d}\frac{1}{e} \tag{21}$$

where A_u is the current palette of u. Denoting by N_u the current set of uncoloured neighbours of u,

$$\text{Ex}[\deg'(u)] = \sum_{v \in N_u} (1 - \Pr[u \text{ colours}]) \sim D\left(1 - \frac{s}{d}\frac{1}{e}\right) = D',$$

since by induction $|N_u| = \deg(u) \sim D$. This shows that the expectation of $\deg'(u)$ is near the ideal value D', thereby establishing Equation (17). This was of course what suggested the recurrence

$$D' = D\left(1 - \frac{s}{d}\frac{1}{e}\right)$$

in the first place.

We now show that the palette sizes also have expectation asymptotically equal to their ideal values. As the neighbours of u colour themselves, the colours they have used are removed from

u's palette. Again denoting by N_{uc} the set of neighbours of u whose palettes contain c at the current round, we see that for each colour c in the palette,

$$\begin{aligned}\Pr[c \text{ decays at } u \mid u \text{ does not colour}] &= \Pr[\text{some } v \in N_{uc}\ c\text{-colours} \mid u \text{ does not colour}] \\ &= 1 - \Pr[\text{no } v \in N_{uc}\ c\text{-colours} \mid u \text{ does not colour}].\end{aligned}$$

(Recall that "c decays at u" means that colour c is removed from u's palette A_u.) The last event can be computed by considering the complementary event. The probability that neither of two neighbours c-colour is, in general, difficult to compute but, in the case where there are neither 3- nor 4-cycles, the 2-neighborhood of u is a tree. Thus we see that the events "x does not c-colour" and "y does not c-colour" are "essentially" independent, since c-colouring only depends on the (independent) tentative colour choices made by each vertex and its neighbours and since any two vertices x and y have no neighbours in common except u. The same is true for any number of neighbours of u. Therefore, from the induction hypothesis, using Lemma 1 and Fact 1,

$$\begin{aligned}\Pr[c \text{ decays at } u \mid u \text{ does not colour}] &\sim 1 - \prod_{v \in N_{uc}} \Pr[v \text{ does not } c\text{-colour} \mid u \text{ does not colour}] \\ &\sim 1 - \left(1 - \frac{1}{ed}\right)^d \\ &\sim 1 - \exp(-1/e).\end{aligned} \tag{22}$$

Then, again by induction,

$$\mathrm{Ex}[s'(u)] = \sum_{c \in A(u)} (1 - \Pr[c \text{ decays at } u]) \sim s\exp(-1/e)$$

which establishes Equation (15) and suggests that palette decay during the dozing-off phase is governed by the recurrence

$$s' = s\exp(-1/e).$$

Now we turn to the $d'(u,c)$'s. For each colour c, the graph G_c is defined as the graph induced by the vertices whose palettes contain c. The probability that a vertex u must be removed from G_c is, recalling Equations (21) and (22),

$$\begin{aligned}\Pr[u \text{ disappears from } G_c] &= \Pr[u \text{ colours}] + \Pr[c \text{ decays at } u \text{ and } u \text{ does not colour}] \\ &= \Pr[u \text{ colours}] + \Pr[c \text{ decays at } u \mid u \text{ does not colour}](1 - \Pr[u \text{ colours}]) \\ &\sim \frac{s}{d}\frac{1}{e} + \left(1 - \frac{s}{d}\frac{1}{e}\right)(1 - \exp(-1/e)) \\ &= 1 - \left(1 - \frac{s}{d}\frac{1}{e}\right)\exp(-1/e).\end{aligned}$$

Since, by induction,

$$\mathrm{Ex}[d'(u,c)] = \sum_{v \in N_{uc}} (1 - \Pr[u \text{ disappears from } G_c]) \sim d\left(1 - \frac{1}{d}\frac{1}{e}\right)\exp(-1/e),$$

suggesting that the random variables $d_i(u,c)$'s should be approximated well by the solution to the recurrence

$$d' = d\left(1 - \frac{s}{d}\frac{1}{e}\right)\exp(-1/e).$$

Therefore we see that Equations (15)–(17) hold throughout the dozing-off phase.

After the algorithm switches to the trivial phase, the recurrences for the ideal values take on new forms, reflecting the fact that $p_i = 1$ instead of s_i/d_i. Their derivations are conceptually the same, however.

Start again with the probability that a vertex colours itself with colour c. Namely,

$$\Pr[u \text{ } c\text{-colours}] \sim \frac{1}{s}\left(1 - \frac{1}{s}\right)^d \sim \frac{1}{s}e^{-d/s}.$$

Thus,

$$\Pr[u \text{ colours}] = \sum_{c \in A_u} \Pr[u \text{ } c\text{-colours}] \sim e^{-d/s}$$

Again both asymptotic equalities follow from Fact 1. This implies that the expected number of neighbours of u which remain uncoloured in the next round is asymptotically equal to $D(1 - e^{-d/s})$, which led to the recurrence

$$D' = D(1 - e^{-d/s}).$$

To derive the recurrence for the palette sizes we again use the fact that, when the girth is at least 5, the events "v c-colours" are essentially independent for any set of neighbours of u.

$$\begin{aligned}
\Pr[c \text{ decays at } u \mid u \text{ does not colour}] &= \Pr[\text{some } v \in N_{uc} \text{ } c\text{-colours} \mid u \text{ does not colour}] \\
&= 1 - \Pr[\text{no } v \in N_{uc} \text{ } c\text{-colours} \mid u \text{ does not colour}] \\
&\sim 1 - \prod_{v \in N_{uc}} \Pr[v \text{ does not } c\text{-colour} \mid u \text{ does not colour}] \\
&\sim 1 - \left(1 - \frac{1}{a}e^{-d/s}\right)^d \sim 1 - \exp\left\{-\frac{d}{a}e^{-d/s}\right\}.
\end{aligned}$$

Thus, the expected number of colours in u's palette which survive is asymptotically equal to $s \exp\left\{-(d/s)e^{-d/s}\right\}$ which led to the recurrence

$$s' = s \exp\left\{-\frac{d}{s}e^{-d/s}\right\}.$$

Finally, for the $d_i(u, c)$'s we have

$$\begin{aligned}
\Pr[u \text{ disappears from } G_c] &= \Pr[u \text{ colours}] + \Pr[c \text{ decays at } u \text{ and } u \text{ does not colour}] \\
&\sim e^{-d/s} + (1 - e^{-d/s})\left(1 - \exp\left\{-\frac{d}{s}e^{-d/s}\right\}\right) \\
&= 1 - \left(1 - e^{-d/s}\right)\exp\left\{-\frac{d}{s}e^{-d/s}\right\},
\end{aligned}$$

which led to

$$d' = d\left(1 - e^{-d/s}\right)\exp\left\{-\frac{d}{s}e^{-d/s}\right\}.$$

4.2 Concentration Results

As in the previous subsection, the current palette size, vertex degree and number of neighbours having a certain colour c in their palettes are denoted by s, D and d, respectively. In what follows we analyse one step of the algorithm assuming by induction that the input graph G satisfies the following:

- G is almost D-regular with girth at least four;
- For all vertices u of G, $|A_u| \sim s$ where A_u is the current palette of vertex u.
- For all colours c and vertices u, the number of u-neighbours which have colour c in their palette is $\deg(u,c) \sim d$, i.e. G_c is almost d-regular for all colours c.

Notice that we are considering the girth-4 case. That is, we will prove the concentration results directly for the most general case, paving the way for the extension of Section 5. Throughout, "$t_u = c$" denotes the event "vertex u's tentative color choice is c". We start with some preliminary facts that will come in handy later.

Definition 1 *Given a set of vertices A, $\textsc{up}(A)$ is the set of vertices in A that wake-up at the current round. When the context allows it, for notational convenience we shall denote $|\textsc{up}(A)|$ simply as $\textsc{up}(A)$.*

Definition 2 *For a vertex v and a colour c, let*

$$N_{vc} := \{w \in N_v : \ c \in A_w\},$$

i.e. the set of v-neighbours that have a particular colour c in their palettes.

The next fact says that the number of neighbours of a vertex u which wake-up is sharply concentrated around its expected value. The proof is a routine application of the Chernoff-Hoeffding bounds.

Fact 4 *During the dozing-off phase*

$$\mathrm{Ex}[\textsc{up}(N_u)] \sim \frac{D}{d}s$$

and

$$\Pr\left[|\textsc{up}(N_u) - \mathrm{Ex}[\textsc{up}(N_u)]| \leq \sqrt{3\varphi s\frac{D}{d}}\right] \leq 2e^{-\varphi}.$$

Proof $\textsc{up}(N_u) = B(N_u, p)$ where, by definition of the algorithm $p = \frac{s}{d}$ and, by the induction hypothesis $|N_u| \sim D$. The claim follows from Fact 2. $\bigtriangledown$

There is no need to prove Fact 4 for the trivial phase, for then $\textsc{up}(N_u) = N_u$.

Consider the set of neighbours of a vertex v which have a particular colour c in their palettes, i.e. N_{vc}. The next fact says that the number of such neighbours which wake-up is sharply concentrated around its expectation. Again, the proof is routine.

Fact 5

$$\mathrm{Ex}[\textsc{up}(N_{vc})] \sim s$$

and , for all $\varphi \geq 0$,

$$\Pr\left[|\textsc{up}(N_{vc}) - \mathrm{Ex}[\textsc{up}(N_{vc})]| \leq \sqrt{3\varphi s}\right] \leq 2e^{-\varphi}.$$

Proof $\textsc{up}(N_{vc}) = B(N_{vc}, p)$ where, by definition, $p = \frac{s}{d}$ and, by the induction hypothesis, $|N_{vc}| \sim d$ for all v and c. The claim follows from Fact 2. $\triangledown$

After these preliminaries we are set to prove the concentration result. For brevity sake, we will prove them only for the dozing-off phase. An essentially identical proof holds for the trivial phase. Intuitively, things improve during the trivial phase, for s grows larger relative to both D and d. The bounds will be strong enough to hold simultaneously for all vertices u, colours c, and rounds i until the termination condition is satisfied.

4.2.1 Concentration of vertex degrees

The theorem we want to prove here is the following.

Theorem 3 *For a vertex u let*

$$h_u := |\{v \in N_u : \ v \ \textit{colours}\}|.$$

Then, for $\varphi \geq 5 \log n$ and $n \geq 2$,

$$\Pr\left[|h_u - \mathrm{Ex}[h_u]| > 3\sqrt{\varphi \frac{sD}{ed}}\right] \leq 2e^{-\varphi}(2 + n + n^3).$$

We start with a definition.

Definition 3 *For a vertex v, let*

$$F_v := \{c \in A_v : \nexists w \in N_v, t_w = c\}$$

i.e. the set of colours in v's current palette which are not chosen tentatively by any v-neighbour. F_v is the set of colours which remain "free" for v after v's neighbours make their tentative colour choice.

The next fact asserts that, almost surely, the number of colours from u's palette not claimed as tentative colours by any neighbour is sharply concentrated around its expectation. Said differently, it states that, for each vertex v, the number of free colours is sharply concentrated around its expectation.

Lemma 2 *Let $f_v := |F_v|$. Then,*

$$\mathrm{Ex}[f_v] \sim \frac{s}{e}$$

and, for all $\varphi \geq 4 \log n$ and $n \geq 2$,

$$\Pr[|f_v - \mathrm{Ex}[f_v]| > 3\sqrt{\varphi s}] \leq 2e^{-\varphi}(1 + n^2).$$

Proof To prove the claim we shall use the $\textsc{mobv}$ as stated in Corollary 1. For a vertex w and a colour c, consider the events

$$\mathcal{B}_{wc} := (|\textsc{up}(N_{wc}) - \mathrm{Ex}[\textsc{up}(N_{wc})]| \leq \sqrt{3\varphi s})$$

and let

$$\mathcal{B} := \wedge_{wc} \mathcal{B}_{wc}.$$

By Fact 5 we have

$$\Pr[\overline{\mathcal{B}}] \leq 2n^2 e^{-\varphi}$$

for all $\varphi \geq 0$. First, we compute the expectation of f_v. Assuming $\mathcal{B}$, let

$$X_c := \begin{cases} 1 & \text{no neighbour in } \textsc{up}(N_{vc}) \text{ picks } c \text{ tentatively} \\ 0 & \text{otherwise} \end{cases}$$

so that

$$(f_v|\mathcal{B}) = \sum_{c \in A_v} X_c.$$

By the induction hypothesis and Facts 1 and 5,

$$\begin{aligned} \mathrm{Ex}[f_v] &= \mathrm{Ex}[f_v|\mathcal{B}]\Pr[\mathcal{B}] + \mathrm{Ex}[f_v|\overline{\mathcal{B}}]\Pr[\overline{\mathcal{B}}] \\ &\sim \mathrm{Ex}[f_v|\mathcal{B}] \\ &= \sum_{c \in A_v} \mathrm{Ex}[X_c] \\ &= \sum_{c \in A_v} \prod_{w \in \mathrm{UP}(N_{vc})} \Pr[t_w \neq c] \\ &= \sum_{c \in A_v} \prod_{w \in \mathrm{UP}(N_{vc})} \left(1 - \frac{1}{|A_w|}\right) \\ &= (1 + o(1))s \left(1 - \frac{1 + o(1)}{s}\right)^{s(1+o(1))} \\ &\sim s\,\frac{1}{e}. \end{aligned}$$

The X_c's are not independent, for if a v-neighbour $w \in N_{va} \cap N_{vb}$ wakes up then X_a and X_b will both be affected. Therefore we resort to Corollary 1 and study the variance of $(f_v|\mathcal{B})$. To estimate the variance of $(f_v|\mathcal{B})$ we query, in any order, the vertices in the set

$$B := \bigcup_c \textsc{up}(N_{vc}),$$

asking for their tentative colour choices. These are the only vertices whose tentative colour choices can belong to v's palette. For each such query, there are only two events to consider: either the tentative colour is in v's palette or it isn't. For $w \in B$, this happens with probability

$$\frac{|A_w \cap A_v|}{|A_v|}.$$

The maximum effect of the query is 1 since at most one colour can be affected. To estimate the global variance of this strategy, consider the following bipartite graph H (we should actually write H_v but we drop the subscript for notational convenience). The two sets of the bipartition are A_v and B. An edge $cw \in E(H)$ if and only if $c \in A_w$ and w wakes up. Assuming $\mathcal{B}$ implies that $d_H(c) = |\textsc{up}(N_{vc})| \sim s$ for all colours c, so that $|E(H)| \sim s^2$. Note also that, for all $w \in B$, $d_H(w) = |A_w \cap A_v|$. (In fact, these statements hold simultaneously for all vertices v and corresponding graphs H_v.) Then, for $w \in B$,

$$V(w) \leq \Pr[t_w \in A_v]c_w^2 = \frac{|A_w \cap A_v|}{|A_w|} = \frac{d_H(w)}{|A_w|} \sim \frac{d_H(w)}{s}$$

where, as remarked, $c_w = 1$ since w's tentative choice can affect at most one colour in A_v. Therefore

$$V(f_v|\mathcal{B}) = \sum_{w \in B} V(w) \sim \frac{|E(H)|}{s} \sim s.$$

Hence, invoking Corollary 1,

$$\Pr[|f_v - \mathrm{Ex}[f_v]| > 2\sqrt{\varphi s} + s\Pr(\overline{\mathcal{B}})] \leq 2e^{-\varphi}(1+n^2)$$

which implies, for $\varphi \geq 4\log n$ and $n \geq 2$,

$$\Pr[|f_v - \mathrm{Ex}[f_v]| > 3\sqrt{\varphi s}] \leq 2e^{-\varphi}(1+n^2).$$

$\bigtriangledown$

We can now prove Theorem 3, again using Corollary 1. Let

$$\mathcal{F}_v := (|f_v - \mathrm{Ex}[f_v]| \leq 3\sqrt{\varphi s})$$

and

$$\mathcal{F} := \bigwedge_v \mathcal{F}_v.$$

Then, from Lemma 2,

$$\Pr[\overline{\mathcal{F}}] \leq 2e^{-\varphi}(n+n^3).$$

Also, let

$$\mathcal{A} := \left(|\textsc{up}(N_u) - \mathrm{Ex}[\textsc{up}(N_u)]| \leq \sqrt{3\varphi s\frac{D}{d}}\right)$$

so that, from Fact 4,

$$\Pr[\overline{\mathcal{A}}] \leq 2e^{-\varphi}.$$

We shall study the variance of $\hat{h}_u := (h|\mathcal{A}\wedge\mathcal{F})$. The strategy is simply this: expose the tentative colour choices of vertices in $\textsc{up}(N_u)$, in any order. Assuming $\mathcal{F}$ we have that a u-neighbour v colours itself succesfully if and only if $t_v \in F_v$. By Lemma 2 and the induction hypothesis, this happens with probability

$$\frac{|F_v|}{|A_v|} \sim \frac{1}{e}.$$

The effect of every such query is 1, since v cann affect only itself. Therefore, the variance of this strategy is given by

$$V(\hat{h}_u) = \sum_{v\in \mathrm{UP}(N_u)} \Pr[t_v \in F_v]c_v^2 \sim \sum_{v\in \mathrm{UP}(N_u)} \frac{1}{e} \sim D\frac{s}{d}\frac{1}{e}. \tag{23}$$

Applying Corollary 1, together with the above gives

$$\Pr\left[|h_u - \mathrm{Ex}[h_u]| > 2\sqrt{\varphi\frac{D}{d}\frac{s}{e}} + D\Pr(\overline{\mathcal{A}\wedge\mathcal{F}})\right] \leq 2e^{-\varphi} + \Pr(\overline{\mathcal{A}\wedge\mathcal{F}}).$$

Plugging in the figures and recalling that $\varphi \geq 5\log n$ yields the claim. As remarked, a very similar argument holds for the trivial phase.

4.2.2 Concentration of palette sizes

Our aim here is to establish the following theorem.

Theorem 4 *For a vertex u let f_u be the number of colours that are removed from A_u after one round of the algorithm (note that we are reusing the symbol f_u, i.e. we are* not *referring to the f_u of Lemma 2). Let $\varphi \geq 5 \log n$ and $n \geq 2$. Then,*

$$\Pr[|f_u - \mathrm{Ex}[f_u]| > 3\sqrt{\varphi s}] < 2e^{-\varphi}(1 + n + n^3).$$

We shall prove the statement for the dozing-off phase of the algorithm. The arguments for the trivial phase are essentially the same, only the computations differ slightly.

Definition 4 *Let*

$$S := \{v \in N_u^1 : \ t_v \in A_u\}.$$

This is the set of u-neighbours whose tentative colour choices can affect u's palette.

Our first goal is to show that

$$|S| \sim \mathrm{Ex}[|S|]$$

with high probability, for all vertices u. Since there is a set S for every u, one should write S_u, but the subscript is dropped for sake of convenience.

Lemma 3 *For any $\varphi \geq 4 \log n$ and $n \geq 2$,*

$$\Pr[||S| - \mathrm{Ex}[|S|]| > 3\sqrt{\varphi s}] \leq 2e^{-\varphi}(1 + n^2).$$

Proof Notice that we will be able to establish the claim without actually computing the value of $\mathrm{Ex}[|S|]$ (however, it can be verified that $\mathrm{Ex}[|S|] \sim s$). Again, consider the events

$$\mathcal{B}_{uc} := |\mathrm{UP}(N_{uc}) - \mathrm{Ex}[\mathrm{UP}(N_{uc})]| \leq \sqrt{3\varphi s}$$

and

$$\mathcal{B} = \bigwedge_{uc} \mathcal{B}_{uc}.$$

By Fact 5,

$$\Pr[\overline{\mathcal{B}}] \leq 2n^2 e^{-\varphi}. \tag{24}$$

To establish that $|S|$ is sharply concentrated we shall study the variance of $(|S| \mid \mathcal{B})$ and invoke Corollary 1. At this point, to estimate the variance it is sufficient to expose, in any order, the tentative colour choices of vertices in the set

$$B := \bigcup_{c \in A_u} \mathrm{UP}(N_{uc})$$

i.e. the set of vertices which wake up and which palettes have nonempty intersection with A_u. Only these vertices can choose a tentative colour in A_u, i.e. $B \supseteq S$. Once a tentative colour choice of a vertex in B is exposed, there are only two relevant events: either the tentative colour belongs to u's palette or it doesn't. For $v \in B$, this happens with probability

$$\frac{|A_v \cap A_u|}{|A_u|}.$$

The effect is one since at most one colour can be affected. To estimate the variance of this strategy, define a bipartite graph H as follows (we should really write H_u but we drop the subscript). One side of the bipartition is A_u, the other side is the set B. The edges of H are defined as follows. Edge vc is inserted if and only if $v \in \text{UP}(N_{uc})$, i.e. if v wakes up and $c \in A_v$. By definition, for a $c \in A_u$, $\deg_H(c) = |\text{UP}(N_{uc})|$ and, for $v \in B$, $\deg_H(v) = |A_v \cap A_u|$. Since we assume that $\mathcal{B}$ holds, $\deg_H(c) \sim s$, for all colours (vertices) $c \in A_u$. This implies that

$$|E(H)| \sim s^2. \tag{25}$$

We now estimate the variance of all vertices in B. Recalling that the maximum effect c_v of every $v \in B$ is 1, for $v \in B$, we have

$$V(v) \leq \Pr[t_v \in A_u] c_v^2 = \frac{\deg_H(v)}{|A_v|} \sim \frac{\deg_H(v)}{s}$$

which, together with Equation (25), gives a total variance of

$$V(B) \sim \frac{1}{s} \sum_{v \in B} \deg_H(v) = \frac{|E(H)|}{s} \sim s.$$

Applying Corollary 1, together with Fact 5, gives

$$\Pr[||S| - \text{Ex}[|S|]| > 2\sqrt{\varphi s} + D \Pr(\overline{\mathcal{B}})] \leq 2e^{-\varphi} + \Pr(\overline{\mathcal{B}}).$$

From Equation (24), the hypothesis $\varphi \geq 4 \log n$ and the fact that $D \leq n$, it follows that

$$\Pr[||S| - \text{Ex}[|S|]| > 3\sqrt{\varphi s}] \leq 2e^{-\varphi}(1 + n^2). \tag{26}$$

$\bigtriangledown$

To conclude the proof of Theorem 4 we apply again Corollary 1 to the random variable $(f|\mathcal{S})$ where $\mathcal{S}$ is defined as follows. Let

$$\mathcal{S}_u := (||S| - \text{Ex}[|S|]| \leq 3\sqrt{\varphi s})$$

and

$$\mathcal{S} := \bigwedge_u \mathcal{S}.$$

It follows from Lemma 3 that

$$\Pr[\overline{\mathcal{S}}] \leq 2e^{-\varphi}(n + n^3). \tag{27}$$

Define a bipartite graph L whose sides of the bipartition are the set S and the set of "relevant" N_u^2-neighbours (as usual, it should be L_u but we drop the subscript). The set in question is

$$T := \bigcup_{v \in S} \text{UP}(N_{vt_v}).$$

This needs some explanation. After conditioning on $\mathcal{S}$ we know which u-neighbours woke up and which tentative colours they chose. If t_v is v's tentative colour then, whether or not v successfully colours, thereby depleting u's palette, only depends on vertices in $\text{UP}(N_{vt_v})$. Clearly then, $(f|\mathcal{S})$ is only a function of the colour choices of vertices in T. Assuming $\mathcal{S}$, we have that for each $v \in S$, $\deg_L(v) = |\text{UP}(N_{vt_v})| \sim s$, by Fact 5. This implies $|E(L)| \sim s^2$. Now, for $w \in T$,

$$V(w) \leq \frac{\deg_L(w)}{|A_w|} c_w^2 \sim \frac{\deg_L(w)}{s}$$

because $c_w = 1$ (at most one colour can be affected). Therefore

$$V(T) \leq \frac{1}{s} \sum_{w \in T} \deg_L(w) = \frac{|E(L)|}{s} \sim s.$$

Invoking Corollary 1 gives

$$\Pr[|f - \mathrm{Ex}[f]| > 2\sqrt{\varphi s} + 2se^{-\varphi}(n + n^3)] \leq 2e^{-\varphi} + 2e^{-\varphi}(n + n^3).$$

Using the fact that $\varphi \geq 5 \log n$, the above implies

$$\Pr[|f - \mathrm{Ex}[f]| > 3\sqrt{\varphi s}] \leq 2e^{-\varphi}(1 + n + n^3)$$

which concludes the proof of Theorem 4.

4.2.3 Concentration of d(u,c)

We now focus on $d_i(u, c)$. We shall adopt the same conventions and notation of the preceding subsections. We want to prove the following theorem. As before we shall give the proof only for the dozing-off phase, that for the trivial phase being essentially identical. Since the arguments are very similar to those already employed, the pace will be somewhat faster.

Focus on a vertex $u \in G_c$, the subgraph induced by all vertices that have c in their palettes. There are two different reasons why a u-neighbour v can disappear from G_c.

- v can colour itself during the current round;
- a v-neighbour can colour itself with c during the current round.

We already know that the first random variable– the number of neighbours that colour themselves at the current round– is sharply concentrated. More precisely it is within $3\sqrt{\varphi s}$ of its expectation with probability at least $1 - 2e^{-\varphi}(2 + n + n^2)$. We focus then on the second. The theorem we want to show is the following.

Theorem 5 *Let g_u be the number of u-neighbours such that colour c is removed from their palettes after the current round. Then, for $\varphi \geq 3 \log n$ and $n \geq 2$,*

$$\Pr[|g - \mathrm{Ex}[g]| > 3\sqrt{\varphi d}] \leq 2e^{-\varphi}(1 + n^2).$$

During the proof, keep in mind that we are working in the graph G_c. The difficulty of the proof is, loosely speaking, that g potentially depends on the tentative colour choices of as many as d^3 vertices. To bound the effect, as in the previous sections, we shall condition on the event $\mathcal{B}$, defined previously, which says that, for all vertices u and colours c, $\mathrm{UP}(N_{uc}) \sim s$. Recall that $\Pr[\overline{\mathcal{B}}] \leq 2n^2 e^{\varphi}$ (Lemma 5). We shall study the variance of $(g|\mathcal{B})$.

Focus on a vertex $w \in N^2_{uc}$; under what circumstances can it affect $(g|\mathcal{B})$? First, w must wake up, i.e. $w \in \mathrm{UP}(N^2_{uc})$. Second, after w wakes up, there are two possible, mutually exclusive outcomes: *(i)* none of its neighbours picks tentative colour c, or *(ii)* some w-neighbour picks tentative colour c. Define w to be *live* in the first case and *killed* in the second. Notice that if w is killed, it cannot affect g. Third, for the live vertices, only two events are relevant: either $t_w = c$ or $t_w \neq c$. Therefore, $(g|\mathcal{B})$ is a function of the tentative colour choices of the live vertices. To bound the variance, we consider the bipartite graph H induced by the set of u-neighbours that can lose c, i.e. N^1_{uc}, and the set of their neighbours that wake up, i.e. $\mathrm{UP}(N^2_{uc})$. Assuming

$\mathcal{B}$, it follows that, for all $v \in N^1_{uc}$, $\deg_H(v) \sim s$ and therefore $|E(H)| \sim sd$. The variance of a vertex $w \in \mathrm{UP}(N^2_{uc})$ is bounded from above by

$$V(w) \leq \Pr[w \text{ live}, t_w = c] \deg_H(w) \leq \Pr[t_w = c] \deg_H(w) \sim \frac{\deg_H(w)}{s}.$$

Therefore,

$$V(\mathrm{UP}(N^2_{uc})) \sim \frac{1}{s} \sum_{w \in \mathrm{UP}(N^2_{uc})} \deg_H(w) = \frac{1}{s}|E(H)| \sim d.$$

Invoking Corollary 1,

$$\Pr[|g - \mathrm{Ex}[g]| > 2\sqrt{\varphi d} + d \Pr(\overline{\mathcal{B}})] \leq 2e^{-\varphi} + \Pr(\overline{\mathcal{B}})$$

which, for $\varphi \geq 3 \log n$ and $n \geq 2$, implies

$$\Pr[|g - \mathrm{Ex}[g]| > 3\sqrt{\varphi d}] \leq 2e^{-\varphi}(1 + n^2).$$

4.3 Error Propagation

We can now go back to the problem of error propagation and derive the recurrence for the error terms e_i given in the Theorem. Assume by induction that at round i each of $a(u)$, $d(u,c)$, and $d(u)$ has a cumulative error factor of $(1 \pm e_i)$. In each of the expectation computations, we did one of the following operations a (small) constant number of times: we used the induction hypothesis, we used Fact 1, we made transformations like $1/(1+o(1)) = (1+o(1))$ and $e^{a(1+o(1))} = e^a(1 + o(1))$. All together, these operations imply that each expectation has an error factor $(1 \pm c_1 e_i)$, for some constant c_1.

Next we apply the concentration results to show that, with high probability, each random variable deviates from its mean by no more than a certain amount, which may be interpreted as another error factor. Of all the concentration results, the one with the worst error factor is the one given by Theorem 3. We will use that error factor, namely $c_2\sqrt{\varphi s \frac{D}{d}}$ uniformly. In fact, we will bound this factor in each round by the value in the very last round, which gives the worst error. Recalling Equations (7) and (8) which, respectively, give the smallest value of s and the worst (biggest) ratio D/d, we see that, choosing $\varphi = \Theta(\log n)$, the error is at most

$$c_2\sqrt{(ke^{2k} \log n)/\Delta}.$$

Thus,

$$e_{i+1} = c_1 e_i + c_2\sqrt{\frac{ke^k \log n}{\Delta}}.$$

Solving this recurrence gives the error term stated in Theorem refmainThm—namely,

$$e_i = C^i\sqrt{\frac{ke^{2k} \log n}{\Delta}},$$

for an appropriate constant $C > 1$.

During the trivial phase things work out in the same way, (in fact, things improve since the ratio s/D becomes better, i.e. larger), giving an error factor of the same form—namely

$$e_i = C^i\sqrt{\frac{ke^k \log n}{\Delta}},$$

where only the constant C differs.

This finishes the proof of the analysis of the algorithm down to the point where condition (5) holds and establishes Thereom 2. Thereafter, the analysis of the trivial algorithm completes the proof of Theorem 1 [13].

5 Girth 4

Now we allow the input graph to have cycles of length 4, but we still require it to be triangle-free. Having cycles of length 4 affects the analysis in the following fundamental way: it doesn't affect the probability that a vertex colours itself, but it does affect the probability that colours decay from its palette.

To see why, consider an extreme example, the complete bipartite graph $K_{d,d}$ with bipartition (A, B). If every vertex has a palette of size a, the probability that a particular colour c decays from the palette of a particular vertex $u \in A$ can be computed as follows, denoting with $\mathcal{E}$ the event "u does not colour."

$$\begin{aligned}
\Pr[c \text{ decays at } u \mid \mathcal{E}] &= 1 - \Pr[\text{no } v \in B \text{ } c\text{-colours} \mid \mathcal{E}] \\
&= 1 - \Pr[\text{some } w \in A \text{ picks } t_w = c \mid \mathcal{E}] - \Pr[\text{no vertex at all picks } c \mid \mathcal{E}] \\
&= 1 - \left(1 - \left(1 - \frac{p}{s}\right)^d\right) - \left(1 - \frac{p}{s}\right)^{2d} \\
&\sim \frac{1}{e} - \frac{1}{e^2} \approx 0.233
\end{aligned}$$

(Here we used $p = \frac{s}{d}$, as in the dozing-off phase.) This is strictly less than the probability of the same event in the absence of 4-cycles, namely $1 - \exp(-1/e) \approx 0.308$.

In general, when 4-cycles are allowed, the probability of colour decay is a function of the local topology. Since the graph may not have the same local topology everywhere, the rate of colour decay may vary from vertex to vertex and even from colour to colour, since it is the topology of G_c, the graph induced by vertices with colour c in their palettes, that counts.

One thing is true however, the presence of 4-cycles can only lower the probability of decay. This is proven in the following Lemma.

Lemma 4 *In an (almost) d-regular triangle-free graph with palettes of size (almost) a, for every vertex u and colour c,* $\Pr[c \text{ decays at } u \mid u \text{ does not colour}] \leq 1 - \exp\left\{-\frac{pd}{s}e^{-pd/s}\right\}$.

Proof The proof compares the probability that the colour c decays at vertex u in the given graph with the same probability in a graph with no 4-cycles.

Since only the first and second neighbourhoods of u can affect whether c decays at u, we ignore the rest of the graph. Similarly, edges between second neighbours play no role and may be ignored. With this local perspective, the graph consists of u, u's neighbours N_u, and second neighbours w, each of which is adjacent to a fixed set of first neighbours. If the graph were to contain no 4-cycles, the graph seen from this local perspective would simply be a depth 2 tree rooted at u.

Starting with the given graph, one can easily split each second neighbour into a set of degree 1 vertices (leaves), producing a 4-cycle-free graph (tree) where each first neighbour keeps the same degree. The other way around, one could start with just such a tree and merge leaves one after another into composite second neighbours to produce the given graph.

The proof follows this second process. We start with the tree where the degrees of the first neighbours of u, the root, are the same as in the given graph. We know already that for the tree the probability that c decays is asymptotic to $1 - \exp\left\{-\frac{pd}{s}e^{-pd/s}\right\}$. We show that as we merge leaves into composite second neighbours, the probability that c decays cannot increase. Thus, in the orignial graph, the probability that c decays is at most that in the 4-cycle-free graph.

We can make one further simplification: We can ignore all neighbours of u which have not chosen c as their tentative colour and all second neighbours which are only adjacent to such first neighbours. So, throughout we will condition on knowing that u does not colour and knowing the set X of neighbours of u which tentatively colour themselves c. This conditioning will be represented by the event $\mathcal{A}$.

If u's tentative colour is c, the fact that u does not colour tells us that X is not empty, and additionally that no vertex in X can colour itself and therefore that c does not decay at u, regardless of the topology of the second neighbourhood. Thus, we may assume that u's tentative colour was not c and hence that any of the vertices in X might colour themselves c and thereby cause c to decay at u.

So, consider an intermediate situation: some of the leaves have already been merged, and now we need to merge a leaf w_1, adjacent to neighbour v_1, with a (possibly already composite) second neighbour w_2. We can assume that w_2 is not adjacent to v_1, since otherwise this operation would create a duplicate edge, none of which exist in the graph given originally.

$$\begin{aligned}\Pr[c \text{ decays at } u \mid \mathcal{A}] &= \Pr\left[(v_1\ c\text{-colours}) \text{ or } \bigcup_{v\in X\setminus v_1} (v\ c\text{-colours}) \,\middle|\, \mathcal{A}\right] \\ &= \Pr\left[v_1\ c\text{-colours} \,\middle|\, \mathcal{A}\right] + \Pr\left[\bigcup_{v\in X\setminus v_1} (v\ c\text{-colours}) \,\middle|\, \mathcal{A}\right] \\ &\quad - \Pr\left[(v_1\ c\text{-colours}) \text{ and } \bigcup_{v\in X\setminus v_1} (v\ c\text{-colours}) \,\middle|\, \mathcal{A}\right]\end{aligned}$$

Considered individually, the merge operation changes neither the value of $\Pr[v_1\ c\text{-colours} \mid \mathcal{A}]$ nor that of $\Pr[\bigcup_{v\in X\setminus v_1} (v\ c\text{-colours}) \mid \mathcal{A}]$. As for

$$\begin{aligned}&\Pr\left[(v_1\ c\text{-colours}) \text{ and } \bigcup_{v\in X\setminus v_1} (v\ c\text{-colours}) \,\middle|\, \mathcal{A}\right] \\ &\qquad = \Pr\left[v_1\ c\text{-colours} \middle| \mathcal{A}\right] \Pr\left[\bigcup_{v\in X\setminus v_1} (v\ c\text{-colours}) \,\middle|\, (v_1\ c\text{-colours}) \text{ and } \mathcal{A}\right],\end{aligned}$$

again, the probability that v_1 c-colours remains unchanged. But the last probability increases, since in the new merged situation, knowing that v_1 c-colours tells us that $w_1 \equiv w_2$ does not tentatively choose c. This slightly increases the probability that one of the vertices in X adjacent to w_2 c-colours since it removes a possible obstruction.

Thus, after merging w_1 with w_2, the joint probability increases, decreasing the probability that c decays at u, as desired. Since the same is true after each merge operation, this probability in the given graph, where all leaves are merged as needed, is at most the probability of the same event in the 4-cycle-free graph. $\bigtriangledown$

Although it would seem to help, the 4-cycles make the analysis very difficult because they systematically destroy the regularity of the graphs G_c and the uniformity of the palette sizes.

One way to correct this unfortunate state of affairs is to apply the following approach, reminiscent of the agricultural policy of the European Union: Each vertex normalizes the probability of colour decay of each colour in its palette by artificially removing colours from its palette even if no neighbour has used the colour successfully. By doing this with the correct probability, the effective decay probability can be made to be exactly $1 - \exp\left\{-(pd/s)e^{-pd/s}\right\}$.

The algorithm given in the previous section is amended in the following way: each vertex maintains a complete description of its neighbourhood out to distance 2. This allows each vertex u to compute for each colour c the natural probability $p_{u,c}$ of decay. This is done before tentative colours are chosen. The round then proceeds as before. At the end, each colour c which was not naturally removed is artificially removed with probability

$$\frac{1-\exp\left\{-(pd/s)e^{-pd/s}\right\}-p_{u,c}}{1-p_{u,c}}.$$

Finally, each vertex broadcasts its updated palettes to all neighbours out to distance 2.

With this additional balancing mechanism, the expectations for the c-degrees and palette sizes are exactly as they were in the analysis of the previous section. The concentration results are also affected in no significant way by the independent artificial colour removals. We therefore get identical performance bounds, with the exception that the communication and space complexity of the algorithm are somewhat increased.

6 Conclusions

We have given a fast, distributed algorithm for vertex colouring Δ-regular triangle-free graphs using Δ/k colours, for $k = O(\log \Delta)$, as long as $\Delta \geq \log^{1+\delta} n$.

The analysis prompts several interesting questions. First of all, is the lower bound on Δ really necessary or is it an artifact of our analysis?

Another possible improvement would be the removal of the regularity assumption to deal with irregular graphs, perhaps with some condition on the minimum degree.

Another question is whether the balancing mechanism used in dealing with girth 4 graphs is necessary at all; it seems counter-intuitive to remove colours from the palette when no neighbour has used them.

Additionally, our methods can cope with a limited number of triangles in the graph. An interesting extension both algorithmic and existential would be to characterize classes of non-triangle-free graphs which admit Brooks-Vizing colourings. For instance, we know that line graphs can be coloured with roughly $\Delta/2$ colours by very fast distributed algorithms [10], but a more complete characterization would be desirable.

These questions might be tackled from a theoretical standpoint, but the best course of action might be a careful experimental study of the algorithm presented in this paper.

Acknowledgement

Thanks to Michał Karoński, Tomasz Łuczak and Andrzej Ruciński for comments helpful to the presentation of this result.

References

[1] B. Awerbuch, A.V. Goldberg, M. Luby, and S. Plotkin, Network decomposition and locality in distributed computing, in Proceedings of the 30th Symposium on Foundations of Computer Science (FOCS 1989), pages 364-369, IEEE, Research Triangle Park, North Carolina.

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

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

[4] B. Bollobás, Chromatic number, girth, and maximal degree, Discrete Math. 24 (1978), 311–314.

[5] S. Chaudhuri and D. Dubhashi, Probabilistic recurrence relations revisited, Theoretical Computer Science, to appear.

[6] D. Dubhashi, D.A. Grable and A. Panconesi, Nearly-optimal, distributed edge-colouring via the nibble method. Special issue of Theoretical Computer Science for ESA95, the 3rd European Symposium on Algorithms (ESA 95). To appear.

[7] D. Dubhashi and A. Panconesi, Concentration of Measure for Computer Scientist, draft available at `http://www.brics.dk/ ale/papers.html`

[8] D.A. Grable, A large deviation inequality for functions of independent, multi-way choices, Combinatorics, Probability and Computing, Nr. 7 (1998), 57–64.

[9] D.A. Grable, On random greedy triangle packing, Electronic Journal of Combinatorics 4 (1997), R11, pp.19.

[10] D.A. Grable and A. Panconesi, Nearly optimal distributed edge colouring in $O(\log\log n)$ rounds, Random Structures and Algorithms 10 (1997), 385–405. Extended Abstract in Proceedings of the Eight Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 97), pages 278-285, New Orleans.

[11] P. Hajnal and E. Szemerédi, Brooks coloring in parallel, SIAM Journal of Discrete Math. 3 (1990), 74–80.

[12] A.R. Johansson, Asymptotic choice number for triangle-free graphs, preprint, DIMACS, September 30, 1996.

[13] Ö. Johansson, Personal communication, May 1997.

[14] N. Karchmer and J. Naor, A faster parallel algorithm to color a graph with Δ colors, Journal of Algorithms 9 (1988), 83–91.

[15] H.J. Karloff, An NC-algorithm for Brooks theorem, Theoretical Computer Science 68(1) (1989), 89–103.

[16] R.M. Karp, Probabilistic recurrence relations, in Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC 91), pages 190–197, New Orleans.

[17] J.H. Kim, On Brooks' Theorem for sparse graphs, Combinatorics, Probability and Computing 4 (1995), 97–132.

[18] M. Luby, Removing randomness in parallel without processor penalty, Journal of Computer and System Sciences, 47(2) (1993), 250–286.

[19] C. McDiarmid, On the method of bounded differences, In *Surveys in Combinatorics*, J. Siemons Ed., London Mathematical Society Lecture Note Series 141 (1989), 148–188

[20] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.

[21] A. Panconesi and A. Srinivasan, The local nature of Δ-coloring and its algorithmic applications, Combinatorica 15(2) (1995), 255–280.

[22] A. Panconesi and A. Srinivasan, On the complexity of distributed network decomposition, Journal of Algorithms 20 (1996), 356–374.

[23] J.M. Steele, Probability theory and combinatorial optimization, CBMS-NSF 69 of Regional Conference Series in Applied Mathematics, SIAM