Nearly optimal distributed edge colouring in $O(\log \log n)$ rounds

David A. Grable*
Institut für Informatik
Humboldt-Universität zu Berlin
Unter den Linden 6
D-10099 Berlin Germany
grable@informatik.hu-berlin.de

Alessandro Panconesi†
Institut für Informatik
Freie Universität Berlin
Takustraße 9
D-14195 Berlin Germany
ale@inf.fu-berlin.de

February 1996

Abstract

An extremely simple distributed randomized edge colouring algorithm is given which, for any positive constants ε and c and a graph G with minimum degree $\Omega(n^{c/\log \log n})$, produces with high probability a proper edge colouring of G using $(1+\varepsilon)\Delta(G)$ colours in only $O(\log \log n)$ communication rounds.

1 Introduction

The edge colouring problem is a much studied problem in the theory of algorithms, graph theory, and combinatorics, whose relevance to computer science stems from its numerous applications to scheduling and resource allocation problems [5, 10, 12, 16, 17, 11, 19, among others]. Given an input graph, the problem consists in finding a *proper colouring* using as few colours as possible. A proper colouring is an assignment of colours to the edges so that no two incident edges have the same colour.

In this paper we give an extremely fast, distributed, randomized, nearly optimal algorithm for edge colouring. "Nearly optimal" means that the number of colours used is $(1 + \varepsilon)\Delta$, where Δ denotes the maximum degree of the input graph and $\varepsilon > 0$ is any arbitrarily small, but fixed, positive real. The algorithm is randomized in that it is allowed to call a random number generator which can generate uniformly distributed random integers in any interval. The algorithm may fail to find a proper colouring, but we show that this almost never happens, *i.e.* the probability of failure is $o(1)$, a function which goes to 0 as n goes to infinity.

One of the main features of the algorithm is that it is arguably the simplest randomized edge colouring algorithm one can consider. Each edge $e = uv$ is initially given a palette of $(1 + \varepsilon) \max\{\deg(u), \deg(v)\}$ colours. The computation takes place in rounds. In each round,

*Supported by Deutsche Forschungsgemeinschaft project number Pr 296/4-1.
†Supported by an Alexander von Humboldt Research Fellowship.

each uncoloured edge independently picks a tentative colour uniformly at random from its current palette. If no neighbouring edge picks the same colour, it becomes final. Otherwise, the edge tries again in the next round. At the end of each round the palettes are updated in the obvious way: colours successfully used by neighbouring edges are deleted from the current palette.

The algorithm succeeds with high probability on any graph, regardless of its structure, as long as certain minimum degree conditions are met. If the degrees are such that every edge e's initial palette has

$$a_0(e) \gg \log n$$

colours (recall that $a_0(uv) = \max\{\deg(u), \deg(v)\}$ and that $f \gg g$ means $g/f = o(1)$), we can show that the algorithm colours the graph within $O(\log n)$ rounds. But more remarkably, if there is a fixed constant $c > 0$ such that for every edge e

$$a_0(e) = \Omega\left(n^{c/\log\log n}\right)$$

then the algorithm colours the graph within $O(\log\log n)$ rounds.

It is apparent that the algorithm is *distributed*—that is, each edge only needs to exchange information with its neighbours. More precisely, the algorithm can be implemented in the standard *synchronous, message-passing* distributed model of computation. Here, a distributed network (or architecture) is modelled as an undirected graph. The vertices of the graph correspond to processors and edges correspond to bi-directional communication links. The network is synchronous in the sense that computation takes place in a sequence of *rounds*; in each round, each processor reads messages sent to it by its neighbours in the graph, does any amount of local computation, and sends messages back to each of its neighbours. The time complexity of a distributed algorithm is then given by the number of rounds needed to compute the desired function.

In the description of the algorithm it is implicit that each edge is a processor. It is easy to see, however, that each step of the algorithm can be implemented in the distributed model in a constant number of rounds. Thus our algorithm enables a distributed network to edge colour itself. Besides being of theoretical interest, such an algorithm has applications to real-life parallel architectures [7, 11].

Note that, unlike the PRAM, the distributed model has no shared memory, something that makes these two models radically different and, in fact, complementary. In a PRAM, each processor can communicate with any other processor in constant time via the shared memory. Hence, the cost of communication is completely neglected and only computation is charged for. In the distributed model the opposite happens; computation is not charged for, but sending messages is expensive. The time required to send a message from one processor to another is proportional to the minimum distance in the graph between sender and receiver. In particular, if a distributed protocol runs in t steps, a processor can communicate only with other processors at distance t or less and therefore all computation must be "local". A message-passing network models the fact that in real distributed or parallel architectures the cost of sending a message is order of magnitudes higher than the cost of performing local computation.

We note that the local computations required by our algorithm are very simple and require only $O(1)$ time in the PRAM model. Simulating the message passing requires an extra $O(\log n)$ factor, however.

Previous Work and Our Solution.

Clearly, if a graph G has maximum degree Δ then at least Δ colours are needed to properly edge colour it. A classical theorem of Vizing shows that $\Delta + 1$ colours are always sufficient and the proof is actually a polynomial-time algorithm to compute such a colouring (see for example [5]). Interestingly, given a graph G, it is NP-complete to decide whether it is Δ or $\Delta+1$ edge colourable, even for cubic or regular graphs [10, 8].

Efforts at parallelizing Vizing's theorem have failed; the best PRAM algorithm known is a randomized algorithm by Karloff and Shmoys that computes an edge colouring using very nearly $\Delta + \sqrt{\Delta} = (1 + o(1))\Delta$ colours in $O(\log n)$ time [12]. This algorithm can be derandomized using standard derandomization techniques [4, 18]. Whether $(\Delta + 1)$-edge colouring is P-complete is an open problem.

In the distributed setting the first non-trivial result was a randomized algorithm by Panconesi and Srinivasan which, with high probabilty, uses roughly $1.58\Delta + \log n$ colours and runs in $O(\log n)$ time [19]. For the interesting special case of bipartite graphs Lev, Pippinger, and Valiant show that Δ-colourings can be computed in NC, but this is provably impossible in the distributed model of computation even if randomness is allowed (see [16, 20]). The result of Panconesi and Srinivasan was greatly improved by Alon [1] and Dubhashi and Panconesi [6] who showed how to compute nearly optimal colourings in $O(\log n)$ rounds with high probability. These solutions are based on a strategy known as the Rödl Nibble, a powerful probabilistic technique which has been used with great success for a large variety of asymptotic packing, covering, and colouring problems [2, 13, 14, 15, 21, 22, among others]. Our result shows that in order to get asymptotically optimal edge colourings, such semi-random techniques are unnecessary.

The main feature of our algorithm, however, is its vastly improved running time. An interesting open question is whether nearly optimal edge colourings can be computed as fast deterministically. Our solution compares favourably to earlier ones also in terms of simplicity, not only because of the utter simplicity of the algorithm but also because it makes no use of a separate "brute force" routine at the end of the recursion.

The rest of this paper is devoted to the analysis of the algorithm described above. We first deal with the regular case, where all vertices have the same degree, and then show how to remove this assumption.

2 A Large Deviation Inequality

A key ingredient of our proof is a large deviation inequality for functions of independent random varaibles. The one we use is a generalization of a result of Alon, Kim, and Spencer [3], which the first author recently developed. Please see [9] for a proof, a more general result, and further discussion.

Assume we have a probability space generated by independent random variables X_i (choices), where choice X_i is from the set A_i, and a function $Y = f(X_1, \ldots, X_n)$ on that probability space. Consider a query game in which we try to determine the value of Y by making queries of the form "What was the i-th choice?" to a truthful oracle. A *strategy* can be expressed as a decision tree whose internal nodes designate queries to be made and whose leaves designate final values for Y.

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

$$v_q = \sum_{a \in A_i} p_{i,a} \mu_{q,a}^2,$$

where

$$p_{i,a} = \Pr[\text{choice } i \text{ was } a]$$

and

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

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 an upper bound on c_q we often take the maximum amount which Y can change if choice i is changed, but all other choices remain the same.

A *line of questioning* 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 is the maximum variance over all lines of questioning. The use of the term variance is meant to be suggestive: the variance of a strategy for determining Y is an upper bound on the variance of Y.

Proposition 1 *If there is a strategy for determining Y with variance at most V then*

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

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

When applying this Proposition, to determine the variance of each query, we must bound the sum $\sum p_{i,a}\mu_{q,a}^2$. There are two "standard" ways to do this. If we only know that the μ's differ by at most c_q, the best we can say is that

$$v_q = \sum_{a \in A_i} p_{i,a}\mu_{q,a}^2 \leq c_q^2/4. \tag{1}$$

If we know, in addition, that the μ's take on two values, one corresponding to choices with total probability p and the other to choices with total probability $1-p$, then we know that

$$v_q = \sum_{a \in A_i} p_{i,a}\mu_{q,a}^2 \leq p(1-p)c_q^2. \tag{2}$$

Both of these bounds can be shown using elementary, but non-trivial, computations, which we omit.

3 Analysis: Overview

The intuition behind the analysis is as follows. The algorithm generates a sequence of graphs $G_0, G_1, \ldots, G_z$, where G_0 is the input graph G and G_i, the input of round i, is the subgraph induced by the edges which are still uncoloured. Each edge $e = uv$ has initial palette $A_0(e) = \{1, \ldots, (1+\varepsilon)\max\{\deg(u), \deg(v)\}\}$ and at round i the palette $A_i(e)$ is composed of e's initial palette minus the colours used by its coloured neighbours.

At round i, each edge e has a probability $p_i(e)$ of being coloured. In the regular case (but similar considerations apply to the general case) these probabilities are sharply concentrated around a value p_i. The initial value of p_i is (very nearly) e^{-2} and, as the algorithm proceeds, p_i very rapidly approaches 1. Intuitively, this is due to the fact that the size of the intersection of the palettes of any two neighbouring edges shrinks at a much faster rate than the size of the palettes themselves. This makes the likelihood of conflict, *i.e.* of two neighbouring edges picking the same tentative colour, very unlikely and in fact soon negligible.

Perhaps the crucial fact about the analysis is to show that, despite the interaction along the edges of the graph, the colour palettes behave "as if" they were random sets of the original palettes, evolving independently of one another. In other words, one has to show that the correlation introduced by the graph is negligible. Similarly, one needs to show that G_i is "essentially" a random subgraph of the original graph.

We divide the analysis of the algorithm into three phases. During the first phase, we control the vertex degrees, palette sizes, and size of palette intersections quite carefully in order to get to the point were these quantities are small enough that p_i is approaching 1 sufficiently quickly that the vertex degrees are decreasing doubly exponentially. This requires only constantly many rounds.

In the second phase, the analysis simplifies considerably, as we need only bound the vertex degrees from above. Surprisingly, the vertex degrees decrease at a doubly exponential rate. This phase lasts until the vertex degrees are of order $\log n$, at which point they no longer have sufficiently good statistical properties. This phase lasts at most $O(\log\log n)$ rounds.

In the last phase, we take a global view and consider all remaining $O(n \log n)$ edges. Although there may be large local deviations, the total number of edges still decreases at a doubly exponential rate (assuming the original palettes were large enough). It therefore takes only $O(\log\log n)$ rounds to colour all remaining edges.

Consider now the regular case (we will comment on the irregular case in Section 7). We wish to colour a d_0-regular graph with $a_0 = (1+\varepsilon)d_0$ colours. Define α as $\varepsilon/(1+\varepsilon)$, so that

$$(1-\alpha)a_0 = d_0.$$

We assume that α is a positive constant less than $1/10$ or, equivalently, that ε is a positive constant less than $1/9$. As noted in the introduction, we also must assume that $a_0 = \Omega\left(n^{c/\log\log n}\right)$, for some positive constant c. The constant c will be reflected in the running time of phase III of the analysis and can be arbitrarily small (but must be fixed).

4 Analysis: Phase I

The first phase of the analysis will follow the algorithm for s steps, s a constant, until

$$\deg_s(v) \le \frac{\alpha^2}{4} a_0 \tag{3}$$

for every vertex v. Unfortunately the analysis is fairly complicated and is only valid for a constant number of steps. Afterwards though, the situation will be such that the simpler analyses of phases II and III can be used. We will show that with high probability

$$\begin{aligned}
\deg_i(v) &= d_i\,(1+o(1)), && (4)\\
|A_i(e)| &= a_0\left(\frac{\alpha a_0 + d_i}{a_0}\right)^2 (1+o(1)), \text{ and} && (5)\\
|A_i(e)\cap A_i(f)| &= a_0\left(\frac{\alpha a_0 + d_i}{a_0}\right)^3 (1+o(1)) && (6)
\end{aligned}$$

for every vertex v, incident edges e and f, and integer $0 \le i \le s$, where d_i is defined by the recurrence

$$d_{i+1} = d_i\left(1 - \exp\left\{\frac{-2d_i}{\alpha a_0 + d_i}\right\}\right). \tag{7}$$

The key to deciphering equations (4) through (6), as well as many other equations used in this paper, is to notice that they describe quantities which are evolving "as if" they were truly random, independently evolving subsets of some original sets. For instance, equation (5) is what we would expect if the palettes $A_i(e)$ were the intersection of two truly random subsets of size $a_0 - (d_0 - d_i)$ of the original palettes. To see this, suppose that we start with two identical sets A and B of size n and, after deleting k colours independently from each set, we ask what the size of their new intersection is. The new expected size would be

$$(|A \cap B| - (|A| - |A'|))\frac{|B'|}{|B|} = \frac{(n-k)^2}{n}$$

Given that the colours are deleted independently, one can apply the standard Chernoff bounds and show that the true size of the intersection is sharply concentrated around its expectation. This is what equation (5) is essentially saying once we plug in the values $A = B = A_0(e)$, $|A_0(e)| = a_0$, $d_0 = a_0(1-\alpha)$ and $k = d_0 - d_i$. The edges of the graph introduce a correlation, however. The crux of the analysis is to show that this effect is negligible, *i.e.* the palettes and the graph evolve "almost" as truly random subsets of the original palettes and a truly random subgraph of the original graph, respectively.

For notational convenience, we define

$$a_i = a_0 \left(\frac{\alpha a_0 + d_i}{a_0}\right)^2 \qquad \text{and} \qquad p_i = \exp\left\{\frac{-2d_i}{\alpha a_0 + d_i}\right\},$$

so that, roughly, a_i is the size of an edge's palette and p_i is the probability that an edge succeeds in colouring itself when going from G_i to G_{i+1}.

Note that in order that this analysis be valid for more than a constant number of steps, we would have to determine the $o(1)$ terms in (4) through (6). While this could be done, it would further complicate the already difficult proof. We prefer to accept the current situation and use the simpler phase II and III analyses for the rounds beyond s.

Equations (4) through (6) are proved by induction with the help of the following lemma. Roughly, it indicates that what happens at one vertex is nearly independent of what happens at another. In particular, this implies that the edge palettes evolve as nearly independent random subsets of the initial palette. With a slight abuse of notation we use $A_i(u)$ to denote the (implicitly defined) palette of colours which, at round i, are available around a vertex u (*i.e.* not used by any edge incident with u).

Lemma 2 *Consider the step from G_i to G_{i+1} and fix three distinct vertices u, v, and w. Let X be a set of colours such that $|X| \gg \sqrt{d_0 \ln n} \gg \ln n$.*

(i) If $X \subseteq A_i(u)$ then the number of edges e incident with u which succeed in colouring themselves from X is $|X|\left(\frac{p_i d_i}{\alpha a_0 + d_i}\right)(1+o(1))$.

(ii) If $X \subseteq A_i(u) \cap A_i(v)$ then the number of pairs of edges e and f incident with u and v, respectively, which succeed in colouring themselves with the same colour from X is $|X|\left(\frac{p_i d_i}{\alpha a_0 + d_i}\right)^2 (1+o(1))$.

(iii) If $X \subseteq A_i(u) \cap A_i(v) \cap A_i(w)$ then the number of triples of edges e, f, and g incident with u, v, and w, respectively, which succeed in colouring themselves with the same colour from X is $|X|\left(\frac{p_i d_i}{\alpha a_0 + d_i}\right)^3 (1+o(1))$.

Before proving the lemma, we'll show that the $i+1$ versions of equations (4) through (6) follow. Of course, the $i=0$ case holds even without the $(1+o(1))$ factor.

To determine the new degree of a vertex, we use Lemma 2(i). By induction, a vertex v has $\deg_i(v) = d_i(1+o(1))$ and there are $|A_i(v)| = a_0 - (d_0 - d_i(1+o(1))) = (\alpha a_0 + d_i)(1+o(1))$ colours which have not been used by any edge incident with v. With $X = A_i(v)$,

$$\deg_{i+1}(v) = \deg_i(v) - |X|\left(\frac{p_i d_i}{\alpha a_0 + d_i}\right)(1+o(1)) = (d_i - p_i d_i)(1+o(1)) = d_{i+1}(1+o(1)).$$

Now consider a fixed edge $e = uv$. Assuming that e does not colour itself, its new palette contains those colours not successfully used at either u or v. Using parts (i) and (ii) of Lemma 2 with $X = A_i(e)$, inclusion-exclusion gives

$$\begin{aligned} |A_{i+1}(e)| &= |A_i(e)| - 2|A_i(e)|\left(\frac{p_i d_i}{\alpha a_0 + d_i}\right)(1+o(1)) + |A_i(e)|\left(\frac{p_i d_i}{\alpha a_0 + d_i}\right)^2 (1+o(1)) \\ &= a_0\left(\frac{\alpha a_0 + d_{i+1}}{a_0}\right)^2 (1+o(1)), \end{aligned}$$

by induction and (7). Similarly, with $X = A_i(e) \cap A_i(f)$, we have

$$\begin{aligned} |A_{i+1}(e) \cap A_{i+1}(f)| &= |A_i(e) \cap A_i(f)|\left(1 - 3\left(\tfrac{p_i d_i}{\alpha a_0 + d_i}\right) + 3\left(\tfrac{p_i d_i}{\alpha a_0 + d_i}\right)^2 - \left(\tfrac{p_i d_i}{\alpha a_0 + d_i}\right)^3\right)(1+o(1)) \\ &= a_0\left(\frac{\alpha a_0 + d_{i+1}}{a_0}\right)^3 (1+o(1)). \end{aligned}$$

Proof of Lemma 2. The statement of the lemma follows from four additional facts, which we prove by induction, and from the induction hypothesis $|A_i(e)| = a_i(1+o(1))$, for every edge $e \in G_i$. This is valid since this lemma is used only in the proof by induction that $A_{i+1}(e) = a_{i+1}(1+o(1))$.

These facts concern a specific, fixed colour γ. Consider the subgraph of G_i induced by the vertices which have no incident edge which has been coloured γ. Call this subgraph $G_{i,\gamma}$. Clearly, the edges of $G_{i,\gamma}$ are exactly those whose palettes contain γ. The four facts are as follows. For every vertex $u \in G_{i,\gamma}$,

$$\deg_{i,\gamma}(u) = d_{i,\gamma}(1+o(1)), \tag{8}$$

where $d_{i,\gamma} = d_i\left(\frac{\alpha a_0 + d_i}{a_0}\right)$, and for every three disjoint edges e, f, and g in $G_{i,\gamma}$,

$$\Pr[e \text{ colours itself } \gamma] = \frac{1}{a_i}\exp\left\{\frac{-2d_{i,\gamma}}{a_i}\right\}(1+o(1)), \tag{9}$$

$$\Pr[e \text{ and } f \text{ colour themselves } \gamma] = \frac{1}{a_i^2}\exp\left\{\frac{-4d_{i,\gamma}}{a_i}\right\}(1+o(1)), \text{ and} \tag{10}$$

$$\Pr[e, f \text{ and } g \text{ colour themselves } \gamma] = \frac{1}{a_i^3}\exp\left\{\frac{-6d_{i,\gamma}}{a_i}\right\}(1+o(1)). \tag{11}$$

Note that $d_i(\alpha a_0 + d_i)/a_0$ is the "right" value for $d_{i,\gamma}$. Indeed, if the graph and palettes were evolving in a truly random fashion the probability that colour γ would belong to an (implicit) vertex palette after removal of $d_0 - d_i$ colours would be

$$\frac{|A_0(u)| - (d_0 - d_i)}{|A_0(u)|} = \frac{\alpha a_0 + d_i}{a_0}$$

for all γ and u. Then, the expected number of u-neighbours still retaining γ would be $d_i(\alpha a_0 + d_i)/a_0$i, for all u and γ.

Equations (9) through (11) follow from our inductive assumption $|A_i(e)| = a_i(1 + o(1))$ and equation (8), which is also proved by induction. We prove (9) through (11) first. Equation (9) is pretty easy. Fix an edge e. The only relevant edges are those in $G_{i,\gamma}$ and each edge picks γ as its tentative colour with probability $1/a_i(1 + o(1))$, since each edge has a palette of $a_i(1 + o(1))$ colours. Edge e succeeds in colouring itself if it, but none of the incident edges, tentatively picks γ. This happens with probability

$$\frac{1}{a_i}\left(1-\frac{1}{a_i}\right)^{2d_{i,\gamma}}(1+o(1)) = \frac{1}{a_i}\exp\left\{\frac{-2d_{i,\gamma}}{a_i}\right\}(1+o(1)).$$

Equation (10) is more or less the same. Fix edges e and f. They are disjoint and so have at most 4 incident edges in common. Therefore, the probability that they both succeed is

$$\frac{1}{a_i^2}\left(1-\frac{1}{a_i}\right)^{4d_{i,\gamma}}(1+o(1)) = \frac{1}{a_i^2}\exp\left\{\frac{-4d_{i,\gamma}}{a_i}\right\}(1+o(1)).$$

Equation (11) is proved similarly. The only thing to note is that there are at most 12 incident edges in common.

Now we make the induction step in the proof of equation (8). (In the $i = 0$ case, it holds even without the $(1 + o(1))$.) We assume the equation holds as shown and prove the $i + 1$ version. It is helpful to think of the transition from $G_{i,\gamma}$ to $G_{i+1,\gamma}$ as occurring in two steps. First only consider the effect of edges being coloured γ. This causes vertices and their surrounding edges to disappear. Then consider the effect of edges being coloured with other colours. This causes the individual successful edges to disappear. Although splitting the transition into two steps increases the length of the proof slightly, it helps clarify the differences in the two effects.

Fix a vertex u. In the first step some of the neighbours of u disappear, taking with them the connecting edge, and in the second step some more of the edges incident to u disappear. The probability that a given neighbour v has a successful edge (and therefore disappears) is $d_{i,\gamma}$ times the probability that a particular edge succeeds with γ, since these are disjoint events. From equation (9) and the other assumptions this is

$$\frac{d_{i,\gamma}}{a_i}\exp\left\{\frac{-2d_{i,\gamma}}{a_i}\right\}(1+o(1)) = \frac{d_i}{\alpha a_0 + d_i}\exp\left\{\frac{-2d_i}{\alpha a_0 + d_i}\right\}(1+o(1)) = \frac{p_i d_i}{\alpha a_0 + d_i}(1+o(1)).$$

Therefore the expected number of u's neighbours which disappear is

$$d_{i,\gamma}\frac{p_i d_i}{\alpha a_0 + d_i}(1+o(1))$$

Let Y be the number of u's neighbours which survive this first step. We've just shown that

$$\mathrm{Ex}[Y] = d_{i,\gamma}\left(1 - \frac{p_i d_i}{\alpha a_0 + d_i}\right)(1+o(1)) = d_{i,\gamma}\left(\frac{\alpha a_0 + d_{i+1}}{\alpha a_0 + d_i}\right)(1+o(1)).$$

Next we'll use the large deviation inequality to show that Y is with high probability equal to its expectation times $(1 + o(1))$. The only thing that was important in determining the value of Y was knowing which edges tentatively chose γ. Another way to say this is that the underlying probability space for the first step is formed by the independent answers to the question "did edge e tentatively pick γ?" The probability of Yes is $p_e = 1/|A_i(e)| = 1/a_i(1 + o(1))$.

Here's a strategy for determining Y: first look at all of the edges incident with all of the neighbours of u. This is $d_{i,\gamma}^2(1 + o(1))$ queries. Each query can affect the disappearance of at most 4 neighbours of u (and this only if e joins two neighbours of u which have only one other Yes edge which itself joins two neighbours of u), so the variance of each query is at most $16/a_i(1 + o(1))$.

The neighbours with exactly one edge which tentatively picked γ may or may not survive, but if the number of incident edges which tentatively pick γ is not 1, the vertex certainly survives. For each of those vertices still in question, look at all edges incident with the other endpoint of the one edge tentatively coloured γ. All of this requires at most $d_{i,\gamma}^2(1 + o(1))$ additional queries. These queries can affect at most 2 neighbours of u (and that only if the edge creates a five cycle), so the variance of each query is at most $4/a_i(1 + o(1))$.

Therefore the total variance is at most $20d_{i,\gamma}^2/a_i(1 + o(1))$. Proposition 1 then implies that

$$\Pr\left[|Y - \mathrm{Ex}[Y]| > 4\sqrt{5\varphi d_{i,\gamma}^2/a_i(1 + o(1))}\right] \leq 2e^{-\varphi}$$

for all $0 < \varphi < (5/4)d_{i,\gamma}^2/a_i(1 + o(1))$. We will require that the exceptional probability be $2/n^4$ so that we can accommodate such an application of the large deviation inequality per edge per colour per round. In any case, this forces $\varphi = 4\ln n$. The condition that $d_0 \gg \ln n$ and the fact that during Phase I d_i, $d_{i,\gamma}$, and a_i all have order d_0 (since α is a constant) imply that the side condition is satisfied and that Y's deviation is negligible compared with its expectation. Therefore, with high probability,

$$Y = \mathrm{Ex}[Y](1 + o(1)) = d_{i,\gamma}\left(\frac{\alpha a_0 + d_{i+1}}{\alpha a_0 + d_i}\right)(1 + o(1)).$$

Turning to the second step of the transition, how many of these Y edges incident with u survive this step? That is, how many of them manage not to colour themselves with some other colour?

We already know from the first step which edges tentatively chose γ. Since the number of edges incident with vertex v which chose γ is binomially distributed with mean $d_{i,\gamma}/a_i$, the Chernoff bounds easily imply that this number is asymptotically negligible to $d_{i,\gamma}$ with probability sufficiently close to one.

Thus, the probability that a given edge e successfully colours itself with a colour other than γ is the sum of disjoint events:

$$\sum_{\delta \in A_i(e)\setminus\{\gamma\}} \frac{1}{a_i}\left(1 - \frac{1}{a_i}\right)^{2d_{i,\delta}(1-o(1))}(1 + o(1)) = \exp\left\{\frac{-2d_{i,\gamma}}{a_i}\right\}(1 + o(1)) = p_i(1 + o(1)).$$

Therefore,

$$\mathrm{Ex}[\deg_{i+1,\gamma}(u)] \quad = \quad (1 - p_i)Y(1 + o(1))$$

$$
\begin{aligned}
&= (1-p_i)d_{i,\gamma}\left(\frac{\alpha a_0 + d_{i+1}}{\alpha a_0 + d_i}\right)(1+o(1)) \\
&= (1-p_i)d_i\left(\frac{\alpha a_0 + d_i}{a_0}\right)\left(\frac{\alpha a_0 + d_{i+1}}{\alpha a_0 + d_i}\right)(1+o(1)) \\
&= d_{i+1}\left(\frac{\alpha a_0 + d_{i+1}}{a_0}\right)(1+o(1)).
\end{aligned}
$$

This is the right expectation, so we just need to show that this quantity is also highly concentrated about its mean. Again we use Proposition 1. This time, the underlying probability space is generated by the independent, tentative colour choices of those edges of G_i known not to have chosen colour γ. Note that now we have to consider the entire graph, and not just $G_{i,\gamma}$.

Here's a strategy for determining how many of the Y neighbours from the first step survive the second: first query all edges joining u and vertices other than those which were discarded in the first step. Each tentative choice can affect the number of successes by at most 2. That's all we know, so we bound the variance of each query by $2^2/4 = 1$. There are of course at most $d_i(1+o(1))$ such queries.

Call an edge connecting u and one of the Y neighbours which still might succeed in colouring itself (because its colour was unique among the edges incident with u), a *half-successful* edge and the neighbour a *half-successful* vertex. Next query all edges joining two half-successful vertices. Each such edge could kill the chances of one half-successful edge or the other, but not both since they have different tentative colours. This effect of 1 happens with probability at most $2/a_i(1+o(1))$ ("at most" since the edge might not even have the necessary colours in its palette). Thus, each such query has variance at most $2/a_i(1+o(1))$.

Lastly, query all remaining edges incident with a half-successful neighbour. Each such edge could kill only one half-successful edge, so each query has variance at most $1/a_i(1+o(1))$.

The strategy's total variance is therefore

$$d_i(1+o(1)) + d_{i,\gamma}d_i/a_i(1+o(1)),$$

which, when we put $\varphi = 4\ln n$, leads to a deviation of

$$4\sqrt{d_i(1+d_{i,\gamma}/a_i)\ln n}(1+o(1)),$$

which is asymptotically negligible compared with $\deg_{i+1,\gamma}(u)$'s expectation. Thus we conclude that, with high probability,

$$\deg_{i+1,\gamma}(u) = (1-p_i)Y(1+o(1)) = d_{i+1}\left(\frac{\alpha a_0 + d_{i+1}}{a_0}\right)(1+o(1)),$$

as desired. That finishes the proof of the $i+1$ version of (8).

Now we can prove the statements made in the lemma. In the first, we are given a fixed vertex u and a set X of colours, none of which have been used by any edge incident with u, and we ask how many edges incident with u succeed in colouring themselves with colours in X. Call this quantity Y.

Fix a colour $\gamma \in X$. There are $d_{i,\gamma}(1+o(1))$ edges incident with u which might colour themselves γ. Equation (9) tells us that the probability that one of these edges succeeds is

$$\frac{d_{i,\gamma}}{a_i} \exp\left\{\frac{-2d_{i,\gamma}}{a_i}\right\}(1+o(1)) = \frac{d_i}{\alpha a_0 + d_i} \exp\left\{\frac{-2d_i}{\alpha a_0 + d_i}\right\}(1+o(1)) = \frac{p_i d_i}{\alpha a_0 + d_i}(1+o(1)).$$

Since there are $|X|$ colours and only one edge can succeed for each colour, the expected number of edges which succeed in colouring themselves from X is

$$\mathrm{Ex}[Y] = |X|\left(\frac{p_i d_i}{\alpha a_0 + d_i}\right)(1+o(1)).$$

Now we use a large deviation argument similar to that used in the second step of the inductive proof of (8). Here we consider the whole probability space generated by all tentative colour choices. First query the $d_i(1+o(1))$ edges incident with u. Each can affect Y by at most 2. Thus, each of these queries has variance at most $2^2/4 = 1$.

Given this partial information, call the edges which might still colour themselves successfully from X (because their colour was unique and in X) and the corresponding neighbours *half-successful*. Next query all edges joining two half-successful vertices. Each such query has effect at most 1 with probability at most $2/a_i(1+o(1))$. Then query all remain edges incident with the half-successful vertices. These queries also have effect at most 1, but probability at most $1/a_i(1+o(1))$.

This strategy then has a total variance of at most $d_i(1+o(1)) + |X|d_i/a_i(1+o(1))$, which leads to a deviation of at most $4\sqrt{d_i(1+|X|/a_i)\ln n}(1+o(1))$. Since $|X| \gg \sqrt{d_0 \ln n}$, this is again negligible compared with the mean, which proves statement (i) of the lemma.

In statement (ii), we have two vertices u and v and a set X of colours, none of which has been used by an edge incident with u or v. We ask how many pairs of edges e and f incident with u and v, respectively, will succeed in colouring themselves with the same colour from X.

For a specific colour $\gamma \in X$, the probability that there is a pair which colours itself γ is the number of pairs times the probability given by (10). There are between $\deg_{i,\gamma}(u) \cdot \deg_{i,\gamma}(v)$ and $(\deg_{i,\gamma}(u) - 1) \cdot (\deg_{i,\gamma}(v) - 2)$ pairs. In any case, the number of pairs is $d_{i,\gamma}^2(1+o(1))$. Therefore the probability that there is a pair which colours itself γ is

$$\frac{d_{i,\gamma}^2}{a_i^2} \exp\left\{\frac{-4d_{i,\gamma}}{a_i}\right\}(1+o(1)) = \left(\frac{p_i d_i}{\alpha a_0 + d_i}\right)^2(1+o(1)).$$

Since there are $|X|$ colours, the expected number of successful pairs is

$$|X|\left(\frac{p_i d_i}{\alpha a_0 + d_i}\right)^2(1+o(1)).$$

A large deviation argument very similar to that given in the proof of statement (i) works here too. First, we query the edges around the vertices u and v. Changing the tentative colour of one of these edges can create at most one new pair or kill at most two pairs (and this only if the edge in question formed a pair) but not both. Hence, we have $2d_i(1+o(1))$ queries with maximum effect of 2. Call the pairs which survive the exposure of these queries the *half-succesful* pairs.

Next, we query the edges not yet queried which connect two distinct half-succesful pairs. There are at most $4|X|^2(1+o(1))$ queries to be made. Changing the tentative colour of one of these edges creates no new pairs and can kill at most one of the two pairs. The maximum effect of each query is then 1 with probability at most $2/a_i(1+o(1))$, since there are two out of $a_i(1+o(1))$ colours to choose from. Finally, we query the remaining edges incident to edges forming a pair. The number of such queries is at most $2|X|d_i(1+o(1))$ minus twice the number of queries made in the previous group, each with a maximum effect of 1 and probability of at most $1/a_i(1+o(1))$.

This strategy's total variance is at most $4d_i(1+2|X|/a_i)(1+o(1))$ leading to a deviation of at most $4\sqrt{\varphi d_i(1+2|X|/a_i)(1+o(1))}$. By choosing $\varphi = 3\ln n$, we get an exceptional probability of n^{-3}, which can accomodate $|E| < n^2$ edges for constantly many rounds.

Statement (iii) is proved almost identically. For a particular colour γ, the number of triples incident with u, v, and w which might potentially all be coloured γ is $d_{i,\gamma}^3(1+o(1))$. Each triple succeeds with the probability given by (11). Since there are $|X|$ colours, this means that the expected number of successful triples is

$$|X|\frac{d_{i,\gamma}^3}{a_i^3}\exp\left\{\frac{-6d_{i,\gamma}}{a_i}\right\}(1+o(1)) = |X|\left(\frac{p_i d_i}{\alpha a_0 + d_i}\right)^3(1+o(1)).$$

The familiar large deviation argument finishes the proof. □

Given all of this, how large must s be? If we define β_i by d_i/a_0, we see that (7) implies that

$$\beta_{i+1} = \beta_i\left(1-\exp\left\{\frac{-2\beta_i}{\alpha+\beta_i}\right\}\right)$$

and from the initial conditions, $\beta_0 = 1-\alpha$. Since we wish to find an s such that

$$\frac{\deg_s(v)}{a_0} = \beta_s(1+o(1)) \le \frac{\alpha^2}{4},$$

it suffices to find an s such that $\beta_s \le \alpha^2/8$. It is then immediately clear that s depends only on α. A little computation reveals that

$$s \approx 15\log_{10}(1/\alpha)$$

suffices.

5 Analysis: Phase II

The second phase of the analysis follows the progress of the algorithm until

$$\deg_t(v) \le 10(1+\alpha)^9 \ln n \tag{12}$$

for every vertex v. The surprising thing is that $t = s + 4 + (3/2)\ln(1/\alpha) + 3\ln\ln a_0$ suffices.

We will show that the quantities defined by $d_s = (\alpha^2/4)a_0$ and the recurrence

$$d_{i+1} = \frac{8(1+\alpha)}{\alpha a_0} d_i^2 + \sqrt{10 d_i \ln n} \tag{13}$$

satisfy with high probability $\deg_i(v) \le d_i$ for every vertex v and integer $s \le i \le t$. Then we will solve the recurrence to determine how large t need to be in order to satisfy (12).

At the end of Phase I we know that $\deg_s(v) \le (\alpha^2/4)a_0$ for every vertex v and that $|A_s(e)| = a_0 \left(\alpha + \frac{d_s}{a_0}\right)^2 (1+o(1)) \ge \alpha^2 a_0$ for every edge e in G_s. Together these imply that the edge palettes can never become empty, since even if every edge incident with edge e chose a different colour from e's palette, there would still be $(\alpha^2/2)a_0$ colours. Formally, for every $i \ge s$ and every edge e in G_i,

$$|A_i(e)| \ge \frac{\alpha^2}{2} a_0.$$

Also from Phase I we know that for any two edges e and f and $i = s$,

$$|A_i(e) \cap A_i(f)| = a_0 \left(\alpha + \frac{d_i}{a_0}\right)^3 (1+o(1)) \le a_0 \left(\alpha + \alpha^2/4\right)^3 (1+o(1)) \le (1+\alpha)\alpha^3 a_0.$$

Furthermore, since the palette intersections can only get smaller over time, the same upper bound holds for any $i \ge s$.

The proof that the d_i's defined by (13) satisfy $\deg_i(v) \le d_i$, proceeds by induction. The base case $i = s$ follows immediately from the definition of $d_s = (\alpha^2/4)a_0$.

Assume that the claim is true for G_i and consider the step from G_i to G_{i+1}. Here, an edge $e = uv$ does not succeed in colouring itself with probability at most

$$\sum_{f \cap e \ne \emptyset} \Pr[e \text{ and } f \text{ choose the same colour}] = \sum_f \frac{|A_i(e) \cap A_i(f)|}{|A_i(e)| \cdot |A_i(f)|} \le (\deg_i(u) + \deg_i(v)) \frac{4(1+\alpha)}{\alpha a_0}$$

Therefore, by induction, the expected new degree of a fixed vertex v is

$$\mathrm{Ex}[\deg_{i+1}(v)] = \sum_{e \ni v} \Pr[e \text{ does not colour itself}] \le \frac{8(1+\alpha)}{\alpha a_0} d_i^2. \tag{14}$$

A large deviation argument very similar to those given in the proof of Lemma 2 shows that $\deg_{i+1}(v)$ deviates from its mean by more than $\sqrt{5\varphi d_i}$ with probability less than $e^{-\varphi}$. We need that the degree is within this deviation for every vertex in every round of Phase II, so we set $\varphi = 2\ln n$. Therefore, with high probability

$$\deg_{i+1}(v) \le \frac{8(1+\alpha)}{\alpha a_0} d_i^2 + \sqrt{10 d_i \ln n} = d_{i+1},$$

as desired.

All that's left is to "solve" the recurrence (13). Of course, we're really only interested in determining how large t must be in order to guarantee that $d_t \le 10(1+\alpha)^9 \ln n$.

A close look at (13) reveals that the first term dominates when d_i is large—that is, initially—and the second term dominates as d_i becomes smaller. We will break the analysis into two sub-phases to reflect this behaviour. In order to optimize the constants later, we define the transition point r as the minimum integer such that

$$\frac{8(1+\alpha)}{\alpha a_0} d_r^2 \leq 4\alpha(1+\alpha)\sqrt{10 d_r \ln n}.$$

This is equivalent to saying that r is the minimum integer such that $d_r^3 \leq (5/2)\alpha^4 a_0 \ln n$.

For $s \leq i < r$,

$$d_{i+1} \leq \frac{8(1+\alpha)}{\alpha a_0} d_i^2 + \frac{2}{\alpha^2 a_0} d_i^2 \leq \frac{3}{\alpha^2 a_0} d_i^2,$$

using the fact that $\alpha \leq 1/10$. Therefore, for $s \leq i \leq r$,

$$d_i \leq \left(\frac{3}{\alpha^2 a_0}\right)^{2^{i-s}-1} d_s^{2^{i-s}} = \left(\frac{3}{\alpha^2 a_0}\right)^{2^{i-s}-1} \left(\frac{\alpha^2 a_0}{4}\right)^{2^{i-s}} = \frac{\alpha^2 a_0}{3}\left(\frac{3}{4}\right)^{2^{i-s}}.$$

Thus, r is at most the solution to the equation

$$\left[\frac{\alpha^2 a_0}{3}\left(\frac{3}{4}\right)^{2^{r-s}}\right]^3 = (5/2)\alpha^4 a_0 \ln n.$$

This is equivalent to

$$\left(\frac{4}{3}\right)^{3\cdot 2^{r-s}} = \frac{2\alpha^2}{135}\frac{a_0^2}{\ln n} < a_0^2,$$

which is implied by

$$r - s - 1 < \frac{\ln\ln a_0}{\ln 2}.$$

So, r is at most $s + 1 + (3/2)\ln\ln a_0$.

For $r \leq i < t$,

$$d_{i+1} \leq (4\alpha(1+\alpha)+1)\sqrt{10 d_i \ln n} \leq (1+\alpha)^4\sqrt{10 d_i \ln n}$$

Thus, d_i is asymptotic to the solution of $x = (1+\alpha)^4\sqrt{10 x \ln n}$. Namely, d_i is asymptotic to $10(1+\alpha)^8 \ln n$. In fact,

$$d_i \leq (1+\alpha)^4\sqrt{10\ln n}\, d_{i-1}^{2^{-1}} \leq \left((1+\alpha)^4\sqrt{10\ln n}\right)^{2-2^{i-r-1}} d_r^{2^{-(i-r)}} \leq 10(1+\alpha)^8(\ln n) d_r^{2^{-(i-r)}}.$$

Since we intend to follow d_i until $d_t \leq 10(1+\alpha)^9 \ln n$, we require that t be no more than the solution to the equation

$$10(1+\alpha)^8(\ln n) d_r^{2^{-(t-r)}} = 10(1+\alpha)^9 \ln n.$$

This is just

$$(1+\alpha)^{2^{t-r}} = d_r < a_0,$$

which is implied by

$$t - r < (\ln\ln a_0 - \ln\ln(1+\alpha))/\ln 2 < (3/2)\ln\ln a_0 + (3/2)\ln(1/\alpha) + 1.$$

Therefore, t is at most $r + 1 + (3/2)\ln(1/\alpha) + (3/2)\ln\ln a_0 \leq s + 2 + (3/2)\ln(1/\alpha) + 3\ln\ln a_0$.

6 Analysis: Phase III

The third and final phase of the analysis follows the algorithm until every edge is coloured. Up to this point we've been able to control the individual vertex degrees, but from now on they are too small to have good statistical properties. For this reason we now take a global perspective and concentrate on the number of edges remaining in the entire graph.

As noted at the beginning of the previous section, already at the end of Phase I we know that for any $i \geq s$ and any edges e and f in G_i, $|A_i(e)| \geq (\alpha^2/2)a_0$ and $|A_i(e) \cap A_i(f)| \leq (1+\alpha)\alpha^3 a_0$. The stopping condition for Phase II guarantees that for any $i \geq t$ and any vertex v, $\deg_i(v) \leq 10(1+\alpha)^9 \ln n$. Therefore, for any $i \geq t$, in the transition from G_i to G_{i+1}, the probability that an edge e does not succeed in colouring itself is at most

$$\sum_{f \cap e \neq \emptyset} \Pr[e \text{ and } f \text{ choose the same colour}] = \sum_f \frac{|A_i(e) \cap A_i(f)|}{|A_i(e)| \cdot |A_i(f)|} \leq \frac{80(1+\alpha)^{10} \ln n}{\alpha a_0}.$$

The expected number of edges in G_i is at most this probability to the $(i-t)$th power times the number of edges in G_t. This is at most

$$\left[\frac{80(1+\alpha)^{10} \ln n}{\alpha a_0}\right]^{i-t} 5(1+\alpha)^9 n \ln n.$$

When the expected number of edges is less than p, Markov's inequality implies that the probability that there exists an uncoloured edge is less than p. Setting $p = 1/\ln n$ and simplifying, we want to determine the minimum i such that

$$\left[\frac{\alpha a_0}{80(1+\alpha)^{10} \ln n}\right]^{i-t} > 5(1+\alpha)^9 n \ln^2 n.$$

This is equivalent to

$$i - t > \frac{\ln n}{\ln a_0}(1 + o(1)).$$

In particular, if $\ln a_0 = \Omega(\ln n / \ln \ln n)$, we require only $O(\ln \ln n)$ rounds to completely edge colour the graph.

Taking a bit more care with the constants, to ensure that Phase III completes within $k \ln \ln n$ rounds, we require

$$\left[\frac{\alpha a_0}{80(1+\alpha)^{10} \ln n}\right]^{k \ln \ln n} > 5(1+\alpha)^9 n \ln^2 n,$$

which, when $k \geq 1$, is implied by

$$a_0 > \frac{400(1+\alpha)^{19}}{\alpha} n^{1/k \ln \ln n} \ln^2 n.$$

Note that it is possible to replace the constant 400 by the constant $100(1+\alpha)^2$ by changing the stopping point of Phase I. If we follow Phase I until $\deg_s(v) \leq \alpha^3 a_0 / 2(1+\alpha)$, we get $|A_i(e)| \geq \alpha^2 a_0/(1+\alpha)$ for every $i \geq s$, giving improvements both here and in Phase II.

7 Analysis: The irregular case

If the value of the maximum degree Δ could be inexpensively distributed to all of the edges, the regular case analysis would also be valid in the general case. However in a distributed architecture this might be too costly, so it is important that the algorithm rely on local information alone. This motivates the initial palette size of $(1+\varepsilon)\max\{\deg(u), \deg(v)\}$ for edge $e = uv$.

In fact what happens in the case when neighbouring edges receive different sized initial palettes is that the probability of conflict is decreased and so the edges succeed in colouring themselves even more rapidly than in the regular case. We will argue that the graph will be completely coloured at least as fast as a regular graph where all edges have the same initial palette size as the edge in our irregular graph with the smallest initial palette size. We'll do this by fixing an edge e and a round i and showing that the probability that e succeeds in colouring itself in this round is at least as high as if the graph were locally as in the regular case.

We modify e's neighbourhood in several ways, all of which decrease the probability of e's success. First of all, for every edge f incident with e, we ignore every colour from f's palette which was not in e's initial palette. This increases the probability of conflict between e and f by forcing f to choose a tentative colour which e at least has a chance of choosing and therefore decreases the probability of e's success. Next we add phantom edges to the vertex of e with lower degree. Say $e = uv$ and $\deg_i(u) \le \deg_i(v)$, so we add phantom edges to u. To create the phantom edges' palettes and fill out the real edges' short palettes, we randomly add colours from e's initial palette until $\deg_{i,\gamma}(u) = \deg_{i,\gamma}(v) = d_{i,\gamma}$ for all colours γ. This only decreases e's probability of success, since it creates more opportunities for conflict.

But now the situation is locally just as if the graph were $\max(\deg(u), \deg(v))$-regular and so the probability of e's success is as in the regular case analysis. And thus, in the original irregular graph, the probability of e's success is at least as high. In particular, if every edge's initial palette size is large enough, the entire graph will be coloured within $O(\log\log n)$ rounds.

8 Concluding Remarks

If we are willing to spend $O(\log n)$ rounds, we can lower the condition on the minimum initial palette size to $a_0 \gg \log n$. In this case, we skip Phase II entirely and jump to an argument similar to Phase III. Already at the end of Phase I, the probability that an edge does not succeed in colouring itself is less than (see equation (14))

$$\frac{8(1+\alpha)}{\alpha a_0} d_i \le 2(1+\alpha)\alpha.$$

Therefore the expected number of edges remaining in G_i is at most

$$[2(1+\alpha)\alpha]^{i-s}\,\alpha^2 a_0 n/8.$$

This quantity is less than $1/\ln n$ when $i - s = \Omega(\ln(a_0 n \ln n)) = \Omega(\log n)$. The other way around, $O(\log n)$ rounds suffice to colour all edges. The condition $a_0 \gg \log n$ comes only from the hypothesis of Lemma 2 in Phase I.

As we pointed out, if it were possible to distribute the maximum degree Δ to all edges, we could assign initial palettes of size $(1+\varepsilon)\Delta$ and so guarantee completion in $O(\log\log n)$ rounds whenever $\Delta = \Omega\left(n^{c/\log\log n}\right)$. This isn't feasible, however, unless the graph has diameter $O(\log\log n)$. Failing this, we could spend a constant number of rounds (or, if n were known, $O(\log\log n)$ rounds) distributing the maximum local degrees before setting the initial palettes and running the colouring algorithm. Thus we would only require that each edge be "near" a vertex of sufficiently high degree to ensure completion within $O(\log\log n)$ rounds.

Finally, note that the condition $a_0 = \Omega\left(n^{c/\log\log n}\right)$ is not an artifact of our analysis. There really are graphs which the algorithm will not colour in $O(\log\log n)$ rounds unless the initial palettes are this large. Take as an example $n/3$ disjoint copies of the path on 3 vertices. For each path, the two edges both succeed as soon as they pick different tentative colours and the paths succeed independently. Thus, the expected number of paths surviving after t rounds is $n/3a_0^t$. This means that unless the initial palettes are of size $\Omega\left(n^{c/\log\log n}\right)$, the expected number of rounds until all pairs are coloured is more than $O(\log\log n)$.

References

[1] N. Alon, personal communications.

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

[3] N. Alon, J.H. Kim, and J. Spencer, Nearly perfect matchings in regular simple hypergraphs, preprint, 1995.

[4] B. Berger and J. Rompel, Simulating $(\log^c n)$-wise independence in NC, *Journal of the ACM* **38**(4) (1991), 1026–1046.

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

[6] D. Dubhashi and A. Panconesi, Near-optimal distributed edge coloring, Proceedings of the 3rd annual European Symposium on Algorithms (ESA 95), Springer Verlag, LNCS 979, 448–459.

[7] D. Durand, R. Jain and D. Tseytlin. Distributed scheduling algorithms to improve the performance of parallel data transfers. Tech. Report 94-38, DIMACS, 1994.

[8] Z. Galil and D. Leven, NP-completeness of finding the chromatic index of regular graphs, *J. of Algorithms* **4** (1983), 35–44.

[9] D.A. Grable, A large deviation inequality for functions of independent, multi-way choices, preprint, 1996.

[10] I. Holyer, The NP-completeness of edge coloring, *SIAM J. Comp.* **10** (1981), 718–720.

[11] 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**(4) (1992), 352–362.

[12] H.J. Karloff and D.B. Shmoys, Efficient parallel algorithms for edge coloring problems, *Journal of Algorithms* **8** (1992), 39–52.

[13] J. Kahn, Asymptotically good list-colorings, *Journal of Combinatorial Theory, Series A*, to appear.

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

[15] J.H. Kim, The Ramsey number $R(3,t)$ has magnitude $t^2/\log t$, *Random Structures and Algorithms* **7** (1995), 173–208.

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

[17] N.A. Lynch, Upper bounds for static resource allocation in a distributed system, *Journal of Computer and System Sciences* **23** (1981), 254–278.

[18] R. Motwani, J. Naor and M. Naor. The probabilistic method yields deterministic parallel algorithms, 30th FOCS, 1989, 8–13.

[19] A. Panconesi and A. Srinivasan, Fast randomized algorithms for distributed edge coloring, in: Proceedings of the ACM Symposium on Principles of Distributed Computing, 1992, 251–262, *SIAM Journal on Computing*, to appear.

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

[21] N. Pippenger and J. Spencer, Asymptotic behavior of the chromatic index for hypergraphs, *Journal of Combinatorial Theory, Series A* **51** (1989), 24–42.

[22] V. Rödl, On a packing and covering problem, *European Journal of Combinatorics* **5** (1985), 69–78.