Basic Research in Computer Science

BRICS RS-97-37 Grable & Panconesi: Fast Distributed Algorithms for Brooks-Vizing Colourings

Fast Distributed Algorithms for Brooks-Vizing Colourings

(Extended Abstract)

David A. Grable
Alessandro Panconesi

BRICS Report Series **RS-97-37**

ISSN 0909-0878 **December 1997**

Copyright © 1997, BRICS, Department of Computer Science University of Aarhus. All rights reserved.

Reproduction of all or part of this work is permitted for educational or research use on condition that this copyright notice is included in any copy.

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/97/37/`

Fast Distributed Algorithms for Brooks-Vizing Colourings
(Extended Abstract)

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

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

*Supported by Deutsche Forschungsgemeinschaft project number Pr 296/4-2.
†This research was done when visiting the Humboldt University in Berlin thanks to financial support of the Alexander von Humboldt foundadtion.

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 [10, 13, 4].

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 [15] 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 $p = 1/2$). 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.

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, [2]).

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 [8, 11, 12]. 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 [17]. (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 [17].)

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, 18].

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 [9] 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) [3].

Johansson's result, as well as an earlier result of Kim [14], 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 [14], 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 almost always produces such a colouring. Furthermore, they 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, D-regular input graph G such that $D \geq \log^{1+\delta} n$, where $\delta > 0$ is any fixed constant, computes with probability $1-o(1)$ a vertex colouring of G using D/k colours, for any $k \leq \alpha \log D$, where α is a constant which depends on δ. Moreover, with probability $1 - o(1)$, the colouring will be completed within $O\left(k + \frac{\log n}{\log D}\right)$ 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 D. For instance, by choosing $D \geq n^{c/\log\log n}$, where $c > 1$ is any constant, and $k = \log\log n$, the algorithm will compute a $(D/\log\log n)$-colouring in just $O(\log\log n)$ rounds. Or, by choosing $D \geq n^{c/\sqrt{\log n}}$ the algorithm will compute a $(D/\sqrt{\log n})$-colouring in $O(\sqrt{\log n})$ rounds. Notice also that the algorithm works for $k = \Omega(\log D)$, 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 $D = \Omega(\log n)$. This, as well as the regularity assumption (also assumed in [9, 14]), 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 D.

Although we stated our result in its most general form, in this abstract we shall present a slightly weaker version, due to lack of space. Namely, we shall show that the above statement holds with the running time replaced by $O(\log n)$.

2 Girth 5 or more

In this section we assume the distributed network to be a D-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 D and of a positive real k. The goal is a colouring with D/k colours. The protocol is as follows.

For each vertex u, in parallel:
$A := \{1, \ldots, D/k\}$; $p := 1/k$. /* initialize palette and wake-up parameter */
Repeat until u is coloured:
Awake with probability p.
If awake /* attempt to colour */
Choose a tentative colour t uniformly from A.
If no neighbour picked t during this round, make colour t permantent and stop.
/* update palette and wake-up parameter */
$A := A - \{c \mid$ a neighbour of u coloured itself with $c\}$.
if $p < 1$ then $p := \min\left\{1, \frac{1}{1/p - 1/e}\right\}$.

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.

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 $a_i(u)$;
- For each vertex u, colour γ and round i, the number of uncoloured neighbours of u which have γ in their palettes; denoted by $d_i(u, \gamma)$.
- 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 exist a round $i = O(k)$ such that, for every vertex u,

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

This will occur after the algorithm has switched to its trivial phase. Since the behaviour of the trivial algorithm in this situation is known [10, 13, 4],

we can then immediately conclude that the trivial algorithm will with certainty complete the colouring and it will do so within $O(\log n)$ rounds with high probability. As stated in the introduction, with a little bit more work it is possible to show that the running time is actually $O(k + \log n/\log D)$. Due to space limitations we shall content ourselves to carry the analysis only up to the point where the termination condition (1) holds. We remark that this is in any case the most difficult and interesting part of the analysis.

In order to establish condition (1), we shall show that the random variables $a_i(u)$, $d_i(u,\gamma)$, and $\deg_i(u)$ are approximated very well by "ideal" values a_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 $a_i \gg D_i$, which implies condition (1). 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,

$$\begin{aligned} a_{i+1} &= a_i \exp(-1/e), \\ d_{i+1} &= d_i \left(1 - \frac{a_i}{d_i}\frac{1}{e}\right) \exp(-1/e), \text{ and} \\ D_{i+1} &= D_i \left(1 - \frac{a_i}{d_i}\frac{1}{e}\right). \end{aligned} \tag{2}$$

with initial conditions $D_0 = d_0 = D$ and $a_0 = D/k$.

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

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

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 $a_i < d_i$. 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 a_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,

$$d_{ek} = a_{ek} = a_0 \exp(-ek/e) = \frac{D}{ke^k}.$$

Since, at all times, the vertex palettes must be nonempty, $D/ke^k \geq 1$ is a necessary condition (on k in terms of D). Asymptotically, this condition is satisfied for all $k = (\log D)/c$ with $c > 1$ and not satisfied for $k = \log D$. Thus we see that $k = O(\log D)$ 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 [3].

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

$$D_{ek} = D/k.$$

Thus at the end of the dozing-off phase, we are still quite far from attaining condition (1), so we must continue our analysis into the trivial phase.

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

$$\begin{aligned} a_{i+1} &= a_i \exp\left\{-\frac{d_i}{a_i} e^{-d_i/a_i}\right\}, \\ d_{i+1} &= d_i \left(1 - e^{-d_i/a_i}\right) \exp\left\{-\frac{d_i}{a_i} e^{-d_i/a_i}\right\}, \text{ and} \\ D_{i+1} &= D_i \left(1 - e^{-d_i/a_i}\right). \end{aligned} \tag{3}$$

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

$$\frac{d_{i+1}}{a_{i+1}} = \frac{d_i}{a_i}\left(1 - e^{-d_i/a_i}\right) < \left(\frac{d_i}{a_i}\right)^2, \tag{4}$$

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

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

$$\begin{aligned} a_{i+1} &\geq a_i \exp\left\{-\frac{d_i}{a_i} e^{-1}\right\} > a_i \exp\left\{-\frac{d_i}{a_i}\right\} \\ D_{i+1} &\leq D_i \left(1 - \left(1 - \frac{d_i}{a_i}\right)\right) = D_i \frac{d_i}{a_i}. \end{aligned} \tag{5}$$

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

Since the ratio d_i/a_i may be identically one at the beginning of the trivial phase ($i = ek$), the inequality (4) 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 $a_{ek+1} > D/ke^{k+1}$, $D_{ek+1} \leq D/k$ and $\rho := d_{ek+1}/a_{ek+1} \leq 1 - 1/e < 1$.

Now, using Equation (4), we get $d_{ek+i}/a_{ek+i} \leq \rho^{2^{i-1}}$. Therefore, using (5) repeatedly, we see that

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

while

$$\begin{aligned} a_{ek+i} &\geq a_{ek+1} \exp\left\{ -\sum_{j=0}^{i-2} \rho^{2^j} \right\} \\ &\geq a_{ek+1} \exp\left\{ -\sum_{j=0}^{i-2} \rho^{j} \right\} \\ &\geq \frac{D}{ke^{k+1}} \exp\left\{ -\frac{\rho}{1-\rho} \right\}. \end{aligned}$$

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

$$D_i < ca_i \tag{6}$$

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

If we could show that the random variables $\deg_i(u)$, $a_i(u)$ and $d_i(u, \gamma)$ were arbitrarily close to the ideal values D_i, a_i and d_i then Equation (6) would imply that the termination condition (1) holds. This is indeed the case. Lack of space denies us the possibility of giving the entire proof of the following theorem, but we will sketch its proof a bit later.

Theorem 1 *With high probability*

$$a_i(u) = (1 \pm e_i)a_i,\ d_i(u,\gamma) = (1 \pm e_i)d_i,\ \textit{and}\ \deg_i(u) = (1 \pm e_i)D_i \tag{7}$$

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

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

for a constant $C > 1$, and $e_0 = 0$.

First, we show that the assumptions $D = \Omega(\log^{1+\delta} n)$ and $k = O(\log D)$ 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

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

which is just the condition $C_2^k \log n \ll D$, for suitable constants C_1 and C_2. Assuming that $D \geq \log^{1+\delta} n$, for any $\delta > 0$, this condition becomes $C_2^k \ll D^{1+1/(1+\delta)}$, which is just that $k \leq \alpha \log D$ for an appropriately small, but constant α.

So, if the e_i's behave as advertised in the Theorem, the statements concerning the running time and the correctness of the algorithm follow.

In the next few pages, we sketch the proof of the Theorem. Although the actual computations differ, the arguments for the dozing-off phase and for the trivial phase are 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: $E[a_{i+1}(u)] \sim a_{i+1}$, $E[d_{i+1}(u,\gamma)] \sim d_{i+1}$, and $E[\deg_{i+1}(u)] \sim D_{i+1}$. Then it is shown that that the random variables $a_{i+1}(u)$, $d_{i+1}(u,\gamma)$ and $\deg_{i+1}(u)$ are sharply concentrated around their expectations. The use of the asymptotic equality in place of explicit error factors is justified provided that the e_i's behave as stated in the Theorem. We shall return to this later.

The concentration results are an important and non trivial part of the proof but, unfortunately, they are quite involved. So we omit their proofs from this extended abstract. They use a powerful large deviation inequality [5], which has already proved useful in the algorithmic setting [7, 6]. Their applications to the present situation are similar, but more involved than, the applications given in the author's [7]. The simplest of the present cases, that of $\deg_i(u) \sim D_i$ for the trivial phase, is given as an example in [5] and the reader is referred to it. The remaining ones can be derived similarly, although they require a combination of old and new techniques. We remark that the large deviation inequalities are the source of the assumption on the size of D, for they cease to give strong enough bounds when the degree is not high enough.

Next in our proof sketch is the derivation of the recurrences governing the behaviour of the algorithm. This is done 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 i-th version of Equation (7) holds. To simplify notation, a, d, and D will denote the values a_i, d_i, and D_i and a', d', and D' will denote the values a_{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 γ–colours" we will refer to the event that vertex u colours itself succesfully with colour γ during the current round, and by "γ decays at u" we will refer to the event that colour γ is deleted from u's palette at the end of the current round.

In the rest of the section, we focus on an arbitrary uncoloured vertex u. During the dozing-off phase, $a < d$ and the vertices awake with probability $p = a/d$.

Lemma 2 *During the dozing-off phase, for each uncoloured vertex u and each colour γ in u's palette,*

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

The proof is straightforward and follows from the induction hypothesis and the following fact (see for instance [16]).

Fact 3 *Let $A(n)$ and $B(n)$ be such that $A(n)^2 B(n) = o(1)$ (n tending to infinity, as always). Then,*

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

¿From the lemma, again using Fact 3 and the induction hypothesis, we see that,

$$\Pr[u \text{ colours}] = \sum_{\gamma \in A(u)} \Pr[u \text{ } \gamma\text{-colours}] \sim \frac{a}{d}\frac{1}{e} \tag{8}$$

where $A(u)$ is the current palette of u. Denoting by $N(u)$ the current set of uncoloured neighbours of u,

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

since by induction $|N(u)| = \deg(u) \sim D$. This shows that the expectation of $\deg'(u)$ is near the ideal value D' and was what suggested the recurrence

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

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. For each colour γ in the palette, we see that, denoting by $N(u,\gamma)$ the set of neighbours of u whose palettes contain γ at the current round,

$$\begin{aligned}&\Pr[\gamma \text{ decays at } u \mid u \text{ does not colour}]\\ &= \Pr[\text{some } v \in N(u,\gamma)\ \gamma\text{-colours} \mid u \text{ does not colour}]\end{aligned}$$

The last event can be computed by considering the complementary event. The probability that neither of two neighbours γ-colour is difficult in general, but in the case where there are neither 3- nor 4-cycles, we see that the events "x does not γ-colour" and "y does not γ-colour" are "essentially" independent, since γ-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. This can be made rigorous and we can conclude that

$$\begin{aligned}&\Pr[\gamma \text{ decays at } u \mid u \text{ does not colour}]\\ &\sim 1 - \prod_{v \in N(u,\gamma)} \Pr[v \in N(u,\gamma) \text{ does not } \gamma\text{-colour} \mid u \text{ does not colour}] \qquad (9)\\ &\sim 1 - \exp(-1/e).\end{aligned}$$

The proof, which follows from Fact (3) and the IH, is omitted. Then, again by induction,

$$E[a'(u)] = \sum_{\gamma \in A(u)} (1 - \Pr[\gamma \text{ decays at } u]) \sim a \exp(-1/e)$$

which suggested that palette decay during the dozing-off phase is governed by the recurrence

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

Now we turn to the $d'(u,\gamma)$'s. For each colour γ, the graph G_γ is defined as the graph induced by the vertices whose palettes contain γ. The probability that a vertex u must be removed from G_γ is, recalling

Equations (8) and (9),

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

Since, by induction,

$$\begin{aligned}
&E[d'(u,\gamma)] \\
&= \sum_{v \in N(u,\gamma)} (1 - \Pr[u \text{ disappears from } G_\gamma]) \\
&\sim d\left(1 - \frac{1}{d}\frac{1}{e}\right)\exp(-1/e),
\end{aligned}$$

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

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

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 $a(u)$, $d(u,\gamma)$, 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 3, 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. As it turns out, of the three concentration results, the one with the worst error factor is the palette size. We will use that error factor, namely $c_2\sqrt{(\log n)/a_i}$ uniformly. In fact, we will bound this factor in

each round by the value in the last round, which gives the worst error: $c_2\sqrt{(ke^k \log n)/D}$. Thus,

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

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

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

for an appropriate constant $C > 1$.

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 a_i/d_i. Their derivations are conceptually the same, however.

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

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

Thus,

$$\Pr[u \text{ colours}] = \sum_{\gamma \in A} \Pr[u \text{ } \gamma\text{-colours}] \sim e^{-d/a}$$

Again both asymptotic equalities follow from Fact 3. This implies that the expected number of neighbours of u which remain uncoloured in the next round is asymptotically equal to $D(1 - e^{-d/a})$, which led to the recurrence

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

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

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

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

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

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

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

which led to

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

Again, the next step in the proof is to prove the concentration results. The situation here is essentially the same, but slightly simpler than in the dozing-off phase.

Then to finish up one has to consider the error factors. Again, this works out to be almost exactly the same as in the dozing-off phase, giving an error factor of the same form—namely

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

where only the constant C differs.

This finishes the proof of the analysis of the trivial phase down to the point where condition (1) holds. Thereafter, the analysis of the trivial algorithm completes the proof [10, 4, 13].

3 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 γ decays from the palette of a particular vertex $u \in A$ can be computed as follows.

$$\begin{aligned}
&\Pr[\gamma \text{ decays at } u \mid u \text{ does not colour}] \\
&= 1 - \Pr[\text{no } v \in B \ \gamma\text{-colours} \mid —] \\
&= 1 - \Pr[\text{some } w \in A \text{ picks } t(w) = \gamma \mid —] \\
&\quad - \Pr[\text{no vertex at all picks } \gamma \mid —] \\
&= 1 - \left(1 - \left(1 - \frac{p}{a}\right)^d\right) - \left(1 - \frac{p}{a}\right)^{2d} \\
&\sim \frac{1}{e} - \frac{1}{e^2} \approx 0.233
\end{aligned}$$

(Here we used $p = a/d$, 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_γ, the graph induced by vertices with colour γ 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 γ,* $\Pr[\gamma$ *decays at* $u \mid u$ *does not colour*$] \leq 1 - \exp\left\{-\frac{pd}{a} e^{-pd/a}\right\}$.

Proof The proof compares the probability that the colour γ 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 γ 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 γ decays is asymptotic to $1 - \exp\left\{-\frac{pd}{a}e^{-pd/a}\right\}$. We show that as we merge leaves into composite second neighbours, the probability that γ decays cannot increase. Thus, in the orignial graph, the probability that γ 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 γ 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 γ. This conditioning will be represented by a dash (—).

If u's tentative colour is γ, 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 γ does not decay at u, regardless of the topology of the second neighbourhood. Thus, we may assume that u's tentative colour was not γ and hence that any of the vertices in X might colour themselves γ and thereby cause γ 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 given graph.

$$
\begin{aligned}
&\Pr[\gamma \text{ decays at } u \mid \text{—}] \\
&= \Pr\Big[(v_1\ \gamma\text{-colours}) \text{ or } \bigcup_{v\in X\setminus v_1} (v\ \gamma\text{-colours}) \;\Big|\; \text{—}\Big] \\
&= \Pr\Big[v_1\ \gamma\text{-colours} \;\Big|\; \text{—}\Big] + \Pr\Big[\bigcup_{v\in X\setminus v_1} (v\ \gamma\text{-colours}) \;\Big|\; \text{—}\Big] \\
&\qquad - \Pr\Big[(v_1\ \gamma\text{-colours}) \text{ and } \bigcup_{v\in X\setminus v_1} (v\ \gamma\text{-colours}) \;\Big|\; \text{—}\Big]
\end{aligned}
$$

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

$$
\begin{aligned}
&\Pr\Big[(v_1\ \gamma\text{-colours}) \text{ and } \bigcup_{v\in X\setminus v_1} (v\ \gamma\text{-colours}) \;\Big|\; \text{—}\Big] = \\
&\Pr\Big[v_1\ \gamma\text{-colours}\Big|\ \text{—}\Big] \Pr\Big[\bigcup_{v\in X\setminus v_1} (v\ \gamma\text{-colours}) \;\Big|\; (v_1\ \gamma\text{-colours}) \text{ and } \text{—}\Big],
\end{aligned}
$$

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

Thus, after merging w_1 with w_2, the joint probability increases, decreasing the probability that γ 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. $\triangledown$

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

One way to correct this unfortunate state of affairs is to apply the following kamikaze approach: 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 $1 - \exp\left\{-(pd/a)e^{-pd/a}\right\}$.

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 γ the natural probability $p_{u,\gamma}$ of decay. This is done before tentative colours are chosen. The round then proceeds as before. At the end, each colour γ which was not naturally removed is artificially removed with probability

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

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

With this additional balancing mechanism, the expectations for the γ-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.

4 Conclusions

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

The analysis prompts several interesting questions. First of all, is the lower bound on D 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 [7], 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] B. Bollobás, Graph Theory, Springer Verlag, New York, 1979.

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

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

[5] D.A. Grable, A large deviation inequality for functions of independent, multi-way choices, Combinatorics, Probability and Computing, to appear. Available at "http://www.informatik.hu-berlin.de/~grable/ldi.ps".

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

[7] 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.

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

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

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

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

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

[13] 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.

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

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

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

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

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

Recent BRICS Report Series Publications

RS-97-37 David A. Grable and Alessandro Panconesi. *Fast Distributed Algorithms for Brooks-Vizing Colourings (Extended Abstract)*. December 1997. 20 pp. To appear in *The Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '98.

RS-97-36 Thomas Troels Hildebrandt, Prakash Panangaden, and Glynn Winskel. *Relational Semantics of Non-Deterministic Dataflow*. December 1997. 21 pp.

RS-97-35 Gian Luca Cattani, Marcelo P. Fiore, and Glynn Winskel. *A Theory of Recursive Domains with Applications to Concurrency*. December 1997. ii+23 pp.

RS-97-34 Gian Luca Cattani, Ian Stark, and Glynn Winskel. *Presheaf Models for the π-Calculus*. December 1997. ii+27 pp. Appears in Moggi and Rosolini, editors, *Category Theory and Computer Science: 7th International Conference*, CTCS '97 Proceedings, LNCS 1290, 1997, pages 106–126.

RS-97-33 Anders Kock and Gonzalo E. Reyes. *A Note on Frame Distributions*. December 1997. 15 pp.

RS-97-32 Thore Husfeldt and Theis Rauhe. *Hardness Results for Dynamic Problems by Extensions of Fredman and Saks' Chronogram Method*. November 1997. i+13 pp.

RS-97-31 Klaus Havelund, Arne Skou, Kim G. Larsen, and Kristian Lund. *Formal Modeling and Analysis of an Audio/Video Protocol: An Industrial Case Study Using* UPPAAL. November 1997. 23 pp. To appear in *The 18th IEEE Real-Time Systems Symposium, RTSS '97 Proceedings*.

RS-97-30 Ulrich Kohlenbach. *Proof Theory and Computational Analysis*. November 1997. 38 pp.

RS-97-29 Luca Aceto, Augusto Burgueño, and Kim G. Larsen. *Model Checking via Reachability Testing for Timed Automata*. November 1997. 29 pp.

RS-97-28 Ronald Cramer, Ivan B. Damgård, and Ueli Maurer. *Span Programs and General Secure Multi-Party Computation*. November 1997. 27 pp.