BRICS
Basic Research in Computer Science

BRICS RS-98-23 D. P. Dubhashi: Martingales and Locality in Distributed Computing

Martingales and Locality in Distributed Computing

Devdatt P. Dubhashi

BRICS Report Series **RS-98-23**

ISSN 0909-0878 **October 1998**

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/98/23/`

Martingales and Locality in Distributed Computing *

Devdatt P. Dubhashi †
Department of Computer Science and Engg.
Indian Institute of Technology, Delhi
Hauz Khas, New Delhi 110016, INDIA.
email: dubhashi@cse.iitd.ernet.in

October 6, 1998

Abstract

We use Martingale inequalities to give a simple and uniform analysis of two families of distributed randomised algorithms for edge colouring graphs.

1 Introduction

The aim of this paper is to advocate the use of certain Martingale inequalities known as "The Method of Bounded Differences" (henceforth abbreviated to MOBD) [9] as a tool for the analysis of distributed randomized algorithms that work in the *locality* paradigm. The form of the inequality we employ

*To appear in the 18th Conference on the *Foundations of Software Technology and Theoretical Computer Science*, Chennai, India December 17–19, 1998.

†Work done partly while at the SPIC Mathematical Institute, Chennai, India and while visiting **BRICS**, Basic Research in Computer Science, Centre of the Danish National Research Foundation, Department of Computer Science, University of Aarhus, Denmark. Partly supported by the Research program of the EU under contract No. 20244 (ALCOM–IT)

and its application here is significantly different from previous successful uses of the method in Computer Science applications in that it exerts much finer control on the effects of the underlying variables to get significantly stronger bounds , and it succeeds in spite of the lack of complete independence. This last feature particularly, makes it a valuable tool in Computer Science contexts where lack of independence is omnipresent. This aspect of the MOBD has, to the best of our knowledge, never been adequately brought out before. Our contribution is to highlight its special relevance for Computer Science applications by demonstrating its use in the context of a class of distributed computations in the locality paradigm.

We give a high probability analysis of a two classes of distributed edge colouring algorithms, [2, 4, 11]. Apart from its intrinsic interest as a classical combinatorial problem, and as a paradigm example for locality in distributed computing, edge colouring is also useful from a practical standpoint because of its connection to scheduling. In distributed networks or architectures an edge colouring corresponds to a set of data transfers that can be executed in parallel. So, a partition of the edges into a small number of colour classes – i.e. a "good" edge colouring– gives an efficient schedule to perform data transfers (for more details, see [11, 2]). The analysis of edge colouring algorithms published in the literature is extremely long and difficult and that in [11] is moreover, based an a certain *ad hoc* extension of the Chernoff-Hoeffding bounds. In contrast, our analysis is a very simple, short and streamlined application of the MOBD, only two pages long, and besides, also yields slightly stronger bounds. These two examples are intended moreover, as a dramatic illustration of the versatility and power of the method for the analysis of locality in distributed computing in general, a framework for which is sketched at the end.

In a **message-passing** distributed network, one is faced with the twin problems of *locality* and *symmetry breaking*. Each processor can gather data only locally so as to minimise communication which is at a premium rather than computation. Moreover all processors are identical form the point of view of the network which makes it hard, if not impossible, to schedule the operations of the various processors at different times in order to avoid congestion and converge toward the common computing goal. An often successful way to circumvent these difficulties– the locality bottleneck and symmetry-breaking– is to resort to randomization. When randomization is used, one is required to prove performance guarantees on the results delivered by a randomised

algorithm. Notice that in truly distributed environments the usual solution available in sequential or other centralized settings such as a PRAM, namely to restart the randomized algorithm in case of bad outcome, is simply not available. This is because of the cost of collecting and distributing such information to every node in the network. In this context then, it becomes especially important to be able to certify that the randomized algorithm will almost surely work correctly i.e. provide a high probability guarantee.
Even for simple algorithms, this is a challenging task, and the analysis of edge colouring algorithms in the literature are very long and complicated, often requiring *ad hoc* stratagems, as noted above. We show how these analyses can be dramatically simplified and streamlined. The main technical tool used in this paper is a version of the "Method of Bounded Differences" that is more powerful and versatile than the generally known and used version. This version typically yields much stronger bounds. Perhaps even more significantly, the restriction on independence on the underlying random variables which is necessary for the usual version is removed and this greatly extends the scope of applicability of the method. Although these facts are implicit in some of the existing literature, they have never been explicitly noted and highlighted, to the best of our knowledge. For instance, in the (otherwise excellent) survey of McDiarmid [9], the simpler version is stated right at the outset (Lemma 1.2) and illustrated with a number of examples from combinatorics. But the more powerful version is buried away in an obscure corollary towards the end (Corollary 6.10) rather than highlighted as we feel it it deserves to be, and finds no applications. It is our intention here to highlight the more powerful version, especially those aspects that are crucial to make it particularly suitable for analysis of the edge colouring algorithms and more generally, for locality in distributed computing.

2 Distributed Edge Colouring

Vizing's Theorem shows that every graph G can be edge coloured sequentially in polynomial time with Δ or $\Delta + 1$ colours, where Δ is the maximum degree of the input graph (see, for instance, [1]).

It is a challenging open problem whether colourings as good as these can be computed fast in a distributed model. In the absence of such a result one might aim at the more modest goal of computing reasonably good colourings, instead of optimal ones. By a trivial modification of a well-known *vertex* colouring algorithm of Luby it is possible to edge colour a graph using $2\Delta - 2$ colours in $O(\log n)$ rounds (where n is the number of processors) [6].

We shall present and analyze two classes of simple localised distributed algorithms that compute near optimal edge colourings. Both algorithms proceed in a sequence of rounds. In each round, a simple randomised heuristic is invoked to colour a significant fraction of the edges successfully. The remaining edges are passed over to succeeding rounds. This continues until the number of edges is small enough to employ a brute–force method at the final step. For example, the algorithm of Luby mentioned above can be invoked when the degree of the graph becomes small i.e. when the condition $\Delta \gg \log n$ is no longer satisfied.

One of the classes of algorithms involves a standard reduction to bipartite graphs described in [11]: the graph is split into two parts T ("top") and B (bottom). The bipartite graph $G[T, B]$ induced by the edges connecting top and bottom vertices is coloured by invoking the Algorithm P described below. The algorithm is then invoked recursively in parallel on $G[T]$ and $G[B]$, the graphs respectively induced by the top and bottom vertices. Both graphs are coloured using the same set of colours. Thus it suffices to describe the algorithm used for colouring bipartite graphs.

We describe the action carried out by both algorithms in a single round. For the second class of algorithms, we describe the action only for bipartite graphs; additionally, each vertex knows whether it is top or bottom. At the beginning of each round, there is a palette of fresh new available colours, $[\Delta]$, where Δ is the maximum degree of the graph at the current stage. For simplicity we will assume that the graph is Δ–regular.

Algorithm I(Independent): Each edge *independently* picks a colour. This *tentative* colour becomes permanent if there are no conflicting edges picking the same tentative colour at either endpoint.

Algorithm P(Permutation): There is a two step protocol:

- Each bottom vertex, in parallel, makes a *proposal* independently of other bottom vertices by assigning a random *permutation* of the colours to their incident edges.
- Each top vertex, in parallel, then picks a *winner* out of every set of incident edges that have the same colour. Tentative colours of winner edges become final.

- The *losers*– edges who are not winners– are decoloured and passed to the next round.

For the purposes of the high probability analysis below, the exact rule used for selecting the winner edge is unimportant – it can be chosen arbitrarily from any of the edges of the relevant colour; we merely require that it should not depend on edges of different colours. This is another illustration of the power of the martingale method.

It is apparent that both algorithms are truly distributed. That is to say, each vertex need only exchange information with the neighbours to execute the algorithm. This and its simplicity make the algorithms amenable for implementations in a distributed environment. Algorithm I is used with some more modifications in a number of edge colouring algorithms [2, 4]. Algorithm P is exactly the algorithm used in [11].

We focus all our attention in the analysis of one round of both algorithms. Let Δ denote the maximum degree of the graph at the beginning of the round and Δ' denote the maximum degree of the leftover graph. One can easily show that both algorithms, $\mathtt{E}[\Delta' \mid \Delta] \leq \beta\Delta$, for some constant $\beta < 1$. For algorithm I, $\beta = 1 - e^{-2}$ while for algorithm P, $\beta = 1/e$. The goal is to show that this holds with high probability. This is done in § 4 after the relevant tools – the Martingale inequalities – are introduced in the next section.

For completeness, we sketch a calculation of the total number of colours $\textsc{bc}(\Delta)$ used by Algorithm P for the bipartite colouring of a graph with maximum degree Δ: with high probability, it is,

$$\begin{aligned} \textsc{bc}(\Delta) &= \Delta + \frac{(1+\epsilon)\Delta}{e} + \frac{(1+\epsilon)^2\Delta^2}{e} + \dots \\ &\leq \frac{1}{1-(1+\epsilon)e}\Delta \approx 1.59\Delta \text{ for small enough } \epsilon. \end{aligned}$$

To this, one should add $O(\log n)$ colours at the end of the process. As can be seen by analyzing the simple recursion describing the number of colours used by the outer level of the recursion, the overall numbers of colours is the same $1.59\Delta + O(\log n)$, [11].

3 Martingale Inequalities

We shall use martingale inequalities in the *avataras* of the "Method of Bounded Differences" [9]. We shall use the following notations and conventions: $X_1, \ldots, X_n$ will denote random variables with X_i taking values in some set A_i for each $i \in [n]$. For each $i \in [n]$, a_i, a_i' will denote arbitrary elements from A_i. For a function f of several arguments, $f(*, a_i, *)$ will denote that all arguments except the indicated one are held fixed. We shall use boldface notation to abbreviate the sequences: so $\mathbf{X}$ denotes $X_1, \ldots, X_n$, and $\mathbf{a}$ denotes $a_1, \ldots, a_n$. Finally, for each $i \in [n]$, we abbreviate $X_1 = a_1, \ldots, X_i = a_i$ by $\mathbf{X}_i = \mathbf{a}_i$.

3.1 Method(s) of Bounded Differences

The most widely known and used form of the "Method of Bounded Differences" is as follows:

Theorem 1 (Method of Bounded Differences)
Let f be a function that is **Lipschitz** *with constants $c_i, i \in [n]$ i.e.*

$$|f(*, a_i, *) - f(*, a_i', *)| \leq c_i, \quad \text{for } i \in [n]. \tag{1}$$

Then, if $X_1, \ldots, X_n$ are **independent** *random variables, for any $t > 0$,*

$$\mathtt{Pr}[|f - \mathtt{E}[f]| > t] \leq 2 \exp\left(\frac{-2t^2}{\sum_i c_i^2}\right).$$

While extremely convenient to use, this version has two drawbacks. First the bound obtained can be quite weak (because $\sum_i c_i^2$ is large) and second, the assumption of independence limits its range of applicability. A stronger form of the inequality that removes both these limitations is the following

Theorem 2 (Method of Bounded Average Differences)
Let f be a function and $X_1, \ldots, X_n$ a set of random variables (not necessarily independent) such that there are constants $c_i, i \in [n]$, for which

$$|\mathtt{E}[f \mid \mathbf{X}_{i-1} = \mathbf{a}_{i-1}, X_i = a_i] - \mathtt{E}[f \mid \mathbf{X}_{i-1} = \mathbf{a}_{i-1}, X_i = a_i']| \leq c_i, \tag{2}$$

for each $i \in [n]$ and for every two assignments $\mathbf{X}_{i-1} = \mathbf{a}_{i-1}, X_i = a_i, a'_i$ that are separately consistent (hence of non–zero probability). Then, for any $t > 0$,

$$\texttt{Pr}[|f - \texttt{E}[f]| > t] \leq 2\exp\left(\frac{-2t^2}{\sum_i c_i^2}\right).$$

For a discussion of the relative strengths of these methods, see [3].

3.2 Coupling

In order to make effective use of the Method of Average Bounded Differences, we need to get a good handle on the bound (2), for the difference in the expected values of a function under two different conditioned distributions. A very useful technique for this is the method of *coupling*. Suppose that we can find a joint distribution $\pi(\mathbf{Y}, \mathbf{Y}')$ such that the marginal distribution for Y is the same as the distribution of $\mathbf{X}$ conditioned on $\mathbf{X}_{i-1} = \mathbf{a}_{i-1}, X_i = a_i$ and the marginal distribution for $\mathbf{Y}'$ is the same as the distribution $\mathbf{X}$ conditioned on $\mathbf{X}_{i-1} = \mathbf{a}_{i-1}, X_i = a'_i$. Such a joint distribution is called a coupling of the two original distributions. Then,

$$\begin{aligned} |\texttt{E}[f \mid \mathbf{X}_{i-1} = \mathbf{a}_{i-1}, X_i = a_i] - \texttt{E}[f \mid \mathbf{X}_{i-1} = \mathbf{a}_{i-1}, X_i = a'_i]| &= \\ |\texttt{E}_\pi[f(\mathbf{Y})] - \texttt{E}_\pi[f(\mathbf{Y}')]| = |\texttt{E}_\pi[f(\mathbf{Y}) - f(\mathbf{Y}')]| & \quad . \end{aligned} \tag{3}$$

If the coupling π is well–chosen so that $|f(\mathbf{Y}) - f(\mathbf{Y}')|$ is usually very small, we can get a good bound on the difference (2). For example, suppose that

- For any sample point $(\mathbf{y}, \mathbf{y}')$ we have $|f(\mathbf{y}) - f(\mathbf{y}')| \leq d$ for some constant $d > 0$; and

- For most sample points $(\mathbf{y}, \mathbf{y}'$, $f(\mathbf{y}) = f(\mathbf{y}')$. That is, $\pi[f(\mathbf{Y}) - f(\mathbf{Y}')] \leq p$, for some $p << 1$.

Then, we can conclude using (3) that

$$|\texttt{E}[f \mid \mathbf{X}_{i-1} = \mathbf{a}_{i-1}, X_i = a_i] - \texttt{E}[f \mid \mathbf{X}_{i-1} = \mathbf{a}_{i-1}, X_i = a'_i]| \leq pd.$$

We shall construct suitable couplings to bound the difference in (2).

4 High Probability Analyses

4.1 Top Vertices

The analysis is particularly easy when v is a top vertex in Algorithm P. For, in this case, the incident edges all receive colours independently of each other. This is exactly the situation of the classical balls and bins experiment: the incident edges are the "balls" that are falling at random independently into the colours that represent the "bins". One can apply the method of bounded differences in the simplest form. Let $T_e, e \in E$, be the random variables taking values in $[\Delta]$ that represent the tentative colours of the edges. Then the number of edges successfully coloured around v is a function $f(T_e, e \in N^1(v))$, where $N^1(v)$ denotes the set of edges incident on v.

It is easily seen that this function has the *Lipschitz* property with constant 1: changing only one argument while leaving the others fixed only changes the value of f by at most 2. Note that this is true *regardless of the rule for choosing winners*, as long as this rule does not depend on edges of different colours. This will also be true of the remaining analyses below and illustrates once again, the power of the martingale methods.

Moreover, the variables $T_e, e \in N^1(v)$ are independent when v is a "top" vertex. Hence, by the method of bounded differences in the simplest form, we get the following sharp concentration result by plugging into Theorem 1:

Theorem 3 *Let v be a top vertex in algorithm P and let f be the number of edges around v that are successfully coloured in one round of the algorithm. Then,*

$$\mathtt{Pr}[|f - \mathtt{E}[f]| > t] \leq \exp\left(\frac{-t^2}{2\Delta}\right),$$

For $t := \epsilon\Delta$ $(0 < \epsilon < 1)$, this gives an exponentially decreasing probability for deviations around the mean. If $\Delta \gg \log n$ then the probability that the new degree of any vertex deviates far from its expected value is inverse polynomial, i.e. the new max degree is sharply concentrated around its mean.

4.2 Other Vertices: The Difficulty

The analysis for the "bottom" vertices in Algorithm P is more complicated in several respects. It is useful to see why so that one can appreciate the need for using a more sophisticated tool such as the Method of Bounded Average Differences. To start with, one could introduce an indicator random variable X_e for each edge e incident upon a bottom vertex v. These random variable are not independent however. Consider a four cycle with vertices v, a, w, b, where v and w are bottom vertices and a and b are top vertices. Let's refer to the process of selecting the winner (step 2 of the algorithm P) as "the lottery". Suppose that we are given the information that edge va got tentative colour red and lost the lottery— i.e. $X_{va} = 0$— and that edge vb got tentative colour green. We'll argue intuitively that given this, it is more likely that $X_{vb} = 0$. Since edge va lost the lottery, the probability that edge wa gets tentative colour red increases. In turn, this increases the probability that edge wb gets tentative colour green, which implies that edge vb is more likely to lose the lottery. So, not only are the X_e's not independent, but the dependency among them is particularly malicious.

One could hope to bound this effect by using the MOBD in it simplest form. This is also ruled out however, for two reasons. The first is that the tentative colour choices of the edges around a vertex are not independent. This is because the edges incident on vertex are assigned a permutation of the colours. The second reason applies also to algorithm I where all edges act independently. The new degree of v, a bottom vertex in algorithm P or an arbitrary vertex in algorithm I, is a function $f = f(T_e, e \in N(v))$, where $N(v)$ is the set of edges at distance at most 2 from v. Thus f depends on as many as $\Delta(\Delta-1) = \Theta(\Delta^2)$ edges. Even if f is Lipshitz with constants $d_i = 2$, this is not enough to get a strong enough bound because $d = \sum_i d_i^2 = \Theta(\Delta^2)$. Applying the method of bounded differences in the simple form, Theorem 1, would give the bound

$$\mathtt{Pr}[|f - \mathtt{E}[f]| > t] \leq 2\exp\left(-\frac{t^2}{\Theta(\Delta^2)}\right).$$

This bound however is useless for $t = \epsilon\mathtt{E}[f]$ since $\mathtt{E}[f] \approx \Delta/e$.

We will use the Method of Bounded Average Differences, Theorem 2, to get a much better bound. We shall invoke the two crucial features of this more general method. Namely that it does not presume that the underlying variables are independent [1] , and that, as we shall see, it allows us to bound

[1]A referee pointed out that one can redefine variables in the analysis of Algorithm P to make them independent; however, this is unnecessary since Theorem to be applied does

the effect of individual random choices with constants much smaller than those given by the MOBD in simple form.

Let's now move on to the analysis. A similar analysis applies to both cases: when v is a bottom vertex in algorithm P or an arbitrary vertex in algorithm I. Let $N^1(v)$ denote the set of "direct" edges– i.e. the edges incident on v– and let $N^2(v)$ denote the set of "indirect edges" that is, the edges incident on a neighbour of v. Let $N(v) := N^1(v) \bigcup N^2(v)$. The number of edges successfully coloured at vertex v is a function $f(T_e, e \in N(v))$. Note that in Algorithm P, even though f seems to depend on edges at distance 3 from v via their effect on edges at distance 2, f can still be regarded as a function of the edges in $N(v)$ only (i.e. f is fixed by giving colours to all edges in $N(v)$ regardless of what happens to other edges) and hence only these edges need be considered in the analysis.

Let us number the variables so that the direct edges are numbered *after* the indirect edges (this will be important for the calculations to follow). We need to compute

$$\lambda_k := |\mathtt{E}[f \mid \mathbf{T}_{k-1}, T_k = c_k] - \mathtt{E}[f \mid \mathbf{T}_{k-1}, T_k = c'_k]|. \tag{4}$$

We decompose f as a sum to ease the computations later. Introduce the indicator functions $f_e, e \in E$:

$$f_e(\mathbf{c}) := \begin{cases} 1; & \text{if edge } e \text{ is successfully coloured in colouring } \mathbf{c}, \\ 0; & \text{otherwise.} \end{cases}$$

Then $f = \sum_{v \in e} f_e$.

Hence we are reduced, by linearity of expectation, to computing for each $e \in N^1(v)$,

$$|\mathtt{Pr}[f_e = 1 \mid \mathbf{T}_{k-1}, T_k = c_k] - \mathtt{Pr}[f_e = 1 \mid \mathbf{T}_{k-1}, T_k = c'_k]|.$$

For the computations that follows we should keep in mind that in algorithm P bottom vertices assign colours independently of each other. This implies that in either algorithm, the colour choices of the edges incident upon a neighbour of v are independent of each other. In Algorithm I, *all* edges have their colours assigned independently.

not need independence. Moreover, in general, it may not always be possible to make such a redefinition of variables. But the general method will still apply.

4.3 General Vertex in Algorithm I

To compute a good bound for λ_k in (4), we shall construct a suitable coupling of the two different conditioned distributions. The coupling $(\mathbf{Y}, \mathbf{Y}')$ is almost trivial: $\mathbf{Y}$ is distributed as $\mathbf{T}$ conditioned on $\mathbf{T}_{k-1}, T_k = c_k$ and $\mathbf{Y}'$ is identically equal to $\mathbf{Y}$ except that $\mathbf{Y}'_k = c'_k$. It is easily seen that by the independence of all tentative colours, the marginal distributions of $\mathbf{Y}$ and $\mathbf{Y}'$ are exactly the two conditioned distributions $[\mathbf{T} \mid \mathbf{T}_{k-1}, T_k = c_k]$ and $[\mathbf{T} \mid \mathbf{T}_{k-1}, T_k = c'_k]$ respectively.

Now let us compute $|\mathtt{E}[f(\mathbf{Y}) - f(\mathbf{Y}')]|$.

- First, let us consider the case when $e_1, \ldots, e_k \in N^2(v)$, i.e. only the choices of indirect edges are exposed. Let $e_k = (w, z)$, where w is a neighbour of v. Then for a direct edge $e \neq vw$, $f_e(\mathbf{y}) = f_e(\mathbf{y}')$ because in the joint distribution space, $\mathbf{y}$ and $\mathbf{y}'$ agree on all edges incident on e. So we only need to compute $|\mathtt{E}[f_{vw}(\mathbf{Y}) - f_{vw}(\mathbf{Y}')]|$. To bound this simply, we observe first that $f_{vw}(\mathbf{y}) - f_{vw}(\mathbf{y}') \in [-1, 1]$ and second that $f_{vw}(\mathbf{y}) = f_{vw}(\mathbf{y}')$ unless $y_{vw} = c_k or c'_k$. Thus we can conclude that

  $$\mathtt{E}[f_{vw}(\mathbf{Y}) - f_{vw}(\mathbf{Y}')]| \leq \mathtt{Pr}[Y_e = c_k \vee Y_e = c'_k] \leq \frac{2}{\Delta}.$$

  In fact one can do a tighter analysis using the same observations. Let us denote $f_e(\mathbf{y}, y_{w,z} = c_1, y_e = c_2)$ by $f_e(c_1, c_2)$. Note that $f_{vw}(c_k, c_k) = 0$ and similarly $f_{vw}(c'_k, c'_k) = 0$. Hence

  $$\begin{aligned} &\mathtt{E}[f_e(\mathbf{Y}) - f_e(\mathbf{Y}') \mid z] = \\ &\quad (f_{vw}(c_k, c_k) - f_{vw}(c'_k, c_k))\mathtt{Pr}[Y_e{=}c_k] + (f_{vw}(c_k, c'_k) - f_{vw}(c'_k, c'_k))\mathtt{Pr}[Y_e{=}c'_k] \\ &= (f_{vw}(c_k, c'_k) - f_{vw}(c'_k, c_k))\frac{1}{\Delta} \end{aligned}$$

  (Here we used the fact that the distribution of colour around v is unaffected by the conditioning around z and that each colour is equally likely.) Hence $|\mathtt{E}[f_e(\mathbf{Y}) - f_e(\mathbf{Y}')]| \leq \frac{1}{\Delta}$.

- Now let us consider the case when $e_k \in N^1(v)$, i.e. choices of all indirect edges and of some direct edges have been exposed. In this case, we merely observe that f is Lipshitz with constant 2: $|f(\mathbf{y}) - f(\mathbf{y}')| \leq 2$ whenever $\mathbf{y}$ and $\mathbf{y}'$ differ in only one co–ordinate. Hence we can easily conclude that $|\mathtt{E}[f(\mathbf{Y}) - f(\mathbf{Y}')]| \leq 2$.

Overall,

$$\lambda_k \le \begin{cases} 1/\Delta; & \text{for an edge } e_k \in N^2(v), \\ 2; & \text{for an edge } e_k \in N^1(v) \end{cases},$$

and we get

$$\sum_k \lambda_k^2 = \sum_{e \in N^2(v)} \frac{1}{\Delta^2} + \sum_{e \in N^1(v)} 4 \le 4\Delta + 1.$$

We thus arrive at the following sharp concentration result by plugging into Theorem 2:

Theorem 4 *Let v be an arbitrary vertex in algorithm I and let Let f be the number of edges successfully coloured around v in one stage of either algorithm. Then,*

$$\mathtt{Pr}[|f - \mathtt{E}[f]| > t] \le 2 \exp\left(-\frac{t^2}{2\Delta + \frac{1}{2}}\right).$$

A referee observed that a similar result can be obtained very simply for Algorithm I by applying Theorem 1: regard f as a function of 2Δ variables: $T_e, v \in e$ and $\mathbf{T}(w), (v, w) \in E$, where $\mathbf{T}(w)$ records the colours of all edges incident on w except vw. Since f is Lipshitz with constant 2 with respect to each of these variables, we get the bound:

$$\mathtt{Pr}[|f - \mathtt{E}[f]| > t] \le 2 \exp\left(-\frac{t^2}{4\Delta}\right).$$

4.4 Bottom Vertices in Algorithm P

Once again, to compute a good bound for λ_k in (4), we shall construct a suitable coupling of the two different conditioned distributions $\mathbf{T}_{k-1}, T_k = c_k$ and $\mathbf{T}_{k-1}, T_k = c'_k$. Suppose e_k is an edge zy where z is a bottom vertex. The coupling $(\mathbf{Y}, \mathbf{Y}')$ in this case is the following: $\mathbf{Y}$ is distributed as $\mathbf{T}$ conditioned on $\mathbf{T}_{k-1}, T_k = c_k$ and $\mathbf{Y}'$ is identically equal to $\mathbf{Y}$ except on the edges incident on z, where the colours c_k and c'_k are switched. We can think of the distribution as divided into two classes: on the edges incident on a vertex other than z, the two variables $\mathbf{Y}$ and $\mathbf{Y}'$ are identically equal. In particular, when z is not v, they have the same uniform distribution on all permutations of colours on the edges around v. However, on the edges incident on z, the two variables differ on exactly two edges where the colours c_k and c'_k are switched. It is easily seen that by the independence of different vertices, the marginal distributions of $\mathbf{Y}$ and $\mathbf{Y}'$ are exactly the two conditioned distributions $[\mathbf{T} \mid \mathbf{T}_{k-1}, T_k = c_k]$ and $[\mathbf{T} \mid \mathbf{T}_{k-1}, T_k = c'_k]$ respectively.

Now let us compute $|\mathtt{E}[f(\mathbf{Y}) - f(\mathbf{Y}')]|$. Recall that f was decomposed as a sum $\sum_{v \in e} f_e$. Hence by linearity of expectation, we need only bound each $|\mathtt{E}[f_e(\mathbf{Y}) - f_e(\mathbf{Y}')]|$ separately.

- First, let us consider the case when $e_1, \ldots, e_k \in N^2(v)$, i.e. only the choices of indirect edges are exposed. Let $e_k = (w, z)$ for a neighbour w of v. Note that since

  $$\mathtt{E}[f(\mathbf{Y}) - f(\mathbf{Y}')] = \mathtt{E}[\mathtt{E}[f(\mathbf{Y}) - f(\mathbf{Y}') \mid \mathbf{Y}_e, \mathbf{Y'}_e, z \in e]],$$

  it suffices to bound $|\mathtt{E}[f(\mathbf{Y}) - f(\mathbf{Y}') \mid \mathbf{Y}_e, \mathbf{Y'}_e, z \in e]|$. Hence, fix some distribution of the colours around z. Recall that $\mathbf{Y}_{w,z} = c_k$ and $\mathbf{Y'}_{w,z} = c'_k$. Suppose $\mathbf{Y}_{z,w'} = c'_k$ for some other neighbour w' of z. Then by our coupling construction, $\mathbf{Y'}_{z,w'} = c_k$ and on the remaining edges $\mathbf{Y}$ and $\mathbf{Y}'$ agree identically. Moreover, by the independence of the other vertices, the distributions of $\mathbf{Y}$ and $\mathbf{Y}'$ on the remaining edges conditioned on the distribution around z is unaffected. let us denote the conditioned joint distribution by $[(\mathbf{Y}, \mathbf{Y}') \mid z]$. We thus need to bound $|\mathtt{E}[f(\mathbf{Y}) - f(\mathbf{Y}') \mid z]|$.

  Then for a direct edge $e \notin vw, vw'$, $f_e(\mathbf{y}) = f_e(\mathbf{y}')$ because in the joint distribution space, $\mathbf{y}$ and $\mathbf{y}'$ agree on all edges incident on e. So we only need to compute $|\mathtt{E}[f_e(\mathbf{Y}) - f_e(\mathbf{Y}')]|$ for $e \in vw, vw'$. To bound this simply, we observe that for either $e = vw$ or $e = vw'$, first, $f_e(\mathbf{y}) - f_e(\mathbf{y}') \in [-1, 1]$ and second that $f_e(\mathbf{y}) = f_e(\mathbf{y}')$ unless $y_e = c_k or c'_k$. Thus we can conclude that

  $$\mathtt{E}[f_e(\mathbf{Y}) - f_e(\mathbf{Y}')]| \leq \mathtt{Pr}[Y_e = c_k \vee Y_e = c'_k] \leq \frac{2}{\Delta}.$$

  Thus taking the two contributions for vw and vw' together, $|\mathtt{E}[f(\mathbf{Y}) - f(\mathbf{Y}') \mid z]| \leq \frac{4}{\Delta}$.

  In fact one can do a tighter analysis using the same observations. Let us denote $f_e(\mathbf{y}, y_{w,z} = c_1, y_e = c_2)$ by $f_e(c_1, c_2)$. Note that $f_{vw}(c_k, c_k) =$

$0 = f_{vw}(c'_k, c'_k)$ and similarly $f_{vw'}(c_k, c_k) = 0 = f_{vw'}(c'_k, c'_k)$. Thus, for $e = vw$ or $e = vw'$,

$$
\begin{aligned}
&\mathtt{E}[f_e(\mathbf{Y}) - f_e(\mathbf{Y}') \mid z] = \\
&\quad (f_e(c_k, c_k) - f_e(c'_k, c_k))\mathtt{Pr}[Y_e{=}c_k] + (f_e(c_k, c'_k) - f_e(c'_k, y'_e{=}c'_k))\mathtt{Pr}[Y_e{=}c'_k] \\
&= (f_e(c_k, c'_k) - f_e(c'_k, c_k))\frac{1}{\Delta}
\end{aligned}
$$

Hence $|\mathtt{E}[f_e(\mathbf{Y}) - f_e(\mathbf{Y}') \mid z]| \leq \frac{1}{\Delta}$. Taking the two contributions for edges vw and vw' together, $|\mathtt{E}[f(\mathbf{Y}) - f(\mathbf{Y}') \mid x]| \leq \frac{2}{\Delta}$.

- Now let us consider the case when $e_k \in N^1(v)$, i.e. choices of all indirect edges and of some direct edges have been exposed. In this case, we observe again that $|f(\mathbf{y}) - f(\mathbf{y}')| \leq 2$ since $\mathbf{y}$ and $\mathbf{y}'$ differ on exactly two edges. Hence we can easily conclude that $|\mathtt{E}[f(\mathbf{Y}) - f(\mathbf{Y}')]| \leq 2$.

Overall,

$$
\lambda_k \leq \begin{cases} 2/\Delta; & \text{for an edge } e_k \in N^2(v), \\ 2; & \text{for an edge } e_k \in N^1(v) \end{cases},
$$

and we get

$$
\sum_k \lambda_k^2 = \sum_{e \in N^2(v)} \frac{4}{\Delta^2} + \sum_{e \in N^1(v)} 4 \leq 4(\Delta + 1).
$$

We thus arrive at the following sharp concentration result by plugging into Theorem 2:

Theorem 5 *Let v be an arbitrary vertex in algorithm I and let f be the number of edges successfully coloured around v in one stage of algorithm P. Then,*

$$
\mathtt{Pr}[|f - \mathtt{E}[f]| > t] \leq 2\exp\left(-\frac{t^2}{2\Delta + 2}\right).
$$

Comparing this with the corresponding bound for Algorithm I, we see that the failure probabilities for both algorithms are almost identical. For $t = \epsilon\Delta$, both a probability that decreases exponentially in Δ. As remarked earlier, if $\Delta \gg \log n$, this implies that the new max degree is sharply concentrated around the mean (with failure probability inverse polynomial in n). The constant in the exponent here is better than the one in the analysis in [11].

4.5 Extensions

It is fairly clear that the method extends more generally to cover similar scenarios in distributed computing. We sketch such a general setting: One has a distributed randomised algorithm that requires vertices to assign labels to themselves and incident edges. Each vertex acts independently of the others, and furthermore is symmetric with respect to the labels (colours). The function of interest, f depends only on a small local neighbourhood around some vertex v, is Lipschitz and satisfies some version of the following *locality* property: the labels on vertices and edges far away from v only effect f if certain events are triggered on nearer vertices and edges; these triggering events correspond to the setting of the nearer vertices and edges to specific values. For example, in edge colouring, the colour of an idirect edge only affects f if the incident direct edge has the same colour. One can extend the same arguments as above virtually intact for this general setting. This encompasses all the edge colouring algorithms mentioned above as well as the vertex colouring algorithms in [10] and [5]. More details of such a framework are given in the appendix.

5 Acknowledgement

Edgar Ramos pointed out an error in the original analysis during a talk at the Max–Planck–Institut für Informatik soon after I had submitted the paper. I showed him the coupling proof the next day. An anonymous referee also spotted the error and independently developed the same coupling argument. I am grateful for his detailed write–up containing several other comments and suggestions (some of which are acknowledged above) which helped immensely in preparing the revision. I also acknowledge the helpful comments of two other anonymous referees. I would like to thank Alessandro Panconesi for his help in drafting this paper, for invaluable discussions and for resolutely prodding me to pursue the "demise of the permanent" from [11]. Now, finally, R.I.P.

References

[1] Bollabas, B. : Graph Theory: An Introductory Course. Springer–Verlag 1980.

[2] Dubhashi, D., Grable, D.A., and Panconesi, A.: Near optimal distributed edge colouring via the nibble method. Theoretical Computer Science 203 (1998), 225–251, a special issue for ESA 95, the 3rd European Symposium on Algorithms.

[3] Dubhashi, D. and Panconesi, A: Concentration of measure for computer scientists. Draft of a monograph in preparation.

[4] Grable, D., and Panconesi, A: Near optimal distributed edge colouring in $O(\log \log n)$ rounds. Random Structures and Algorithms **10**, Nr. 3 (1997) 385–405

[5] Grable, D., and Panconesi, A: Brooks and Vizing Colurings, SODA 98.

[6] Luby, M.: Removing randomness in parallel without a processor penalty. J. Computer and Systems Sciences **47**:2 (1993) 250–286.

[7] Marton, K.: Bounding $\overline{d}$ distance by informational divergence: A method to prove measure concentration. Annals of Probability **24** (1996) 857–866.

[8] Marton, K.: On the measure concentration inequality of Talagrand for dependent random variables. submitted for publication. (1998).

[9] McDiarmid, C.J.H.: On the method of bounded differences. in J. Siemons (ed): Surveys in Combinatorics, London Mathematical Society Lecture Notes Series 141, Cambridge University Press, (1989).

[10] Molloy, M. and Reed, B.: A bound on the strong chromatic index of a graph. J. Comb. Theory (B) **69** (1997) 103–109.

[11] Panconesi, A. and Srinivasan, A.: Randomized distributed edge coloring via an extension of the Chernoff–Hoeffding bounds. SIAM J. Computing **26**:2 (1997) 350–368.

[12] Spencer, J.: Probabilistic methods in combinatorics. In proceedings of the International Congress of Mathematicians, Zurich, Birkhauser (1995).

[13] Talagrand, M.: Concentration of measure and isoperimetric inequalities in product spaces. Publ. math. IHES, **81**:2 (1995) 73–205..

A A General Framework

Let f be a function to be computed by a randomised local algorithm in a distributed environment represented by a graph $G = (V, E)$. We shall lay down conditions on f and on algorithms computing f locally that will enable the methods in the previous section to be extended to to derive a sharp concentration result on f. In this way, the resulting theorem can be applied in a cook–book substitution style to get sharp concentration results on other problems and algorithms in a localised distributed setting. In particular, we indicate how the edge colouring algorithms from [2, 4, 11] discussed above fit the framework and hence are special cases of it.

Suppose that f is a function determined by labels $\ell(e), e \in E$ on the edges of G which are assigned by the randomised algorithm. In the edge colouring problem, the labels on the edges are their colours.

First, f must be *local*: that is, there is a vertex v and a radius r such that f is completely determined by the labels of the edges in the neighbourhood of radius r centered around v. More precisely, let $d(v, e)$ denote the distance of e from v in the graph G and let us define the neighbourhood $N(v, r) := \{e \in E \mid d(v, e) \leq r\}$. Then $f = f(\ell(e), e \in N(v, r))$. In the edge colouring example, f is the number of edges coloured successfully around a fixed vertex v, and the radius $r = 2$.

Second, is a *symmetry* condition on the algorithm: the elementary events $\ell(e) = v$ must occur with probability at most $1/\Delta$ where Δ is the maximum degree of a vertex in the network. This condition is satisfied by almost any conceivable algorithm in a distributed setting, due to symmetry, in particular, the two algorithms for edge colouring discussed above.

Third, is a combined locality property of the function as well as a *dependence* property of the computing algorithm: the influence of edges on f decreases as one goes further away from v. Edges are divided into two classes: *primary*, which are in the close neighbourhood of v and *secondary*, which are further away. The function f is *Lipschitz* with respect to the primary edges: that is, if all labels are held fixed except for that of a single primary edge e, then the function value changes by at most c for some constant c. In the edge colouring problem, the edges in $N^1(v)$ are the primary edges and $c = 2$.

On the other hand, changing the label of a secondary edge only has an effect on f if certain triggering events on nearer edges are enabled. The further away an edge is, the more such events need to be triggered. In the edge colouring example, the edges in $N^2(v)$ are secondary. Changing the colour of an edge in $N^2(v)$ only has an effect if the neighbouring edge in $N^1(v)$ has the same colour.

All these conditions are natural and intuitive, and all except this last one are also simple to formulate precisely. For the last, we ask for the forbearance of the reader. For each secondary edge e at distance $d > 1$ from v, and each label c, corresponding to the event $\ell(e) = c$, there exists a set $\mathcal{E}(e, c)$ consisting of at least $d/2$ pairs (e^*, c^*) corresponding to the event $\bigwedge_{(e^*,c^*)\in\mathcal{E}(e,c)} \ell(e^*) = c^*$,(also denoted by $\mathcal{E}(e, c)$ for economy of notation) involving only edges e^* of distance less than d from v satisfying the following two properties:

- For any two labels c, c',

  $$f(*, \ell(e) = c, *) = f(*, \ell(e) = c', *), \quad \text{if } \neg\mathcal{E}(e, c) \bigwedge \neg\mathcal{E}(e, c').$$

  In words, changing the value of $\ell(e)$ while keeping all other labels the same has no effect on f if the setting of the other labels does not enable the two corresponding triggering sets of events on nearer edges. This can also be viewed as a locality property of the function – it is a strengthening of the Lipschitz condition. In the edge colouring problem, $\mathcal{E}(e, c)$ consists of the single event $\ell(e^*) = c$ where e^* is the edge connecting e to v.

- The events in each set $\mathcal{E}(e, c)$ are *independent*. This is a property of the algorithm. In fact, the following weaker condition is sufficient: the events in each set $\mathcal{E}(e, c)$ should be *negatively dependent* so that

  $$\mathtt{Pr}[\mathcal{E}(e, c)] \leq \prod_{\mathcal{E}(e,c)} \mathtt{Pr}[\ell(e) = c].$$

  Both the edge colouring algorithms satisfy this property trivially.

A similar analysis to that in the previous section leads to the following general sharp concentration result.

Theorem 6 *Let f be a function with Lipschitz constant c determined locally by a symmetric algorithm as above in a neighbourhood of radius r and p primary edges. Then if $\mathtt{E}[f] \geq \Delta^{\delta}\sqrt{4r + c^2 p}$ for some $\delta > 0$, then*

$$\mathtt{Pr}[|f - \mathtt{E}[f]| > \epsilon \mathtt{E}[f]] \leq 2 \exp\left(-\frac{\epsilon^2}{2}\Delta^{2\delta}\right).$$

Recent BRICS Report Series Publications

RS-98-23 Devdatt P. Dubhashi. ***Martingales and Locality in Distributed Computing*****. October 1998. 19 pp.**

RS-98-22 Gian Luca Cattani, John Power, and Glynn Winskel. ***A Categorical Axiomatics for Bisimulation*****. September 1998. ii+21 pp. Appears in Sangiorgi and de Simone, editors,** ***Concurrency Theory: 9th International Conference*****, CONCUR '98 Proceedings, LNCS 1466, 1998, pages 581–596.**

RS-98-21 John Power, Gian Luca Cattani, and Glynn Winskel. ***A Representation Result for Free Cocompletions*****. September 1998. 16 pp.**

RS-98-20 Søren Riis and Meera Sitharam. ***Uniformly Generated Submodules of Permutation Modules*****. September 1998. 35 pp.**

RS-98-19 Søren Riis and Meera Sitharam. ***Generating Hard Tautologies Using Predicate Logic and the Symmetric Group*****. September 1998. 13 pp.**

RS-98-18 Ulrich Kohlenbach. ***Things that can and things that can't be done in PRA*****. September 1998. 24 pp.**

RS-98-17 Roberto Bruni, José Meseguer, Ugo Montanari, and Vladimiro Sassone. ***A Comparison of Petri Net Semantics under the Collective Token Philosophy*****. September 1998. 20 pp. To appear in** ***4th Asian Computing Science Conference*****, ASIAN '98 Proceedings, LNCS, 1998.**

RS-98-16 Stephen Alstrup, Thore Husfeldt, and Theis Rauhe. ***Marked Ancestor Problems*****. September 1998.**

RS-98-15 Jung-taek Kim, Kwangkeun Yi, and Olivier Danvy. ***Assessing the Overhead of ML Exceptions by Selective CPS Transformation*****. September 1998. 31 pp. To appear in the proceedings of the** ***1998 ACM SIGPLAN Workshop on ML*****, Baltimore, Maryland, September 26, 1998.**

RS-98-14 Sandeep Sen. ***The Hardness of Speeding-up Knapsack*****. August 1998. 6 pp.**

RS-98-13 Olivier Danvy and Morten Rhiger. ***Compiling Actions by Partial Evaluation, Revisited*****. June 1998. 25 pp.**