Further Algorithmic Aspects of the Local Lemma

Michael Molloy*
Department of Computer Science
University of Toronto
Toronto, Canada
molloy@cs.toronto.edu

Bruce Reed*
Equipe Combinatoire
CNRS
Université Pierre et Marie Curie
Paris, France
reed@lug.ibp.fr

February 8, 2001

Abstract

We provide a method to produce an efficient algorithm to find an object whose existence is guaranteed by the Lovász Local Lemma. We feel that this method will apply to the vast majority of applications of the Local Lemma, unless the application has one of four problematic traits. However, proving that the method applies to a particular application may require proving two (possibly difficult) concentration-like properties.

1 Introduction

The probabilistic method is used to prove the existence of objects with desirable properties by showing that a randomly chosen object from an appropriate probability distribution has the desired properties with positive probability.

For example, it has been used to prove the existence of efficient routing procedures [15, 6], good sorting networks [1] and various types of graph colourings [12, 13, 14, 17, 18]. Often, the probability that a randomly chosen object has the desired properties is reasonably large and hence the method yields a randomized algorithm for constructing a object with the desired properties: we simply pick objects at random until we find one. Under fairly general conditions, such a method can be derandomized using, for example, the method of conditional probabilities due to Erdős and Selfridge [7].

More sophisticated tools, such as the Lovász Local Lemma [9], allow us to prove the existence of objects with properties which occur with exponentially small probability. To turn such proofs into algorithms, even random ones, requires more refined approaches. In [5], Beck showed that certain applications of the Local Lemma led to polynomial-time construction algorithms (with some sacrifices made with regards to the constants in the original application). Alon[3] provided a parallel variant of the algorithm and remarked that it was not clear how widely applicable the technique was, citing Acyclic Edge Colouring (defined below) as one of the applications of the Local Lemma for which it seemed difficult to find a corresponding algorithm.

In this paper we

1. Provide a general set of conditions, similar to those of the Local Lemma, such that for any problem satisfying these conditions we can not only guarantee the existence of the desired object, we can actually construct such a object in polynomial-time.

2. Introduce a technique to develop constructive versions of applications of the Local Lemma which do not meet the first set of conditions. We show how this variant can be applied to develop algorithms

*This work was supported by NATO Collaborative Research Grant #CRG950235. The work of the first author is supported by an NSERC Research Grant.

for previously intractable applications of the Local Lemma such as Acyclic Edge Colouring, and we outline a set of conditions which will allow this technique to be applied.

We remark that this first set of conditions is satisfied by a wide variety of problems. For example, it holds for the applications of the Local Lemma in [12, 13, 14, 17, 18] and yields efficient algorithms for the corresponding construction problems. Furthermore, proving that a certain application satisfies these conditions is typically fairly straightforward.

In contrast, the second set of conditions, while more general than the first, are much harder to verify. We pinpoint the difficulty later in the paper. Nevertheless, we believe that the vast majority of the applications of the Local Lemma can be made algorithmic by an appropriate application of the second theorem, unless the application has one of four problematic traits outlined in Section 7.

2 The Local Lemma

First we state the Local Lemma in a rather general form.

The Local Lemma *Suppose that $\mathcal{A} = \{A_1, ..., A_n\}$ is a set of random events, such that each A_i is mutually independent of $\mathcal{A} - (\{A_i\} \cup \mathcal{D}_i)$, for some $\mathcal{D}_i \subseteq \mathcal{A}$. Suppose further that we have $x_1, ..., x_n \in [0, 1)$ such that*

$$\mathbf{Pr}(A_i) \leq x_i \prod_{A_j \in \mathcal{D}_i} (1 - x_j) \qquad 1 \leq i \leq n. \tag{1}$$

Then

$$\mathbf{Pr}\left(\overline{A_1} \wedge ... \wedge \overline{A_n}\right) > 0.$$

Two common forms of the Local Lemma are obtained by replacing (1) by the following conditions:

Simple Asymmetric Case

$$\mathbf{Pr}(A_i) \leq \frac{1}{8} \qquad \text{and} \qquad \sum_{A_j \in \mathcal{D}_i} \mathbf{Pr}(A_j) \leq \frac{1}{4}, \qquad 1 \leq i \leq n. \tag{2}$$

Symmetric Case

$$d = \max_i |\mathcal{D}_i|, p = \max_i \mathbf{Pr}(A_i), \qquad \text{and} \qquad pd \leq \frac{1}{4}. \tag{3}$$

That (2) suffices follows by setting $x_i = 4\mathbf{Pr}(A_i)$. That (3) suffices follows from the fact that (2) suffices and that the case $d = 1$ is trivial.

In a typical application of the Local Lemma, we construct an object (eg. a colouring of a graph or a routing network) via a random procedure. Typically $\mathcal{A}$ is a set of "bad" events, and our procedure is successful if none of them hold. Unfortunately, while the Local Lemma guarantees the *existence* of the desired object, the probability that our procedure is successful can be exponentially small in n, and so the proof does not immediately yield an efficient constructive algorithm, not even a randomized one.

We now present two applications which illustrate the uses of various forms of the Local Lemma. Both of these applications can be made constructive using the techniques of Sections 4 and 5.

2.1 Frugal Colouring

We say that a proper vertex-colouring of a graph is β*-frugal*, if for each vertex v and colour c, the number of times that c appears in N_v, the neighbourhood of v, is less than β. Frugal colouring was introduced in [10] and played an important role in the bound on the total chromatic number in [11].

Consider any fixed β. Alon (see [10]) has shown that for each Δ, there exist graphs with maximum degree Δ which require $\Omega(\Delta^{1+\frac{1}{\beta-1}})$ colours in any β-frugal colouring. We provide here the proof from [10] that this is best possible.

Theorem 2.1 *If G has maximum degree $\Delta > \beta^{\beta-1}$ then G has a β-frugal proper vertex colouring using at most $12\Delta^{1+\frac{1}{\beta-1}}$ colours.*

Proof Set $C = 12\Delta^{1+\frac{1}{\beta-1}}$. We assign to each vertex of G a uniformly random colour from $\{1,...,C\}$. For each edge (u,v) we define the Type A event $A_{u,v}$ to be the event that u,v both receive the same colour. For each $\{u_1,...,u_\beta\}$ all in the neighbourhood of one vertex, we define the Type B event $B_{u_1,...,u_\beta}$ to be the event that $u_1,...,u_\beta$ all receive the same colour. Note that if none of these events hold, then our random procedure has successfully found a β-frugal colouring of G.

The probability of any Type A event is at most $1/C$, and the probability of any Type B event is at most $1/C^{\beta-1}$. Note that each event is mutually independent of all but at most $\beta\Delta$ Type A events and $\beta\Delta\binom{\Delta}{\beta-1}$ Type B events. Thus this application satisfies (2) but not (3). □

2.2 Acyclic Edge Colouring

We say that a proper edge colouring of a graph is *acyclic* if the union of any two colour classes is a forest. The following result was proved in [4].

Theorem 2.2 *If G has maximum degree Δ then G has an acyclic proper edge colouring using at most 16Δ colours.*

Proof Set $C = 16\Delta$. We assign to each edge of G a uniformly random colour from $\{1,...,C\}$. For each pair of incident edges e,f, we define the Type 1 event $A_{e,f}$ to be the event that e,f both receive the same colour. For each $2k$-cycle C, we define the Type k event A_C to be the event that the edges of C become properly 2-coloured. If none of these events hold, then the resulting colouring is proper and acyclic.

The probability of each Type 1 event is $1/C$ and the probability of each Type k event, $k \geq 2$, is $1/C^{2(k-1)}$. It is straightforward to show that for each $k \geq 2$, no edge lies in more than $\Delta^{2(k-1)}$ different $2k$-cycles. Each Type k event, $k \geq 1$ is mutually independent of all but at most $4k\Delta$ Type 1 events and $2k \times \Delta^{2(\ell-1)}$ Type ℓ events, $\ell \geq 2$. It is readily seen that this application does not satisfy (2). However, by setting $x_{u,v} = 2/C$ for each edge (u,v), and $x_C = (2/C)^{2(k-1)}$ for each k-cycle C, we satisfy (1). □

3 The First Algorithmic Version

In this section, we present a theorem which seems to capture all applications to which Beck's technique in [5] will apply.

In what follows, $\mathcal{F} = \{f_1,...,f_m\}$ is a set of independent random trials. $\mathcal{A} = \{A_1,...,A_n\}$ is a set of events such that each A_i is determined by the outcome of the trials in $F_i \subseteq \mathcal{F}$. We say that F_i *intersects* F_j and A_i *intersects* A_j ($A_i \sim A_j$) if $F_i \cap F_j \neq \emptyset$.

For any $f_{j_1},...,f_{j_k} \in F_i$ and any $w_{j_1},...,w_{j_k}$ in the domains of $f_{j_1},...,f_{j_k}$ respectively, we define $\mathbf{Pr}^*(A_i|f_{j_1} \to w_{j_1},...,f_{j_k} \to w_{j_k})$ to be the probability of A_i conditional on the event that the outcomes of $f_{j_1},...,f_{j_k}$ are $w_{j_1},...,w_{j_k}$ respectively. We sometimes just say $\mathbf{Pr}^*(A_i)$ if it causes no ambiguity, always meaning $f_{j_1},...,f_{j_k}$ to be the set of trials already carried out and $w_{j_1},...,w_{j_k}$ to be their outcomes. We allow $k=0$ in which case $\mathbf{Pr}^*(A_i) = \mathbf{Pr}(A_i)$.

Theorem 3.1 *If we have the following:*

1. *for each $1 \leq i \leq n, \mathbf{Pr}(A_i) \leq p$;*
2. *each F_i intersects at most d other F_j's;*
3. *$pd^9 < 1/8$;*
4. *for each $1 \leq i \leq n, |F_i| \leq \omega$;*
5. *for each $1 \leq j \leq m$, the size of the domain of f_i is at most γ, and we can carry out the random trial in time t_1;*

6. *for each* $1 \le i \le n$*,* $f_{j_1}, \ldots, f_{j_k} \in F_i$ *and* $w_{j_1}, \ldots, w_{j_k}$ *in the domains of* $f_{j_1}, \ldots, f_{j_k}$ *respectively, we can compute* $\mathbf{Pr}^*(A_i)$ *in time* t_2*;*

then we have a randomized $\mathrm{O}(m \times d \times (t_1 + t_2) + m \times \gamma^{\omega d \log\log m})$*-time algorithm which will find outcomes of*
$f_1, \ldots, f_m$ *such that none of the events in* $\mathcal{A}$ *hold.*

So, for example, if $d, \omega, \gamma = \mathrm{O}(\log^{1/3} m)$, $t_1 = \mathrm{poly}(\gamma)$, and $t_2 = \mathrm{O}(\gamma^{\omega} \times \mathrm{poly}(\omega))$ (see Remark 3 below), then the running time of our algorithm is $\tilde{O}(m)$.

Remarks:

1. Each f_j lies in at most $d+1$ F_i's, and so $n < m(d+1)/\omega$. Therefore our expression of the running time depends implicitly on n.

2. Note that in condition 4 we can always assume $\omega \leq d$.

3. Taking $k = |F_i|$ in condition 6 implies that given any set of possible outcomes to all the trials in F_i, we need to be able to test whether A_i holds within time $t_2^{'} \leq t_2$. Given that we can carry out this test in time $t_2^{'}$, we can usually take $t_2 = \mathrm{O}(\gamma^{\omega} \times t_2^{'})$ by simply testing each of the possible combinations of outcomes for the remaining trials.

4. In many applications (see for example [5], [3], [10]) we can replace the $\mathrm{O}(\gamma^{\omega d \log\log m})$ term in the running time by an $\mathrm{O}(2^{\log\log m})$ term, thus yielding a polynomial running time for d, ω, γ arbitrarily large. We will elaborate on this in a full version of the paper.

Thus, roughly speaking, as long as an application of the Local Lemma is in some sense well-behaved, and we can replace"$pd < \frac{1}{4}$" by "$pd^9 < \frac{1}{8}$", then condition 3 holds and so we can apply Theorem 3.1 to obtain an efficient constructive algorithm. While the exponent 9 in condition 3 can be somewhat improved, we don't expect that it be made to be near 1 using this algorithm or a simple variant. Thus, condition 3 indicates the approximate limits of the original technique of [5].

As mentioned earlier, Theorem 3.1 can be applied to the main results in [12, 13, 14, 17, 18], as well as many others to obtain, for example, an efficient algorithm to produce a $\Delta + \mathrm{O}(1)$ total colouring of any graph on n vertices and with maximum degree $\Delta = \mathrm{o}(\log^{1/3} n)$. However, it is not strong enough to apply to Theorems 2.1 and 2.2. For these, we must apply the techniques in later sections.

The algorithm and proof are essentially the same as those in [5] (see also [3]). The only new ideas are to consider the conditional probabilities $\mathbf{Pr}^*(A_i)$ and to occassionally "undo" a trial. We include an outline here for completeness and to introduce the ideas required in later sections.

Proof In our **First Sweep**, we carry out the trials $f_1, ..., f_t$ in sequential order. After each trial f_j, we compute $\mathbf{Pr}^*(A_i)$ for each i such that $f_j \in F_j$. If $\mathbf{Pr}^*(A_i) > p^{2/3}$, then we say that A_i is *dangerous*, and we (a) "undo" f_j - i.e. we cancel its outcome and carry out the trial again at a later time; (b) freeze f_j and all other remaining trials in F_i - i.e. we will not carry out those trials during this sweep and so when it comes to their turn we will skip them.

Note that at the end of the First Sweep, $\mathbf{Pr}^*(A_i) \leq p^{2/3}$ for all i. Therefore, by the Local Lemma (Symmetric Case), upon carrying out the remaining trials, $\mathbf{Pr}^*(\overline{A_1} \wedge ... \wedge \overline{A_n}) > 0$. That is, there is a feasible solution extending the partial solution given by the trials that have already been carried out. We will see that so few of the events became dangerous, that it is now nearly feasible to find the good set of outcomes for the remaining trials using exhaustive search.

Claim 3.2 *For each $1 \leq i \leq n$, the probability that A_i becomes dangerous is at most $p^{1/3}$.*

Proof If the probability that $\mathbf{Pr}^*(A_i)$ will ever exceed $p^{2/3}$ is greater than $p^{1/3}$, then $\mathbf{Pr}(A_i) > p$. □

We denote by $\mathcal{H}$ the hypergraph with $V(\mathcal{H}) = \mathcal{F}$, and $E(\mathcal{H}) = \{F_1, ..., F_n\}$, and we denote by $\mathcal{L}$ the line graph of $\mathcal{H}$. $\mathcal{L}^{(a,b)}$ is the graph with vertex set $V(\mathcal{L})$ $(= E(\mathcal{H}))$, and where two vertices are adjacent iff they are at distance exactly a or b in $\mathcal{L}$.

Following the notation of [5, 3], we call $T \subseteq E(\mathcal{H})$ a $(1,2)$-*tree* if the subgraph induced by T in $\mathcal{L}^{(1,2)}$ is connected. We call $T \subseteq E(\mathcal{H})$ a $(2,3)$-*tree* if the subgraph induced by T in $\mathcal{L}^{(2,3)}$ is connected *and* no two vertices of T are adjacent in $\mathcal{L}$ (i.e. no two vertices intersect in $\mathcal{H}$). We call an (a,b)-tree dangerous if all of its vertices correspond to dangerous events.

The key observation is this: No A_i intersects two events which "belong" to different maximal dangerous $(1,2)$-trees. Thus, we can deal with the frozen trials contained in each maximal dangerous $(1,2)$-tree independently, and so as long as they are sufficiently small, an exhaustive search for each tree of all the combinations of possible outcomes of the corresponding trials is feasible. □

Claim 3.3 *With probability at least $\frac{1}{2}$, there are no dangerous $(1,2)$-trees of size greater than $d\log_2 m$.*

Proof The proof follows that of Lemma 2.1 of [3], and we refer the reader to that paper for more details.

It is straightforward to show that every dangerous $(1,2)$-tree of size dK contains a dangerous $(2,3)$-tree of size K. For each f_i, the number of $(2,3)$-trees of size K in $\mathcal{H}$ that f_i lies in is at most $(ed^3)^K$. The hyperedges F_i of $\mathcal{H}$ lying in any such tree are disjoint. Therefore, by Claim 3.2, the probability that all of the events A_i corresponding to these hyperedges become dangerous is at most $(p^{1/3})^K$. Thus, the expected number of dangerous $(2,3)$-trees of size K is at most $m(ed^3p^{1/3})^K$ which is less than $\frac{1}{2}$ for $K = \log m$. The claim now follows from Markov's Inequality. □

If after the First Pass we have any dangerous $(1,2)$-trees of size greater than $d\log_2 m$, then we repeat the First Pass. The expected number of repetitions is constant, and each repetition takes time $O(m\times d\times(t_1+t_2))$. At this point, an exhaustive search for the satisfactory outcomes to the frozen trials corresponding to each dangerous $(1,2)$-tree takes time $O(\gamma^{\omega d\log m})$. Thus, if d,γ,ω are constant, this can be done in polynomial time. To improve the running time, we run a Second Pass in the same manner as the First Pass, where we carry out the frozen events in sequence, and an event becomes dangerous if its conditional probability exceeds $p^{1/3}$. Within an expected linear number of repetitions of the Second Pass, there will be no dangerous $(1,2)$-trees of size greater than $d\log\log m$, and so we can complete our exhaustive search of each one in time $O(\gamma^{\omega d\log\log m})$. Thus the total expected running time is $O(m\times d\times t_1\times t_2 + m\times\gamma^{\omega d\log\log m})$. □

4 b-Frugal Colouring

4.1 $\beta = 3$

We now find an algorithm to construct 3-frugal colourings as guaranteed by Theorem 2.1. That is, a polytime algorithm which will provide a 3-frugal colouring of any graph G on n vertices and with maximum degree $\Delta = O(\log^{1/3} n)$, using $O(\Delta^{3/2})$ colours. We first observe that Theorem 3.1 fails to apply here since Condition 3 fails to hold.

We proceed as follows. We begin with $20\Delta^{3/2}$ colours and assume Δ to be sufficiently large. For each vertex v we maintain lists Bad_v of forbidden colours and L_v of available colours.

During **Phase 1**, we colour the vertices one-at-a-time, giving each vertex v in turn a uniformly random colour from L_v. When v receives a colour c, we place c into Bad_u for each uncoloured $u \in N_v$, and if neccesary, we remove c from L_u, replacing it with a new colour. More specifically, we initialize $L_v = \{1, ..., 12\Delta^{3/2}\}$ for each v, and whenever a colour c is added to Bad_v, if $c \in L_v$ then we remove c from L_v and add to L_v the lowest colour in $\{1, ..., 20\Delta^{3/2}\} - (L_v \cup \mathrm{Bad}_v)$. Note that this guarantees that no 2 adjacent vertices will receive the same colour, i.e. no Type A events will hold.

We prevent Type B events from holding in a similar manner. For each $\{u_1, u_2, u_3\}$ in a common neighbourhood, if 2 of them, say u_1, u_2 ever receive the same colour c, then we place c in Bad_{u_3} and update L_{u_3} accordingly.

The only concern here is that Bad_v might grow too large for some vertex v; if $|\mathrm{Bad}_v| > 8\Delta^{3/2}$ then we will no longer be able to keep $|L_v| = 12\Delta^{3/2}$. Note that at most Δ colours will enter Bad_v because of Type A events, but it is possible that every colour enters Bad_v because of Type B events. In fact, if enough colours were available, up to $\frac{\Delta(\Delta-1)}{2}$ could enter Bad_v. However, the expected number which will enter is at most $\Delta\binom{\Delta}{2} \times \frac{1}{16\Delta^{3/2}} = \frac{\Delta^{3/2}}{32}$, and in fact we can show that the probability that $|\mathrm{Bad}_v|$ exceeds $\Delta^{3/2}$ is less than $e^{-\theta(\Delta)}$.

We say that v is *dangerous* if $|\mathrm{Bad}_v| > 3\Delta^{3/2}$. If a vertex becomes dangerous, then we undo the last trial and freeze it along with v and all vertices within distance 2 of v, delaying their colourings until later phases. Note that this ensures that Bad_v will not increase any further during Phase 1.

The algorithm then proceeds similarly to that in Section 2. In **Phase 2** we repeat this process on uncoloured vertices, this time v becomes dangerous if $|\mathrm{Bad}_v|$ exceeds $6\Delta^{3/2}$, and again in this case we freeze all vertices within distance 2. In **Phase 3** we use exhaustive search to find a satisfactory colouring for the remaining uncoloured vertices. The following property is important:

Property 4.1 *Suppose that at the end of either Phase 1 or Phase 2, we assign to each uncoloured vertex v, a uniformly random colour from L_v. Then the probability of any Type A event is either 0 or $1/(12\Delta^{3/2})$, and the probability of any Type B event is either 0 or $1/(12\Delta^{3/2})^2$.*

Proof $\mathbf{Pr}^*(A_{u,v}) = 1/(12\Delta^{3/2})$ if either u or v is uncoloured and $\mathbf{Pr}^*(A_{u,v}) = 0$ otherwise. $\mathbf{Pr}^*(B_{u_1,u_2,u_3}) = 1/(12\Delta^{3/2})^2$ if at most one of u_1, u_2, u_3 is coloured and $\mathbf{Pr}^*(B_{u_1,u_2,u_3}) = 0$ otherwise. □

Property 4.1 implies the existence of a satisfactory completion of our colouring, as in the proof of Theorem 2.1.

The analysis of Phase 1 is a little more delicate than that in Section 2. Recall that it was important in the proof of Claim 3.3 that if $F_{i_1}, ..., F_{i_K}$ are disjoint then the events that $A_{i_1}, ..., A_{i_K}$ become dangerous are independent. The analogous property does not hold here. The problem is that the colour assigned to v affects L_u for each u adjacent to v, and so can eventually affect L_u for every vertex u in the graph. Thus the choices of colours assigned to (virtually) any two vertices in the graph are dependent. Nevertheless, we can still prove:

Claim 4.2 *If $v_1, ..., v_K$ are all at distance at least 5 then the probability that they all become dangerous during a Phase is at most $\left\{ (\frac{1}{400})^{\Delta^{3/2}} \right\}^K$.*

Proof For each v_i, at most Δ colours can enter Bad_{v_i} because they appear on a neighbour of v_i. Therefore, if v_i becomes dangerous, then there must be at least $2\Delta^{3/2} < 3\Delta^{3/2} - \Delta$ colours $c_1^i, ..., c_{2\Delta^{3/2}}^i$ such that each c_j^i appears on two vertices u_j^i, w_j^i both in the neighbourhood of the same neighbour of v_i. Furthermore, because $v_1, ..., v_K$ are all at distance 5, the u_j^i, w_j^i are all distinct.

For any choice of $c_1^1, u_1^1, w_1^1, ..., c_{2\Delta^{3/2}}^K, u_{2\Delta^{3/2}}^K, w_{2\Delta^{3/2}}^K$, the probability that each u_j^i, v_j^i both get c_j^i is at most $(\frac{1}{12\Delta^{3/2}})^{2K\times 2\Delta^{3/2}}$. Therefore, the expected number of sets of such colours and vertices for $v_1, ..., v_K$ is at most:

$$
\begin{aligned}
& \binom{20\Delta^{3/2}}{2\Delta^{3/2}}^K \left(\Delta \times \binom{\Delta}{2}\right)^{K\times 2\Delta^{3/2}} \left(\frac{1}{12\Delta^{3/2}}\right)^{2K\times 2\Delta^{3/2}} \\
< & \binom{20\Delta^{3/2}}{2\Delta^{3/2}}^K \left(\frac{1}{288}\right)^{K\times 2\Delta^{3/2}} \\
< & \left(\frac{10e}{2\times 288}\right)^{K\times 2\Delta^{3/2}} \\
< & \left(\frac{1}{400}\right)^{K\times \Delta^{3/2}},
\end{aligned}
$$

and so the Claim follows from Markov's Inequality. □

For any vertex v, there are at most Δ^4 vertices within distance 4 of v. Since for Δ sufficiently large, $(\frac{1}{400})^{\Delta^{3/2}} \times (\Delta^4)^9 < \frac{1}{8}$, the analysis in the proof of Theorem 3.1 applies here to show that with high probability the completion of the colouring in Phase 3 can be found by exhaustive search.

4.2 $\beta \geq 4$

For $\beta \geq 4$, we apply essentially the same algorithm as for $\beta = 3$, to obtain a β-frugal colouring using $O(\Delta^{1+\frac{1}{\beta-1}})$ colours. (Recall from the proof of Theorem 2.1 that $C = 12\Delta^{1+\frac{1}{\beta-1}}$.) There is one major complication: the analogue of Claim 4.1 does not hold. This is because if exactly $1 \leq i \leq \beta - 2$ of $u_1, ..., u_\beta$

are coloured, all with the same colour, then $\mathbf{Pr}^*(B_{u_1,...,u_\beta}) = 1/(12\Delta^{3/2})^{\beta-i}$. This makes it difficult to ensure that it is possible to successfully complete the colouring at the end of Phase 1. For example, it is possible that we have $\theta(\Delta^3)$ mutually intersecting Type B events each of which has conditional probability $1/C^2 = \theta(\Delta^{-(2+\frac{2}{\beta-1})})$ (i.e. each of which corresponds to a set of vertices all but two of which have the same colour). If this happens, the Local Lemma will not apply.

There are 2 ways to handle this problem. First we present the easy way:

Method 1: At the beginning of each Phase, we start with a new set of $20\Delta^{1+\frac{1}{\beta-1}}$ colours, thus using a total of $60\Delta^{1+\frac{1}{\beta-1}}$ colours. Note that the analogue of Property 4.1 now holds as if at least one of $u_1, ..., u_\beta$ are coloured then $\mathbf{Pr}^*(B_{u_1,...,u_\beta}) = 0$.

This method applies very well to many graph colouring problems.

Now we describe the difficult way. The reason that we present it is that we feel that the technique will apply to most applications of the Local Lemma.

Method 2: Consider any set $u_1, ..., u_\beta$ all lying in a common neighbourhood. Suppose that exactly $1 \leq i \leq \beta - 2$ of them are coloured. Roughly speaking (i.e. ignoring any conditioning on the fact that exactly i of them are coloured), the probability that all i receive the same colour is at most $1/C^i$, and so $\mathbf{Pr}^*(B_{u_1,...,u_\beta})$ is equal to $1/C^{\beta-i}$ with probability at most $1/C^i$, and is equal to 0 otherwise. Thus, $\mathbf{Exp}(\mathbf{Pr}^*(B_{u_1,...,u_\beta})) \leq 1/C^\beta = \mathbf{Pr}(B_{u_1,...,u_\beta})$.

For each event E (Type A or Type B), we denote by $\mathcal{F}_E$ the set of vertices which determine E (and so $|\mathcal{F}_E| = 2$ or β) and we denote by $\mathcal{H}_E$ the set of events which intersect E, that is the events E' such that $\mathcal{F}_E \cap \mathcal{F}_{E'} \neq \emptyset$.) At any step of the algorithm, we define $\mathbf{Pr}^*(E)$ to be $1/C^{|\mathcal{F}_E|-1}$ if none of $\mathcal{F}_E$ has been coloured; $1/C^{|\mathcal{F}_E|-i}$ if exactly $i \geq 1$ of the vertices in $\mathcal{F}_E$ have been coloured and all have received the same colour; 0 otherwise, i.e. if some two vertices in $\mathcal{F}_E$ have different colours.

Initially, for Δ sufficiently high, $\sum_{E' \in \mathcal{H}_E} \mathbf{Pr}(E') \leq \frac{1}{12}$. It follows that, roughly speaking, the expected value of $P_E^* = \sum_{E' \in \mathcal{H}_E} \mathbf{Pr}^*(E')$ at the end of Phase 1 is at most $\frac{1}{12}$. Furthermore, this sum is a function of $O(\Delta^2)$ colour assignments and so for large Δ, we can show that it is highly concentrated around its expected value.

We follow the same algorithm as in Section 3.1, with one modification: if P_E^* ever exceeds $\frac{1}{8}$ then we call E *dangerous*, uncolour the most recently coloured vertex, and freeze all uncoloured vertices in $\cup_{E' \in \mathcal{H}_E} \mathcal{F}_{E'}$. (Of course, we also treat a vertex v as being dangerous if $|\mathrm{Bad}_v|$ ever exceeds $3\Delta^{1+\frac{1}{\beta-1}}$.)

The second phase follows in the same manner, this time treating E as dangerous if P_E^* ever exceeds $\frac{1}{4}$. Note that at the end of each phase, $p_E^* \leq \frac{1}{4}$ for each E and so the Simple Asymmetric Case of the Local Lemma implies the existence of a successful completion of our vertex colouring. All we need is to prove that with high probability, all the "clumps" of uncoloured vertices are small enough that we can find this completion using exhaustive search. This follows from the following analogues of Claim 4.2

Claim 4.3 *If $v_1, ..., v_K$ are all at distance at least 5 then the probability that they all become dangerous during a Phase is at most* $\left\{ \left(\frac{1}{400}\right)^{\Delta^{1+\frac{1}{\beta-1}}} \right\}^K$.

Claim 4.4 *If $E_1, ..., E_K$ are all Type B events no two of which correspond to vertices of distance less than 5 apart then the probability that they all become dangerous during a Phase is at most* $\left\{ \left(\frac{1}{400}\right)^{\Delta^{1+\frac{1}{\beta}}} \right\}^K$.

Note that each of these lemmas is essentially a proof of the concentration of a set of random variables. The proof of Claim 4.3 follows along the same lines as the proof of Claim 4.2. The proof of Claim 4.4 is much more difficult, and we omit it here, saving it for a full version of the paper.

5 A Sequential Approach

Method 2 of Section 4.2 is very general, and we expect that it will apply to the vast majority of applications of the Local Lemma. However, *proving* that it applies may often be difficult. In a typical application of the Local Lemma, proving that this method yields an efficient algorithm reduces to proving two concentration results analogous to Claims 4.3 and 4.4, which will usually be intuitively true but might be very difficult to prove. In this section we place the method in a more general setting.

Suppose that we have a set $\mathcal{F} = \{f_1, ..., f_m\}$ of independent random trials and events $A_1, ..., A_n$, each A_i determined by the outcomes of the trials in $F_i \subseteq \mathcal{F}$ (sometimes also referred to as F_{A_i}. For each A_i, we set $\mathcal{H}_i = \cup_{F_j \cap F_i \neq \emptyset} F_j$ to be the set of all trials which determine events on which A_i is dependent.

The key is to carry the trials out sequentially in a manner that none of the A_i can hold. To do this, we will have to occasionally change the distribution of a trial f_j (eg. modify the lists Bad_v and L_v). In particular, upon carrying out a trial f_k, we may have to change the distributions of some of the trials f_j for which f_j, f_k both lie in some F_i. Sometimes the distribution of some f_j will become so skewed (eg. Bad_v becomes too large) that we must declare the trial to be *dangerous*, undo the previous trial and freeze all events in $\cup_{f_j \in A_i} F_i$. More specifically, for each trial f_j, we have a set Good_j of good distributions for f_j. If the distribution of f_j ever leaves Good_j then we declare f_j to be dangerous. Note that this ensures that the distribution of every f_j will always be good.

Earlier, we defined $\mathbf{Pr}^*(A_i)$ to be the probability of A_i conditional on the outcomes of any previous trials. However, here we must be a little more careful as the distributions of the trials are shifting. Here, for any event E we define $\mathbf{PR}(E)$ to be the maximum probability of E over all choices of good distributions for the trials in F_E. Similarly, we define $\mathbf{PR}^*(E)$ to be the maximum conditional probability of E over all choices of good distributions for the uncompleted trials in F_E.

Recall that to ensure the *existence* of a set of outcomes to $\mathcal{F}$ for which none of the A_i hold, it is enough that for each i we have $\sum_{F_j \cap F_i \neq \emptyset} \mathbf{Pr}(A_i) \leq \frac{1}{4}$. Here, to be able to *efficiently find* such a set of outcomes, we will require $\sum_{F_j \cap F_i \neq \emptyset} \mathbf{PR}(A_j) \leq \frac{1}{12}$. (In fact, $\frac{1}{4}$ will suffice here rather than $\frac{1}{12}$, but we use $\frac{1}{12}$ for ease of exposition.)

During the First Pass, we will declare A_i to be dangerous if $P_i^* = \sum_{F_j \cap F_i \neq \emptyset} \mathbf{PR}^*(A_j)$ ever exceeds $\frac{1}{8}$, and we will then undo the previous trial and freeze all the trials in $\mathcal{H}_i$. Similarly, during the Second Pass, we declare A_i to be dangerous if P_i^* exceeds $\frac{1}{4}$. Note that this ensures that at the end of the First Pass $P_i^* \leq \frac{1}{8}$ for every i and at the end of the Second Pass $P_i^* \leq \frac{1}{4}$ for every i. This in turn ensures that at the end of the Second Pass, it will be possible to choose outcomes for the uncompleted trials such that none of the events A_i hold.

The main work is to show that with high probability, we can find these outcomes efficiently using exhaustive search. To do this, it suffices to prove the following two lemmas for suitable values of p_1, p_2:

Lemma 5.1 *For any set of trials $f_{i_1}, ..., f_{i_t}$ all sufficiently far apart,*

$$\mathbf{PR}(f_{i_1}, ..., f_{i_t} \text{ all become dangerous}) < p_1^t$$

Lemma 5.2 *For any set of events $A_{i_1}, ..., A_{i_t}$ such that $\mathcal{H}_{i_1}, ..., \mathcal{H}_{i_t}$ are all disjoint,*

$$\mathbf{PR}(A_{i_1}, ..., A_{i_t} \text{ all become dangerous}) < p_2^t$$

Typically, we require p_1, p_2 to be relatively small compared to the maximum number of $\mathcal{H}_j$'s that any $\mathcal{H}_i$ intersects.

Note that roughly speaking, for any A_i, at any point during Phase 1 $\mathbf{Exp}(P_i^*) \leq \frac{1}{12}$ and so Lemma 5.2 is essentially a concentration result. Furthermore, P_i^* is typically a function of a large number of trials each of which has a small effect on P_i^*, and so it should be very highly concentrated. Thus, Lemma 5.2 will usually be at least intuitively true for a very small value of p_2. However, these concentration results can often be notoriously difficult to prove.

Generally, the distribution of a trial f_i is altered by removing values from their domains. Lemma 5.1 is essentially a statement about the concentration of the number of values removed, or more generally the 'amount' by which the distribution must be altered. Again, in a typical application it is intuitively clear that Lemma 5.1 should hold, but proving it can be difficult.

6 A few quick remarks

The method of Sections 4 and 5 applies to provide an efficient algorithm for finding an acyclic edge colouring of a graph with maximum degree Δ using at most 20Δ colours. The proof is very similar to that in Section 4, but is more difficult. One of the complications is in adjusting the technique to apply to the general form of the Local Lemma. We postpone the details for a full version of the paper.

The guarantee that with high probability the algorithms perform well is based on a first moment analysis. Thus, in most cases, the algorithm can be derandomised using the method of conditional probabilities introduced by Erdős and Selfridge [7] to yield polytime deterministic algorithms.

The methods presented in this paper can all be implemented efficiently as parallel algorithms. Again, we postpone the details for a full version of the paper.

7 Shortcomings of the methods

It should be noted that the running time of these algorithms is a polynomial in the number of bad events (see Remark 1 following Theorem 3.1). There are applications of the Local Lemma, most notably to Ramsey Theory, in which the number of bad events is exponential in the size of the input. In such cases, these methods do not yield algorithms which are polynomial in the size of the input.

The other case in which these methods may not apply well is when some bad events are determined by a relatively large number of random trials, i.e. when γ is too large in Theorem 3.1. This is the case, for example, in [6]. Often this problem can be overcome, as mentioned in Remark 4 following Theorem 3.1.

Of course, these methods may not apply if the probability space being considered is not easily expressed as a large set of independent random trials.

Finally, these techniques do not seem to apply to applications of the Lopsided Local Lemma (see [8], [2], and [16]).

References

[1] M. Ajtai, J. Komlós and E. Szemerédi, *An* $O(n \log n)$ *sorting network*, Proceedings of the 15th ACM Symposium on Theory of Computing (1983), 1 - 9.

[2] M. Albert, A. Frieze and B. Reed, *Multicoloured Hamilton Cycles*, Electronic Journal of Combinatorics **2** (1995) #R10.

[3] N. Alon, *A parallel algorithmic version of the Local Lemma*, Random Structures and Algorithms, **2** (1991), 367 - 378.

[4] N. Alon, C. McDiarmid and B. Reed, *Acyclic colouring of graphs*, Random Structures and Algorithms **2** (1991) 277 - 288.

[5] J. Beck, *An algorithmic approach to the Lovász Local Lemma*, Random Structures and Algorithms, **2** (1991), 343 - 365.

[6] A. Broder, A. Frieze and E. Upfal, *Static and Dynamic Path Selection on Expander Graphs: A Random Walk Approach*, Proceedings of the 29th ACM Symposium on Theory of Computing (1997), 531 - 539.

[7] P. Erdös and J. Selfridge, *On a combinatorial game*, J. Comb. Th. (A) **14** (1973), 298 - 301.

[8] P. Erdös and J. Spencer, *Lopsided Lovász Local Lemma and Latin transversals*, Disc. App. Math. **30** (1990) 151 154.

[9] P. Erdös and L. Lovász, *Problems and results on* 3*-chromatic hypergraphs and some related questions*, in: "Infinite and Finite Sets" (A. Hajnal et. al. Eds), Colloq. Math. Soc. J. Bolyai 11, North Holland, Amsterdam, 1975, pp. 609-627.

[10] H. Hind, M. Molloy and B. Reed, *Colouring a graph frugally*, Combinatorica, to appear.

[11] H. Hind, M. Molloy and B. Reed, *Total colouring with* $\Delta + \mathrm{poly}(\log \Delta)$ *colours.* SIAM J. of Comp., to appear.

[12] A. Johansson, *The choice number of sparse graphs*, manuscript.

[13] J. Kahn, *Asymptotically good list-colorings*, J. Combinatorial Th. (A), **73** (1996), 1 - 59.

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

[15] F. T. Leighton, B. Maggs and S. Rao *Packet routing and job-shop scheduling in O(congestion + dilation) steps*, Combinatorica **14** (1994), 167 - 180.

[16] C. McDiarmid, *Hypergraph colouring and the Lovász Local Lemma*, manuscript.

[17] M. Molloy and B. Reed, *A bound on the total chromatic number*, submitted.

[18] B. Reed, $\chi, \Delta,$ *and* ω, Journal of Graph Theory, to appear.