Combinatorics, Probability and Computing (2009) 18, 819–834. © 2009 Cambridge University Press
doi:10.1017/S0963548309990290 Printed in the United Kingdom

Conflict-Free Colourings of Graphs and Hypergraphs

JÁNOS PACH[†] and GÁBOR TARDOS[‡]

EPFL-SB-IMB-DCG, CH-1015 Lausanne, Switzerland
and
Department of Computer Science, City College, 138th Street at Convent Avenue, NY, NY 10031, USA
(e-mail: pach@cims.nyu.edu)

School of Computing Science, Simon Fraser University,
8888 University Drive, Burnaby, BC, V5A 1S6, Canada
and
Rényi Institute, 13–15 Reáltanoda utca Budapest, Hungary
(e-mail: tardos@cs.sfu.edu)

Received 14 May 2009; revised 29 May 2009

A colouring of the vertices of a hypergraph H is called *conflict-free* if each hyperedge E of H contains a vertex of 'unique' colour that does not get repeated in E. The smallest number of colours required for such a colouring is called the *conflict-free chromatic number* of H, and is denoted by $\chi_{\mathrm{CF}}(H)$. This parameter was first introduced by Even, Lotker, Ron and Smorodinsky (*FOCS* 2002) in a *geometric* setting, in connection with frequency assignment problems for cellular networks. Here we analyse this notion for general hypergraphs. It is shown that $\chi_{\mathrm{CF}}(H) \leqslant 1/2 + \sqrt{2m + 1/4}$, for every hypergraph with m edges, and that this bound is tight. Better bounds of the order of $m^{1/t} \log m$ are proved under the assumption that the size of every edge of H is at least $2t - 1$, for some $t \geqslant 3$. Using Lovász's Local Lemma, the same result holds for hypergraphs in which the size of every edge is at least $2t - 1$ and every edge intersects at most m others. We give efficient polynomial-time algorithms to obtain such colourings.

Our machinery can also be applied to the hypergraphs induced by the neighbourhoods of the vertices of a graph. It turns out that in this case we need far fewer colours. For example, it is shown that the vertices of any graph G with maximum degree Δ can be coloured with $\log^{2+\epsilon} \Delta$ colours, so that the neighbourhood of every vertex contains a point of 'unique' colour. We give an efficient deterministic algorithm to find such a colouring, based on a randomized algorithmic version of the Lovász Local Lemma, suggested by Beck, Molloy and Reed. To achieve this, we need to (1) correct a small error in the Molloy–Reed approach, (2) restate and re-prove their result in a *deterministic* form.

[†] Supported by NSF grant CCF-08-30272, and by grants from the NSA, the PSC-CUNY, the Hungarian Research Foundation (OTKA), and the BSF.
[‡] Supported by NSERC grant 329527, and by OTKA grants T-046234, AT-048826, and NK-62321.

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

1. Introduction

Let H be a hypergraph with vertex set $V(H)$ and (hyper)edge set $E(H)$, and let $c : V(H) \to \{1, 2, 3, \ldots\}$ be a colouring of its vertex set. We say that c is a *proper colouring* if no edge $E \in E(H)$ consisting of at least two points is monochromatic. The smallest number of colours required for such a colouring is usually called the *chromatic number* of H, and is denoted by $\chi(H)$. A colouring is a *rainbow colouring* if, for every edge $E \in E(H)$, no two vertices of E receive the same colour. The minimum number of colours, $\chi_{\mathrm{RB}}(H)$, used in a rainbow colouring is the *rainbow chromatic number* of H. For a special class of hypergraphs defined by geometric means, Even, Lotker, Ron and Smorodinsky [16] introduced an intermediate notion: a colouring of H is called *conflict-free* if every non-empty edge $E \in E(H)$ contains a vertex whose colour does not get repeated in E. The minimum number of colours in such a colouring is the *conflict-free chromatic number*, denoted by $\chi_{\mathrm{CF}}(H)$. Obviously, every rainbow colouring is conflict-free and every conflict-free colouring is proper, therefore we have

$$\chi(H) \leqslant \chi_{\mathrm{CF}}(H) \leqslant \chi_{\mathrm{RB}}(H)$$

for every hypergraph H.

For graphs (2-uniform hypergraphs), the above three chromatic numbers coincide. However, they can wildly differ in general. For instance, for the complete 4-uniform hypergraph $K_7^{(4)}$, consisting of all 4-element subsets of a 7-element set, we have $\chi(K_7^{(4)}) = 3$, $\chi_{\mathrm{CF}}(K_7^{(4)}) = 5$, and $\chi_{\mathrm{RB}}(K_7^{(4)}) = 7$.

The study of conflict-free colourings was originally motivated by a frequency assignment problem for cellular networks [5]. We regard the base stations that form the backbone of a network as vertices of a hypergraph H, and the frequencies used by the base stations as colours. The range of communication of a mobile agent (client), that is, the set of base stations it can communicate with, is represented by a hyperedge $E \in E(H)$. To avoid interference among radio signals, we want to assign frequencies to the base stations so that every agent can tune to a frequency that is used by a unique base station within its range. Frequencies are expensive; therefore we want to minimize the number of frequencies used.

With this model in mind, Even, Lotker, Ron and Smorodinsky [16], as well as the authors of many later contributions, considered this question in *geometric* settings [25, 17, 10, 11, 26, 1, 6]. In most cases, H is a hypergraph defined by taking non-empty intersections of a finite point set in $\mathbb{R}^d$ with a family of geometric objects (balls, half-spaces, Jordan regions, boxes), or the dual of such a hypergraph [3]. Various deterministic and randomized, static and on-line versions of the question have also been considered; see [9] for a survey.

The conflict-free chromatic number exhibits some unique properties and raises a number of combinatorial and algorithmic questions, interesting in their own right. The aim of the present paper is to establish some basic results on this new parameter for *general hypergraphs* and for some special classes ('neighbourhood hypergraphs').

Our starting point is the following simple observation that can be obtained by generalizing and strengthening an argument of Cheilaris [9]. The *degree* of a vertex v in a hypergraph is the number of (hyper)edges containing v.

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

Theorem 1.1. *Let H be a hypergraphs with $|E(H)| < \binom{s}{2}$ edges for some positive integer s, and let Δ be the maximum degree of a vertex in H. Then the conflict-free chromatic number of H satisfies:*

(a) $\chi_{\mathrm{CF}}(H) < s$;
(b) $\chi_{\mathrm{CF}}(H) \leqslant \Delta + 1$.

Both bounds are optimal and the corresponding colourings can be found in linear deterministic time.

It follows from part (a) that, for every hypergraph with m edges,

$$\chi_{\mathrm{CF}}(H) \leqslant \frac{1}{2} + \sqrt{2m + \frac{1}{4}}. \tag{1.1}$$

To see that the above bounds cannot be improved, consider the complete graph K_s as a 2-uniform hypergraph. It has $\binom{s}{2}$ edges and maximum degree $s-1$, yet all conflict-free colourings must assign different colours to all of its vertices.

This example also suggests that the presence of many small edges, particularly edges of size *two*, may force the conflict-free chromatic number to be large. If we have a lower bound on the size of the edges of our hypergraph, Theorem 1.1 can be substantially improved.

Theorem 1.2. *For any positive integers t and Γ, the conflict-free chromatic number of any hypergraph in which each edge is of size at least $2t-1$ and each edge intersects at most Γ others, is $O(t\Gamma^{1/t}\log\Gamma)$.*

There is a randomized polynomial-time algorithm to find such a colouring.

It follows that if H is a hypergraph of m edges, each of size at least $2t-1$, then

$$\chi_{\mathrm{CF}}(H) \leqslant C_t m^{1/t}\log m, \tag{1.2}$$

where $C_t > 0$ is a constant depending only on t. Moreover, a colouring with this property can be constructed by a *deterministic* polynomial-time algorithm.

For any fixed $t \geqslant 3$, this bound is better than (1.1). The first interesting question here is to decide whether (1.2) remains true under the weaker assumption that the size of every edge is at least t. If for any fixed $t \geqslant 3$, the answer is yes, then the result is nearly optimal: the conflict-free chromatic number of $K_s^{(t)}$, the complete t uniform hypergraph on s vertices with $m = \binom{s}{t}$ edges ($s \geqslant 2t$) is at least $\frac{s}{2} = \Omega(m^{1/t})$.

In what follows, we analyse the conflict-free chromatic numbers of some special hypergraphs associated with graphs. We have to introduce some notation and terminology.

Given a graph G and a vertex $x \in V(G)$, the *neighbourhood* $N_G(x) = N(x)$ of x is defined as the set consisting of x and all vertices in G connected to x. The set $\dot{N}_G(x) = \dot{N}(x) = N(x) \setminus \{x\}$ is called the *pointed neighbourhood* of x. The *conflict-free chromatic parameter* $\kappa_{\mathrm{CF}}(G)$ of G is defined as $\chi_{\mathrm{CF}}(H)$ for the hypergraph H with

$$V(H) = V(G), \quad E(H) = \{N_G(x) : x \in V(G)\}.$$

The *pointed* version of this parameter, $\dot{\kappa}_{\mathrm{CF}}(G)$, is defined analogously, except that instead of H we have to consider the hypergraph $\dot{H}$ with edge set $E(\dot{H}) = \{\dot{N}_G(x) : x \in V(G)\}$. Studying

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

these parameters for a given graph (network) G is not irrelevant to the frequency assignment problem described at the beginning of this paper.

We start with an example. Let K_s' be the graph obtained from the complete graph K_s on s vertices by subdividing each edge with a new vertex. Each pair of the s original vertices form the pointed neighbourhood of a new vertex, so all original vertices must receive different colours in any conflict-free colouring of the corresponding hypergraph $\dot{H}$. Thus, we have $\dot{\kappa}_{\mathrm{CF}}(K_s') \geqslant s$ and it is easy to see that equality holds here. On the other hand, K_s' is bipartite and any proper colouring of a graph is a conflict-free colouring of the hypergraph formed by the neighbourhoods of its vertices. This shows that $\kappa_{\mathrm{CF}}(K_s') = 2$, for any $s \geqslant 2$.

The example illustrates that the pointed conflict-free chromatic parameter of a graph cannot be bounded from above by *any* function of its non-pointed variant. For many other graphs, the latter parameter is larger. For instance, let H denote the graph obtained from the complete graph K_4 by subdividing a single edge with a vertex. It is easy to check that $\kappa_{\mathrm{CF}}(H) = 3$, while $\dot{\kappa}_{\mathrm{CF}}(H) = 2$. However, it is not difficult to verify that

$$\kappa_{\mathrm{CF}}(G) \leqslant 2\dot{\kappa}_{\mathrm{CF}}(G), \tag{1.3}$$

for any graph G. This inequality holds, because in a conflict-free colouring of the pointed neighbourhoods, each neighbourhood $N(x)$ also has a vertex whose colour is not repeated in $N(x)$, unless x has degree *one* in the subgraph spanned by one of the colour classes. One can fix these offending neighbourhoods by carefully splitting each colour class into two.

Cheilaris [9] proved that the pointed conflict-free chromatic parameter of any graph with n vertices satisfies $\dot{\kappa}_{\mathrm{CF}}(G) \leqslant 2\sqrt{n}$. Inequality (1.1) provides a slightly better bound.

Corollary 1.3. *The pointed conflict-free parameter of any graph G of n vertices satisfies*

$$\dot{\kappa}_{\mathrm{CF}}(G) \leqslant \frac{1}{2} + \sqrt{2n + \frac{1}{4}}.$$

This bound is asymptotically tight, as shown by the graph K_s' defined above, which has $n = \binom{s}{2} + s$ vertices and for which $\dot{\kappa}_{\mathrm{CF}}(K_s') = s$.

Using Theorem 1.2 the last result can be substantially improved if we assume that the minimum degree $\delta = \delta(G)$ of the vertices in G is not too small.

Corollary 1.4. *The pointed conflict-free chromatic parameter of any graph G with n vertices and minimum degree $\delta(G) \geqslant 2t - 1$ ($t \geqslant 3$ integer) satisfies $\dot{\kappa}_{\mathrm{CF}}(G) = O(tn^{1/t} \log n)$.*

It is an interesting open problem to decide whether the last statement remains true for all graphs G with n vertices and minimum degree $\delta(G) \geqslant t$. For example, does $\dot{\kappa}_{\mathrm{CF}}(G) = o(\sqrt{n})$ hold for all graphs on n vertices with minimum degree 3?

We can also bound the pointed conflict-free chromatic parameter of a graph G using Theorem 1.2 and setting $t = \lceil \delta/2 \rceil$ and $\Gamma = \Delta^2$, where δ and Δ denote the minimum and maximum degrees of G, respectively. If these parameters are not too widely separated, we obtain a polylogarithmic bound.

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

Corollary 1.5. *The pointed conflict-free chromatic parameter of any graph G with maximum degree Δ satisfies $\chi_{CF}(G) = O(\log^2 \Delta)$, provided that the minimum degree δ of G satisfies $\delta = \Omega(\log \Delta)$.*

One may conjecture that there is very little difference between the *pointed* conflict-free chromatic parameter and its *non-pointed* counterpart. After all, the two neighbourhood hypergraphs in question are almost the same: each edge of the second hypergraph can be obtained from the corresponding edge of the first one by adding a single vertex. This may give us slightly more leeway in how to colour the neighbourhoods of vertices of small degree, but this cannot dramatically change the situation. To our surprise, this intuition turned out to be wrong. In sharp contrast with Corollary 1.3, the *non-pointed* conflict-free chromatic parameter of a graph with n vertices is at most polylogarithmic in n, without making any restriction on the degrees.

Theorem 1.6. *The conflict-free chromatic parameter of any graph G with n vertices satisfies $\chi_{CF}(G) = O(\log^2 n)$. The corresponding colouring can be found by a deterministic polynomial-time algorithm.*

The last bound is not far from optimal.

Theorem 1.7. *There exist graphs on n vertices with conflict-free chromatic parameter $\Omega(\log n)$.*

It is an interesting open problem to close the gap between the last two bounds.

For the (non-pointed) conflict-free parameter of graphs of arbitrary size but limited degree, we establish an upper bound which is only slightly worse than the bound in Theorem 1.6.

Theorem 1.8. *The conflict-free chromatic parameter of any graph G with maximum degree Δ satisfies $\chi_{CF}(G) = O(\log^{2+\varepsilon} \Delta)$ for any $\varepsilon > 0$. The corresponding colouring can be found by a deterministic polynomial-time algorithm.*

Unlike in Corollary 1.5, in the last statement we did not have to make any assumption on the *minimum* degree of the vertices. All logarithms used in this paper are meant to be *base* 2.

Theorems 1.1 and 1.2 are proved in Sections 2 and 3, respectively. Section 4 contains the proofs of Theorems 1.6 and 1.7. The existential part of Theorem 1.8, based on Lovász's Local Lemma [15], is presented in Section 5.

Turning the existence proof of Theorem 1.8 into a constructive one poses a challenge. The difficulty lies in making the use of the Lovász Local Lemma efficient. Following the pioneering work of Beck [7], several algorithmic versions of the Local Lemma have been developed [2, 20, 12, 13, 19, 21]. A very recent *randomized* algorithm of [24] also applies here.

To obtain a *deterministic* algorithm, we have to use the approach of Molloy and Reed [20]. However, we have to make several changes in the original argument to avoid the possible pitfalls. We list the problems we have to deal with.

(1) The theorem of Reed and Molloy requires a somewhat stronger assumption than Lovász's condition $dp < e^{-1}$ in the original lemma. To satisfy this stronger inequality, we have to *modify the values* of our parameters.

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

(2) For the algorithm, we need to be able to *compute* the probability of the bad events conditioned on some of the variables being fixed. In Section 7, we give an efficient algorithm for this purpose.

(3) The proof of Reed and Molloy [20] has a small *error*. This problem does not affect most applications of the theorem, where the underlying random variables have a bounded range and take each of their values with a probability bounded away from 0. We use the geometric distribution; therefore, in our case these conditions are not satisfied. In Section 6, we outline the Molloy–Reed result, describe a problem with its original proof, and restate and prove the result in a correct form (Theorem 6.1), explicitly dealing with the small probabilities in the distribution of the random variables.

(4) Finally, the Molloy–Reed theorem claims only the existence of an efficient *randomized* algorithm and not a *deterministic* one. They mention that in most applications their algorithm can be derandomized. To avoid dealing with the derandomization separately, our Theorem 6.1 (Section 6) is formulated and proved as a general statement, which guarantees the existence of a deterministic algorithm. (As usual, this algorithm can be best understood as a derandomized version of a more natural probabilistic one.)

The constructive proof of Theorem 1.8, based on our Theorem 6.1, is presented in Section 7.

2. Proof of Theorem 1.1

Our argument is based on Cheilaris's proof [9] of the fact that the pointed conflict-free chromatic parameter of any graph with n vertices is at most $2\sqrt{n}$.

First we establish (b). Consider a hypergraph H with maximum degree at most Δ. Order the vertices of H arbitrarily, and one by one assign colours to them from a set of $\Delta + 1$ colours. Make sure that the colour of the *first* vertex of any edge E does not get repeated in E. When we process a vertex v of degree d, this requirement may exclude at most d colours. Thus, there is still at least one colour left for v.

To prove (a), consider a hypergraph with fewer than $\binom{s}{2}$ edges. We describe an algorithm which produces a conflict-free colouring of H with at most $s-1$ colours. At step i ($i = 1, 2, \ldots$), we find a vertex v_i of degree at least $s - i$, provided that such a vertex exists. We colour v_i with a new colour, and remove it from H together with all edges containing it. Then we proceed to step $i + 1$. As no other vertex will be coloured to the colour of v_i, all edges removed at step i will end up having a vertex of unique colour, namely v_i. If there is no vertex v_i of degree at least $s - i$, the maximum degree of the vertices of the remaining hypergraph H' is smaller than $s - i$. Now, by part (b) of the theorem, there is a conflict-free colouring of H' with $s - i$ previously unused colours.

This algorithm must terminate at step $s - 1$ or before. Otherwise, the total number of edges removed during the first $s - 1$ steps of the procedure would be at least $\sum_{i=1}^{s-1}(s - i) = \binom{s}{2}$, but H had fewer edges to begin with. If the algorithm terminates at step i, then $i - 1$ colours were used in the earlier steps and at most $s - i$ colours at the last step, so the colouring uses fewer than s colours altogether.

Clearly, the above proofs are algorithmic. Furthermore, they can be implemented in time $O(nm)$ for any hypergraph with n vertices and m edges. □

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

3. Proof of Theorem 1.2

The *geometric distribution* with parameter p ($0 < p \leqslant 1$) is the distribution on positive integers that assigns probability $p(1-p)^{i-1}$ to the value i.

We start with an auxiliary result.

Lemma 3.1. *Let us colour each element of a set V independently, according to the geometric distribution with parameter p. If $|V| \geqslant 2t-1$ for a positive integer t, then the probability that no element of V receives a unique colour (one that is not received by any other element of V) is at most $2(etp)^t$.*

Proof. It will be more convenient to consider the randomized colouring of V as a gradual process. We order the elements arbitrarily. In the first phase, we process the elements of V one by one in this order, and for each element v we make an independent choice. With probability p we colour v with colour 1 and with probability $1-p$ we leave it uncoloured. In phase i, $i \geqslant 2$, we consider the uncoloured elements in their pre-assigned order and for each of them we make another independent choice: with probability p they receive colour i and with probability $1-p$ they remain uncoloured. This process gets repeated as long as there is an uncoloured element. It is easy to verify that at the end of this process the elements of V are coloured independently, each according to the geometric distribution with parameter p.

Let us first assume that $|V| = 2t-1$. Notice that if no colour is unique in V, then there are at most $t-1$ colours used. To bound the probability of this let σ be a partition of V into k non-empty classes. The probability that the partition induced by the colouring is σ is at most p^{2t-1-k}. Indeed, in order to obtain the partition σ, we have to choose to colour at the right phase for each of the $|V|-k$ elements that are not the first ones in their equivalence classes. Since the total number of k-partitions of n elements is $S(2t-1,k) \leqslant k^{2t-1}/k! \leqslant (e/k)^k k^{2t-1}$, the probability of using precisely k colours is at most $X_k = e^k(kp)^{(2t-1)-k}$. We can assume that $p < 1/(2t)$, otherwise the claimed bound is meaningless. This implies that the sequence X_k is exponentially increasing for $k \leqslant t-1$. The total probability of using at most $t-1$ colours is at most $\sum_{k=1}^{t-1} X_k < 2X_{t-1} \leqslant (ept)^t$, proving the claim.

Now assume that $|V| \geqslant 2t$, and let V' consist of the $2t$ elements of V that receive their colours last during the multi-phase colouring process described above. We consider the set V' and the colouring of the elements in $V \setminus V'$ fixed and concentrate on the random colouring of V'. Notice that if no colour is unique in V, then there is at most a single unique colour in V' and thus at most t colours are used in V'. We bound the probability of this similarly to the previous case. For a partition σ of $|V'|$ with k classes, the probability that the random colouring of V' induces this σ is at most p^{2t-k}. The number of k-partitions is bounded by $k^{2t}/k! \leqslant (e/k)^k k^{2t}$, so the probability that V' receives exactly k colours is at most $Y_k = e^k(kp)^{2t-k}$. With the $p < 1/(2t)$ assumption this is exponentially increasing for $k \leqslant t$, and the total probability of V' receiving at most t colours is at most $\sum_{k=1}^{t} Y_k < 2Y_t = 2(etp)^t$, as claimed. □

Proof of Theorem 1.2. We start by selecting $2t-1$ vertices of each edge and delete all vertices not selected for any edge. As a result all edges of the remaining hypergraph have between $2t-1$ and $(2t-1)(\Gamma+1)$ vertices. Once a suitable conflict-free colouring is found for this hypergraph,

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

we can extend it to the original graph by colouring all previously deleted vertices with a single unused colour. We can assume that $t < \Gamma$, as for larger values of t the bound claimed in the theorem becomes worse. For the rest of this proof we assume no edge has more than $2\Gamma^2$ vertices.

Let us colour each vertex of the hypergraph independently according to the geometric distribution with parameter $p = \Gamma^{-1/t}/(30t)$.

By Lemma 3.1, the probability that an edge does not have a unique colour is at most $1/(5\Gamma)$. We set a threshold $T = O(t\Gamma^{1/t} \log \Gamma)$ such that the probability of any given vertex getting a colour above T is less than $1/(20\Gamma^3)$. For an edge e of the hypergraph we let A_f be the event that either the colouring of f is not conflict-free or some vertex of the edge receives a colour above the threshold T. We have $P[A_f] < e^{-1}/(\Gamma + 1)$ and the event A_f is independent from the collection of all other event A_g with the edge g disjoint from the edge f. The Lovász Local Lemma states that under these conditions all the events A_f can be simultaneously avoided. This means the existence of a conflict-free colouring with T colours. □

4. Proofs of Theorems 1.6 and 1.7

Proof of Theorem 1.6. Let G be a graph with n vertices. We colour the vertices of G recursively. First, we find an independent set I_1 and colour all of its vertices with colour 1. This colour will not be used for any other vertex of G. Next we remove all vertices from G that belong to I_1 or have precisely one neighbour in I_1. No matter how we extend this colouring, every vertex that has been removed so far will have a neighbour of unique colour: namely, a neighbour of colour 1. Repeating this procedure for the remaining graph, as long as at least one vertex is left, we define colour classes I_2, I_3, *etc*. Finally, we obtain a partial colouring with independent colour classes $I_1, \ldots, I_k$ such that the neighbourhood of every vertex contains a point of unique colour. Using a new colour for all uncoloured vertices, we obtain a conflict-free colouring of G. Notice that the last colour class typically does not form an independent set in G.

The efficiency of this procedure, as well as the number of colours used, depends on the number of vertices removed at each step. We claim that, by a proper choice of the independent set I, at each step one can remove an at least $\Omega(1/\log n)$ fraction of the remaining vertices. Moreover, such an independent set I can be found in randomized linear and deterministic polynomial time. This yields a deterministic polynomial-time algorithm to construct a vertex colouring of G with $O(\log^2 n)$ colours such that every neighbourhood has a vertex of unique colour.

First we select a maximal independent set S of vertices in G. For the randomized procedure, let I be obtained by picking an integer i uniformly randomly from the interval $0 \leqslant i \leqslant \log n$, and then selecting $I \subseteq S$ by setting $v \in I$ with probability 2^{-i}, independently for each vertex $v \in S$. It is easy to see that, for any vertex w of G, the probability that w gets removed at the first step is $\Omega(1/\log n)$. Indeed, for any vertex $w \in S$, w gets removed if we pick $i = 0$. For any $w \notin S$ that has $j \geqslant 1$ neighbours in S, if we pick $i = \lfloor \log j \rfloor$, then w gets removed with probability at least e^{-2}. Thus, the expected number of removed points is $\Omega(n/\log n)$.

We can turn this randomized procedure into a deterministic one in the standard way: we set i first, and then we decide $v \in I$ separately for each vertex $v \in S$, every time computing the expected value (for the remaining choices) of the number of vertices removed, and making sure that this number does not decrease. This way, we surely find a set I for which the number of removed vertices is at least as large as its expected value. □

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

Proof of Theorem 1.7. A graph G is called *k-super-universal* for some parameter $k \geqslant 1$ if, for any set of vertices $A \subseteq V(G)$ with $|A| \leqslant k$ and for any $B \subseteq A$, there is a vertex $x \in V(G)$, $x \notin A$, which is connected to no element of B, but to all elements of $A \setminus B$.

We claim that if a graph G is k-super-universal, then $\kappa_{\mathrm{CF}}(G) > k/2$. To see this, let us colour the vertices of G with at most $k/2$ colours. We will show that there is a neighbourhood $N(x)$ in which no colour appears precisely once. Let B be the set of all vertices x that have a 'unique' colour, that is, a colour not given to any vertex other than x. Further, let A be the set obtained from B by adding two representative vertices for each 'non-unique' colour. Clearly, $|A| \leqslant k$ and, by super-universality, G has a vertex x not in A which has no neighbour in B and which is connected to every vertex in $A \setminus B$. Clearly, each colour occurring in $N(x)$ appears at least twice.

To show the existence of super-universal graphs, consider the random graph $G = G(n, 1/2)$ on n vertices with edge probability $1/2$. It is well known (and easy to show) that G is almost surely *k-super-universal* for some $k = \Omega(\log n)$. This establishes the existence of n-vertex graphs G with $\kappa_{\mathrm{CF}}(G) = \Omega(\log n)$. □

5. Proof of existence for Theorem 1.8

Let $f(\Delta)$ denote the maximum of $\kappa_{\mathrm{CF}}(G)$ over all graphs G with maximum degree at most Δ. As the chromatic number of any such graph is at most $\Delta + 1$, we have $f(\Delta) \leqslant \Delta + 1$.

To prove the much stronger bound on $f(\Delta)$ claimed in Theorem 1.8, we use recursion.

Let G be a graph with maximum degree at most $\Delta \geqslant 6$. We colour the vertices of G independently, according to the geometric distribution with parameter $q = 1/(33 \log \Delta)$. We use the Lovász Local Lemma [4] with the following two types of 'bad' events.

(1) For any vertex v of degree at least $2t - 1$, where $t = 3 \log \Delta$, let B_v denote the event that the neighbourhood of v has no unique colour.
(2) For any vertex v, let B'_v denote the event that the colour of v is larger than $100 \log^2 \Delta$.

The probability of each of these events is at most $p = 1/(6\Delta^2)$. For the events B_v, this follows from Lemma 3.1.

Since the vertices were coloured independently, the events B_v and B'_v are independent of the collection of events B_x and B'_y whenever x is at distance at least 3 from v and y is at distance at least 2 from v. As the maximum degree in G is Δ, each bad event is independent from all but fewer than $d = 2\Delta^2$ other bad events. We have $dp < e^{-1}$. In view of the Local Lemma, this implies that there exists a colouring that avoids all bad events. Let us fix such a colouring χ. This is a colouring with at most $100 \log^2 \Delta$ colours, for which the neighbourhood of any vertex of degree at least $2t - 1$ has a unique colour.

We use recursion to fix the potential problems with the neighbourhoods of small-degree vertices. Let G' be the subgraph of G induced by the vertices whose degrees in G are smaller than $2t - 1$. Clearly, the maximum degree of G' is at most $\Delta' = 2t - 2$. Let χ' be a (not necessarily proper) vertex colouring establishing $\kappa_{\mathrm{CF}}(G') \leqslant f(\Delta')$. This colouring uses at most $f(\Delta')$ colours and the neighbourhood of any vertex has a unique colour.

First, extend χ' to a vertex colouring χ'' of G by adding a new colour to all vertices of G that do not belong to G'. Then we combine χ and χ'' by assigning to a vertex v of G the pair $(\chi(v), \chi''(v))$.

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

This combined colouring uses at most $100\log^2\Delta(f(\Delta')+1)$ colours. In view of the properties of χ, the neighbourhood of each vertex v with degree larger than Δ' has a unique colour. Because of the properties of χ'', the same is true for the remaining (low-degree) vertices. Thus, we obtain $\kappa_{\mathrm{CF}}(G) \leqslant 100\log^2\Delta(f(\Delta')+1)$, and, since G was an arbitrary graph with maximum degree at most Δ, we have

$$f(\Delta) \leqslant 100\log^2\Delta(f(6\log\Delta-2)+1).$$

This recursion solves to $f(\Delta) = O(\log^2\Delta\log^{2+\varepsilon}\log\Delta) = O(\log^{2+\varepsilon}\Delta)$, for any $\varepsilon > 0$. □

6. Algorithmic Local Lemma

Before fixing the problems related to the Molloy–Reed algorithm, listed at the end of the Introduction, we give a brief overview of this algorithm and the algorithm of Beck on which it is based.

Both algorithms assume that the probability space under consideration is determined by mutually independent random variables and each of the bad events B_i is determined by a subset A_i of the variables. Let p be an upper bound on the probability of any one bad event and let d be an upper bound on the number of sets A_j intersected by a single set A_i. The algorithm finds an assignment of values to the random variables that makes none of the bad events occur if $d^9p < 1/8$ (or if a similar alternative inequality holds).

The simplest form of the algorithm consists of two sweeps. In the first sweep, we fix the values of the random variables one by one in an arbitrary order. Each variable is assigned a random value according to its distribution. While doing so, we keep track of the conditional probability of each bad event (that is, the probability that it occurs if we finish the first sweep by keeping the values of all variables already fixed and choosing the values of the remaining variables according to their probabilities). We also choose a threshold $p < T < 1$ and proclaim a bad event *dangerous* if its probability gets T or higher.

The algorithms of Beck and Molloy–Reed differ in their treatment of the dangerous events. If the bad event B_i becomes dangerous, Beck freezes all the variables in A_i, that is, the variables in A_i that have not yet been fixed will not get fixed during the first sweep. Clearly, the conditional probability of a bad event will not change during the first sweep, after it becomes dangerous. Beck's algorithm is designed for the case when all the elementary random variables are uniform 0–1 variables. Fixing one of them to any value will increase the probability of any event by a factor of at most 2. Thus, dangerous events have a probability between T and $2T$ at the end of the first sweep, while the probability of all other bad events clearly remain below T. It is not hard to show that no bad event becomes dangerous with probability larger than p/T.

Molloy and Reed, however, allow arbitrary random variables. Therefore, they cannot bound the 'jump' in the conditional probability caused by fixing a single random variable. Hence, they 'undo' the fixing of the random variable that caused some bad event to turn dangerous. They still label the offending bad event as dangerous and freeze all its variables not fixed earlier; among them they freeze the last variable that they tried to fix but could not. This guarantees that all bad events, dangerous or otherwise, have probability at most T at the end of the first sweep. They fail to notice, however, that the innocent-looking act of 'undoing' *does increase* the probability of a bad event if we tend to do it in cases when the conditional probability would decrease. For a toy

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

example to illustrate this phenomenon, consider $s \log s$ independent and identically distributed 0–1 random variables F_i, for some large real s, such that $P[F_i = 1] = 1/s$. For every i, let B_i denote the event that $F_i = 1$ and let B_0 denote the event that $F_i = 0$ for all i. The probability of each of these events is around $1/s$. If we try and fix the values of these random variables, most will be fixed to 0, but those F_i whose values we try to fix to 1 make the corresponding B_i dangerous. Thus, the value of none of the variables will be fixed to 1 during the first sweep. This makes B_0 become dangerous with overwhelming probability.

In what follows, we restate the result of Beck, Molloy, and Reed in a correct form, taking care of small probabilities in the distribution and constructing a deterministic algorithm.

Theorem 6.1. *Let $\mathcal{F} = \{F_1, \ldots, F_m\}$ be a collection of mutually independent discrete random variables. For $1 \leqslant i \leqslant n$, let A_i be a subset of $\mathcal{F}$ and let B_i be an event determined by the values of the variables $F_j \in A_i$. Assume that:*

(1) *for each B_i, we have $P[B_i] \leqslant p$,*
(2) *each A_i intersects at most d other A_j,*
(3) *the range of a variable F_i contains at most k values and the probability of each of them is at least δ,*
(4) *$|A_i| \leqslant s$, for each i,*
(5) *$pd^9 < \delta^2/200$,*
(6) *for each $1 \leqslant i \leqslant n$, $F_{j_1}, \ldots, F_{j_l} \in A_i$, and for any values w_u in the range of F_{j_u} $(1 \leqslant u \leqslant l)$, one can compute the conditional probability $P[B_i | F_{j_1} = w_1, \ldots, F_{j_l} = w_l]$ in time t.*

Then we have a deterministic $O(dkmt + kmn^4 + ntk^{s(d^2+1)\log\log n})$ time algorithm that finds evaluations of the variables F_j such that none of the events B_i occur. If $pd^{10} \log\log n < \delta^2/64$ holds, the running time can be reduced to $O(dkmt + kmn^4)$.

As mentioned above, the proof is based on sweeps fixing the values of *some* but not all random variables. First we establish the properties of a single sweep. Let F_j, A_i, B_i, p, d, k, and δ be as in Theorem 6.1, and assume that they satisfy conditions (1), (2), (3) and (6) there. Let G be the graph defined on the n vertices $B_1, \ldots, B_n$, connecting B_i and B_j by an edge if A_i and A_j intersect. Recall that, by condition (2), this graph has maximum degree at most d.

Lemma 6.2. *For $p < T < 1$ and for any positive integer r satisfying $(T/(4pd^3))^r > n$, one can find suitable values of* some *of the variables F_j in $O(km(dt + n(4d^3)^{r-1}))$ deterministic time, which satisfy the following two conditions.*

(i) *The conditional probability of all events B_i remain below T/δ.*
(ii) *Let G' be the subgraph of G spanned by the vertices B_i that are not fully evaluated, that is, for which A_i contains unevaluated variables. All connected components of G' have at most $(d^2+1)(r-1)$ vertices.*

Proof. Following Beck's approach, we evaluate the variables F_j one by one, always recomputing the probabilities of the events B_i conditioned on the variables evaluated so far. If the probability of an event B_i becomes at least T, we declare this event *dangerous*, and *freeze* all variables in A_i that have not yet been evaluated. We never evaluate frozen variables, and

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

continue until there exists a variable F_j that is neither frozen nor evaluated. This procedure and condition (3) guarantee that at the end of the procedure (i) is true. To satisfy condition (ii), we need to specify how the individual variables are evaluated.

A set of exactly r events B_i that form an independent set in G but span a connected subgraph in G^3, is called a $(2,3)$-*tree*. Elementary calculations show that there are at most $n(4d^3)^{r-1}$ (2,3)-trees and they can be efficiently enumerated. We call a (2,3)-tree *dangerous* if all of its vertices are dangerous. The *probability of a* $(2,3)$-*tree* at a given moment is the probability that all its elements will hold, conditioned on the values of the variables already evaluated. This is simply the product of the conditional probabilities of the elements of the (2,3)-tree. We maintain the sum S of the probabilities of all (2,3)-trees throughout the algorithm. When evaluating a variable F_j, we choose a value that does not lead to an increase of S. As S is the expected number of (2,3)-trees with all their elements satisfied, the linearity of expectations ensures that such a choice is possible.

The probability of a (2,3)-tree at the start of the algorithm is at most p^r, by condition (1). Thus, the inequality $S \leqslant n(4pd^3)^r < T^r$ holds at the start of the algorithm, and, by our choice of the evaluations, it also holds at the end. The probability of a dangerous (2,3)-tree is at least T^r, so no dangerous (2,3)-tree is created during the algorithm.

To prove condition (ii), consider a component C of G', and let C' be a maximal independent set of dangerous vertices in C. It is easy to see that C' is connected in G^3. If $|C'| \geqslant r$ holds, one can find a subset of C' of size exactly r that is still connected in G^3. That subset would be a dangerous (2,3)-tree, a contradiction. Thus $|C'| < r$. But every vertex of C is connected to a dangerous vertex of C and every dangerous vertex of G is connected to an element of C'. So the degree bound on G implies the bound on $|C|$ claimed in (ii).

To estimate the running time note that we evaluate a subset of the m independent variables by trying each of at most k possible values for them. At every try we have to update the conditional probability of the at most $d+1$ events affected, and update S by recomputing the probabilities of the $(2,3)$-trees. □

Proof of Theorem 6.1. Setting $T = 8pd^4$ and $r = \lfloor \log n / \log(2d) \rfloor + 1$, Lemma 6.2 gives us an $O(dkmt + kmn^4)$ algorithm for evaluating some of the variables in such a way that the conditional probability of no event B_i is more than $p' = 8d^4p/\delta$. For some of the events B_j, all variables in A_j will be evaluated in this first sweep, and, as $p' < 1$, these events B_i do not occur. The remaining events B_i span the subgraph G' of G. By Lemma 6.2, all connected components of G' have fewer than $n' = (d^2+1)r$ vertices.

As the events in different components do not share variables, we can apply Lemma 6.2 to each component separately. For this second sweep we set $T' = 8d^4p'$ and $r' = \lfloor \log n' / \log(2d) \rfloor + 1$. Without increasing the bound on the running time, we evaluate some more random variables such that the conditional probability of all events B_i are at most $p'' = 8p'd^4/\delta$ and the components of the subgraph G'' of G spanned by the events B_i still not fully evaluated have size at most $n'' = (d^2+1)\log n' / \log(2d)$. By condition (5), we have $p'' < 1$, so the fully evaluated events B_i do not hold.

Finally, we evaluate the remaining variables in each component separately, by exhaustive search. By condition (4), each component contains at most $n''s$ variables, and, by condition (3), each of those variables has at most k possible values. We can test in time t if an event B_i

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

occurs under a given evaluation. To prove that the exhaustive search will actually find a solution for which none of the events B_i occur, we use the Local Lemma. The setting of our parameters and condition (5) ensures that the condition $(d+1)p'' < e^{-1}$ of the Local Lemma is satisfied.

In the case when $pd^{10} \log\log n < \delta^2/64$, the expected number of the events B_i satisfied in any single component of G'' is at most $p''n'' < 1$. Therefore, we can replace the costly exhaustive search by a sequential evaluation of the remaining random variables, each time making sure that this expected number does not grow. □

7. Efficient algorithm for conflict-free colouring

A forthcoming paper of Moser and Tardos [24], based on [23], presents a general algorithmic version of the Local Lemma. This version applies without any significant restrictions, and it provides an efficient randomized algorithm to find the right setting of the underlying independent random variables, for which no 'bad' events occur, whenever the existence of such a setting is guaranteed by the Local Lemma. This yields an efficient *randomized* algorithm to find a vertex colouring with the properties claimed in Theorem 1.8. This process also parallelizes well, yielding an $O(\log^2 n)$ time randomized parallel algorithm with a polynomial number of processors for the same job.

It seems, however, that the same process does not lend itself easily to derandomization, unless the bad events are independent from all but a constant number of other bad events. Unfortunately, this condition is not satisfied in our case. To obtain a *deterministic* algorithm for finding a colouring with the properties described in Theorem 1.8, we have to turn to Theorem 6.1.

Theorem 7.1. *For any $\varepsilon > 0$, there exists a polynomial-time deterministic algorithm for finding a vertex colouring of any graph G with maximum degree Δ, which shows that $\kappa_{\mathrm{CF}}(G) = O(\log^{2+\varepsilon} \Delta)$.*

Proof. As in the proof of Theorem 1.8, we use recursion. If Δ is small enough, we properly colour G with $\Delta + 1$ colours and this vertex colouring serves as our conflict-free colouring of the neighbourhoods. For every sufficiently large Δ, we define a threshold $\Delta' = \Theta(\log \Delta)$ and find a vertex colouring χ with $O(\log^2 \Delta)$ colours, for which the neighbourhood of any vertex of degree at least Δ' has an element of unique colour. Then we use recursion on the subgraph of G spanned by the vertices of degree smaller than Δ' to obtain a colouring χ', which is conflict-free on the neighbourhoods. Finally, we extend χ' by a single new colour given to the high-degree vertices, and obtain our final colouring as the product of this extended colouring and χ.

The challenge is to turn the probabilistic existence proof for the colouring χ based on the Local Lemma into an efficient algorithm for finding χ.

We set $t = 30 \log \Delta$, $\Delta' = 2t - 1$, and $q = 1/(11t)$. Consider a random variable F with geometric distribution of parameter q, and let k be the smallest integer with $P[F > k] \leqslant 1/(2\Delta + 2)$. Note that $k < \log(2\Delta + 2)/q = O(\log^2 \Delta)$. Let F' be the distribution of F conditioned on $F \leqslant k$. This distribution has a range of k possible values, and the least likely value, k, is taken with probability at least $\delta = q/(2\Delta + 2)$.

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

Let us take independent random variables $\chi(v)$ distributed as F', for each vertex v of G. The value of $\chi(v)$ is the colour of the vertex v. For any vertex v of degree at least Δ', let B_v be the event that no unique colour appears in the neighbourhood of v. Clearly, B_v is determined by $\chi(w)$ for $w \in N(v)$. As G has maximum degree Δ, any neighbourhood $N(v)$ has at most $s = \Delta + 1$ elements and intersects at most $d = \Delta^2$ other neighbourhoods $N(w)$.

We could bound the probability of B_v by $2(eqt)^t < 2^{-t} = \Delta^{-30}$, using Lemma 3.1, if the distributions of the colours $\chi(v)$ were according to the geometric distribution with parameter q. The true distribution is obtained by conditioning on the value being at most k. With probability at least $1/2$, the colours of the vertices in $N(v)$ would still not exceed k, even if we allowed the unbounded geometric distribution, so the probability of B_v is at most twice what it would be with the geometric distribution: $p = 2\Delta^{-30} \geqslant P[B_v]$.

We can now apply Theorem 6.1 to the random variables $\chi(v)$ and to the events B_v. Conditions (1)–(4) are satisfied. Condition (5) is satisfied for large enough Δ. Condition (6) is also satisfied for some t polynomial in the size of G; see Lemma 7.2 below. By Theorem 6.1, we find the required vertex colouring χ. The running time of the algorithm is polynomial, provided that $d = O(\log\log n)$. In the case when $d > 1000 \log\log n$, the stronger inequality $pd^{10}\log\log n < \delta^2/64$ also holds. So we obtain a polynomial-time algorithm in every case. □

It remains to prove that the conditional probability of a unique colour is efficiently computable.

Lemma 7.2. *Let a discrete distribution D be given by specifying the probabilities of all values in its range, let s and t be non-negative integers, $x_1, \dots, x_s$, constants, $y_1, \dots, y_t$ independent random variables distributed according to D. The probability of having no value that appears exactly once in the sequence $x_1, \dots, x_s, y_1, \dots, y_t$ can be computed in deterministic polynomial time in s, t and the size of the range of D.*

Proof. We assume without loss of generality that the range of D is $1, \dots, u$, and that the constants x_i belong to this range, for $1 \leqslant i \leqslant s$. We use dynamic programming to compute the probability $P(v, w)$ that there is no value z with $z \geqslant v$ that appears exactly once in the sequence $x_1, \dots, x_s, y'_1, \dots, y'_w$, where $y'_1, \dots, y'_w$ are independent random variables distributed according to the distribution D conditioned on $y'_i \geqslant v$.

We compute $P(v, w)$, for $1 \leqslant v \leqslant u$ and $0 \leqslant w \leqslant t$. It is easy to compute $P(u, w)$, as the distribution is deterministic there. For $P(v, w)$ with $v < u$, one computes the probability $P_{v,w,i}$ that exactly i of the w random variables take the value v, and calculates $P(v, w) = \sum P_{v,w,i} P(v + 1, w - i)$, where the summation extends over all $0 \leqslant i \leqslant w$ with the possible exception of the (at most one) value of i that makes v appear exactly once in the sequence. Finally, $P(1, t)$ is the probability we had to compute. □

Acknowledgement

We are grateful to Panagiotis Cheilaris, Andreas Holmsen, and Radoš Radoičić for many interesting remarks and suggestions. The idea of considering conflict-free colourings for hypergraphs formed by the neighbourhoods of the vertices in graphs emerged in discussions with them at the geometry seminar at NYU.

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

References

[1] Ajwani, D., Elbassioni, K., Govindarajan, S. and Ray, S. (2007) Conflict-free coloring for rectangle ranges using $O(n^{0.382})$ colors. In *Proc. 19th Annual ACM Symposium on Parallelism in Algorithms and Architectures* (SPAA), pp. 181–187.

[2] Alon, N. (1991) A parallel algorithmic version of the local lemma. *Random Struct. Alg.* **2** 367–378.

[3] Alon, N. and Smorodinsky, S. (2006) Conflict-free colorings of shallow discs. In *22nd Ann. ACM Symposium on Computational Geometry* (SoCG), pp. 41–43.

[4] Alon, N. and Spencer, J. (2000) *The Probabilistic Method*, 2nd edn, Wiley, New York.

[5] Aardal, K., van Hoesel, S., Koster, A., Mannino, C. and Sassano, A. (2003) Models and solution techniques for frequency assignment problems. *4OR* **1** 261–317.

[6] Bar-Noy, A., Cheilaris, P. and Smorodinsky, S. (2008) Deterministic conflict-free coloring for intervals: From offline to online. *ACM Trans. Alg.* **4** #44.

[7] Beck, J. (1991) An algorithmic approach to the Lovász local lemma. *Random Struct. Alg.* **2** 343–365.

[8] Behzad, M. (1971) The total chromatic number of a graph: A survey. In *Combinatorial Mathematics and its Applications* (Oxford 1969), Academic Press, London, pp. 1–8.

[9] Cheilaris, P. (2008) Conflict-free coloring. PhD thesis, City University of New York.

[10] Chen, K. (2006) How to play a coloring game against a color-blind adversary. In *Proc. 22nd Annual ACM Symposium on Computational Geometry* (SoCG), pp. 44–51.

[11] Chen, K., Fiat, A., Kaplan, H., Levy, M., Matoušek, J.,Mossel, E., Pach, J., Sharir, M., Smorodinsky, S., Wagner, U. and Welzl, E. (2006) Online conflict-free coloring for intervals. *SIAM J. Comput.* **36** 1342–1359.

[12] Czumaj, A. and Scheideler, C. (2000) A new algorithm approach to the general Lovász local lemma with applications to scheduling and satisfiability problems. In *Proc. 32nd Annual ACM Symposium on Theory of Computing*, ACM, New York, pp. 38–47.

[13] Czumaj, A. and Scheideler, C. (2000) Coloring nonuniform hypergraphs: A new algorithmic approach to the general Lovász local lemma. In *Proc. 9th International Conference 'Random Structures and Algorithms'* (Poznan 1999), *Random Struct. Alg.* **17** 213–237.

[14] Diestel, R. (2005) *Graph Theory*, 3rd edn, Vol. 173 of *Graduate Texts in Mathematics*, Springer, Berlin.

[15] Erdős, P. and Lovász, L. (1975) Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and Finite Sets* (*to Paul Erdős on his 60th birthday*), Vol. II (A. Hajnal *et al.*, eds), North-Holland, Amsterdam, pp. 609–627.

[16] Even, G., Lotker, Z., Ron, D. and Smorodinsky, S. (2003) Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks. *SIAM J. Comput.* **33** 94–136.

[17] Har-Peled, S. and Smorodinsky, S. (2005) Conflict-free coloring of points and simple regions in the plane. *Discrete Comput. Geometry* **34** 47–70.

[18] Hind, H., Molloy, M. and Reed, B. (1997) Colouring a graph frugally. *Combinatorica* **17** 469–482.

[19] Leighton, T., Lu. C.-J., Rao, S. and Srinivasan, A. (2001) New algorithmic aspects of the local lemma with applications to routing and partitioning. *SIAM J. Comput.* **31** 626–641.

[20] Molloy, M. and Reed, B. (1998) Further algorithmic aspects of the local lemma. In *Proc. 30th Annual ACM Symposium on Theory of Computing*, pp. 524–529.

[21] Molloy, M. and Reed, B. (2002) *Graph Colouring and the Probabilistic Method*, Vol. 23 of *Algorithms and Combinatorics*, Springer, Berlin.

[22] Molloy, M. and Salavatipour, M. R. (2005) A bound on the chromatic number of the square of a planar graph. *J. Combin. Theory Ser. B* **94** 189–213.

[23] Moser, R. A constructive proof of the Lovász Local Lemma. In *Proc. 41st Annual ACM Symposium on Theory of Computing*, pp. 343–350.

[24] Moser, R. and Tardos, G. (2009) A constructive proof of the general Lovász Local Lemma. arXiv:0903.0544v3.

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press

[25] Pach, J. and Tóth, G. (2003) Conflict-free colorings. In *Discrete and Computational Geometry*, Vol. 25 of *Algorithms and Combinatorics*, Springer, Berlin, pp. 665–671.
[26] Smorodinsky, S. (2007) On the chromatic number of some geometric hypergraphs. *SIAM J. Discrete Math.* **21** 676–687.
[27] Wegner, G. (1977) Graphs with given diameter and a coloring problem. Technical Report, University of Dortmund.
[28] Woldar, A. (2002) Rainbow graphs. In *Codes and Designs* (Columbus 2000), Vol. 10 of *Ohio State Univ. Math. Res. Inst. Publ.*, de Gruyter, Berlin, pp. 313–322.

https://doi.org/10.1017/S0963548309990290 Published online by Cambridge University Press