Conflict-free colorings of graphs and hypergraphs

János Pach*
EPFL, Lausanne and CCNY, New York

Gábor Tardos†
Simon Fraser University, Burnaby

Abstract

It is shown that, for any $\epsilon > 0$, every graph of maximum degree Δ permits a *conflict-free coloring* with at most $\log^{2+\epsilon} \Delta$ colors, that is, a coloring with the property that the neighborhood $N(v)$ of any vertex v, contains an element whose color differs from the color of any other element of $N(v)$. We give an efficient deterministic algorithm to find such a coloring, based on an algorithmic version of the Lovász Local Lemma suggested by Beck, Molloy and Reed. We need to correct a small error in the Molloy-Reed approach, restate it in a deterministic form, and re-prove it.

The problem can be extended to arbitrary hypergraphs H: a coloring of the vertices of H is called *conflict-free* if each hyperedge E of H contains a vertex whose color does not get repeated in E. Every hypergraph with fewer than $\binom{s}{2}$ edges permits a conflict-free coloring with fewer than s colors, and this bound cannot be improved. Moreover, there is a linear time deterministic algorithm for finding such a coloring. The concept of conflict-free colorings was first raised in a geometric setting in connection with frequency assignment problems for cellular networks.

*Supported by NSF Grant CCF-05-14079, and by grants from NSA, PSC-CUNY, Hungarian Research Foundation OTKA, and BSF. Email: pach@cims.nyu.edu

†Supported by NSERC grant 329527, and by OTKA grants T-046234, AT-048826, and NK-62321. Email: tardos@cs.sfu.edu

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 coloring of its vertex set. A coloring of the vertices of H is called *conflict-free* if every edge $\emptyset \neq E \in E(H)$ contains a vertex whose color does not get repeated in E. The minimum number of colors in such a coloring is the *conflict-free chromatic number*, denoted by $\chi_{\mathrm{CF}}(H)$.

The study of this relatively new hypergraph parameter was motivated by a frequency assignment problem for cellular networks [AHK03]. We think of the base stations that form the backbone of a network as vertices of a hypergraph H, and of the frequencies used by the base stations as colors. 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.

The first paper [ELRS03] in the subject, written by Even, Lotker, Ron, and Smorodinsky (FOCS 2002), as well as the majority of later contributions considered this question in various geometric settings [PT03], [HS05], [Ch06], [ChFK06], [S07], [AEG07], [BCS08]. In most cases, H is a hypergraph defined by taking nonempty intersections of a finite point set in $\mathbf{R}^d$ with a family of geometric objects (balls, half-spaces, Jordan regions, boxes). In other cases, H is the dual of such a hypergraph (e.g., in [AS06]). Various deterministic and randomized static and on-line versions of the question have been considered; see [C08], for a survey. Here we study the same problem for arbitrary graphs.

Given a graph G and a vertex $x \in V(G)$, the *neighborhood* $N_G(x) = N(x)$ 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 neighborhood* 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

$$V(H) = V(G),$$

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

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 $E(\dot{H}) = \{\dot{N}_G(x) : x \in V(G)\}$.

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 neighborhood of a new vertex, so all original vertices must receive different colors in any conflict-free coloring of the corresponding hypergraph $\dot{H}$. Thus, we have $\dot{\kappa}_{\mathrm{CF}}(K'_s) \geq s$ and it is easy to see that equality holds here. On the other hand, K'_s is bipartite and any proper coloring of a graph is a conflict-free coloring of the hypergraph formed by the neighborhoods of its vertices. This shows that $\kappa_{\mathrm{CF}}(K'_s) = 2$, for any $s \geq 2$.

The above 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

$$\kappa_{\mathrm{CF}}(G) \leq 2\dot{\kappa}_{\mathrm{CF}}(G), \qquad (1)$$

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

An elegant argument of Cheilaris [C08] implies that the pointed conflict-free chromatic parameter of any graph with n vertices satisfies

$$\dot{\kappa}_{\mathrm{CF}}(G) \leq 2\sqrt{n}. \tag{2}$$

As is shown by the graph $G = K'_s$ defined above, with $|V(G)| = n = \binom{s}{2} + s$, the order of magnitude of this bound cannot be improved.

It is worth adapting Cheilaris's proof to arbitrary hypergraphs with n edges, because in this setting, with a small modification we can obtain an exact bound. In the last section, we establish

Theorem 1 *(a) Let s be a positive integer. Then every hypergraph with fewer than $\binom{s}{2}$ edges admits a conflict-free coloring with fewer than s colors.*
(b) Every hypergraph of maximum degree Δ admits a conflict-free coloring with at most $\Delta + 1$ colors.

Both bounds are best possible and both colorings can be found in linear deterministic time.

As a consequence, we obtain a slightly stronger version of Cheilaris's result: the pointed conflict-free chromatic parameter of any graph with fewer than $\binom{s}{2}$ vertices or any graph of maximum degree at most $s - 2$ is smaller than s. This bound is asymptotically tight, as is shown by the graph K'_s defined above, which has $\binom{s}{2} + s$ vertices, maximum degree $s - 1$, and for which $\dot{\kappa}_{\mathrm{CF}}(K'_s) = s$.

In the case when we have a lower bound on the size of the edges of our hypergraph, Theorem 1 can be substantially strengthened.

Theorem 2 *For any positive integers t and m, the conflict-free chromatic number of every hypergraph with m edges, each of which has size at least $2t - 1$, is $O(tm^{1/t} \log m)$.*

There are randomized linear time and deterministic polynomial time algorithms to find such a coloring.

The last theorem gives a better bound than Theorem 1, whenever $t \geq 3$, that is, when the size of each edge is at least *five*. In particular, we obtain

Corollary 3 *The pointed conflict-free chromatic parameter of any graph G with n vertices and minimum degree at least 5 satisfies $\dot{\kappa}_{\mathrm{CF}}(G) = O(n^{1/3} \log n)$.*

It is an interesting open problem to decide whether the last statement remains true for all graphs with minimum degree at least *three*. More generally, is it true that the conflict-free chromatic number of every hypergraph with m edges of size at least *three* is $o(\sqrt{m})$. The best lower bound we know is given by the complete 3-uniform hypergraph with s vertices and $\binom{s}{3}$ edges, whose conflict-free chromatic number is $\lceil s/2 \rceil = \Omega(m^{1/3})$.

Cheilaris [C08] proved the same $2\sqrt{n}$ upper bound for the (*non-pointed*) conflict-free chromatic parameter of all graphs with n vertices as for its pointed variant (cp. (1) and (2)). Surprisingly (to us), this bound can be replaced by a polylogarithmic one.

Theorem 4 *The conflict-free chromatic parameter of any graph G with n vertices satisfies $\kappa_{\mathrm{CF}}(G) = O(\log^{2+\varepsilon} n)$, for any $\varepsilon > 0$. Such a coloring can be found by a deterministic polynomial time algorithm.*

Here we show that the last bound is not far from optimal: there exist graphs of n vertices with conflict-free chromatic parameter $\Omega(\log n)$.

A graph G is called *k-super-universal* for some parameter $k \geq 1$ if for any set of vertices $A \subseteq V(G)$ with $|A| \leq k$ 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 color the vertices of G with at most $k/2$ colors. We will show that there is a neighborhood $N(x)$ in which no color appears precisely once. Let B be the set of all vertices x that have a "unique" color, that is, a color 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" color. Clearly, $|A| \leq k$ and by the super-universality G has a vertex x not in A which has no neighbor in B and which is connected to every vertex in $A \setminus B$. Clearly, each color 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 $\kappa_{\mathrm{CF}}(G) = \Omega(\log n)$.

It is an interesting open problem to close the gap between the $O(\log^{2+\varepsilon} n)$ upper bound and the $\Omega(\log n)$ lower bound for the maximum conflict-free chromatic parameter of an n-vertex graph.

Theorems 2 and 4 are proved in Sections 2 and 3, respectively. In fact, we establish a much stronger version of Theorem 4, which is stated in the abstract: we show that the bound in Theorem 4 holds not only for all graphs with n vertices, but also for all graphs of *maximum degree n*. (See Theorem 6.) The proof is based on the Lovász Local Lemma [EL75], [D05].

Turning the existence proof of Theorem 6 and Theorem 4 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 [Bec91], several algorithmic versions of the Local Lemma have been developed [A91], [MR98], [CzS00a], [CzS00b], [LLRS01], [MR02]. For our purposes, the version of Molloy and Reed [MR98] is most suitable, but 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' condition $dp < e^{-1}$ in the original lemma. To satisfy this stronger inequality, we have to *modify the values* of our parameters.

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 4, we give an efficient algorithm for this purpose.

3. The proof of Reed and Molloy [MR98] has a small *error*. This problem does not effect 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 4, we outline the Molloy-Reed result, describe a problem with its original proof, and restate and prove the result in a correct form (Theorem 7), 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 7 was formulated as a general statement which guarantees the existence of a deterministic algorithm. (Of course, we agree that this algorithm can be best understood as a derandomized version of a more natural probabilistic one.)

The constructive proof of Theorem 6, based on our Theorem 7, is presented in Section 5.

2 Proof of Theorem 2

The *geometric distribution* with parameter p $(0 < p \leq 1)$ 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 5 *Let us color each element of a set V independently, according to the geometric distribution with parameter p. If $|V| \geq 2t-1$ for a positive integer t, then the probability that no element of V receives a unique color (one that is not received by any other element of V) is at most $(11tp)^t$.*

Proof: It will be more convenient to consider the randomized coloring 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 color v with color 1 and with probability $1-p$ we leave it uncolored. In phase i, $i \geq 2$, we consider the uncolored elements in their preassigned order and for each of them we make another independent choice: with probability p they receive color i and with probability $1-p$ they remain uncolored. This process gets repeated as long as there is an uncolored element. It is easy to verify that at the end of this process the elements of V are colored independently, each according to the geometric distribution with parameter p.

Let $n = |V|$. First we claim that the probability that the above coloring uses at most $n/2$ colors is at most $2(enp/2)^{\lceil n/2 \rceil}$. To see this, let σ be a partition of V into k nonempty classes. The probability that the partition induced by the coloring is σ is at most p^{n-k}. Indeed, in order to obtain the partition σ, we have to choose to color at the right phase for each of the $n-k$ elements that are not the first ones in their equivalence classes. Since the total number of k-partitions of n elements is $S(n,k) \leq k^n/k! \leq (e/k)^k k^n$, the probability of using precisely k colors is at most $X_k = e^k (kp)^{n-k}$. We can assume that $p < 1/n$, otherwise the claimed bound is meaningless. This implies that the sequence X_k is exponentially increasing for $k \leq n/2$. The total probability of using at most $n/2$ colors is at most $\sum_{k=1}^{\lfloor n/2 \rfloor} X_k < 2X_{\lfloor n/2 \rfloor} \leq 2(enp/2)^{\lceil n/2 \rceil}$, as claimed.

Obviously, if no color is unique in V, then there are at most $n/2$ colors, so the above bound applies. This implies that, for $n = 2t$ or $n = 2t-1$, the probability that there is no unique color is at most $2(etp)^t$. If n is large, then the above bound is trivial, so we use the following observation instead. For $1 \leq j \leq n$, let V_j consist of the j elements of V that receive their colors last during the multi-phase coloring process described above. Suppose $n \geq 2t$. If no element of V receives a unique color, then either all colors appearing in V_{2t} are repeated in V_{2t} or all colors in V_{2t-1} are repeated in V_{2t-1}. We can bound the probability of either of these events by $2(etp)^t$, as at any point during the coloring process, the distribution of the partition induced by the coloring of the set of yet uncolored vertices is the same as if the whole procedure was performed for that set only. The claimed bound follows. □

Proof of Theorem 2: Our randomized algorithm finding a suitable coloring assigns a random color to each vertex of the hypergraph independently according to the geometric distribution with parameter $p = m^{-1/t}/(2tt)$. Clearly, this distribution can be sampled in $O(nm)$ time (here n is the number of vertices) and this time is also enough to check if the coloring is conflict-free and does not use too many colors.

By Lemma 5, the probability that an edge does not have a unique color is at most $1/(2m)$. We obtain a conflict-free coloring with probability at least $1/2$. If n is polynomially bounded in m, then with high probability the random coloring uses $O(p^{-1} \log m)$ colors. This completes the proof for this case.

It remains to reduce the general case to the case when the number of vertices is polynomial in the number of edges. We can assume that $t \leq m$, as for larger values of t the bound claimed in the theorem becomes worse. Select $2t - 1$ vertices from each edge of the hypergraph and restrict the hypergraph to the selected vertices. That is, remove all vertices that are not selected for any edge of the hypergraph. The resulting hypergraph has fewer than $2m^2$ vertices, so there exists a conflict-free coloring with the required number of colors. Introducing a new color for all previously removed vertices, we obtain a conflict-free coloring of the original hypergraph.

Finally, to derandomize this algorithm we simply set the color of the vertices one by one making sure that the expected number of edges with no unique color does not increase. The expectation is for the setting the remaining colors randomly, according to the geometric distribution with parameter p truncated at a threshold $T = O(p^{-1} \log m)$. This expectation is computable in polynomial time, see Lemma 10. □

3 Proof of existence for Theorem 4

We prove the following stronger statement.

Theorem 6 *The conflict-free chromatic parameter of any graphs G with maximum degree Δ satisfies $\kappa_{\mathrm{CF}}(G) = O(\log^{2+\varepsilon} \Delta)$, for any $\varepsilon > 0$.*

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

In what follows, we establish a recursive bound on $f(\Delta)$, which is better than the trivial bound $f(\Delta) \leq \Delta + 1$, for large values of Δ.

Let G be a graph with maximum degree at most $\Delta \geq 6$. We color the vertices of G independently, according to the geometric distribution with parameter $q = 1/(66 \log \Delta)$. We use the Lovász Local Lemma [AS00] 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 neighborhood of v has no unique color.
2. For any vertex v, let B'_v denote the event that the color of v is larger than $200 \log^2 \Delta$.

The probability of each of these events is at most $p = 1/(6\Delta^2)$. For the events B_v, this follows by Lemma 5.

Since the vertices were colored 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 coloring that avoids all bad events. Let us fix such a coloring χ. This is a coloring with at most $200 \log^2 \Delta$ colors, for which the neighborhood of any vertex of degree at least $2t - 1$ has a unique color.

We use recursion to fix the potential problems with the neighborhoods 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 coloring establishing $\kappa_{\rm CF}(G') \leq f(\Delta')$. This coloring uses at most $f(\Delta')$ colors and the neighborhood of any vertex has a unique color.

First, extend χ' to a vertex coloring χ'' of G by adding a new color 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 $(\chi(v), \chi''(v))$. This combined coloring uses at most $200 \log^2 \Delta (f(\Delta') + 1)$ colors. In view of the properties of χ, the neighborhood of each vertex v with degree larger than Δ' has a unique color. Because of the properties of χ'', the same is true for the remaining (low degree) vertices. Thus, we obtain $\kappa_{\rm CF}(G) \leq 200 \log^2 \Delta (f(\Delta') + 1)$, and, since G was an arbitrary graph with maximum degree at most Δ, we have

$$f(\Delta) \leq 200 \log^2 \Delta (f(6 \log \Delta - 2) + 1).$$

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

4 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 it is based on.

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 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 $d^9 p < 1/8$ (or if a similar other 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 gets 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 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 7 *Let $\mathcal{F} = \{F_1, \ldots, F_m\}$ be a collection of mutually independent discrete random variables. For $1 \le i \le n$, let A_i be a subset of $\mathcal{F}$ and B_i an event determined by the values of the variables $F_j \in A_i$. Assume that*

1. *for each B_i, we have $P[B_i] \le p$;*
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. *$|A_i| \le s$, for each i;*
5. *$pd^9 < \delta^2/200$;*
6. *for each $1 \le i \le n$, $F_{j_1}, \ldots, F_{j_l} \in A_i$, and for any values w_u in the range of F_{j_u} $(1 \le u \le l)$, 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 $O(dkmt + kmn^4 + mtk^{s(d^2+1)\log\log n})$ time algorithm that finds evaluations of the variables F_j such that none of the events B_i occur. In case $pd^{10} \log\log n < \delta^2/64$ 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 7, and assume that they satisfy the 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 8 *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 $O(km(t+n(4d^3)^{r-1}))$ 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 been evaluated yet. We never evaluate frozen variables and 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. For satisfying 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 $S \leq n(4pd^3)^r < T^r$ 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 $|C'| \geq r$ 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). □

Proof of Theorem 7: Setting $T = 8pd^4$ and $r = \lfloor \log n/log(2d) \rfloor + 1$, Lemma 8 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 8, 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 8 to each component separately. For this second sweep we set $T' = 8d^4p'/\delta$ 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 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. □

5 Efficient algorithm for conflict-free coloring

To complete the proof of Theorem 6, we apply Theorem 7 to the existence proof given in Section 3.

Theorem 9 *For any $\varepsilon > 0$, there exists a polynomial time deterministic algorithm for finding a vertex coloring 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 6, we use recursion. If Δ is small enough, we properly color G with $\Delta + 1$ colors and this vertex coloring serves as our conflict-free coloring of the neighborhoods. For every sufficiently large Δ, we define a threshold $\Delta' = \Theta(\log \Delta)$ and find a vertex coloring χ with $O(\log^2 \Delta)$ colors, for which the neighborhood of any vertex of degree at least Δ' has an element of unique color. Then we use recursion on the subgraph of G spanned by the vertices of degree less than Δ' to obtain a coloring χ' that is conflict-free on the neighborhoods. Finally, we extend χ' by a single new color given to the high degree vertices and obtain our final coloring as the product of this extended coloring and χ.

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

We set $t = 30 \log \Delta$, $\Delta' = 2t - 1$, and $q = 1/(22t)$. Consider a random variable F with geometric distribution of parameter q, and let k be the smallest integer with $P[F > k] \leq 1/(2\Delta + 2)$. Note that $k < \log \Delta / q = O(\log^2 \Delta)$. Let F' be the distribution of F conditioned on $F \leq k$. This distribution has a range of k possible values, and the least likely value, k, is taken with probability at least $\delta = q/(2\Delta + 2)$.

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

We could bound the probability of B_v by $(11qt)^t = 2^{-t} = \Delta^{-30}$, using Lemma 5, if the distributions of the colors $\chi(v)$ 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 colors 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 of what it would be with the geometric distribution: $p = 2\Delta^{-30} \geq P[B_v]$.

We can now apply Theorem 7 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 10 below. By Theorem 7, we find the required vertex coloring 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. $\square$

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

Lemma 10 *Let a discrete distribution D be given by specifying the probabilities of all values in its range, let s and t be nonnegative integers, $x_1, \ldots, x_s$, constants, $y_1, \ldots, y_t$ independent random variables distributed according to D. The probability of having no value that appears exactly once in the sequence $x_1, \ldots, x_s, y_1, \ldots, y_t$ 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 $1, \ldots, u$, and that the constants x_i belong to this range, for $1 \leq i \leq s$. We use dynamic programming to compute the probability $P(v, w)$ that there is no value z with $z \geq v$ that appears exactly once in the sequence $x_1, \ldots, x_s, y'_1, \ldots, y'_w$, where $y'_1, \ldots, y'_w$ are independent random variables distributed according to the distribution D conditioned on $y'_i \geq v$.

We compute $P(v, w)$, for $1 \leq v \leq u$ and $0 \leq w \leq t$. 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 $0 \leq i \leq w$ 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. $\square$

6 Proof of Theorem 1

This proof is an adaptation of the proof of 2 from [C08].

We start with proving (b). Consider a hypergraph H with maximum degree at most Δ. We order the vertices of H arbitrarily, and one by one we assign colors to the vertices from an $\Delta + 1$ element set of colors. We make sure that the color of the *first* vertex of any edge E does not get repeated in E. When we reach a vertex v of degree d, this requirement may exclude at most d colors, but as we have $d \leq \Delta$, we can still assign a suitable color to v.

To prove (a), consider a hypergraph with fewer than $\binom{s}{2}$ edges. We give a coloring algorithm that finds a conflict-free coloring of H with at most $s - 1$ colors. The algorithm consists of steps starting with step 1. At step i, we find a vertex v_i of degree at least $s - i$ if such a vertex exists. We color v_i with a new color and remove it with from H, together with all edges containing v_i. This concludes step i and we proceed to step $i + 1$. As no other vertex will be colored to the color of v_i, all edges removed at step i will end up having a vertex of unique color – namely, v_i. If we do not find a suitable vertex v_i, the remaining hypergraph has maximum degree less than $s - i$, and we can find a conflict-free coloring with $s - i$ colors not used earlier using the algorithm for part (a).

This algorithm must terminate in step $s - 1$ or before. Otherwise, the total number of edges removed during the first $s - 1$ steps 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 in step i, then $i - 1$ colors were used in the earlier steps and at most $s - i$ colors in the last step, so the coloring uses fewer than s colors 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. To see that the bounds in parts (a) and

(b) 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 colorings must assign different colors to all of its s vertices. □

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

References

[AEG07] D. Ajwani, K. Elbassioni, S. Govindarajan, and S. Ray: 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)*, 2007, pages 181==187.

[A91] N. Alon: A parallel algorithmic version of the local lemma, *Random Structures & Algorithms* **2** (1991), 367–378.

[AS06] N. Alon and S. Smorodinsky: Conflict-free colorings of shallow discs, in: *22nd Ann. ACM Symposium on Computational Geometry (SoCG)*, 2006, 41–43.

[AS00] N. Alon and J. Spencer: *The Probabilistic Method. 2nd ed.*, John Wiley & Sons], New York, 2000.

[AHK03] K. Aardal, S. van Hoesel, A. Koster, C. Mannino, and A. Sassano: Models and solution techniques for frequency assignment problems, *4OR: A Quarterly Journal of Operations Research* **1** (2003), 261–317.

[BCS08] A. Bar-Noy, P. Cheilaris, and S. Smorodinsky: Deterministic conflict-free coloring for intervals: from onine to online, *ACM Transactions on Algorithms* **4** (2008), 44:1–44:18.

[Bec91] J. Beck: An algorithmic approach to the Lovász local lemma, *Random Structures & Algorithms* **2** (1991), 343–365.

[Beh71] M. Behzad: The total chromatic number of a graph: A survey, in: *Combinatorial Mathematics and its Applications (Proc. Conf., Oxford, 1969)*, Academic Press, London, 1971, 1–8.

[C08] P. Cheilaris: *Conflict-Free Coloring*, PhD thesis, City University of New York, 2008.

[Ch06] K. Chen: How to play a coloring game against a color-blind adversary, in: *Proceedings of the 22nd Annual ACM Symposium on Computational Geometry (SoCG)*, 2006, 44–51.

[ChFK06] K. Chen, A. Fiat, H. Kaplan, M, Levy, J. Matoušek, E. Mossel, J. Pach, M. Sharir, S. Smorodinsky, U. Wagner, and E. Welzl: Online conflict-free coloring for intervals, *SIAM J. Comput. (SICOMP)* **36** (2006), 1342–1359.

[CzS00a] A. Czumaj and C. Scheideler: A new algorithm approach to the general Lovász local lemma with applications to scheduling and satisfiability problems, in: *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, ACM, New York, 2000, 38–47.

[CzS00b] A. Czumaj and C. Scheideler: Coloring nonuniform hypergraphs: a new algorithmic approach to the general Lovsz local lemma, in: *Proceedings of the Ninth International Conference "Random Structures and Algorithms" (Poznan, 1999), Random Structures Algorithms* **17** (2000), 213–237.

[D05] R. Diestel: *Graph Theory (3rd ed.)*, Graduate Texts in Mathematics **173**, Springer-Verlag, Berlin, 2005.

[EL75] P. Erdős and L. Lovász: 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, 1975, 609-627.

[ELRS03] G. Even, Z. Lotker, D. Ron, and S. Smorodinsky: Conflict–free colorings of simple geometric regions with applications to frequency assignment in cellular networks, *SIAM J. Comput.* **33** (2003), 94–136.

[HS05] S. Har-Peled and S. Smorodinsky: Conflict-free coloring of points and simple regions in the plane, *Discrete and Computational Geometry* **34** (2005), 47–70.

[HMR97] H. Hind, M. Molloy, and B. Reed: Colouring a graph frugally, *Combinatorica* **17** (1997), 469–482.

[LLRS01] T. Leighton, C.-Y. Lu. S. Rao, S, and A. Srinivasan: New algorithmic aspects of the local lemma with applications to routing and partitioning, *SIAM J. Comput.* **31** (2001), 626–641.

[MR98] M. Molloy and B.Reed: Further algorithmic aspects of the local lemma, in: *Proceedings of the 30th Annual ACM Symposium on Theory of Computing* (1998), 524–529.

[MR02] M. Molloy and B. Reed: *Graph Colouring and the Probabilistic Method*, Algorithms and Combinatorics **23**, Springer-Verlag, Berlin, 2002.

[MS05] M. Molloy and M. R. Salavatipour: A bound on the chromatic number of the square of a planar graph, *J. Combin. Theory Ser. B* **94** (2005), 189–213.

[PT03] J. Pach and G. Tóth: Conflict-free colorings, in: *Discrete and Computational Geometry*, Algorithms Combin. **25**, Springer, Berlin, 2003, 665–671.

[S07] S. Smorodinsky: On the chromatic number of some geometric hypergraphs, *SIAM J. Discrete Math.* **21** (2007), 676–687.

[W77] G. Wegner: Graphs with given diameter and a coloring problem, Technical Report, University of Dortmond, 1977.

[W00] A. Woldar: Rainbow graphs, in: *Codes and Designs (Columbus, OH, 2000)*, Ohio State Univ. Math. Res. Inst. Publ. **10**, de Gruyter, Berlin, 2002, 313–322.