Almost all graphs with average degree 4 are 3-colorable

Dimitris Achlioptas
Microsoft Research, Redmond, Washington
optas@microsoft.com

Cristopher Moore*
Computer Science Department, University of New Mexico, Albuquerque
and the Santa Fe Institute, Santa Fe, New Mexico
moore@cs.unm.edu

Abstract

The technique of approximating the mean path of Markov chains by differential equations has proved to be a useful tool in analyzing the performance of heuristics on random graph instances. However, only a small family of algorithms can currently be analyzed by this method, due to the need to maintain uniform randomness within the original state space. Here, we significantly expand the range of the differential equation technique, by showing how it can be generalized to handle heuristics that give priority to high- or low- degree vertices. In particular, we focus on 3-coloring and analyze a "smoothed" version of the practically successful Brelaz heuristic. This allows to prove that almost all graphs with average degree d, i.e. $G(n, p = d/n)$, are 3-colorable for $d \leq 4.03$, and that almost all 4-regular graphs are 3-colorable. This improves over the previous lower bound of 3.847 on the 3-colorability threshold for $G(n, p = d/n)$ and gives the first non-trivial result on the colorability of random regular graphs. In fact, our methods can be used to deal with *arbitrary* sparse degree distributions and in conjunction with general graph algorithms that have a preference for high- or low-degree vertices.

*Supported by NSF grant PHY-0071139, the Sandia University Research Program, and Los Alamos National Laboratory.

1 Introduction

Let $G(n,p)$ denote the random graph on n vertices where each pair of vertices has an edge between them with probability p. We will say that a property holds *for almost all graphs of degree* d if it holds with probability $1-o(1)$ in $G(n,p=d/n)$. Erdős and Rényi [8] observed that such random graphs display a number of threshold phenomena as d is increased. The best-known of these is the emergence of a giant component, where G possesses a component of size $\Omega(n)$ w.h.p. if and only if $d>1$. The intuition behind this fact is that random graphs behave like expanders, and are locally tree-like.

Achlioptas and Friedgut [2] showed that k-colorability also has such a sharp, albeit non-uniform, threshold for all $k>2$. That is, there exists a function $d_k(n)$ such that if $p=(1-\epsilon)d_k(n)$ then $G(n,p=d/n)$ is k-colorable with probability $1-o(1)$, but for $p=(1+\epsilon)d_k(n)$ it is non-k-colorable with probability $1-o(1)$. It is widely conjectured that $d_k(n)$ converges to a constant d_k as $n\to\infty$. The existence and location of d_k for $k\geq 3$ remains one of the major open question in random graph theory (see [4]).

Numerical experiments (e.g. [6]) suggest that $d_3\approx 4.7$. Probabilistic counting arguments [11] give an upper bound of 4.99. Since a graph is k-colorable if it has no subgraph of minimum degree k, i.e. a *k-core*, lower bounds for d_3 were given by bounding the appearance of a 3-core. These culminated in the work of Pittel, Spencer and Wormald [20] who determined exactly when a k-core appears, implying $d_3\geq 3.35$.

The best known lower bound on d_3 prior to this work was given by Achlioptas and Molloy [3] who, going beyond the 3-core, analyzed a simple list-coloring algorithm, and showed that it colors random graphs of degree d with positive probability whenever $d<3.847$. Combining this with the sharp threshold of [2] implies $d_3\geq 3.847$. Their analysis relies on the fact that, throughout the algorithm's execution, the uncolored part of the graph is distributed as $G(n',p)$ where n' is the number of uncolored vertices. Thus they can treat the algorithm as a Markov process in a very low-dimensional state space, and closely approximate its progress with the trajectory of pair of differential equations.

Our original motivation was to improve upon this bound by analyzing algorithms that give priority to highest degree vertices. Since such algorithms clearly fail to maintain uniform randomness in $G(n',p)$, we needed to extend the differential equations method to handle information about the degree sequence of the graph. Specifically, we now demand that the uncolored vertices form a graph that is uniformly random conditional on its degree sequence. This requires several key ideas to deal with the multiple issues that arise.

1. Since the maximum degree is a highly volatile random variable, we introduce a "smoothed" version of the Brelaz heuristic. Namely, we color a vertex of degree i with probability proportional to a preference function $h(i)$ which increases sharply with i, such as $h(i)=i^\alpha$ for some constant $\alpha\gg 0$.

   This smoothing idea allows us to give priority to high-degree vertices without creating complicated correlations, and makes the expected effect of each step of the algorithm a smoothly varying function of the degree sequence. It can easily be applied to other algorithms with a preference for high- or low-degree vertices, giving us a great deal of flexibility in algorithm design and analysis.

2. Since we have a potentially infinite system of differential equations, one for each degree, we need to reduce them to a finite-dimensional system by truncating the degree sequence at some finite degree Δ. We show how very-high-degree vertices can be colored using a simple argument that creates a small number of 2-color low-degree vertices. These are then taken into account in the initial conditions of the differential equations.

3. To avoid dealing with states with $o(n)$ vertices, which cannot be handled by the differential equations technique, we run our algorithm for enough rounds so that the great majority of the vertices are colored, and then apply a simple endgame strategy to the leftover uncolored vertices.

4. Since general monotonicity and sharp threshold arguments such as [2] do not exist for random graphs on degree sequences, we need to introduce backtracking in order to obtain an algorithm that succeeds with high probability rather than with constant probability.

5. We need to establish symmetry, or "list-stability", among the three color lists of 2-color vertices. As we can not use the lazy-server technique of [1] here, we allow these lists to diverge, model each step of the algorithm as a multitype branching process, and show how list-stability follows from the variational stability of the latter. Our tools for analyzing such branching processes are of independent interest.

We then obtain the following improvement to the previous lower bound on d_3:

Theorem 1 *Almost all graphs with average degree 4.03 are 3-colorable.*

Having developed this machinery we can now study the colorability not just of $G(n,p)$ but of general random graphs with any well-behaved degree sequence with finite average degree. Thus, for example, we can also prove the first non-trivial result about the 3-colorability of random regular graphs:

Theorem 2 *Almost all 4-regular graphs are 3-colorable.*

Up to now, very few algorithms have been analyzed in the degree sequence setting and previously only the "low-degree" prefix of the degree sequence has been treated specially (for instance when one repeatedly removes all vertices of degree smaller than k to find the k-core). Ours is the first algorithmic analysis that allows us to give priority to *high* degree vertices, and also that allows us to analyze the performance of heuristics on graphs with arbitrary sparse degree distributions.

2 Related work and Motivation

Our results are based on analyzing the performance of algorithm A below, which is a probabilistic generalization of the Brelaz heuristic [7], one of the most practically successful ideas for graph coloring.

Algorithm A proceeds by maintaining for each uncolored vertex v a list of available colors $\ell(v)$. In each step, some uncolored vertex w is chosen and permanently assigned a random color from $\ell(w)$. Initially, all lists contain the same set of three colors which we will refer to as $\{R, G, B\}$. At any moment, for each uncolored vertex v, its list $\ell(v)$ consists of the originally available colors minus the colors used by the colored neighbors of v. Thus, the algorithm fails if we ever have $\ell(v) = \emptyset$ for some uncolored vertex. We will say that v is a "q-color vertex" if $|\ell(v)| = q$. Throughout the paper, unless specified otherwise, the degree of a vertex v, denoted by $\deg(v)$, will be the number of its *uncolored* neighbors. The function $h(i)$, which will determine the algorithm's preference for vertices with i uncolored neighbors, will be specified later.

Algorithm A

```
procedure color(v)
    pick c ∈ ℓ(v) uniformly at random; for all w adjacent to v set ℓ(w) ← ℓ(w) − c;

while there are uncolored vertices
    if there are 2-color vertices
        then
            pick a 2-color vertex v at random with probability proportional to h(deg(v));
            color(v)
        else
            pick a 3-color vertex v uniformly at random;
            color(v)
    while there are 1-color vertices
            pick a 1-color vertex v uniformly at random;
            color(v)
```

Remark: Observe that we've implemented A so that it is oblivious to 0-color vertices. If such a vertex is ever generated, then A simply goes on forever. This (impractical) choice has the benefit of allowing us to analyze each iteration of the `while` loop without having to condition on A not having failed already.

It is crucial to note that A's focus on the selection of 2-color vertices is not misguided. Our choices regarding 3-color vertices are virtually inconsequential: we only get to color one such vertex per connected component. Our choices regarding 1-color vertices are truly inconsequential: no matter how, i.e. in what order, we choose to color the 1-color vertices, we get to the same state after all of them have been colored.

In [3], Achlioptas and Molloy considered an algorithm called 3-GL which, when 2-color vertices exist, chooses such a vertex uniformly at random and colors it. Thus, 3-GL is equivalent to A if $h(i)$ is a constant independent of i. This uniform randomness in the choice of 2-color vertices has the following very expedient property: if the original graph is distributed as $G(n,p)$, then the graph induced by the uncolored vertices is also distributed as $G(n',p)$ where n' is the number of uncolored vertices. To see this, assume that initially the outcomes of all $\binom{n}{2}$ experiments determining the graph are concealed and that, in each step, we are only allowed to "expose" the outcomes regarding the edges incident to the vertex being colored. With a moment's reflection it is not hard to see that it is in fact possible to carry out 3-GL in this restricted model. As a result, after coloring each vertex, no information about the graph induced by the uncolored vertices has been exposed. Exploiting this property, Achlioptas and Molloy gave a tight analysis of 3-GL on $G(n,p)$ in [3].

While selecting 2-color vertices uniformly at random is very expedient for probabilistic analysis, it is not at all clear that this is a desirable rule. For example, it is very natural to give priority to the high-degree 2-color vertices. Such vertices, prima facie, are more constrained, so it makes a lot of sense to color them before we attempt to color less constrained vertices. Indeed, the Brelaz heuristic is exactly this idea taken to the extreme: $h(i) = 1$ if i is the maximum degree in the graph, otherwise $h(i) = 0$.

There are two main technical difficulties in analyzing an algorithm that gives priority to high degree vertices. The obvious one is that even if the input graph is distributed as $G(n,p)$, we cannot hope that the graph induced by the uncolored vertices remains distributed as $G(n',p)$. Rather, we will need a significantly more refined notion of "state" for the uncolored part of the graph. The theory of random graphs on a *fixed degree sequence* will be most useful in that respect. The study of such graphs was pioneered by Bender and Canfield [5] and Wormald [22] and was significantly developed by Molloy in [18] and Molloy and Reed in [19]. Recently, such random graphs have gained popularity as a model of "massive" graphs, since, unlike $G(n,p)$, they offer full flexibility over the degree distribution. By developing our coloring machinery for this model we can handle arbitrary sparse degree sequences, e.g. random regular graphs, not just $G(n,p)$.

The second difficulty is more technical and, perhaps, more subtle. Assume for a moment that we chose h to be as in the Brelaz heuristic, i.e. placing all the probability mass on maximum degree vertices. While this certainly promotes our goal of preferring high degree vertices, it results in a probabilistic process that is extremely hard to analyze. This is because the maximum degree among 2-color vertices is a very volatile random variable whose value changes rapidly in the course of the algorithm. For example, since vertices of the same (high) degree in the original graph can take varying amounts of time to become 2-color vertices, the maximum degree among 2-color vertices is non-monotonic in time. Maintaining this maximum degree information as part of the state would require an extremely "microscopic", and hence cumbersome, representation of the random process.

To overcome this difficulty, we observe that we can alternatively think of the Brelaz heuristic as setting $h(i) = i^\alpha$, where $\alpha \to \infty$. By taking $h(i) = i^\alpha$ where $\alpha \gg 0$ but finite, we get a "soft" version of the rule. Observe now that in this case the probability that a given vertex is chosen is a function of its degree and a smoothly changing function of the degree distribution, i.e. the sum of f_i over all vertices. Thus, to carry out our choices we don't need to track of any additional information, such as the current maximum degree. Considering larger and larger values of α will allow us to put more and more weight on high-degree vertices while always maintaining a tractable process. In fact, we will see that taking $\alpha = 13$ already seems to come very close to the limiting performance.

We expect this smoothing idea to be of general use in analyzing the performance of other heuristics on random graphs, e.g. heuristics for finding small dominating sets [9]. Also, we can use this idea in random for 3-SAT by giving priority to variables that appear in many clauses. In so far, very few results have been proven for heuristics with preferences for high- or low-degree vertices. In each case, ingenuity and a lot of hard work was required. The idea of smoothing, along with the other technical tools developed in this paper, come very close to being a uniform, complete method for analyzing such algorithms.

3 The random configuration model and outline of the proof

In order to analyze A's performance on random graphs with arbitrary degree distributions, we will use the *random configuration* model, introduced independently by Bender and Canfield [5] and Wormald [22]. Suppose we have a list V of vertices and their degrees, such that $\sum_v \deg(v)$ is even. In steps 1 and 2 below, we form a random configuration on this degree sequence; in step 3 we use that configuration to form a random (multi)graph with that degree sequence.

1. Form a set V' consisting of $\deg(v)$ copies of each vertex $v \in V$.
2. Pick a uniformly random perfect matching E' on V'.
3. For each matching pair in V', add an edge between the corresponding vertices in V.

The idea is to make one copy of v in V' for each of the "half-edges" protruding from v. By choosing a random matching of V', we connect each copy to a random partner, chosen uniformly among all the copies of all the vertices. It follows that the probability that a random edge contains a given vertex v is proportional to $\deg(v)$, and two vertices have an edge between them with probability proportional to the product of their degrees. Naturally, a graph formed in this way may contain self-loops or multiple edges. Nonetheless, it is possible to show [3] that if the average degree remains bounded as n grows (as will always be the case throughout this paper), then the expected number of multiple edges and self-loops is $O(1)$ and, in fact, with positive probability we get a *simple* graph. Most importantly, simplicity does not skew the distribution.

Fact 1 *For any degree sequence $\mathcal{D}$, if a random configuration on $\mathcal{D}$ gives rise to a simple graph G, then G is a uniformly random graph among all graphs with that degree sequence.*

To analyze the performance of A on random graphs with arbitrary degree distributions it will be convenient to think of A as running directly on random configurations per se (see below). By establishing that the vertex copies corresponding to uncolored vertices form a random configuration whose degree sequence evolves with time, we get that the graph induced by the uncolored vertices is always uniformly random conditional on its degree sequence. Similarly to the case for 3-GL, the easiest way to see this claim is to think of the edges in the configuration as exposed only when we color a vertex: that is, when a vertex v is colored, the partners of all the unexposed copies of v are exposed. More precisely, to carry out A in the configuration model we need to redefine the procedure `color` as follows.

After assigning color c to vertex $v \in V$, for each of its copies $v' \in V'$ do the following:

1. Expose the edge $\{v', u'\} \in E'$. Note that u' is a copy of some $u \in V$.
2. Remove c from $\ell(u)$
3. Remove $\{v', u'\}$ from E'.
4. Remove v' and u' from V'.

We will refer to a step of the algorithm in which a 2- or 3-color vertex is colored as a *free step*, while a step in which a 1-color vertex is colored will be a *forced step*. We will call a single iteration of A's `while` loop a *round*. Thus, a round consists of a single free step followed by an ensuing sequence of forced steps, and no 1-color vertices remain at the end of a round.

By performing `color` in this manner, we see that after each round only edges incident to colored vertices have been exposed. By the method of deferred decisions it follows that if we start with a random configuration, then at any point in the algorithm's execution the unexposed edges form a uniformly random matching on the vertex copies of uncolored vertices. Note that if the edge $e = \{u', v'\}$ in the configuration corresponds to a multiple edge or self-loop in the multigraph, the algorithm still proceeds as desired, i.e. e has no effect on the coloring or the lists. Also, we note that when running on a random configuration $\deg(v)$ is always equal to the number of unexposed copies of v. Naturally, as long as there are no self-loops or multiple edges incident to v, $\deg(v)$ is equal to v's original degree minus the number of its colored neighbors.

At each step of the algorithm, our state will consist of the number of vertices of each (color list, degree) pair at the beginning of each round. Intuitively, a state is good if the average number of 1-color vertices generated per forced step is less than 1. If a state is good then we can relate the round to a subcritical branching process and show that the probability that a 0-color vertex is created in that round is $O(1/n)$. If we can prove this for all rounds then A succeeds with positive probability, giving us the following.

Theorem 3 *A constant fraction of all graphs with average degree 4.03 are 3-colorable.*

Theorem 4 *A constant fraction of all 4-regular graphs are 3-colorable.*

In fact, this analysis is tight, since A actually does fail with positive probability on such graphs. To get the high probability results of Theorems 1 and 2 we have to introduce a *backtracking* version of A. To preserve conditional independence, we can only afford a limited form of backtracking, akin to that introduced by Frieze and Suen in [10] for random 3-SAT. Since each round of A fails only if certain local events occur, such as a cycle in the graph induced by the forced steps, our limited backtracking is w.h.p. enough.

Since backtracking adds a fair amount of complexity to the details of the analysis (but not much to its substance) in this extended abstract we focus on presenting a complete analysis of A, while in Appendix F we outline the analysis of the backtracking version, focusing on why limited backtracking is sufficient in this context. Finally, we note that in the case of $G(n,p)$, Theorem 1 follows immediately from Theorem 3, by using the following corollary of the main result in [2]: for all $k > 2$, if $G(n, p = d^*/n)$ is k-colorable with positive probability, then for all $d < d^*$, $G(n, d/n)$ is k-colorable w.h.p. However, this approach does not work for regular graphs and backtracking gives a fully algorithmic proof of both Theorem 1 and Theorem 2.

We will model the state with the technique of differential equations. In particular, given a random graph G on a degree sequence with *bounded* maximum degree, we will model A's execution on G with a finite system of differential equations, with one equation for each (color list, degree) pair. The idea of using differential equations to approximate discrete random processes goes back at least to Kurtz [13, 14]. It was first applied in the analysis of algorithms by Karp and Sipser [12] and was greatly expanded by since then Mitzenmacher [15, 16] and Wormald [23].

In order to deal with random graphs with unbounded maximum degree, such as $G(n, d/n)$, we will need to deal with the high-degree vertices by other means. The good news is that since G has bounded average degree, for every $\delta > 0$ there exists a constant $\Delta = \Delta(\delta)$ such that fewer than δn vertices have degree greater than Δ. If we take Δ to be large enough, the subgraph induced by these high-degree vertices and their neighbors fails to percolate; since random graphs in the configuration model are expanders, this subgraph consists mostly of trees. We color these easily, leaving us with a finite degree sequence on which to run A.

Here we show only how to deal with high-degree vertices to the extent that it is neede to prove Theorem 3. However, our technique applies to any degree distribution whose high-degree tail is sufficiently well-behaved, and whose average degree is constant. Moreover, we note that for $G(n,p)$ in particular, we could alternatively exploited the fact that the high-degree tail of the degree sequence follows a truncated Poisson distribution.

The remainder of the paper is organized as follows. In Section 4 we give our notation and state the key lemmas we use to prove Theorems 1, 2, 3 and 4. In Section 5 we define multitype branching processes and calculate the expected effect of a single round. In Section 6 we convert this expectation into a system of differential equations. In Section 7 we integrate our differential equations for initial conditions corresponding to $G(n, d/n)$ and to 4-regular graphs.

4 Preliminaries, notation, and key lemmas

Theorem 4 follows from Lemma 1 below and Fact 1 which relates random graphs and random configurations.

Lemma 1 *Let $\mathcal{D}^4$ be the degree sequence where all vertices have degree 4. Let G be a random multigraph formed by picking a random configuration on $\mathcal{D}^4$. With positive probability G is simple and 3-colorable.*

To prove Theorem 3 we will use Fact 1 along with the fact that the the degree sequence of $G(n, d/n)$ is very tightly concentrated around its expectation. In particular, it is folklore that for any constant d, w.h.p. $G(n, d/n)$ has $(e^{-d}d^i/i!) \cdot n + o(n^{2/3})$ vertices of degree $i \leq 2 \log n / \log \log n$ and no vertices of higher degree.

Lemma 2 *For $d = 4.03$, let $\mathcal{D}^*$ be any degree sequence with $(e^{-d}d^i/i!) \cdot n + o(n^{2/3})$ vertices of degree $i \leq 2 \log n / \log \log n$ and no vertices of higher degree. Let G be a random multigraph formed by picking a random configuration on $\mathcal{D}^*$. With positive probability G is simple and 3-colorable.*

To prove Lemma 2 we will prove two lemmata, dealing with the high degree and low degree vertices respectively. Before doing so we will need to introduce some definitions.

Definition 5 *We will say that a vertex v has* high degree *if its initial degree is greater than Δ and that it has* low degree *otherwise. We will set $\Delta = 30$, and let $\phi = \sum_{i>\Delta} i\, e^{-d} d^i / i!$.*

Definition 6 *For a random configuration C let E_H be the set of edges in C incident to high degree vertices. Let H be the multigraph induced by E_H. Let Y be the set of low-degree vertices that lie in cyclic components of H. Let E_Y be the set of edges in C incident to vertices in Y. Let K be the multigraph induced by $E_H \cup E_Y$. Let $B = C - \{E_H \cup E_Y\}$. Let L be the set of all low degree vertices in K which are not in Y.*

Lemma 3 *Let $\mathcal{D}^*$ be any degree sequence as in Lemma 2 and let C be a random configuration on $\mathcal{D}^*$.*

1. *With positive probability, the multigraph K is simple and can be 3-colored so that all vertices in L have monochromatic neighborhoods.*
2. *With high probability B has $b_i \cdot n + o(n^{2/3})$ vertices of each degree $0 \le i \le \Delta$, where*

$$b_i = \sum_{j=i}^{\Delta} \binom{j}{i} \left(\frac{\phi}{d}\right)^{j-i} \left(1 - \frac{\phi}{d}\right)^{i} \frac{e^{-d} d^j}{j!} \quad .$$

3. *With high probability $|L| < \phi \cdot n$.*

Proof. See Appendix A.

Lemma 4 *Let $\mathcal{B}$ be any degree sequence with $b_i \cdot n + o(n^{2/3})$ vertices of degree $0 \le i \le \Delta$ with b_i as in part 2 of Lemma 3, and no vertices of higher degree. Assign of lists to the vertices of $\mathcal{B}$ such that at most ϕn vertices have 2 available colors and all others have all 3 colors. Let G be a random multigraph formed by picking a random configuration on $\mathcal{B}$. With positive probability G is simple and list-colorable.*

To combine Lemmata 3 and 4 to get Lemma 2 we observe that they are concerned with disjoint sets of edges. While this does not quite make the corresponding events independent (since the partition is not fixed a priori) it still allows for a rather straightforward argument (presented in Appendix B).

To prove Lemmata 1 and 4 we will analyze the performance of A on the corresponding random configurations. In both cases it will be technically convenient to only let A run for a predetermined number of rounds, rather than until completion, and argue that it is easy to complete the coloring of the "leftover" vertices remaining at that time. This allows us to avoid the rather hairy analysis of A's last few rounds and use a much simpler (and more general) argument instead. We note that if left to run until completion, A would actually succeed in coloring the entire graph with positive probability. For the purposes of the analysis it will also be convenient, rather than leaving 0-color vertices uncolored, to process them like ordinary vertices as follows: if a 0-color vertex v exists, assign v a random color $c \in \{R, G, B\}$, expose v's remaining neighbors, remove c from their list and label v `bad`. Clearly, the algorithm fails if a `bad` vertex is ever created but this trick has the advantage that at the beginning of each round, only 2- and 3-color vertices exist.

Since we wish to analyze A on arbitrary degree sequences, we will divide the uncolored vertices present at the beginning of each round according to their (current) color list and degree. We will say "at time t" to refer to the moment right before the tth round starts. For each $i \ge 0$ and each color $c \in \{R, G, B\}$ we will denote by $C_i(t)$ the number of 2-color vertices which at time t have degree i and *do not* contain color c in their list. By $W_i(t)$ we will denote the number of 3-color vertices of degree i at time t and we will let $U_i(t) = R_i(t) + G_i(t) + B_i(t)$, the total number of 2-color vertices of degree i. We will sometimes drop the reference to t in our random variables when that does not lead to confusion. We will use the term *list sequence* to refer to a degree sequence where each vertex v has been assigned a list $\ell(v) \subseteq \{R, G, B\}$.

Thus, to prove Lemma 1 and Lemma 4 we will show that in each case, in the allotted number of rounds, A properly colors a (large) piece of the graph and leaves a piece that can be colored easily. To make this precise, we need to introduce the following definitions.

Definition 7 *A list sequence on n vertices is* (δ, ϵ)-easy *if it has bounded maximum degree, all vertices have at least 2 colors, at least δn vertices have 3 colors, and $\sum_i i(i-2)(U_i + W_i) < -\epsilon n$.*

The reason we call such list sequences easy is the following lemma.

Lemma 5 *For all $\delta, \epsilon > 0$, a (δ, ϵ)-easy list sequence is simple and 3-colorable with positive probability.*

Proof. See Appendix A.

Definition 8 *For a list sequence $\mathcal{L}$ and an integer t, let $G(t)$ be the random multigraph induced by the edges exposed after running A on $\mathcal{L}$ for t rounds. Let $\mathcal{L}(t)$ be the list sequence of the uncolored vertices at time t and let $E(t)$ denote the number of unexposed edges at time t.*

Armed with Lemma 5, the main thrust of our argument is captured by Lemmata 6 and 7 below. For technical reasons it is easier to prove things about A's performance when, initially, for each i, an equal number of 2-color vertices have each of the 3 color pairs. Since adding colors to the lists of a graph cannot hurt its list-colorability Lemmata 1 and 4 follow immediately from Lemma 5 and Lemmata 6 and 7.

Lemma 6 *Let $\mathcal{L}^4$ be any list sequence where all vertices have degree 4 and where for each i, $R_i = G_i = B_i = \phi n + o(n)$, while all other vertices have 3 available colors. There exist $\delta, \epsilon, T > 0$ such that with positive probability: i) $G(T)$ is simple and contains no* bad *vertices, and ii) $\mathcal{L}(T)$ is (δ, ϵ)-easy.*

Lemma 7 *Let $\mathcal{L}^*$ be any list sequence with degree sequence as in Lemma 4 and where for each i, $R_i = G_i = B_i = \phi n + o(n)$ while all other vertices have 3 available colors. There exist fixed $\delta, \epsilon, T > 0$ such that with positive probability: i) $G(T)$ is simple and contains no* bad *vertices, and ii) $\mathcal{L}(T)$ is (δ, ϵ)-easy.*

To get a rough idea of how the first part of each of Lemmata 6 and 7 will be proved, it is useful to observe the following. If at least δn unexposed copies remain in the graph being colored and a given round has z forced steps, then the probability that a bad vertex is generated during that round is roughly proportional to z^2/n. To see this note that each time we expose a single copy in that round there is at most a $(z \times \Delta)/(\delta n)$ chance that its match lies among the yet unmatched copies of 1-color vertices waiting to be colored in that round. Clearly, if such a match never occurs, then no bad vertex is created. From this observation we see that as long as $\Omega(n)$ copies remain unexposed, if the expected squared length of a round is bounded, then A runs a $O(1/n)$ probability of failure in that round. Since there are at most n rounds, bounding the second moment of the length of each round is the key to proving positive probability of success.

To bound that second moment it will be useful to view each round as similar to a branching process where the progenitor is the 2-color vertex, chosen on the free step, and its progeny consists of the 1-color vertices colored by the forced steps. Using this viewpoint, the heart of the matter becomes showing that w.h.p. each such branching process is subcritical. Subcriticality implies that there exists some $\rho > 0$ such that the probability that a round lasts more than i steps is bounded by $(1-\rho)^i$ which, in turn, is more than enough to guarantee a bounded second moment. Naturally, the criticality of the branching process is related to the expected number of 1-color vertices generated per vertex colored. Moreover, here, not all vertices are of equal "potency", since their degree and assigned color affects their expected progeny. As a result, rather than standard branching processes, we need to use multitype branching processes, where a type amounts to a (color, degree) pair.

Another key element is that the expected progeny of each type shift very smoothly in the course of the algorithm. In particular, if the list sequence at the beginning of a round corresponds to a subcritical branching process, then with overwhelming probability the list sequence at the end of the round will correspond to a very similar branching process. This is because the length of the round will be polylogarithmic, while the number of vertices with each (color list, degree) pair is $\Omega(n)$. Naturally, this suggests an inductive approach where we make sure that the list sequence during the course of the algorithm stays away from the region giving rise to supercritical branching processes. This is precisely what the differential equations approach enables us to do. That is, we will model the scaled random evolution of the list sequence with a deterministic trajectory in $\mathbb{R}^{4\times(\Delta+1)}$ and argue that w.h.p. the random process stays very close to this trajectory.

5 A single round as a multitype branching process

We now analyze what happens in a single round of A. We focus on the case where at the beginning of the round there are $\Omega(n)$ unexposed copies belonging to 2-color vertices. That certainly holds initially and we will see that w.h.p. it will also hold for all T rounds that we'll run of A, by our choice of T.

It will be convenient to think of all the vertices having a given (color list, degree) pair as grouped together in a "bucket". Hence, coloring a vertex v with color c removes v from its bucket and moves each of its neighbors w from bucket $(\ell(w), \deg(w))$ to bucket $(\ell(w) - c, \deg(w) - z)$, where z is the number of edges between v and w (almost always $z = 1$). Specifically, any 3-color neighbor of degree i becomes a 2-color vertex of degree $i - z$, while a 2-color neighbor becomes a 1-color vertex if $x \in \ell(w)$, and stays a 2-color vertex if $x \notin \ell(w)$. In either case $\deg(w)$ becomes $\deg(w) - z$.

Now, in a round, we start by picking some 2-color vertex v and assigning it a color c from its list. We then expose the unexposed copies of v. Some of these copies belong to 3-color vertices, while others belong to 2-color vertices. In either case, we update the lists of those vertices by removing c from their lists. This might lead to the creation of some new 1-color vertices which will we process just like we did v, and the round proceeds until no 1-color vertices remain. To analyze this process precisely we will need to introduce multitype branching processes.

5.1 Multitype branching processes

In the standard Galton-Watson (GW) branching process we have an initial (progenitor) vertex which gives rise to X children, where X is an arbitrary integer-valued random variable. Each of those children proceeds to procreate independently, its offspring distribution being the same as that of the progenitor, and so on. The fundamental theorem of branching processes tells us that if $\mathbf{E}(X) < 1$, i.e. the branching process is subcritical, then extinction is certain.

A natural generalization of the Galton-Watson process is one in which there are b vertex "types", the type of a vertex determining the probability distribution of its progeny. More precisely, for each type $1 \leq j \leq b$ there is a probability distribution $f_j : \mathbb{N}^b \to \mathbb{R}$, telling us the probability that a vertex of type j will have progeny $(x_1, \ldots, x_b)$, i.e. x_i children of each type $1 \leq i \leq b$. The progenitor also has a probability distribution, in which it is of type j with probability p_i. The evolution is similar to the GW process: the progenitor procreates according to f_j; for each $1 \leq i \leq b$, each of its children of type i procreate independently according to f_i; and so on. Such multitype branching processes are well understood (e.g. [17]).

Similarly to the Galton-Watson branching process, in order to determine whether extinction is certain, it suffices to only consider expectations. Rather than the scalar $\mathbf{E}(X)$, the key here is the matrix M where $M(i, j)$ is the expected type-i progeny of a type-j vertex. The criterion is whether the largest eigenvalue λ_1 of M is strictly smaller than 1. In that case the process is subcritical, its expected total progeny is bounded and, moreover, it can be read off directly from M. The following lemma establishes exactly how the total progeny of each type relates to M and also gives two crucial variational properties of the total progeny.

Lemma 8 *Consider a multitype branching process with b types and let M be the $b \times b$ matrix where $M(i, j)$ is the expected type-i progeny of a type-j vertex. Let the probability that the progenitor is of type j be given by the vector $p = (p_1, \ldots, p_b)$ and let $m = m(p) = (m_1, \ldots, m_b)$ be the total expected progeny of each type.*

If the largest eigenvalue $\lambda_1 = \lambda_1(M)$ of the matrix M, satisfies $\lambda_1 < 1 - \delta$ for some $\delta > 0$ then

1. *For all p, $m(p) = (I - M)^{-1} p$ where I is the identity matrix.*
2. *Let $n(q)$ be the total expected progeny of a multitype branching process with b types, matrix N, and progenitor distribution q. If $|M - N| = \epsilon < \delta/2$ and $|p - q| = \zeta$ then $|m(p) - n(q)| \leq 2\zeta/\delta + 8/\delta^2$.*

Proof: See Appendix B.

Since (the coloring of) each vertex gives rise to new vertices to be colored, we would like to map each round to a multitype branching process, with the types corresponding to (received color, degree) pairs. There are two main issues complicating such a mapping. The first issue is that when we expose the copies adjacent to the vertex being colored we might find that some of them belong to uncolored 1-color vertices already colored in the same round; i.e. the graph induced by the vertices colored in a round might contain cycles, self-loops or multiple edges. In that case we cannot always equate the number of 1-color vertices generated when a vertex is colored with its number of children in the branching process. The other issue is that the total number of vertices with a given (color list, degree) pair shifts, albeit slightly, in the course of a round. Therefore, we do not quite have a fixed probability distribution governing the progeny of each type for the course of each round. To overcome these difficulties we will rely heavily on the fact that in the subcritical regime, with overwhelming probability, a round exposes no more than a polylogarithmic number of copies. As a result, with a fair amount of work, we can prove that the issues above can all be absorbed in lower-order terms. We give a proof of the corresponding lemma (Lemma 9 below) in Appendix C.

If we assume that the probability distributions governing progeny did not change in the course of a round, and if we ignore the possibility of cycles, self-loops and multiple edges, then we can work everything out immediately. When we color a vertex v, the partners matching its $\deg(v)$ copies are chosen uniformly among all other unexposed copies, so the probability that a given uncolored vertex w contains each of these copies is proportional to $\deg(w)$. In particular, if there are E unexposed copies, the probability that w gets connected to v is $\deg(w)/E$. For $x, y \in \{R, G, B\}$ and $0 \le i, j \le \Delta$, let us write the matrix entry $M_{(x,i),(y,j)}$ for the expected number of x-color vertices of degree i generated by assigning color y to a vertex of degree j. Thus, we associate round t with a multitype branching process whose matrix is

$$M_{(x,i),(y,j)} = (i+1)j\, C_{i+1}(t)/E(t) \ , \tag{1}$$

where C_{i+1} is the number of 2-color vertices with $\ell = \{x, y\}$ and degree $i+1$. This gives us a square matrix M of size $3(\Delta+1)$, corresponding to a branching process with one type for each (color, degree) pair. Parts 2a and 2b of Lemma 9 below asserts that as long as things are bounded away from criticality, this multitype branching process is an excellent approximation of the behavior of a single round. More precisely:

Definition 9 *A list sequence $\mathcal{L}(t)$ is (α, β)-subcritical if $\sum_{i=0}^{\Delta} U_i(t) > \alpha n$ and the largest eigenvalue $\lambda_1 = \lambda_1(M)$ of the matrix M defined in Equation* (1) *satisfies $\lambda_1 < 1 - \beta$. A list sequence $\mathcal{L}(t)$ is subcritical if it is (α, β)-subcritical for some $\alpha, \beta > 0$.*

Definition 10 *For any sequence $\{s_i\}_{i=0}^{\Delta}$, let $s^{(q)} = \sum_i s_i\, i^q$ denote its q-th moment.*

In Lemma 9 below, part 1 is unrelated to branching processes and comes from a straightforward calculation of the expected effect of the free step. In part 2a we use the branching process to determine the expected number of copies exposed while coloring vertices of each color in the course of a round. Note that number is the expected total progeny of vertices of a given color, summed over all degrees. The expression in 2a is exactly what we get from the branching process along with a $o(1)$ term that absorbs the effects of any potential cycles, self-loops or multiple edges. Having determined these expectations, in part 2b we determine the expected change in our list sequence in the course of a single round by distributing the aforementioned copies to the vertices present at time t. Analogously to part 2a, the expression we get (up to the $o(1)$ term) represents what these changes would be if the list sequence had remained constant in the course of a round. Finally, part 2c follows from the fact that the total progeny in the branching process is the sum of independent variables each with bounded range. Thus, the geometric dropoff follows from standard large deviations inequalities. The $O(1/n)$ term, essentially, corresponds to the event that the branching process fails to dominate the round.

Lemma 9 *For $c \in \{R, G, B\}$ let $c_i = c_i(t) = C_i(t)/n$. Let $w_i = w_i(t) = W_i(t)/n$. Let $u_i = u_i(t) = U_i(t)/n$.*

1. *For each $c \in \{R, G, B\}$, the probability $p_{(c,i)}$ that the progenitor receives color c and has degree i is*

$$p_{(r,i)} = \frac{1}{2}\frac{h(i)(g_i + b_i)}{\sum_j h(j)\, u_j} \ , \quad p_{(g,i)} = \frac{1}{2}\frac{h(i)(b_i + r_i)}{\sum_j h(j)\, u_j} \ , \quad p_{(b,i)} = \frac{1}{2}\frac{h(i)(r_i + g_i)}{\sum_j h(j)\, u_j} \ . \tag{2}$$

   Let $p \in \mathbb{R}^{3\times(\Delta+1)}$ be the vector with entries $p_{(c,i)}$ for $c \in \{R, G, B\}$ and $0 \le i \le \Delta$.

2. *Let M be the matrix given by Equation* (1)*. If $\mathcal{L}(t)$ is subcritical then*

   (a) The expected number of copies exposed while coloring vertices with color c in round t is

$$k_c = \sum_{i=0}^{\Delta} i \times \left((I - M)^{-1}\, p\right)_{(c,i)} + o(1) \ . \tag{3}$$

(b) Let $k = k_R + k_G + k_B$. For all $c \in \{R, G, B\}$ and for all $0 \le i \le \Delta$,

$$\mathbf{E}(W_i(t+1) - W_i(t)) = -k \frac{iw_i}{w^{(1)} + u^{(1)}} + o(1) \tag{4}$$

$$\mathbf{E}(C_i(t+1) - C_i(t)) = k_C \frac{(i+1)(w_{i+1} + c_{i+1})}{w^{(1)} + u^{(1)}} - k \frac{ic_i}{w^{(1)} + u^{(1)}} - \frac{h(i)\, c_i}{\sum_j h(j)\, u_j} + o(1) \ . \tag{5}$$

(c) There exists $\rho > 0$ such that

$$\Pr[\textit{More than } s \textit{ copies are exposed in round } t] < (1-\rho)^s + O(1/n) \ .$$

Proof. See Appendix C.

In the following section we will use Lemma 9 to establish that we can model the evolution of $\mathcal{L}(t)$ by a system of differential equations as long as $\mathcal{L}(t)$ remains subcritical. In particular, using that system, we will be able to establish that if we apply A to certain initial list sequences of interest for T rounds, then w.h.p. $\mathcal{L}(t)$ is subcritical for all $0 \le t \le T$. Along with the following lemma, this implies that in each such case, there are no bad vertices at the end of round T.

Lemma 10 *Assume that for a list sequence $\mathcal{L}$ and integer T the following holds: if we apply A to $\mathcal{L}$ for T rounds then w.h.p. $\mathcal{L}(t)$ is subcritical for all $0 \le t \le T$. Then, if we apply A to $\mathcal{L}$ for T rounds with positive probability the resulting graph $G(T)$ is simple and contains no* bad *vertices.*

Proof. See Appendix B.

6 Differential equations

In this section we show how to employ a theorem of Wormald [23] (stated for completeness as Theorem 11 in Appendix D) to trace the evolution of $\mathcal{L}(t)$. That is, for $0 \le i \le \Delta$, we will trace the random variables $R_i(t)$, $G_i(t)$, $B_i(t)$ and $W_i(t)$ for $0 \le t \le T$, where T is some a priori determined number of rounds.

Given a finite collection of random variables $Y_1, \ldots, Y_k$ the theorem allows us to construct a set of (deterministic) real-valued functions $y_1, \ldots, y_k$ with the property that w.h.p. for all t considered, $Y_i(t) = y_i(t/n) \cdot n + o(n)$. This is achieved by taking a collection of equations describing the per-round conditional expected change of each random variable, such as Equations (5) and (5), and transforming them into a system of differential equations. The functions y_i are the solutions to that system. Naturally, there are a number of issues that need to be addressed before this intuitive transformation can yield rigorous mathematical results. The statement of Theorem 11, while appearing rather technical, simply formalizes those issues.

As one might anticipate we would like the joint evolution of the random variables $Y_1, \ldots, Y_k$ to have certain nice, i.e. smoothness, properties. Let $\mathbf{H}(t)$ be the history of the process, i.e. the matrix $\langle \vec{Y}(0), \ldots, \vec{Y}(t) \rangle$, where $\vec{Y}(t) = (Y_1(t), \ldots, Y_k(t))$. We would like the following:

- For every possible history, we want the conditional expected change of each random variable in a round to be bounded. Moreover, we want to be able to approximate that expectation within $o(1)$, even if we are only given the current value of each random variable within $o(n)$. That is, $o(n)$ perturbations of the variables should "not matter". Alternatively, we can view this as being able to approximate the conditional expected change without having access to a "microscopic" view of the process.

- For every possible history, we want the conditional change of each random variable in a round to have reasonable tail behavior. If that is true then over periods of $n^{2/3}$ rounds, say, the total change will be sharply concentrated around its expectation (enabling the inductive approach).

- The two conditions above already allow us to convert the probabilistic dynamics of the process into a deterministic, algebraic map. We would like this map to be smooth which, probabilistically, translates to a certain stability of the underlying random process. In particular, we want there to be an absolute bound on how much a small perturbation of the state (in any direction) can change the dynamics.

These are precisely conditions (i), (ii) and (iii) of Theorem 11. Note that in the above we demanded that the property behaves "nicely" for any possible history. In many cases, this is too much to ask for. In particular, while the process is expected to behave nicely in the vast majority of runs, one cannot definitively exclude the possibility that the process enters a bad regime in which "all bets are off." To deal with this issue, one can prespecify a set of "good" states such that for states in that set, all the desiderata are met. In particular, one can carve out a set in $\mathbb{R}^{k+1}$ (since t is also part of the state) and establish that as long as the (normalized) state lies in that set, the conditions are met. This is precisely, the notion of the domain D in Theorem 11 and, as the remark following the theorem asserts, it suffices for the process to be nice for states inside the domain.

In our case, the set of good states amounts to the set of subcritical list sequences. More precisely, let us say that a point $p = (r_i, g_i, b_i, w_i)_{i=0}^{\Delta} \in \mathbb{R}^{4\times(\Delta+1)}$ is subcritical if there exist $\alpha, \beta > 0$ such that the list sequence defined by taking $C_i = c_i \cdot n$ for $c \in \{R, G, B, W\}$ and $0 \le i \le \Delta$ is (α,β)-subcritical. For $\theta > 0$, let us say that such a point p is θ-subcritical if every point within θ of p is subcritical. In particular, for fixed $\theta > 0$ (to be specified) we will take our domain to be $D = D(\theta) = \{p \in \mathbb{R}^{4\times(\Delta+1)} : p \text{ is } \theta\text{-subcritical}\}$. Below we will also speak of θ-subcritical list sequences in the obvious sense.

(i): Combining parts 1, 2a, and 2b of Lemma 9 we see that indeed, for any subcritical list sequence we satisfy condition (i) of Theorem 11: Equations (5) and (5) express, within $o(1)$, the expected conditional change per round as a function of the normalized list sequence.

(ii): Part 2c of Lemma 9 establishes condition (ii) with room to spare.

(iii): To see that we satisfy condition (iii) we have to be a bit more careful. In particular, it is here that we will use the variational properties we proved for the total progeny of multitype branching processes. Since our degree-preference function is $h(j) = j^{\alpha}$ for some finite integer α, to prove a a Lipschitz condition for the expressions in part 2b of Lemma 9 it suffices to focus on the following quantities: k_C for each $c \in \{R, G, B\}$ and $u^{(1)}, w^{(1)}, u^{(\alpha)}$ where $u^{(\alpha)} = \sum_j h(j)\, u_j$.

For $1/(w^{(1)} + u^{(1)})$ recall that if a list configuration is subcritical then, by definition, $u^{(1)}$ is bounded away from 0. Since here we have θ-subcritical list configurations this implies that there is an absolute Lipschitz constant $L = L(\theta)$ for this term. Note now that since Δ is bounded, if $u^{(1)}$ is bounded away from 0, then the same must hold for $u^{(\alpha)}$ for any $\alpha \ge 0$, implying a Lipschitz condition for $1/u^{(\alpha)}$.

For the k_C we will show that for all $c \in \{R, G, B\}$ and $0 \le i \le \Delta$, each of the coordinates of vector $(I - M)^{-1}\, p$ satisfies a Lipschitz condition. For that we first observe that by Equation (2) the coordinates of the vector p describing the probability distribution for the progenitor satisfy a Lipschitz condition. Moreover, the entries of the matrix M defined in Equation (1) also satisfy a Lipschitz condition since $E(t) \ge U^{(1)}(t)$ and we saw that $u^{(1)}$ is bounded away from 0 for θ-subcritical list configurations. Thus, our claim follows from part 2 of Lemma 8.

This gives us the following system of differential equations for $c \in \{R, G, B, W\}$ and $0 \le i \le \Delta$:

$$\frac{\mathrm{d}w_i}{\mathrm{d}x} = -k\,\frac{iw_i}{w^{(1)} + u^{(1)}} \tag{6}$$

$$\frac{\mathrm{d}c_i}{\mathrm{d}x} = k_C\,\frac{(i+1)(w_{i+1} + c_{i+1})}{w^{(1)} + u^{(1)}} - k\,\frac{ic_i}{w^{(1)} + u^{(1)}} - \frac{i^{\alpha}\, c_i}{u^{(\alpha)}}\ . \tag{7}$$

(Note that $u_i \equiv r_i + g_i + b_i$ here so this is indeed a system of differential equations on r_i, g_i, b_i, w_i.)

By inspection we see that Equation (7) is symmetric with respect to the colors $c \in \{R, G, B\}$. So, under symmetric initial conditions, i.e. $R_i(0) = G_i(0) = B_i(0)$ for all i, we get that the solutions to the differential equations are identical, i.e. $c_i = u_i/3$ and $k_C = k/3$ for each $c \in \{R, G, B\}$. Therefore, if we know that we will only consider symmetric initial conditions we can rewrite the system as

$$\frac{\mathrm{d}w_i}{\mathrm{d}x} = -k\,\frac{iw_i}{w^{(1)} + u^{(1)}} \tag{8}$$

$$\frac{\mathrm{d}u_i}{\mathrm{d}x} = k\,\frac{(i+1)(w_{i+1} + u_{i+1}/3) - iu_i}{w^{(1)} + u^{(1)}} - \frac{i^{\alpha}\, u_i}{u^{(\alpha)}} \tag{9}$$

Now, it is easy to see that the matrix M defined in Equation (1) has rank 3 in general; when, though,

$R_i = G_i = B_i$, it has rank 1, and its largest (and only nonzero) eigenvalue is

$$\lambda_1 = \frac{2}{3} \frac{u^{(2)} - u^{(1)}}{w^{(1)} + u^{(1)}} \ . \tag{10}$$

To maintain subcriticality we require that $\lambda_1 < 1$ at all times throughout the algorithm. This is reminiscent of the Molloy-Reed criterion [19] for the emergence of a giant component in the configuration model which, in terms of the total degree sequence $a_i = w_i + u_i$, can be expressed as

$$\frac{a^{(2)} - a^{(1)}}{a^{(1)}} < 1 \ .$$

Equation (10) allows us to calculate k explicitly from Equation (3) and get

$$k = \frac{3(w^{(1)} + u^{(1)})}{3w^{(1)} + 5u^{(1)} - 2u^{(2)}} \frac{u^{(\alpha+1)}}{u^{(\alpha)}} \ .$$

Assume now that T is such that for $0 \le x \le T/n$, the solution to the differential equations remains strictly inside the domain, i.e. $\lambda_1 < 1 - \xi$ for some constant $\xi = \xi(\theta)$. Then Theorem 11 then implies that w.h.p. for all $0 \le t \le T$ and all $c \in \{R, G, B, W\}$,

$$C_i(t) = c_i(t/n) \cdot n + o(n) \ .$$

To determine initial conditions and T for which the above holds we need to solve the system of equations (8)–(9). For that it is beneficial to rescale time by taking

$$\mathrm{d}v = \frac{k}{w^{(1)} + u^{(1)}} \, \mathrm{d}x \ .$$

That is, we measure time not by how many rounds A has run, but by how many edges have been exposed, and scale that as well according to the number of remaining edges. (Exposing all the edges now takes infinite time, but this is no cause for worry.) This gives differential equations in terms of v:

$$\frac{\mathrm{d}w_i}{\mathrm{d}v} = -iw_i \tag{11}$$

$$\frac{\mathrm{d}u_i}{\mathrm{d}v} = (i+1)(w_{i+1} + u_{i+1}/3) - \left(i + \frac{3w^{(1)} + 5u^{(1)} - 2u^{(2)}}{3u^{\alpha+1}} \, i^{\alpha} \right) u_i \ . \tag{12}$$

In Appendix E we show how solving equations (11)–(12) with $\alpha = 0$ allows us to recover analytically the original list-coloring result of Achlioptas and Molloy [3]. For $\alpha > 0$, however, while parts of this system are solvable analytically (for instance, $w_i(v) = w_i(0)\, e^{-iv}$) we have to resort to high-precision numerical integration. We note that perhaps some other form for the preference function $h(i)$, e.g. $h(i) = \exp(si)$, may result in a system that has a closed form solution, but we have not been able to find one.

Remark: Since, in the end, we are only interested in the case of the symmetric initial conditions, our development of multitype branching processes might appear to be somewhat redundant. This is not so. Being able to consider the dynamics for the non-symmetric case (and prove that the process still behaves reasonably) is what allows us to define a domain in which we can apply Theorem 11. Without such "wiggle room" (around the trajectory corresponding to symmetric initial conditions) that would be impossible. Multitype branching processes give us an algebraic way of establishing "list-stability" [3], i.e. that throughout the course of the algorithm, for each degree there is approximately the same number of 2-color vertices with each color pair.

7 Integrating the differential equations

To prove Lemma 6, and complete the proof of Theorem 4, we integrate the differential equations (11)–(12) with initial conditions corresponding to a list sequence where all vertices have degree 4, $(1-\epsilon n)$ vertices have all 3 available colors and $(\epsilon/3)n$ vertices have each of the 3 different 2-color lists. Clearly, proving positive probability of success for this list sequence implies Lemma 6 where all vertices have initially all 3 colors. Recall that the reason for stripping off these colors is so that initially there are $\Omega(n)$ copies belonging to 2-color vertices, i.e. so that $\mathcal{L}(0)$ is (ϵ, β)-subcritical. Even with $\alpha = 0$ we find that λ_1 is never more than 0.91283. We define the number of rounds T implicitly by setting the rescaled time $v = 2$, when the uncolored vertices are (δ, ϵ)-easy with $\delta = 0.00033$ and $\epsilon = 0.00044$.

In fact, even if the initial degree distribution has κn vertices of degree 5 and $(1-\kappa)n$ of degree 4, for $\kappa = 0.219$ we have $\lambda_1 < 0.99973$ at all times for $\alpha = 0$, so we claim that these graphs are 3-colorable as well. Setting $\alpha = 20$ improves this to $\kappa = 0.3$, but even $\alpha = 50$ only increases this to 0.302.

Similarly, to prove Lemma 7, and complete the proof of Theorem 3, we start with initial conditions corresponding to a list sequence where the degree sequence is as in Lemma 7 while $R_i = G_i = B_i = \phi n$ for each $0 \le i \le \Delta$. Note that since w.h.p. $|L| < \phi n$, the list configuration in Lemma 7 w.h.p. has fewer than ϕn 2-color vertices in total, so our initial conditions, again, correspond to imposing a handicap on the algorithm. With $d = 4.03$, $\Delta = 30$ and $\alpha = 13$ we find that λ_1 is never more than 0.99909. At $v = 1.2$ the uncolored vertices are (δ, ϵ)-easy with $\delta = w^{(0)} = 0.059$ and $\epsilon = 2(w^{(1)} + u^{(1)}) - (w^{(2)} + u^{(2)}) = 0.010$.

By varying α and requiring that $\lambda_1 < 1$ at all times we obtain the following values of feasible d_α:

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13
d_α	3.847	3.899	3.936	3.961	3.980	3.993	4.003	4.010	4.016	4.020	4.024	4.027	4.029	4.030

In all cases we integrated to 16 digits of precision (the above values are rounded down).

The first of these values is familiar from the original list-coloring algorithm of [3]. In Appendix E we show how one can also derive this special case analytically from our differential equations. The values d_α appear to converge somewhat slower than geometrically, but we conjecture that as $\alpha \to \infty$ the only improvement is in the third decimal digit. We also note that reducing the graph to its 3-core first (by repeatedly removing all vertices of degree smaller than 3) does not help much. Using the techniques of [20] to get the degree sequence of the 3-core and plugging it in our differential equations gives a small improvement in d, to 4.04. So, in fact, this gives us a slight improvement to Theorem 1.

References

[1] Dimitris Achlioptas. Setting two variables at a time yields a new lower bound for random 3-SAT. In *32nd Annual ACM Symposium on Theory of Computing (Portland, OR, 2000)*, pages 28–37. ACM, New York, 2000.

[2] Dimitris Achlioptas and Ehud Friedgut. A sharp threshold for k-colorability. *Random Structures Algorithms*, 14(1):63–70, 1999.

[3] Dimitris Achlioptas and Michael Molloy. The analysis of a list-coloring algorithm on a random graph. In *38th Annual Symposium on Foundations of Computer Science (Miami, FL, 1997)*, pages 204–212. IEEE Comput. Soc. Press, Los Alamitos, CA, 1997.

[4] Noga Alon and Joel H. Spencer. *The probabilistic method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., New York, 1992. with an appendix by Paul Erdős, a Wiley-Interscience Publication.

[5] E.A. Bender and E.R. Canfield. The asymptotic number of labelled graphs with given degree sequences. *J. Combin. Theory Ser. A*, (24):296–307, 1978.

[6] Stefan Boettcher and Allon Percus. Optimization with extremal dynamics. *Phys. Rev. Lett.*, (86):5211–5214, 2001.

[7] D. Brelaz. New methods to color the vertices of a graph. *Comm. of the ACM*, (22):251–256, 1979.

[8] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 5:17–61, 1960.

[9] Alan M. Frieze and Colin J.H. McDiarmid. Algorithmic theory of random graphs. *Random Structures Algorithms*, 10(1-2):5–42, 1997.

[10] Alan M. Frieze and Stephen Suen. Analysis of two simple heuristics on a random instance of k-SAT. *J. Algorithms*, 20(2):312–355, 1996.

[11] Alexis C. Kaporis, Lefteris M. Kirousis, and Yannis C. Stamatiou. A note on the non-colorability threshold of a random graph. *Electron. J. Combin.*, 7:R29, 2000.

[12] Richard Karp and Michael Sipser. Maximum matchings in sparse random graphs. In *22nd Annual Symposium on Foundations of Computer Science*, pages 364–375. IEEE Comput. Soc. Press, Los Alamitos, CA, 1981.

[13] Thomas G. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *J. Appl. Probability*, 7:49–58, 1970.

[14] Thomas G. Kurtz. *Approximation of population processes*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa., 1981.

[15] M. Mitzenmacher. Analyses of load stealing models based on families of differential equations. *Theory Comput. Syst.*, 34(1):77–98, 2001.

[16] Michael Mitzenmacher. Studying balanced allocations with differential equations. *Combin. Probab. Comput.*, 8(5):473–482, 1999.

[17] Charles J. Mode. *Multitype branching processes. Theory and applications*. American Elsevier Publishing Co., Inc., New York, 1971.

[18] Michael Molloy. *Random graphs on a fixed degree sequence*. Ph.D Thesis, Carnegie Mellon University, 1995.

[19] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Structures Algorithms*, 6(2-3):161–179, 1995.

[20] Boris Pittel, Joel H. Spencer, and Nicholas C. Wormald. Sudden emergence of a giant k-core in a random graph. *J. Combin. Theory Ser. B*, 67(1):111–151, 1996.

[21] Michel Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Insitute Des Hautes Études Scientifiques, Publications Mathématiques*, (81):73–205.

[22] Nicholas Wormald. *Some Problems in the Enumeration of Labelled Graphs.* Ph.D. Thesis, Newcastle University, 1978.

[23] Nicholas C. Wormald. Differential equations for random processes and random graphs. *Ann. Appl. Probab.*, 5(4):1217–1235, 1995.

A Handling the high-degree and the leftover vertices

A.1 Proof of Lemma 3

To prove part 1 of Lemma 3 we consider the following procedure, called `High`:

1. Let E_H be the set of edges incident to high-degree vertices. Let H be the graph induced by E_H.
2. Fail if any of the following is true:
   (a) H contains a component with more than one cycle.
   (b) H contains more than $\log n$ cyclic components.
   (c) H contains a component of size greater than $\log^2 n$.
3. Let Y be the set of low-degree vertices in the cyclic components of H. Let E_Y be the set of edges incident to vertices in Y. Let K be the graph induced by $E_H \cup E_Y$.
4. Fail if either of the following is true:
   (a) Some edge in E_Y connects two vertices in Y.
   (b) Some vertex of K that is not a vertex of H is incident to more than one edges of E_Y.
5. Let L be the set of low-degree vertices in K that are not in Y. Find a 3-coloring η of K in which every vertex in L has a monochromatic neighborhood.

Remark: Recall that our goal is to color the high-degree vertices and leave us with a small number of 2-color low-degree vertices. Now that if H were acyclic, i.e. the high-degree vertices and their neighborhoods formed a forest, then we could accomplish this by 2-color H. Then every low-degree neighbor of a high-degree vertex automatically has a monochromatic neighborhood, so we can give it the list consisting of the other two colored. Unfortunately, with positive probability H has at least one unicyclic component and this approach can fail if low-degree vertices occur in a cycle. We deal with this issue by adding one more layer of neighbors to the cyclic components of H to form K.

Lemma 11 `High` *never fails to find a suitable coloring in step 5.*

To prove this we will use the following lemma.

Lemma 12 *Let G be a 3-colorable graph and S an independent set of G such that no cycle passes through any vertex in S. Then G can be 3-colored so that every vertex in S has a monochromatic neighborhood.*

Proof. Consider the components C_i of $G \setminus S$. If we define a graph K on the C_i where C_i, C_j are connected if they share (in G) a common neighbor $u \in S$, then K is acyclic since no $u \in S$ appears in any cycle of G. Also, any two such components share a unique $u \in S$, and that u is a neighbor (in G) of at most one vertex v_i in each C_i. Since G is 3-colorable, so is each C_i. Moreover, for any $v \in C_i$ and $c \in \{R, G, B\}$, there is a 3-coloring η_i of C_i such that $\eta_i(v) = c$. Therefore, we can proceed by 3-coloring the C_i one-by-one. Each time we color some C_i with a coloring η_i, consider all uncolored C_j connected to C_i in K through some $u \in S$ and its neighbors $v_i \in C_i$ and $v_j \in C_j$. Set $\eta_j(v_j) = \eta_i(v_i)$ and color C_j accordingly. Since K is acyclic, this procedure will never specify the color of more than one vertex of any component, so we will succeed in coloring every component. Clearly, each $u \in S$ has a monochromatic neighborhood since we require its neighbors v_i in every C_i to have the same color. □

Proof of Lemma 11. We first claim that if we reach step 5, then K consists of unicyclic components and is therefore 3-colorable. For this, observe that if we pass condition 2a, every component of H must be unicyclic. K cannot have a multicyclic component unless some edge of E_Y lies inside a cyclic component of H, or unless some edges of E_Y connect two cyclic components of H. Passing condition 4 implies that neither of these possibilities occurs. We also claim that L is an independent set in K. For that note that L and Y are disjoint by definition and that the only edges between low-degree vertices in K have at least one endpoint in Y. Finally, note each vertex in L has degree 1 in K and, thus, no cycle passes through any vertex in L. Thus, the lemma follows by Lemma 12. □

Lemma 13 *If step 2 of* `High` *succeeds, then step 4 succeeds w.h.p.*

Proof. Since we have passed conditions 2b and 2c, the total number of vertices in H lying on cyclic components is bounded by $\log^3 n$. Since the maximum degree of the graph is bounded by $2 \log n / \log \log n$, this implies $|E_Y| < \log^4 n$. Therefore, with certainty, there are $\Omega(n)$ unexposed copies in the rest of the graph, i.e. not in H, and those vertices have bounded degree. As a result, the probability of each of the events in step 4 is $O(|E_Y|^2/n) = O(\log^8 n/n)$. □

Lemma 14 *Step 2 of* `High` *succeeds w.h.p.*

Proof. Recall that we are considering a random multigraph on a degree sequence $\mathcal{D}^*$ and that $\Delta = 30$. Pick a random copy uniformly at random among all copies belonging to high-degree vertices. Let v be the corresponding vertex and let $C(v)$ be the number of copies belonging in the same connected component as v. We claim that there exists a constant $\rho > 0$ such that

$$\Pr[C(v) = s] < (1 - \rho)^s + O(1/n) \ . \tag{13}$$

Using this claim and arguing as in Lemma 10 we get that the probability that v lies in a cyclic component is $O(1/n)$ and the probability that it lies in a multicyclic component is $O(1/n^2)$. Therefore, the expected number of cyclic components in H is $O(1)$ while the expected number of multicyclic components is $o(1)$. Markov's inequality and the union bound, respectively, imply that w.h.p. H has no more than $\log n$ cyclic components and no multicyclic components. Moreover, since the number of vertices in $C(v)$ cannot be greater than its number of copies plus 1, (13) readily implies that H has no component of size $\log^2 n$.

To prove (13) we argue as in the proof of Lemma 9 to control the shift in the degree sequence in the course of discovering the component of v. Again, we can add a $o(1)$ term to the expected progeny of each vertex and get that with probability $1 - O(1/n)$ this yields a branching process that dominates the breadth-first search tree. Ignoring $o(1)$ terms we use the following branching process.

Recall that the configuration model consists of a matching between copies of vertices, where each vertex of degree i has i partner copies. Consider a two-type branching process defined on these copies, in which a copy of a high-degree vertex gives birth to all the partner copies of its match, while a copy of a low-degree vertex gives birth to the partner copies of its match only if that match is of high degree. These birth rates form a 2×2 matrix,

$$M = \begin{pmatrix} p_{\text{high}}\, d_{\text{high}} & p_{\text{high}}\, d_{\text{low}} \\ p_{\text{low}}\, d_{\text{high}} & 0 \end{pmatrix}$$

where p_{high} and p_{low} are the probabilities that a random copy is of high- or low-degree respectively, while d_{high} and d_{low} and are the average number of copies born by a high- or low-degree copy. The largest eigenvalue of M is $\lambda = \frac{1}{2}\, d_{\text{high}}\, p_{\text{high}} \left(1 + \sqrt{1 + 4\,\frac{d_{\text{low}}\, p_{\text{low}}}{d_{\text{high}}\, p_{\text{high}}}}\right)$. For the degree sequence $\mathcal{D}^*$ and with $\Delta = 30$ we get $\lambda < 10^{-15}$, giving a subcritical branching process. $\square$

Since `High` succeeds w.h.p. and since, by Fact 1, K is simple with positive probability we, get that with positive probability K is simple and has a 3-coloring such that all vertices in L have monochromatic neighborhoods. This completes the proof of part 1 of Lemma 3.

To prove parts 2 and 3 of the lemma we first observe that the vertices in B are the original low degree vertices minus the vertices in Y. Moreover, as we proved above, w.h.p. $|E_Y| < \log^4 n$.

Part 2 of the lemma now follows from observing that a vertex in B has degree i if its original degree was $j \geq i$ and $j - i$ of its edges were incident to vertices in K, i.e. to either H or Y. Now, each copy of a low-degree vertex is matched to a high-degree vertex with probability $p_{\text{high}} = \phi/d$. Summing over the $\binom{j}{i}$ ways for this to happen gives the sum of part 2. For Y it suffices to observe that w.h.p. $|E_Y| < \log^4 n$ and hence its effect is captured by the $o(n^{2/3})$ term. This calculation gives us the expected number of vertices of degree i in B. Concentration now follows from standard arguments.

Part 3 of the lemma follows from observing that the vertices in L are either neighbors of high-degree vertices or adjacent to some vertex in Y. Clearly, the number of vertices adjacent to high-degree vertices cannot exceed ϕn, while the number of vertices adjacent to Y cannot exceed $|E_Y|$. Moreover, it is clear that w.h.p. $\Omega(n)$ of the ϕn edges incident to high-degree vertices are between high-degree vertices. $\square$

Numerically, for $d = 4.03$ and $\Delta = 30$, we have $\phi = 4.475 \times 10^{-16}$. Thus in the case studied here, the degree sequence b_i of Lemma 3 is almost identical to the original Poisson distribution with mean d.

A.2 Proof of Lemma 5

In [19] it was shown that if a degree sequence $\mathcal{D}$ is (δ, ϵ)-easy, then with positive probability a random configuration on $\mathcal{D}$ gives rise to a simple graph with: i) no multicyclic component, and ii) no more than $C \log n$ cycles, for some C depending on the maximum degree. Here, we claim a slightly sharper result, namely that with positive probability such a degree sequence gives rise to a simple graph with no multicyclic component and no more than B cycles, for some *constant* B. Given this result the lemma follows trivially since with constant probability each such cycle contains at least one of the ϵn 3-color vertices and, hence, the graph is list-colorable.

To prove this claim we argue as in Lemma 14, picking a random copy belonging to a maximum degree vertex v and showing that the probability that v lies in a cyclic component is bounded by $O(1/n)$. For that, again, we consider the breadth-first tree from v and prove that with probability $1 - O(1/n)$ it is dominated by a branching process whose total progeny has a distribution with geometric dropoff. Arguing as in Lemma 10, we use this dropoff to prove that the probability of getting a cyclic component is $O(1/n)$ and the probability of getting a multicyclic component is $O(1/n^2)$. The geometric dropoff holds because the branching process has bounded expected progeny since it is subcritical, and the total progeny is concentrated as the sum of independent variables with bounded range (i.e. the progeny of each copy). In fact, the eigenvalue of our branching process is bounded away from 1 iff $\mathcal{D}$ is (δ, ϵ)-easy for some $\delta > 0$. $\square$

B Miscellaneous proofs

Proof of Lemma 2. Let $\mathcal{D}^*$ be any degree sequence as in Lemma 2 and let C be a random configuration on $\mathcal{D}^*$. Let $\mathcal{E}_1$ be the event that K is simple and can be 3-colored so that all vertices in L have monochromatic neighborhoods. Let $\mathcal{E}_2$ be the analogous event for the multigraph induced by B. Clearly, if both $\mathcal{E}_1$ and $\mathcal{E}_2$ hold, then then the multigraph induced by C is simple and 3-colorable. Note that the events $\mathcal{E}_1, \mathcal{E}_2$ depend on disjoint sets of edges (but are not independent) and Lemma 3 asserts that $\mathcal{E}_1$ holds with positive probability.

Whenever $\mathcal{E}_1$ holds we will first 3-color K so that all vertices in L have monochromatic neighborhoods. We then uncolor each vertex in L and assign it the 2-color list avoiding the color assigned to its neighbors. We assign all 3 available colors to the other vertices in B. By parts 2 and 3 of Lemma 3 w.h.p. this yields a degree sequence and a list assignment which satisfy the conditions of Lemma 4. Therefore, with positive probability the graph induced by B is list-colorable and simple. Since list-colorability here implies that each vertex in L has a monochromatic neighborhood, we get that $\mathcal{E}_1 \wedge \mathcal{E}_2$ must hold with positive probability. □

Proof of Lemma 8. The miracle of linearity of expectation make it trivial to see that the expected population of type j after z generations is $M^z p$ and therefore the total expected progeny is given by

$$m(p) = \sum_{z=0}^{\infty} M^z p \ .$$

Note now that the sum $\sum_{z=0}^{\infty} M^z$ converges to $(I-M)^{-1}$ iff all eigenvalues of M have modulus less than one. Since the entries of M are real and nonnegative, its dominant eigenvalue is real and positive, so this amounts to requiring that $\lambda_1 < 1$.

To prove the variational part we will need the following two facts from linear algebra:

$$|(I-A)^{-1}| = \frac{1}{1-|A|} \ , \qquad \text{valid if } \lambda_1(A) < 1 \tag{14}$$

$$|A^{-1} - B^{-1}| \leq 2 \times |A-B| \times \max\{|A^{-1}|^2, |B^{-1}|^2\} \ . \tag{15}$$

Moreover, we note that $|N| = |M + N - M| \leq |M| + |N - M| \leq 1 - \delta + \delta/2 < 1$ which by (14) implies

$$|(I-N)^{-1}| = \frac{1}{1-|N|} < 2/\delta \ . \tag{16}$$

Using the triangle inequality, the fact $|Ax| \leq |A| \times |x|$ and the fact $|p| \leq 1$, we get

$$\begin{aligned}
|m(p) - n(q)| &= |(I-M)^{-1}p - (I-N)^{-1}q| \\
&= \left|\left((I-M)^{-1} - (I-N)^{-1}\right)p + (I-N)^{-1}(p-q)\right| \\
&\leq \left|\left((I-M)^{-1} - (I-N)^{-1}\right)p\right| + |(I-N)^{-1}(p-q)| \\
&\leq \left|(I-M)^{-1} - (I-N)^{-1}\right| + |(I-N)^{-1}| \times |p-q| \\
&\leq \left|(I-M)^{-1} - (I-N)^{-1}\right| + (2/\delta) \times \zeta \ . \qquad (17)
\end{aligned}$$

To bound $|(I-M)^{-1} - (I-N)^{-1}|$ we will use (15) and (16) as follows:

$$\begin{aligned}
|(I-M)^{-1} - (I-N)^{-1}| &\leq 2 \times |M-N| \times \max\{|(I-M)^{-1}|^2, |(I-N)^{-1}|^2\} \\
&\leq 2 \times |M-N| \times \frac{4}{\delta^2} \\
&= (8/\delta^2) \times \epsilon \ . \qquad (18)
\end{aligned}$$

Putting together (17) and (18) we get $|m(p) - n(q)| \leq 2\zeta/\delta + 8/\delta^2$. □

C Relating rounds and branching processes

Proof of Lemma 9. We will refer to the random process corresponding to a single round as P. Recall that we are only interested in P for subcritical list sequences.

We start by observing that P proceeds by repeatedly determining which unexposed vertex copies are adjacent to the vertex being colored. Since an unexposed vertex copy is *always* picked uniformly at random among all unexposed vertex copies, we see that the probability of a given copy being picked in each step is exactly $1/E$ where E is the current number of unexposed copies. Thus, the coloring of each vertex gives rise to a probability distribution governing the resulting number of 1-color vertices of each (color, degree) pair. Moreover, this probability distribution depends only on the current list sequence.

Now, just before a round begins, for every color $c \in \{R, G, B\}$ and every degree $0 \le i \le \Delta$ there is a probability distribution $f_{(c,i)} : \mathbb{N}^{3\times\Delta} \to \mathbb{R}$, governing the number of 1-color vertices of each color and degree generated by assigning color c to a vertex of degree i. Note that the random experiment corresponding to $f_{(c,i)}$ allows for the possibility of self-loops and multiple edges; events which tend to diminish the 1-color "progeny" of the colored vertex. On the other hand, as long as there are $\Omega(n)$ unexposed copies, since Δ is bounded, each one of these events has probability $O(1/n)$. Let $f^{\circ}_{(c,i)} : \mathbb{N}^{3\times\Delta} \to \mathbb{R}$ be another probability distribution and let $X_{(c,i)} \in \mathbb{N}^{3\times\Delta}$ be a random variable whose value is determined as follows: we draw, independently, one sample from $f_{(c,i)}$ and one from $f^{\circ}_{(c,i)}$; for each color and degree we take the value of X to be the of sum the corresponding values in $f_{(c,i)}, f^{\circ}_{(c,i)}$. From our observation above regarding self-loops and multiple edges we see that it is trivial to construct a $f^{\circ}_{(c,i)}$ with $f^{\circ}_{(c,i)}(0, \ldots, 0) = 1 - O(1/n)$ so that $X_{(c,i)}$ dominates the probability distribution of 1-color vertices generated by assigning color c to an i-degree vertex, *regardless* of the presence of self-loops and multiple edges.

Consider now a pair of list sequences $\mathcal{L}_1, \mathcal{L}_2$ with the property that i) $E_1 \ge \delta n$, for some $\delta > 0$, and ii) for every (color list, degree) pair, the difference in the corresponding number of vertices between $\mathcal{L}_1$ and $\mathcal{L}_2$ is bounded by $h(n) = o(n)$. Let $f^1_{(c,i)}, f^2_{(c,i)}$ be the probability distributions governing 1-color progeny in $\mathcal{L}_1, \mathcal{L}_2$, respectively and let $f^{\star}_{(c,i)} : \mathbb{N}^{3\times\Delta} \to \mathbb{R}$ be another probability distribution. Similarly to our construction above, let $X_{(c,i)}$ be a random variable whose value is determined as follows: we draw, independently, one sample from $f^1_{(c,i)}$ and one from $f^{\star}_{(c,i)}$; for each color and degree we take the value of X to be the of sum the corresponding values in $f^1_{(c,i)}, f^{\star}_{(c,i)}$. Again, it is trivial to construct a probability distribution $f^{\star}_{(c,i)}$ in which $f^{\star}_{(c,i)}(0, \ldots, 0) = 1 - O(h(n))$ so that $X_{(c,i)}$ dominates the probability distribution for 1-color vertices generated by assigning color c to an i-degree vertex in $\mathcal{L}_2$.

We will consider now a multitype branching process Q with $3 \times (\Delta + 1)$ types, one for each (c, i) pair, where $c \in \{R, G, B\}$ and $0 \le i \le \Delta$. The probability that the progenitor of Q is of a given type is set to be equal to the corresponding value for P. Let $h(n) = \log^3 n$. For each (c, i), the progeny of vertices of type (c, i) in Q is determined as follows: we pick, independently, one sample from $f_{(c,i)}$, one from $f^{\circ}_{(c,i)}$ and one from $f^{\star}_{(c,i)}$. We take the progeny for each type of offspring to be the of sum of the values for that type given by $f_{(c,i)}, f^{\circ}_{(c,i)}, f^{\star}_{(c,i)}$. We make the following claim: P and Q can be coupled so that whenever the total progeny of Q is smaller than $\log^2 n$, the progeny of Q dominates the progeny of P. To see this observe that if the progeny of P is smaller than $\log^2 n$ then, since Δ is bounded, throughout P's run for each $c \in \{R, G, B, W\}$, C_i cannot have changed by more than $O(\log^2 n)$. Therefore, since $h = \log^3 n$, every time we color a vertex in P, its 1-color progeny is dominated by the progeny in Q.

Now, clearly, the expected number of vertices of each type colored in P is bounded by the expected number of vertices of that type in Q plus the expected number of vertices in P whenever Q has total progeny greater than $\log^2 n$. We will prove below that the probability of Q having total progeny greater than $\log^2 n$ is $o(1/n)$. Since there are never more than $O(n)$ unexposed copies in the graph, we see that the expected progeny of P for each type is within $o(1)$ of the expected progeny for Q.

To bound the expected total progeny of each type in Q we define a multitype branching process F identical to B, except that in F in order to determine the progeny of each vertex we only take into account the sample from $f_{(c,i)}$ each time. Now, to bound the progeny of B we first observe that the entries in its matrix of expectations are within $o(1)$ of the corresponding entries for F. Therefore, since F is subcritical, part 2 of Lemma 8 readily implies that for each type, the total progenies of B and F differ by $o(1)$. Combining this with the $o(1)$ term from the possibility that B has more than $\log^2 n$ total progeny we readily get Equation (3).

To prove our claim about the probability of B having more than $\log^2 n$ total progeny, we will prove the following stronger claim: there exists $\rho > 0$ such that the probability that B's total progeny equals s is smaller than $(1 - \rho)^s$. Note that this claim readily implies part 2c of Lemma 9. Moreover, note that this claim is trivial to prove for single-type subcritical branching processes. Here, the proof of our claim, which we omit, relies on using Talagrand's inequality [21] and showing how we can inductively dominate any subcritical multitype process with a (potentialy infinite) sequence of single-type processes.

Finally, to prove Equations (5) and (5) we argue as follows. Since in the configuration model copies are matched uniformly at random, the probability that a vertex v is matched to the copy being exposed is proportional to $\deg(v)/E$. So, in particular, each copy of the vertex being colored "hits" a 3-color vertex with probability $iw_i/(w^{(1)}+u^{(1)})$, turning it into a 2-color vertex. Since, in expectation, there are a total of k copies colored per round, Equation (5) follows.

Equation (5) is derived in a similar manner, except that we need to sum together several effects. The first term comes from the fact that when we color a vertex with color c, each of its copies has probability $(i+1)(w_{i+1}+c_{i+1})/(w^{(1)}+u^{(1)})$ of creating a 2-color vertex of degree i whose list lacks c. Multiplying this probability by the number, k_C, of copies colored c during the round gives the first term. The second term is simply the probability that any copy hits a vertex in bucket $(\{r,g,b\}-c,i)$, multiplied by the total number of copies colored per round. Finally, the third term is the probability that the free step of A chooses a vertex in bucket $(\{r,g,b\}-c,i)$, i.e. the sum of $h(i)$ over the vertices in that bucket divided by the same sum over all 2-color vertices. $\square$

Proof of Lemma 10. Before we proceed with the proof, let us note that it is not obvious how to get a (stronger) lemma asserting that if $\mathcal{L}(t)$ is subcritical for all $0 \le t \le T$ then with positive probability there are no bad vertices at time T. This is because the evolution of $\mathcal{L}(t)$ is almost, but not perfectly, independent of whether bad vertices are generated in $0 \le t \le T$. Having said that, we prove the lemma as follows.

We start by observing that every time we expose a copy in the course of a round the probability that a bad event occurs, i.e. that we get a self-loop, a multiple edge, or a 0-color vertex, is bounded by the following ratio: the number of unexposed copies belonging to vertices that have lost at least one color during the current round, divided by the total number of unexposed copies belonging to all other vertices. Moreover, note that this fact holds independently of the rest of the history of the process, i.e. these two are the only relevant parameters for determining the probability of a bad event. Yet, as mentioned in the remark, the total number of copies exposed in a round is not independent from discovering something bad in that round. Thus, to dominate the bad events we proceed as follows.

Throughout the algorithm's execution let us say that a copy is "dangerous" if it belongs to a vertex that has lost at least one color during the current round. Let $s = 0, 1, \ldots$ enumerate the copies exposed in the course of the algorithm and let $X(s)$ be the number of "dangerous" copies just before we expose the s-th copy. Recall now that if $\mathcal{L}(t)$ is subcritical there is some $\alpha > 0$ such that there are at least αn unexposed copies at time t. So, if $\mathcal{L}(t)$ is subcritical for all $0 \le t \le T$, there exists some $\gamma > 0$ such that when we expose the s-th copy in the course of the T rounds, the probability of a bad event occuring is at most $X(s)/(\gamma n)$.

Imagine now that before the algorithm starts we perform Z Bernoulli trials, each one having probability of success $p = 1/(\gamma n)$, and "conceal" their outcomes. When we run the algorithm, when we expose the s-th copy, we also expose the outcome of $2X(s)$ of the Bernoulli trials. We will say that this procedure fails if any of the following occurs: i) some Bernoulli trial succeeds, ii) $X(s) > \gamma n/2$, iii) we run out of Bernoulli trials. Recalling that $\Pr[\mathrm{Bin}(2n,p) > 0] \ge np$ for all $np < 1/2$, we see that the failure of this procedure dominates the occurrence of a bad event. Moreover, we can bound the probability of $\sum_s 2X(s) > Z$ as follows.

Observe that $\sum X(s)$ over the course of a single round cannot be greater than square of the number of copies exposed in that round. This is because $X(s)$ decreases by at most 1 with each copy exposed. Moreover, note that if $\mathcal{L}(t)$ is subcritical for all $0 \le t < T$, part 2c of Lemma 9 implies that there exists some $\rho > 0$ such that for all $0 \le t < T$, the length of round t is dominated by a geometric random variable with parameter $1-\rho$. Thus, analogously to the Bernoulli trials above, we can construct a set of T independent, identically distributed geometric random variables $F_0, \ldots, F_{T-1}$, so that F_t dominates the length of round t. Note that under this construction we also get that the event $X(s) > \gamma n/2$ is dominated by the event that at least one of these geometric random variables is greater than $\gamma n/2$. Now, by standard arguments, it is easy to show that if $T = \Theta(n)$ then w.h.p. $\sum_{t<T} F_t^2 < \frac{2}{1-\rho} \times T$ and no $F_t > \gamma n/2$. Therefore, as long as $T = \Theta(n)$, we can take $Z = O(n)$ to guarantee that w.h.p. we do not run out of Bernoulli trials. Thus, the lemma follows by observing that $\mathcal{L}(t)$ is indeed subcritical w.h.p. for all $0 \le t \le T$ and that $\mathrm{Bin}(O(n), O(1/n))$ equals 0 with positive probability. $\square$

D Wormald's theorem

In the statement of Theorem 11, below, asymptotics denoted by o and O, are for $n \to \infty$ but uniform over all other variables. In particular, "uniformly" refers to the convergence implicit in the $o()$ terms. For a random variable X, we say $X = o(f(n))$ *always* if $\max\{x \mid \Pr[X = x] \neq 0\} = o(f(n))$. We say that a function f satisfies a *Lipschitz condition* on $D \subseteq \mathbb{R}^j$ if there exists a constant $L > 0$ such that $|f(u_1, \ldots, u_j) - f(v_1, \ldots, v_j)| \leq L \sum_{i=1}^{j} |u_i - v_i|$, for all $(u_1, \ldots, u_j)$ and $(v_1, \ldots, v_j)$ in D.

Theorem 11 (Wormald [23]) *Let $Y_i(t)$ be a sequence of real-valued random variables, $1 \leq i \leq k$ for some fixed k, such that for all i, all t and all n, $|Y_i(t)| \leq Cn$ for some constant C. Let $\mathbf{H}(t)$ be the history of the sequence, i.e. the matrix $\langle \vec{Y}(0), \ldots, \vec{Y}(t) \rangle$, where $\vec{Y}(t) = (Y_1(t), \ldots, Y_k(t))$.*

Let $I = \{(y_1, \ldots, y_k) : \Pr[\vec{Y}(0) = (y_1 n, \ldots, y_k n)] \neq 0$ for some $n\}$. Let D be some bounded connected open set containing the intersection of $\{(s, y_1, \ldots, y_k) : s \geq 0\}$ with a neighborhood of $\{(0, y_1, \ldots, y_k) : (y_1, \ldots, y_k) \in I\}$.[1]

Let $f_i : \mathbb{R}^{k+1} \to \mathbb{R}$, $1 \leq i \leq k$, and suppose that for some $m = m(n)$,

(i) for all i and uniformly over all $t < m$,

$$\mathbf{E}\left(Y_i(t+1) - Y_i(t) | \mathbf{H}(t)\right) = f_i(t/n, Y_0(t)/n, \ldots, Y_k(t)/n) + o(1) \ , \ \textit{always};$$

(ii) for all i and uniformly over all $t < m$,

$$\Pr\left[|Y_i(t+1) - Y_i(t)| > n^{1/5} \middle| \mathbf{H}(t) \right] = o(n^{-3}) \ , \ \textit{always};$$

(iii) for each i, the function f_i is continuous and satisfies a Lipschitz condition on D.

Then

(a) for $(0, \hat{z}^{(0)}, \ldots, \hat{z}^{(k)}) \in D$ the system of differential equations

$$\frac{dz_i}{ds} = f_i(s, z_0, \ldots, z_k), \qquad 1 \leq i \leq k$$

has a unique solution in D for $z_i : \mathbb{R} \to \mathbb{R}$ passing through $z_i(0) = \hat{z}^{(i)}$, $1 \leq i \leq k$, and which extends to points arbitrarily close to the boundary of D;

(b) almost surely

$$Y_i(t) = z_i(t/n) \cdot n + o(n) \ ,$$

uniformly for $0 \leq t \leq \min\{\sigma n, m\}$ and for each i, where $z_i(s)$ is the solution in (a) with $\hat{z}^{(i)} = Y_i(0)/n$, and $\sigma = \sigma(n)$ is the supremum of those s to which the solution can be extended.

Note: The theorem remains valid if the reference to "always" in (i),(ii) is replaced by the restriction to the event $(t/n, Y_0(t)/n, \ldots, Y_k(t)/n) \in D$.

E Coloring with $\alpha = 0$

Here we show that equations (11) and (12) reproduce the dynamics of the original list-coloring algorithm of Achlioptas and Molloy [3] when $\alpha = 0$, i.e. when $h(i)$ is a constant independent of i. As mentioned in Section 2, when $\alpha = 0$, the graph induced by the uncolored vertices is distributed as $G(n', p)$, where n' is the number of uncolored vertices. Therefore, the degree distribution of the uncolored vertices is Poisson with

[1] That is, after taking a ball around the set I, we require D to contain the part of the ball in the halfspace corresponding to $s = t/n \geq 0$.

a time-varying mean δ and the degree of each vertex is independent of its list. Therefore, recalling that w_i and u_i correspond to 3- and 2-color vertices, respectively, we have

$$w_i = \frac{\beta\delta^i}{i!} \quad \text{and} \quad u_i = \frac{\gamma\delta^i}{i!}$$

while their derivatives obey

$$\frac{\mathrm{d}w_i}{\mathrm{d}v} = w_i\left(\frac{1}{\beta}\frac{\mathrm{d}\beta}{\mathrm{d}v} + i\,\frac{1}{\delta}\frac{\mathrm{d}\delta}{\mathrm{d}v}\right) \quad \text{and} \quad \frac{\mathrm{d}u_i}{\mathrm{d}v} = u_i\left(\frac{1}{\gamma}\frac{\mathrm{d}\gamma}{\mathrm{d}v} + i\,\frac{1}{\delta}\frac{\mathrm{d}\delta}{\mathrm{d}v}\right)$$

Since $w^{(1)} = \beta\delta e^{\delta}$, $u^{(1)} = \gamma\delta e^{\delta}$, and $u^{(2)} = \gamma\delta(\delta+1)e^{\delta}$, Equations (11) and (12) now take the simple form

$$\frac{\mathrm{d}\beta}{\mathrm{d}v} = 0 \ , \quad \frac{\mathrm{d}\gamma}{\mathrm{d}v} = (\delta-1)(\beta+\gamma) \ , \quad \frac{\mathrm{d}\delta}{\mathrm{d}v} = -\delta$$

With the initial conditions corresponding to $G(n, d/n)$, namely $\beta(0) = e^{-d}$, $\gamma(0) = 0$, and $\delta(0) = d$, the solution to the differential equations is

$$\beta(v) = e^{-d} \ , \quad \gamma(v) = e^{-(v+de^{-v})} - e^{-d} \ , \quad \delta(v) = de^{-v} \ .$$

Using the facts $w^{(0)} = \beta e^{\delta}$ and $u^{(0)} = \gamma e^{\delta}$ and substituting $x = 1 - e^{-v}$ we get

$$w^{(0)}(x) = e^{-dx} \ , \quad u^{(0)}(x) = 1 - x - e^{-dx}$$

which is precisely the solution given in [3]. The reader can verify that Equation (10) becomes

$$\lambda_1 = \frac{2\gamma\delta}{3(\beta+\gamma)} = \frac{2}{3}\,du^{(0)} \ .$$

This is maximized at $x = \ln d/d$, at which point $\lambda_1 = (2/3)(d - \ln d - 1)$. Setting this to 1 gives $d - \ln d = 5/2$, so $d = -W_{-1}(-e^{-5/2}) = 3.847+$ where W_{-1} is the -1'th branch of Lambert's W function, just as in [3].

F Backtracking

In this section we show how to use a backtracking procedure to improve A so that it succeeds w.h.p. rather than just with positive probability. This allows us to change the "constant fraction" of Theorems 3 and 4 to "almost all" in Theorems 1 and 2.

To do this, we expand our notion of "state" so that it also includes the *edges* between 2-color vertices of the *same* color pair. A connected component of 2-color vertices with the same color pair will be called a "clump." We will maintain the invariant that these clumps are trees and hence 2-colorable.

This time we get a Markov process defined on the number of clumps of each size and degree, where the degree of a clump is defined as the number of unmatched copies it contains. Because w.h.p. we only change (backtrack) the color of $o(n)$ vertices, it turns out that the differential equations for this clump-degree sequence are essentially equivalent to those presented here, inducing the same evolution for the list sequence (up to $o(n)$ terms). Below we outline how our backtracking analysis works. The outline elucidates both why the backtracking has minimal effect on the list sequence dynamics and why it gives success w.h.p.

The rough idea of the analysis is as follows. There are three events we consider bad to happen in the course of a round: (1) the creation of a 0-color vertex, (2) the creation of a unicyclic clump (occurring when some newly 2-color vertex is connected to the clump by exactly two edges), and (3) the creation of a 1-color vertex from a 3-color vertex (that lost 2 colors in the round). Whenever (1) or (2) take place, we backtrack as follows: we reverse the free step at the beginning of the round giving the chosen 2-color vertex the other color; we similarly reverse all forced steps made in the round; we take the forced steps implied by the new choice for the free step and the new choices for the (previously) forced steps. Whenever (2) or (3) occur we actually also color all 3-color vertices that lost a color in the round, and take all the ensuing forced steps.

We use a Boolean variable `strike` to tell us if any one of the above events has occurred in the round; we fail if any two of them occur, or if any one occurs both in our first attempt and after backtracking (two strikes and we're out). We show that the probability of having two strikes in a single round is $\mathrm{polylog}(n)/n^2$. There are also three sudden death events which cause us to fail immediately, each one having probability $\mathrm{polylog}(n)/n^2$ per round: creating more than one 0-color vertex, hitting a 3-color vertex three times, or having a clump become multicyclic. If no round has two strikes or a sudden death then we succeed. Summing over the $O(n)$ rounds of the algorithm gives a $\mathrm{polylog}(n)/n$ probability of failure, so we succeed w.h.p.

We first state our procedure in $G(n, p)$, where we can select a pair of vertices and expose whether there is an edge between them. We then explain how to carry out the procedure in the (slightly more cumbersome) configuration model.

1. (Free step) Select a 2-color vertex v, choose one of its colors randomly and make it a 1-color vertex with that color. Set `strike` = `false`.
2. (Forced steps) While there are 1-color vertices, choose one such vertex randomly, color it with its color c, expose its neighbors containing c in their lists (and only those), and remove c from their lists.
3. If step 2 creates more than one 0-color vertex, fail.
4. (Event 1) If step 2 creates one 0-color vertex:
   - If `strike` = `true`, fail.
   - Else set `strike` = `true` and go to step 5.

   If no 0-color vertices are created go to step 7.
5. (Backtracking) If a 0-color vertex exists let C be its clump (as of the beginning of the round). Reverse the coloring of all other clumps colored in this round by flipping each vertex to the other color in its former color list, and then 2-color C. If there is no 0-color vertex, flip the coloring of every clump.
6. If step 5 creates any new 1-color vertices, go to step 2. Else continue to step 7.
7. Expose the 3-color neighbors of the vertices colored so far. Call this set of vertices U.
8. If any $u \in U$ is a neighbor of three colored vertices, fail.
9. (Event 2) If any $u \in U$ is now a 1-color vertex:
   - If `strike` = `true`, fail.
   - Else set `strike` = `true` and go to step 2.
10. For all $u \in U$, expose all edges between u and 2-color vertices v with $\ell(v) = \ell(u)$. This connects u to clumps of 2-color vertices.
11. If one or more clumps become multicyclic, fail.
12. (Event 3) If one or more clumps become cyclic:
    - If `strike` = `true`, fail.
    - Else set `strike` = `true`, make every vertex in U a 1-color vertex, its color being the one that was just removed from its list, and go to step 5.

Let H be the graph consisting of the vertices colored in the first attempt of the round on both its free and forced steps. If we backtrack, let H' be the the graph consisting of the vertices colored by forced steps in the second attempt. Observe that the new coloring of H created by step 5 is valid. In particular, switching the coloring of all the clumps other than the one containing the 0-color vertex allows us to 2-color this clump, and all the *relevant* edges (those that converted a 2-color vertex to 1-color vertex) are still satisfied. Since we have eliminated the possibility that any 3-color vertex is hit three times in step 8, no 3-color vertex becomes a 0-color vertex on either the first or second attempt.

Note that clumps only become cyclic if they are connected to vertices in u by more than one edge. Then the trick in step 12 of coloring each former 3-color vertex $u \in U$ with exactly the color it lost in the first attempt does two things for us: first, it disconnects each u from the clump it would have become a part of if we adopted the coloring of the first attempt, and thus avoids making that clump cyclic. Second, now that the edges between u and that clump have been exposed, by coloring u we "destroy the evidence" and return to a situation where the only exposed edges are between 2-color vertices with the same color pair. The cost is that coloring u sets off another round of forced steps, and if this round presents any difficulty we fail.

To bound the probability of failure we need the following rather easy lemmata.

1. The probability of event (1) on either the first or second step is $\text{polylog}(n)/n$.
   Idea: Note that a 0-color vertex can only be created on the first or second attempt if H or $H \cup H'$, respectively, has a cycle. Since H is essentially generated by a subcritical branching process, w.h.p. $|H| = \text{polylog}(n)$ and the probability that H has a cycle is $\text{polylog}(n)/n$. If we backtrack, the new coloring of H is valid as observed above. The danger then is that either H' has a cycle of its own, or that H' re-connects with H at some point during the second attempt in the round, creating a cycle in $H \cup H'$. However, note that *none of the edges between clumps that were relevant in the first attempt are relevant in the second.* That is, if an edge between 2-color vertices u, v with different color pairs is relevant for one choice of u's color, it is not relevant for the other. Therefore, while both attemps color the clump containing the vertex chosen on the free step, the sets of relevant edges leading out of that first clump on the two attempts are disjoint. If there are s such edges on the second attempt, we can then think of H' as consisting of the first clump plus, essentially, the progeny of s independent subcritical branching processes. Since $s = \text{polylog}(n)$ and each one has $\text{polylog}(n)$ size, the probability that any of them intersect with each other or with H, creating a cycle in H or $H \cup H'$ is $\text{polylog}(n)/n$.

2. The probability of events (2) and (3) is $\text{polylog}(n)/n$.
   Idea: W.h.p. $|H|$ and $|U|$ are $\text{polylog}(n)$.

3. The probability that we get two strikes is, thus, $\text{polylog}(n)/n^2$.

4. The probability that we fail in steps 3 or 8 is $\text{polylog}(n)/n^2$.

To convert the above procedure to the configuration model we proceed as follows. Unlike in $G(n, p)$, we are not allowed to query whether a given pair of vertices is connected; we can only ask for the partner of a given copy of a given vertex. Therefore, instructions like "expose edges between u and 2-color vertices v with $\ell(v) = \ell(u)$" in step 10, and similar instructions in steps 2 and 7, are not allowed in the configuration model. On the other hand, we cannot afford to expose all of u's neighbors in step 10, since we need to end each round with the only exposed edges being those between 2-color vertices with the same color pair.

To carry out the instructions of steps 2, 7 and 10, we apply the method of deferred decisions to the matching of the unexposed copies. In step 10, for instance, let M be the set of unmatched copies with $\ell = \ell(u)$. Choose the number q of u's copies which will be matched with copies in M (we can write the probability distribution for q explicitly as a function of $|M|$ and $\deg(u)$), choose q copies uniformly at random from M and q from the copies of u, and choose one of the $q!$ permutations uniformly at random by which to match them. This maintains a uniformly random configuration conditioned on the set of clumps. We apply a similar approach to steps 2 and 7, letting M be the appropriate set of unexposed copies in each case.

Finally we note that, analogously to the non-backtracking case it is easier to only let the algorithm run until it colors most of the graph but not all of it. Here, the endgame is even easier. Observe that the backtracking procedure maintains deterministically that the clumps are acyclic and hence 2-colorable. Assume now that we run the algorithm long enough so that when we stop and expose all the edges among the uncolored vertices, the resulting graph consists of trees and unicyclic components (just as we did in the non-backtracking case). Then, deterministically, every cycle in that graph contains either a 3-color vertex, or two vertices with different lists. In either case, each component is list-colorable.