Coloring random graphs

Michael Krivelevich * Benny Sudakov †

Abstract

We present a randomized polynomial time algorithm that colors almost every graph on n vertices in $n/(\log_2 n + c\sqrt{\log_2 n})$ colors for every positive constant c.

Keywords: Random graphs; Graph coloring; Analysis of algorithms

1 Introduction

A random graph $G(n,p)$ is a graph on n labeled vertices, in which every pair of vertices is chosen to be an edge of G randomly and independently with probability p. In particular, when $p = 1/2$, we get a probability space where all labeled graphs on n vertices are equiprobable, thus making this case especially interesting for study. A subject of the theory of random graph is the investigation of the asymptotic behavior of various graph invariants. We say that a graph property A holds *almost surely* (a.s.) if the probability that $G(n,p)$ has A tends to one as n tends to infinity.

One of the most important parameters of a random graph $G(n,p)$ is its chromatic number, which we denote by $\chi(G(n,p))$. The problem of determining the asymptotic behavior of $\chi(G(n,p))$ was one of the major open problems in the theory of random graphs. Trivially $\chi(G) \geq |V(G)|/\alpha(G)$, where $\alpha(G)$ denotes the size of the largest independent set in G. It has been known for a long time that $\alpha(G(n,1/2)) \leq 2\log n$ a.s. (all logarithms in this paper are in base two) and therefore $\chi(G(n,1/2)) \geq n/2\log n$. Grimmett and McDiarmid [6] and Bollobás and Erdős [4] proved that a naturally defined greedy algorithm uses almost surely at most $(1 + 5\log\log n/\log n)n/\log n$ colors to color $G(n,1/2)$. (See also [2], Ch. 11.3.) Finally, Bollobás proved in [3] that the above mentioned lower bound on $\chi(G(n,1/2))$ gives its true asymptotic behavior, thus solving completely this important question. His proof uses martingales and can be translated into an $n^{\Theta(\log n)}$-time algorithm.

However, no polynomial time algorithm is known to color the random graph $G(n,1/2)$ in at most $n/(1+\epsilon)\log n$ colors for any fixed $\epsilon > 0$. Moreover, even the question of finding in polynomial time

*Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel. Email: krivelev@math.tau.ac.il. Research supported in part by a Charles Clore Fellowship.

†Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel. Email: sudakov@math.tau.ac.il. This research forms part of a Ph.D thesis written by the author under the supervision of Prof. N. Alon.

an independent set of size $(1+\epsilon)\log n$ is open. This problem was posed more than twenty years ago by Karp [7]. Despite the efforts of various researchers (see, e.g., [10], [8]), no real progress on this question has been achieved so far. It is only known that the greedy algorithm, constructing an independent set by looking at the vertices of $G(n,1/2)$ in some fixed order and adding a vertex to the current independent set whenever possible, outputs almost surely an independent set of size at least $\log n - \log\log n$ and at most $\log n + \omega(n)$ for any function $\omega(n) \to \infty$ (see [6],[10]). Jerrum conjectures in [8] that there is no polynomial time algorithm that with probability greater than $1/2$ finds an independent set of size $(1+\epsilon)\log n$. A thorough discussion of this problem as well as of other algorithmic aspects of the theory of random graphs can be found in a survey paper of Frieze and McDiarmid [5].

In this paper we describe a new graph coloring algorithm, which incorporates the greedy procedure with some additional ideas. The analysis of this algorithm yields the following theorem.

Theorem 1 *For any constant $c \geq 1$ there exists a randomized algorithm that colors almost every graph in $G(n,1/2)$ in at most*

$$\frac{n}{\log n + c\sqrt{\log n}} = \frac{n}{\log n}\left(1 - \frac{(1+o(1))c}{\sqrt{\log n}}\right)$$

colors in time $O(n^{6c^2+1})$.

We make no attempt to estimate precisely or to optimize the time complexity of our algorithm.

The size of the average color class found by our algorithm is $\log n + c\sqrt{\log n}$, thus giving a (slight) improvement over the performance of the greedy algorithm.

2 The algorithm

Our algorithm is a mixture of the standard greedy procedure and the "expose-and-merge" technique introduced by Matula [9]. At each iteration of the algorithm we try to find an independent set of size $\log n + c\sqrt{\log n}$ in the subgraph of G spanned by yet uncolored vertices. While finding it, we check a relatively small number of pairs of vertices. Then at the following iterations we "forget" the already exposed part of G (by flipping a fair coin for each pair we have seen) and treat the remaining subgraph as truly random. The resulting independent sets can actually contain some already exposed edges of G, but this will happen rarely.

Algorithm
Input: A random graph $G=(V,E) \in G(n,1/2)$.

$U := V$; $F := \emptyset$; $E_0 = \emptyset$; $k = \lceil \log n + c\sqrt{\log n} \rceil + 1$; $k_1 = \lceil \log n - 2c\sqrt{\log n} \rceil$;
(U is the set of yet uncolored vertices; F is the set of exposed pairs of vertices; $E_0 \subseteq F$ is the set of exposed edges of G.)

1. Choose a random permutation σ of U; let U_1 be the set of the first $\lceil U/2 \rceil$ vertices of U according to σ and let $U_2 = U \setminus U_1$.

2. For each pair of vertices u, v in $F \cap [U]^2$, add/remove the edge (u, v) to/from G randomly and independently with probability $1/2$, where $[U]^2$ denotes the set of all pairs of distinct elements of U..

3. Run the greedy algorithm on $G[U_1]$: construct an independent set by looking at the vertices of U_1 in the order defined by σ and add a vertex to the current independent set I whenever possible. As soon as $|I| = k_1$ stop, otherwise FAIL.

4. Let V_0 be the set of all vertices of U_2, having no neighbors in I. If $|V_0| > 2^{2c\sqrt{\log n}}$ or $|V_0| < 2^{1.9c\sqrt{\log n}}$ FAIL.

5. Choose randomly an independent set J from all independent sets of $G[V_0]$ of size $k - k_1$; if there is no such set FAIL.

6. Denote $R = I \cup J$. Check if there exists an edge from E_0 with both endpoints in R. If yes, color each vertex of R by a distinct fresh color, otherwise color all vertices of R by the same fresh color.

7. $U := U \setminus R$, $F := F \cup [V_0]^2$; $E_0 := E_0 \cup E(G[V_0 \setminus J])$. If $|U| \geq n/\log^3 n$ return to 1.

8. Color the vertices of the remaining set U by distinct fresh colors.

Output: A coloring of G.

The analysis of the above algorithm uses the following claims.

Claim 2.1 *A random graph $G(n, 1/2)$ contains an independent set of size at least $2\log n - 3\log\log n$ with probability at least $1 - \exp(-n^2/\log^9 n)$.*

Proof. See, e.g., Theorem 3.2 of Chapter 7 of [1]. $\square$

Claim 2.2 *The greedy algorithm applied to $G(n, 1/2)$ produces an independent set of size at least $\log n - \sqrt{\log n}$ with probability at least $1 - 1/n^2$.*

Proof. An independent set I output by the greedy algorithm is maximal under inclusion, that is, every vertex outside I has a neighbor in I. The probability of the existence in $G(n, 1/2)$ of a maximal (under inclusion) independent set of size less than $s = \log n - \sqrt{\log n}$ is at most

$$\binom{n}{s}\left(1 - \left(\frac{1}{2}\right)^s\right)^{n-s} \leq \binom{n}{s} e^{-(\frac{1}{2})^s(n-s)} \leq \left(\frac{en}{s}\right)^s e^{-(1+o(1))2^{\sqrt{\log n}}} \ll \frac{1}{n^2}\,. \qquad \square$$

Note that at Step 2 we flip a fair coin for every pair of vertices of G that we have already exposed. This ensures that after having performed this step, the subgraph of G spanned by yet uncolored vertices can be treated as a truly random graph.

By Claim 2.2 and as $c \geq 1$, the probability of ever failing at Step 3 is at most $n \cdot (\log^3 n/n)^2 = o(1)$. Turning to Step 4, we observe that the number of non-neighbors of I in U_2 is binomially distributed with parameters $|U_2|$ and $(1/2)^{|I|}$, where $n/2\log^3 n \leq |U_2| \leq n/2$. Using standard Chernoff-type bounds on the tails of a binomial random variable (see, e.g., [1], Appendix A), it can be easily shown that the probability of ever failing at Step 4 is $o(1)$. By Claim 2.1, the probability of failure at Step 5 is also $o(1)$. Hence our algorithm succeeds almost surely.

Now we estimate the number of colors used by the algorithm. We call an iteration of the algorithm *bad* if at Step 6 every vertex of R is colored by a distinct color. Let X be the number of bad iterations. Then the total number of colors used is at most $n/k + kX + n/\log^3 n$. At each iteration E_0 contains edges spanned by different sets V_0, constructed at Step 4. Since each V_0 has size at most $2^{2c\sqrt{\log n}} < n^{0.05}$, the number of edges in E_0 is at most $n \cdot \binom{n^{0.05}}{2} < n^{1.1}$. Also, the size of U is at least $n/\log^3 n$. A crucial observation here is that due to Steps 1,2 and 5 of the algorithm every k-subset of U is equally likely to become the independent set R formed at Step 6. Note that each edge of E_0 is in at most $\binom{|U|-2}{k-2}$ k-tuples of vertices of U. Therefore the probability that a randomly chosen k-subset of U contains an edge of E_0, that is, the probability that the corresponding iteration is bad, is at most

$$\frac{\binom{|U|-2}{k-2}|E_0|}{\binom{|U|}{k}} \leq \frac{k^2|E_0|}{|U|^2} \leq \frac{1}{n^{0.8}} \; .$$

Hence the expectation of X is less than $n \cdot n^{-0.8} = n^{0.2}$, and by Markov's inequality we get that almost surely $X < n^{0.3}$. Thus the total number of colors used by the algorithm is a.s. at most

$$\frac{n}{k} + kn^{0.3} + \frac{n}{\log^3 n} \leq \frac{n}{k-1} \leq \frac{n}{\log n + c\sqrt{\log n}} \; .$$

To complete the analysis of the algorithm, we need to estimate its complexity. It is easy to see that the most time-consuming part is Step 5, which can be done by an exhaustive search over all $(k-k_0)$-subsets of V_0 and takes $O\left(k^2\binom{|V_0|}{k-k_0}\right) \leq O(n^{6c^2})$ units of time at each iteration. This completes the proof of Theorem 1. $\square$

3 Concluding remarks

1. The above algorithm can be easily adapted for coloring a random graph $G(n,p)$ for any fixed value of p. It will color $G(n,p)$ almost surely in at most $\log(1/(1-p))\frac{n}{\log n + c\sqrt{\log n}}$ colors for any positive constant c.

2. Let $f(k)$ denote the expected number of independent sets of size k in $G(n,1/2)$. It is immediate that $f(k) = \binom{n}{k}2^{-\binom{k}{2}}$. Using the second moment method it can be proven that this random variable is concentrated around its expectation (see. e.g. [2], Ch. 11), thus the number of independent sets of size k in $G(n,1/2)$ is almost surely $(1+o(1))f(k)$. A simple calculation shows that the function $f(k)$ attains its maximum at $k_0 = \log n - (1+o(1))\log\log n$, therefore the independent sets of this size are the most frequent ones in $G(n,1/2)$. This observation can serve as an explanation of the performance of the greedy algorithm, which colors the random graph using independent sets of the most popular size. Also, if $k=\log n+c\sqrt{\log n}$, then the ratio $f(k)/f(k_0)$ is at least one over a polynomial in n. Thus our algorithm uses independent sets of size k which are much less popular than those of size k_0 but are still not too rare. It is worth noting that the above estimate for k is the maximal for which $f(k)/f(k_0)$ is polynomially small, as already for $k=\log n+\omega(n)\sqrt{\log n}$, with $\omega(n)$ being any function tending to infinity arbitrarily slowly, the inverse of the ratio is superpolynomial. Recall that Karp posed the problem of finding in polynomial time an independent set of size $(1+\epsilon)\log n$. The above argument indicates that it may be substantially more difficult even to find an independent set of size $\log n+\omega(n)\sqrt{\log n}$ for any $\omega(n)\to\infty$. Some support to this paradigm is given by the paper of Jerrum [8], who shows (implicitly) that the Metropolis algorithm may require at least $n^{\Omega(\omega^2(n))}$ steps to reach an independent set of size $\log n+\omega(n)\sqrt{\log n}$.

References

[1] N. Alon and J. Spencer, **The probabilistic method**, Wiley, New York, 1992.

[2] B. Bollobás, **Random Graphs**, Academic Press, New York, 1985.

[3] B. Bollobás, *The chromatic number of random graphs*, Combinatorica 8 (1988), 49–55.

[4] B. Bollobás and P. Erdős, *Cliques in random graphs*, Math. Proc. Camb. Phil. Soc. 80 (1976), 419–427.

[5] A. Frieze and C. McDiarmid, *Algorithmic theory of random graphs*, Random Str. Algor. 10 (1997), 5–42.

[6] G. Grimmett and C. McDiarmid, *On colouring random graphs*, Math. Proc. Cam. Phil. Soc. 77 (1975), 313–324.

[7] R. Karp, The probabilistic analysis of some combinatorial search algorithms, in: *Algorithms and Complexity: New Directions and Recent Results*, J. F. Traub, Ed., Academic Press, New York, 1976, pp. 1–19.

[8] M. Jerrum, *Large cliques elude the Metropolis process*, Random Str. Algor. 3 (1992), 347–359.

[9] D. Matula, *Expose-and-merge exploration and the chromatic number of random graphs*, Combinatorica 7 (1987), 275–284.

[10] C. McDiarmid, *Colouring random graphs*, Ann. Operations Res. 1 (1984), 183–200.