Deciding k-colorability in expected polynomial time

Michael Krivelevich *

Abstract

For every fixed $k \geq 3$ we describe an algorithm for deciding k-colorability, whose expected running time in polynomial in the probability space $G(n,p)$ of random graphs as long as the edge probability $p = p(n)$ satisfies $p(n) \geq C/n$, with $C = C(k)$ being a sufficiently large constant.

1 Introduction

A graph $G = (V, E)$ is called *k-colorable* if there exists a coloring $f : V \to \{1, \ldots, k\}$ so that no edge of G is monochromatic under f. The *chromatic number* of G, denoted by $\chi(G)$, is the minimal $k \geq 1$ for which G is k-colorable.

Graph coloring ([9]) has long been one of the central notions in Graph Theory and Combinatorial Optimization. Great many diverse problems can be formulated in terms of finding a coloring of a given graph in a small number of colors or calculating, exactly or approximately, the chromatic number of the graph. Unfortunately, it turns out that these computational problems are very hard. Karp proved already in 1972 [11] that is it NP-complete to decide, for any fixed $k \geq 3$, whether a given graph G is k-colorable. Recent results show that one should not even hope to obtain an efficient algorithm which approximates the chromatic number within a non-trivial approximation ratio. Specifically, Feige and Kilian proved [6] that, unless $coRP = NP$, there is no approximation algorithm for the chromatic number whose approximation ratio over graphs on n vertices is less than $n^{1-\epsilon}$, for any fixed $\epsilon > 0$.

These hardness results, combined with the extreme importance and wide applicability of graph coloring, have been stimulating the development of algorithms, performing well on *average*, as opposed to the somewhat pessimistic worst case scenario. When discussing the average performance of a particular algorithm A, it is usually assumed that a probability distribution on the set of all inputs of A is defined. In this case, the *expected running time* of A is $\sum_G Pr[G] R_A(G)$, where the summation runs over all possible inputs of A, $Pr[G]$ is the probability of a graph G in the

*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 by a USA Israeli BSF grant and by a Bergmann Memorial Award.

Keywords: Algorithm, graph coloring, average complexity, vector chromatic number.

underlying probability space, and $R_A(G)$ is the running time of A on G. A crucial advantage of the notion of average running time is that in order for A to have a polynomial expected running time, A should be polynomial for most (in the probability sense) input graphs, while it can allow an exponential slowdown for an exponentially small fraction of the inputs.

Several papers ([15], [2], [5], [14], [7], [12], to mention just a few) discussed coloring algorithms with expected polynomial time. In some of them, the input space was composed of all k-colorable graphs with certain probability distribution defined on them, and the task was to find a k-coloring.

In this paper, we consider a different probability distribution, that corresponding to the *binomial random graph* $G(n,p)$. In this distribution, the ground space is composed of all labeled graphs on n vertices $\{1,\ldots,n\}$, where the probability assigned to a graph $G=(V,E)$ is $Pr[G]=p^{|E(G)|}(1-p)^{\binom{n}{2}-|E(G)|}$. Another way of describing the random graph $G(n,p)$ is to say that each pair of vertices $1\le i\ne j\le n$ is chosen to be an edge of G independently and with probability p. Due to this description the parameter p is usually called the *edge probability.* Note that p may by a function of the number of vertices n: $p=p(n)$. When $p=1/2$, all labeled graphs on n vertices are equiprobable in $G(n,1/2)$, and thus the task of determining the probability of a property P is essentially equivalent to that of counting the number of graphs having P. Usual asymptotic assumptions and notation apply when considering $G(n,p)$. In particular, the number of vertices n is assumed to tend to infinity. Also, a graph property P holds *almost surely*, or a.s. for brevity, in $G(n,p)$ if the probability that a graph G, drawn according to the distribution $G(n,p)$, possesses P tends to 1 as n tends to infinity. A recent monograph [8] can be used as a basic reference for the theory of random graphs.

While designing coloring algorithms that work well on average over the probability space $G(n,p)$, it is useful to keep in mind an asymptotic behavior of the chromatic number of $G(n,p)$. Bollobás [3] and Łuczak [13] showed that almost surely the chromatic number of G satisfies $\chi(G)=(1+o(1))n\log_2(1/(1-p))/\log_2 n$ for a constant p, and $\chi(G)=(1+o(1))np/(2\ln(np))$ for $C/n\le p(n)\le o(1)$, where the $o(1)$ term tends to 0 as np tends to infinity.

The algorithmic task we consider here is formulated as follows: given a fixed integer $k\ge 3$, design an algorithm for deciding k-colorability in expected polynomial time over the probability space $G(n,p)$, for various values of the edge probability $p=p(n)$. Note that if $p\ge C(k)/n$ with a large enough constant $C(k)>0$ then a.s. $G(n,p)$ is not k-colorable, and thus almost all graphs should be rejected by the algorithm. This observation however still leaves the task of distinguishing between the typical instances (i.e. those G with $\chi(G)>k$) and non-typical ones open. It is easy to see that, for a constant edge probability p, the probability that $G(n,p)$ is k-colorable is extremely small (about $e^{-\Theta(n^2)}$) and an algorithm should reject an overwhelming majority or input graphs. Thus we may hope to devise a very efficient algorithm for this case. And indeed, Wilf showed in [15] that for $p=1/2$ a simple backtracking algorithm for k-colorability has average running time bounded by an absolute constant that depends on k only. In particular, when $k=3$, a backtrack search tree has about 197 nodes on average. Another possible way of designing such an algorithm would be to search for small instances of non-k-colorable subgraphs (for example, for cliques of size $k+1$). Once such a subgraph is found, we know that the input graph is not k-colorable, and the algorithm may safely reject it. The probability that no such subgraph will be found is only $e^{-\Theta(n^2)}$, which gives us enough time to check all potential k-colorings exhaustively.

The situation becomes more complicated when the edge probability $p(n)$ tends to zero with n tending to infinity. Bender and Wilf proved [2] that in this case the backtrack algorithm has expected running time $e^{\Theta(1/p)}$, i.e., becomes exponential in n. Also, one can easily show that for every fixed $k \geq 3$, if the edge probability p satisfies $p(n) = o(n^{-2/k})$, then a.s. every subgraph of $G(n,p)$ with a bounded number of vertices is k-colorable, and thus one cannot hope to find a certificate for non-k-colorability by performing local search only.

Here we present an algorithm for deciding k-colorability in expected polynomial time in $G(n,p)$ for every fixed $k \geq 3$, as long as $p(n) \geq C/n$, where $C = C(k) > 0$ is a sufficiently large constant. Our algorithm can be immediately extended for larger values of $p(n)$. Note that if C is sufficiently large, the random graph $G(n,p)$ is not k-colorable with probability $1 - e^{-\Theta(n)}$. Therefore the algorithm still rejects most of the graphs from $G(n,p)$. In order to be able to reject an input graph, the algorithm needs some graph parameter whose value can serve as a certificate for non-k-colorability. This parameter should be computable in polynomial time. The parameter we will use in our algorithm is the so called *vector chromatic number* of a graph [10]. Besides being computable in polynomial time, the vector chromatic number turns out to be extremely robust, and the probability that its value is small is exponentially small in n. This will enable us to invest exponential time in "exceptional" graphs, i.e. those with small vector chromatic number. More details are given in the subsequent sections.

The rest of the paper is organized as follows. In the next section we provide a necessary background on the vertex chromatic number. In Section 3 we present our algorithm and analyze its properties. Section 4 is devoted to concluding remarks.

Throughout the paper, the number of vertices n is assumed to be as large as necessary. We routinely omit floor and ceiling signs for the sake of clarity of presentation. No serious attempt is made to optimize the involved constants.

2 Vector chromatic number

As we have mentioned in the introduction, the key technical notion for our algorithm is that of a vector chromatic number, introduced by Karger, Motwani and Sudan in [10]. We start this section with relevant definitions.

Suppose we are given a graph $G = (V,E)$ with vertex set $V = \{1,\ldots,n\}$. A *vector k-coloring* of G is an assignment of unit vectors $v_i \in R^n$ to the vertices of G so that for every edge $(i,j) \in E(G)$ the standard scalar product of the corresponding vectors v_i, v_j satisfies the inequality $(v_i, v_j) \leq -\frac{1}{k-1}$. The graph G is called *vector k-colorable* if such a vector k-coloring exists. Finally, the *vector chromatic number* of G, which we denote by $v\chi(G)$, is the minimal real $k \geq 1$ for which G is vector k-colorable.

Karger, Motwani and Sudan established the connection between the usual chromatic number of a graph, $\chi(G)$, and its vector chromatic number, $v\chi(G)$. Below we repeat some of their arguments and conclusions.

Proposition 2.1 *If $\chi(G) = k$, then G is vector k-colorable. Thus, $v\chi(G) \leq \chi(G)$.*

Proof. The statement will follow easily from the claim below.

Claim 2.2 *For every $k \leq n+1$, there exists a family $\{v_1, \ldots, v_k\}$ of k unit vectors is R^n satisfying $(v_i, v_j) = -\frac{1}{k-1}$ for every $1 \leq i \neq j \leq k$.*

Proof. The existence of such a family can be proven by induction on n, as in [10]. Here we present an alternative proof.

Clearly it is enough to prove the claim for the case $k = n+1$ (if $k < n+1$, find such a family in R^{k-1} and complete the found vectors by zeroes in the last $n-k+1$ coordinates to get the desired family). Define an n-by-n matrix $A = (a_{ij})$ by setting $a_{ii} = 1$ for $1 \leq i \leq n$, and $a_{ij} = -1/n$ for all $1 \leq i \neq j \leq n$. Then A is a symmetric positive definite matrix (the eigenvalues of A are $\lambda_1 = \ldots = \lambda_{n-1} = 1 + 1/n$, $\lambda_n = 1/n$). Therefore it follows from standard linear algebra results that there exists a family $\{v_1, \ldots, v_n\}$ of n vectors in R^n so that $a_{ij} = (v_i, v_j)$ for all $1 \leq i, j \leq n$. In particular, $(v_i, v_i) = a_{ii} = 1$, so all members of this family are unit vectors. Also, $(v_i, v_j) = a_{ij} = -1/n$ for all $1 \leq i \neq j \leq n$. Set now $v_{n+1} = -(v_1 + \ldots + v_n)$. Then $(v_{n+1}, v_{n+1}) = (v_1 + \ldots + v_n, v_1 + \ldots + v_n) = n \cdot 1 + 2\binom{n}{2}(-1/n) = 1$, and v_{n+1} is a unit vector as well. Also, for all $1 \leq i \leq n$, $(v_i, v_{n+1}) = (v_i, -v_1 - \ldots - v_n) = -1 + (n-1)/n = -1/n$. Hence, $\{v_1, \ldots, v_n, v_{n+1}\}$ forms the desired family. $\Box$.

Returning to the proof of the proposition, we argue as follows. Let $V = C_1 \cup \ldots \cup C_k$ be a k-coloring of G. Based on the above claim, we can find a family $\{v_1, \ldots, v_k\}$ of unit vectors in R^n so that $(v_i, v_j) = -1/(k-1)$ for all $1 \leq i \neq j \leq k$. Now, for each color class C_i, every vertex from C_i gets the vector v_i assigned to it. The obtained assignment is clearly a vector k-coloring of G. $\Box$

It is interesting to note that the vector chromatic number of a graph G is always upper bounded by the so called Lovász θ-function of its complement, $\theta(\bar{G})$, as shown in [10] (see also [1] for further discussion). However we will not use this connection in our analysis.

An important feature of the vector chromatic number, noticed by Karger et al., is stated in the following proposition.

Proposition 2.3 *If a graph G on n vertices is vector k-colorable, then a vector $(k+\epsilon)$-coloring of the graph can be constructed in time polynomial in k, n and $\log \frac{1}{\epsilon}$.*

In particular, it follows from the above proposition that the vector chromatic number of G can be approximated within any prescribed precision $\epsilon > 0$ in time polynomial in n and $\log(1/\epsilon)$.

Karger et al. combined Proposition 2.3 with clever rounding arguments to produce an algorithm for approximate coloring of k-colorable graphs, for any fixed $k \geq 3$. We will use their algorithmic result in its existential form, as described below.

Definition 2.4 *Given a graph $G = (V, E)$ on n vertices and an integer $1 \leq t \leq n$, a* semi-coloring *of G in t colors is a family $(C_1, \ldots, C_t)$, where each $C_i \subseteq V(G)$ is an independent set in G, the subsets C_i are pairwise disjoint, and $|\bigcup_{i=1}^{t} C_i| \geq \frac{n}{2}$.*

Proposition 2.5 *For any $k \geq 3$, there exist $c = c(k) > 0, n_0 = n_0(k) > 0$ so that the following holds. For any $n > n_0$ and for any graph G on n vertices and with $m > n$ edges, if $v\chi(G) \leq k$ then there exists a semi-coloring of G in t colors, where*

$$t \leq c \left(\frac{m}{n}\right)^{\frac{k-2}{k}} \ln^{1/2}\left(\frac{m}{n}\right) .$$

Thus the assumption that the vector chromatic number of G is small enables to claim the existence of many pairwise disjoint and large on average independent sets in G.

3 Algorithm

In this section we describe our algorithm for deciding k-colorability. As the reader will see immediately, the algorithm is extremely simple and in a sense just calculates the vector chromatic number of an input graph.

ALGORITHM

Input: *An integer $k \geq 3$ and a graph $G = (V, E)$ on n vertices.*

Step 1. *Calculate the vector chromatic number $v\chi(G)$ of the input graph G;*

Step 2. *If $v\chi(G) > k$, output "G is not k-colorable";*

Step 3. *Otherwise, check exhaustively all k^n potential k-colorings of G. If a proper k-coloring of G is found, output "G is k-colorable", else output "G is not k-colorable".*

The correctness of the algorithm follows immediately from the properties of the vector chromatic number, discussed in Section 2.

Proposition 3.1 *The above algorithm always outputs a correct answer.*

Proof. If the answer is output at Step 2, then $v\chi(G) > k$. Therefore, by Proposition 2.1 G is not k-colorable, and the output answer is indeed correct. Otherwise the answer is output at Step 3 after having performed the exhaustive search and is thus obviously correct. $\square$

Now we need to estimate the algorithm's expected running time over the probability space $G(n,p)$.

Theorem 3.2 *If the edge probability $p(n)$ satisfies $p = C/n$ and $C > 0$ is large enough, then the expected running time of the above algorithm is polynomial in n.*

Due to Proposition 2.3 Step 2 of the algorithm takes polynomial time. (An alert reader can remark at this point that Proposition 2.3 talks about only approximating the value of $v\chi(G)$ instead of calculating it precisely. However, this does not pose any additional difficulty as instead of comparing $v\chi(G)$ with k we could have compared it with $k+\epsilon$ for some small $\epsilon > 0$. The only price we would have to pay for it is a slight increase in the value of C. We prefer to "hide" this technicality in order to make the suggested algorithm more transparent.) Notice that we get to Step 3 only if $v\chi(G) \leq k$. At Step 3 we check exhaustively all k^n potential k-colorings of G, and checking each potential coloring costs us time polynomial in n. Therefore it takes at most $k^n poly(n)$ time to perform Step 3. The statement of the theorem is thus immediately implied by the following proposition.

Proposition 3.3 *If $C = C(k) > 0$ is large enough, and G is distributed according to $G(n,p)$ with $p = C/n$, then*

$$Pr[v\chi(G) \leq k] \leq k^{-n} .$$

Proof. The proof is based on the following technical claims about the probability space $G(n,p)$.

Claim 3.4 *If $C > 0$ is large enough then*

$$Pr[|E(G)| \leq 2n^2p] \geq 1 - o(k^{-n}) .$$

Claim 3.5 *For every fixed $c > 0$, $k \geq 3$, if $C > 0$ is large enough then the following is true in $G(n,p)$ with $p = C/n$. Let $t = c(2C)^{\frac{k-2}{k}} \ln^{1/2}(2C)$. Then*

$$Pr[G \text{ has a semi-coloring in } t \text{ colors}] = o(k^{-n}) .$$

Assuming the above two claims hold, we prove now Proposition 3.3 and thus Theorem 3.2. By Claim 3.4 we may assume that G has at most $2n^2p = 2Cn$ edges. If the vector chromatic number of such a graph is at most k, then by Proposition 2.5 G has a semi-coloring in $t = c(m/n)^{(k-2)/k)} \ln^{1/2}(m/n) \leq c(2C)^{(k-2)/k} \ln^{1/2}(2C)$ colors. However, by Claim 3.5 this happens in $G(n,p)$ with probability $o(k^{-n})$.

It remains only to prove Claims 3.4 and 3.5.

Proof of Claim 3.4.

$$\begin{aligned} Pr[|E| \geq 2n^2p] &\leq \binom{\binom{n}{2}}{2n^2p} p^{2n^2p} \leq \left(\frac{\frac{en^2}{2}}{2n^2p}\right)^{2n^2p} p^{2n^2p} \leq \left(\frac{e}{4p}\right)^{2n^2p} p^{2n^2p} \\ &= \left(\frac{e}{4}\right)^{2n^2p} = \left(\frac{e}{4}\right)^{2Cn} = o(k^{-n}) \end{aligned}$$

for large enough $C = C(k) > 0$.

Proof of Claim 3.5. If G has a semi-coloring with t colors, then there is a collection $(C_1, \ldots, C_t)$ of pairwise disjoint independent sets in G, whose cardinalities sum up to $n/2$. For a given such collection $(C_1, \ldots, C_t)$, the probability that each C_i is independent can be estimated from above as follows:

$$\begin{aligned} Pr[C_1, \ldots, C_t \text{ are independent}] &= \prod_{i=1}^{t} (1-p)^{\binom{|C_i|}{2}} \leq \exp\left\{-p \sum_{i=1}^{t} \binom{|C_i|}{2}\right\} \\ &\leq \exp\left\{-pt \binom{\frac{n}{2t}}{2}\right\} \leq \exp\left\{-\frac{pn^2}{9t}\right\} \\ &= \exp\left\{-\frac{Cn}{9c(2C)^{\frac{k-2}{k}} \ln^{1/2}(2C)}\right\} \leq \exp\left\{-\frac{C^{\frac{2}{k}} n}{\ln C}\right\} \end{aligned}$$

(we used the convexity of the function $f(x) = \binom{x}{2}$ in the second inequality above). As the total number of such collections is at most $t^n < C^n$, the union bound shows that the probability in question is at most $C^n \exp\{-C^{2/k} n / \ln C\} = o(k^{-n})$ for large enough $C = C(c, k)$. $\square$

4 Concluding remarks

For every fixed $k \geq 3$ we have presented an algorithm for deciding k-colorability. The algorithm always produces a correct answer and its expected running time is polynomial over the probability space $G(n,p)$, where $p(n) \geq C(k)/n$.

It would be very interesting to devise an algorithm (or a family of algorithms) for deciding k-colorability, whose expected running time is polynomial for *any* value of the edge probability $p(n)$. Note that if $p(n) \leq c/n$ with $c = c(k) > 0$ sufficiently small, then $G(n,p)$ is k-colorable almost surely, and the algorithm should thus accept most of the input graphs. The problem becomes especially challenging when $p(n)$ is close to the threshold probability for non-k-colorability. This is due to the widespread belief that typical instances at the uncolorability threshold are computationally hard (see, e.g. [4] for a relevant discussion). We have come close to this critical point by presenting an algorithm which works well on average for edge probabilities of the same order as the threshold probability for non-k-colorability.

Another possible direction of future research is to develop algorithms for deciding k-colorability on graphs on n vertices, where the parameter k is a function on n: $k = k(n)$. Unfortunately, the method of this paper, based on the notion of a vector chromatic number, seems to be no longer applicable in this case as Proposition 2.5 degenerates to a trivial statement in this case. A different approach is needed to tackle this problem.

Finally, it seems to be interesting to understand the asymptotic behavior of the vector chromatic number $v\chi(G)$ in the probability space $G(n,p)$, for various values of $p(n)$.

References

[1] N. Alon and N. Kahale, *Approximating the independence number via the θ-function*, Math. Programming 80 (1998), 253–264

[2] E. Bender and H. Wilf, *A theoretical analysis of backtracking in the graph coloring problem*, J. Algorithms 6 (1985), 275–282.

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

[4] J. Culberson and I. Gent, *Well out of reach: why hard problems are hard*, Technical report APES-13-1999, APES Research Group, 1999. Available from: http://apes.cs.strath.ac.uk/apesreports.html.

[5] M. Dyer and A. Frieze, *The solution of some random NP-hard problems in polynomial expected time*, J. Algorithms 10 (1989), 451–489.

[6] U. Feige and J. Kilian, *Zero knowledge and the chromatic number*, Proc. 11^{th} IEEE Conf. Comput. Complexity, IEEE (1996), 278–287.

[7] M. Fürer, C. R. Subramanian and C. E. Veni Madhavan, *Coloring random graphs in polynomial expected time*, Algorithms and Comput. (Hong Kong 1993), Lecture Notes Comp. Sci. 762, Springer, Berlin, 1993, 31–37.

[8] S. Janson, T. Łuczak and A. Ruciński, **Random graphs**, Wiley, New York, 2000.

[9] T. Jensen and B. Toft, **Graph coloring problems**, Wiley, New York, 1995.

[10] D. Karger and R. Motwani and M. Sudan, *Approximate Graph coloring by semidefinite programming*, Journal of the ACM, 45 (1998), 246–265.

[11] R. Karp, Reducibility among combinatorial problems, in: *Complexity of computer computations* (E. Miller and J. W. Thatcher, eds.) Plenum Press, New York, 1972, 85–103.

[12] M. Krivelevich and V. H. Vu, *Approximating the independence number and the chromatic number in expected polynomial time*, Proc. 27^{th} Int. Colloq. on Automata, Languages and Programming (ICALP'2000), Lecture Notes in Computer Science 1853, Springer, Berlin, 13–24.

[13] T. Łuczak, *The chromatic number of random graphs*, Combinatorica 11 (1991), 45–54.

[14] H. J. Prömel and A. Steger, *Coloring clique-free graphs in polynomial expected time*, Random Str. Algor. 3 (1992), 275–402.

[15] H. Wilf, *Backtrack: a $O(1)$ expected time algorithm for the graph coloring problem*, Inform. Proc. Letters 18 (1984), 119–121.