Available online at www.sciencedirect.com
SCIENCE DIRECT®
Discrete Applied Mathematics 153 (2005) 25–47
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

The resolution complexity of random graph *k*-colorability

Paul Beame[a,1], Joseph Culberson[b,2], David Mitchell[c,3], Cristopher Moore[d,4]

[a] *Computer Science and Engineering, University of Washington, Box 352350, Seattle, WA 98195, USA*
[b] *Department of Computer Science, University of Alberta, Edmonton, Alta, Canada*
[c] *Department of Computer Science, Simon Fraser University, Burnaby, BC, Canada*
[d] *Computer Science Department, University of New Mexico, Albuquerque, NM 87131, USA*

Received 19 December 2003; received in revised form 11 August 2004; accepted 6 May 2005
Available online 24 August 2005

Abstract

We consider the resolution proof complexity of propositional formulas which encode random instances of graph k-colorability. We obtain a tradeoff between the graph density and the resolution proof complexity. For random graphs with linearly many edges we obtain linear-exponential lower bounds on the size of resolution refutations. For random graphs with n vertices and any $\varepsilon > 0$, we obtain a lower-bound tradeoff between graph density and refutation size that implies subexponential lower bounds of the form 2^{n^δ} for some $\delta > 0$ for non-k-colorability proofs of graphs with n vertices and $O(n^{3/2-1/k-\varepsilon})$ edges. We obtain sharper lower bounds for Davis–Putnam–DPLL proofs and for proofs in a system considered by McDiarmid.

These proof complexity bounds imply that many natural algorithms for k-coloring or k-colorability have essentially the same exponential tradeoff lower bounds on their running times. We also show

E-mail addresses: beame@cs.washington.edu (P. Beame), joe@cs.ualberta.ca (J. Culberson), mitchell@cs.sfu.ca (D. Mitchell), moore@cs.unm.edu (C. Moore).

[1] Research supported by NSF Grants CCR-9800124 and CCR-0098066.
[2] Research supported by NSERC Grant OGP8053.
[3] Research supported by NSERC Grant RGPIN238987.
[4] Research supported by NSF Grant PHY-0200909.

doi:10.1016/j.dam.2005.05.004

that very simple algorithms for k-colorability have upper bounds on their running times that are qualitatively similar to the lower bounds as a function of the graph density.

Keywords: Graph coloring; Proof complexity; Resolution proofs; Random graphs; Chromatic number

1. Introduction

The problem of coloring graphs has a long history. The decision problem, k-colorability, being an important NP-complete problem, has generated significant interest for random graphs as well. Since the transcript of any complete algorithm for k-coloring on input G also provides a proof of non-k-colorability in the case that G is not k-colorable, the study of the proof complexity of k-colorability also yields bounds on the running time of such algorithms. We consider here the question of the proof complexity of non-k-colorability for random graphs.

With sufficiently many edges, a random graph is not k-colorable almost certainly. McDiarmid [35] studied the complexity of deciding non-k-colorability in random graphs using a particular natural class of proof procedures that emulates a variety of coloring algorithms. Here, we consider natural encodings of the k-colorability of random graphs as CNF formulas and examine the resolution complexity of these formulas.

It is easy to see how many natural algorithms for k-coloring can be simulated as resolution proof procedures. Indeed, many natural algorithms, including the most efficient current procedures can naturally be phrased in terms of resolution procedures in which the graph of resolution inferences is a tree. This is a subclass of resolution proofs known collectively as the Davis–Putnam or DPLL procedure. (Resolution itself comes with no underlying search procedure for proofs, whereas DPLL procedures include a search strategy as well.) In our analysis, we build on the ideas that have been used to analyze the resolution proof complexity of random k-CNF formulas together with results about the colorability properties of random graphs.

For any graph G, we denote by $\chi(G,k)$ the formula for the "natural" encoding of the statement "G is k-colorable", as a propositional formula. (That is, using the usual notation of $\chi(G)$ for chromatic number, $\chi(G,k)$ expresses "$\chi(G)\leqslant k$.") For any CNF formula ϕ, we denote by $Res(\phi)$ the size of the shortest resolution refutation of ϕ. By convention, we let $Res(\phi)=\infty$ if ϕ is satisfiable.

The general scheme of our lower-bounds proof is similar to that of the proofs of resolution lower bounds for random k-SAT [17,10,9,13]. We make particular use of the recent relationship between clause width and proof size for resolution proofs shown by Ben-Sasson and Wigderson [13]. The argument is as follows:

1. Almost surely, every small subgraph of G has low degree, and thus is k-colorable. From this, it can be shown that in any refutation of $\chi(G,k)$ there is a "complex clause" that cannot be derived from $\chi(H,k)$, for any H which is a very small subgraph of G.
2. Almost surely, every small subgraph of G has many vertices of degree less than k. From this it can be shown that any "complex clause" contains many literals. The theorem of Ben-Sasson and Wigderson relating minimum refutation width to minimum refutation size then implies that any refutation of $\chi(G,k)$ must be long.

The class of procedures considered by McDiarmid produces tree-like proofs but these are not DPLL procedures. Nonetheless, it is possible to show using arguments similar to those of [18,13] that if these procedures produce short proofs then they have efficient simulations by resolution proofs of small width. Thus, all our lower bounds apply to these procedures as well.

2. Preliminaries

For a set of boolean variables $V=\{v_1,\ldots,v_n\}$, a literal is any variable $x\in V$, or its negation $\neg x$, a clause is a set of literals, and a formula is a set of clauses. The interpretation of a formula is as a conjunctive normal form (CNF) formula of propositional calculus, that is, a conjunction of disjunctions. A truth assignment for V is a function $\tau: V\mapsto\{T,F\}$. Assignment τ satisfies a clause C iff $\tau(l)=T$ for at least one literal $l\in C$, and τ satisfies a set of clauses ϕ, written $\tau\vDash\phi$, iff it satisfies every clause in ϕ. A formula is satisfiable iff there is an assignment to its variables which satisfies it. For any clause or formula A, $vars(A)$ denotes the set of variables which appear (negated or otherwise) in A. The width of a clause C denoted $w(C)$ is the number of variables occurring in C. The width of a formula is the maximum of the widths of its clauses. We will consider only clauses that are not tautologies, thus $w(C)\equiv|vars(C)|$.

2.1. *k*-colorability formulas

For any graph $G=(V,E)$, we view each edge $e\in E$ and a size two subset of the vertex set V and let $n\stackrel{\text{def}}{=}|V|$. A *k-coloring* of G is a function *col* which maps each vertex onto an integer in $[k]=\{1,\ldots,k\}$. A *proper coloring* is a coloring for which $(u,v)\in E\Rightarrow col(u)\neq col(v)$, and we say that G is *k-colorable* iff there is a proper k-coloring of G. We define $\phi=\chi(G,k)$ to be the formula on kn variables with n positive clauses of width k, and $\binom{k}{2}n+k|E|$ negative clauses of width 2 as follows:

1. For each $v\in V$, $vars(\phi)$ has k propositional variables, $\{x_{v,1},\ldots,x_{v,k}\}$, and ϕ has one positive clause of width k
$$\left(\bigvee_{j\in[k]} x_{v,j}\right)\in\phi.$$
2. For each $v\in V$, ϕ has $\binom{k}{2}$ negative clauses of width 2, $\forall i<j\in[k]$ $(\neg x_{v,i}\vee\neg x_{v,j})\in\phi$.
3. For each edge $(u,v)\in E$, ϕ has k negative clauses of width 2,
$$\forall l\in[k]\quad(\neg x_{u,l}\vee\neg x_{v,l})\in\phi.$$

Clearly, $\chi(G,k)$ is satisfiable iff G is k-colorable. Corresponding to each coloring of G is a unique truth assignment for $\chi(G,k)$. (The reverse is not the case, since some assignments give a vertex no color or multiple colors.) We sometimes fail to distinguish between colorings and assignments. For example, we may say that a coloring makes a clause true, with the obvious meaning.

2.2. Random graphs

In the study of random graphs there are three natural models one could consider. The most commonly considered models are $\mathcal{G}(n, p)$, where each of the $\binom{n}{2}$ edges is chosen independently with probability p, and $\mathcal{G}_{n,m}$, where a set of precisely m distinct edges is chosen uniformly at random. We find it most convenient to express our lower-bound proofs in terms of a third distribution, $\hat{\mathcal{G}}_{n,m}$, where m edges are chosen independently with replacement (and duplicates are ignored), although we use the usual $\mathcal{G}(n, p)$ distribution for our upper bounds. As shown, for example, in [8] if $p = m/\binom{n}{2}$ then, when considering properties that are monotone (or anti-monotone), the almost certain properties under all three distributions are the same up to a change from m to $m \pm o(m)$. Our results will therefore apply to $\mathcal{G}(n, p)$ and $\mathcal{G}_{n,m}$ as well.

We will consider the dependence of our results on a graph density parameter $\Delta = \Delta(n) = m/n$, the ratio of edges to vertices.

2.3. Resolution complexity

Resolution is a rule of inference for clauses, which allows one to derive the clause $C \cup D$ from two clauses $C \cup l$ and $D \cup \neg l$. A resolution derivation of a clause C from a set of clauses ϕ is a sequence of clauses, $\pi = C_0, \ldots, C_m = C$, where each clause C_i is either an element of ϕ or is derived by the resolution rule from two clauses C_j, C_k occurring in π, for $j, k < i$. The derivation is of size m. A resolution derivation of the empty clause (denoted Λ) from ϕ is called a *refutation* of ϕ. The fundamental property of resolution is that there is a refutation of a set ϕ of clauses if and only if ϕ is unsatisfiable. The resolution complexity of ϕ, here, denoted $Res(\phi)$, is the size of the shortest refutation of ϕ.

A related method for CNF formula satisfiability is the Davis–Putnam–DPLL procedure [23]. Such a procedure can be described recursively as follows: first, check whether F is trivially satisfiable (has no clauses) or is trivially unsatisfiable (contains an empty clause) and if so stop. Otherwise, select a literal l and search for a satisfying assignment for the formula $F\lceil_{l=1}$ obtained by setting l to true in F (eliminating all clauses containing l and removing $\neg l$ from those clauses that contain it). Otherwise, repeat the search with the formula $F\lceil_{\neg l=1}$. If neither of these searches finds a satisfying assignment then F is not satisfiable. DPLL algorithms will typically select a literal appearing in a clause of length 1, called a *unit clause*, if one exists since that literal must be set to true to satisfy the formula. In the case that there are no unit clauses there are many heuristics, called splitting rules, for the selection of the next literal l that have been used in the literature. It is not hard to show that any DPLL algorithm actually produces a resolution refutation and moreover that the form of this refutation is *tree-like*, in that the graph of inferences forms a binary tree. Let $DPLL(\phi)$ denote the size of the shortest DPLL refutation of ϕ. The following key relationship between the proof size and resolution width was shown by Ben-Sasson and Wigderson.

Proposition 1 (*Ben-Sasson and Wigderson [13]*). *Let $w^*(F)$ be the minimum over all resolution refutations Π of F of the largest width of a clause in Π. Then $DPLL(F) \geqslant 2^{w^*(F)-w(F)}$ and $Res(F) \geqslant 2^{c(w^*(F)-w(F))^2/n}$ for $c = 1/(9\ln 2)$.*

The class of algorithms considered by McDiarmid is similar in spirit to DPLL algorithms, except that instead of trying assignments to a particular boolean variable (akin to choosing the specific color for a vertex), one chooses whether or not two non-adjacent vertices will be colored the same or differently. This is represented by graph operations that either identify the non-adjacent vertices or add an edge between them.

More formally, McDiarmid's proof system for non-k-colorability has as its objects graphs H derived from the input graph G. The axioms of the proof system are the k'-cliques for any $k' > k$. Given a graph H with two non-adjacent vertices $u, v \in H$, then H follows from $H \cup \{(u, v)\}$ and H_{uv}, where H_{uv} is the graph obtained by identifying u and v and naming the resulting vertex v. H can also follow from any H' such that H' is a subgraph of H. McDiarmid only considered proofs whose inference graph forms a tree.

Lemma 1. *Let $k \geqslant 2$. If the non-k-colorability of a graph G can be proven by a size S tree-like proof in McDiarmid's proof system then there is a resolution refutation of $\chi(G, k)$ of width at most $k(k+1) + 2k\log_2 S$.*

Proof. The proof follows a general argument due to Russell Impagliazzo (personal communication) that extends the width-size relationship for tree-like resolution in [13] to decisions involving bounded numbers of variables. Write $F \vdash_w F'$ if and only if there is a resolution derivation of F' from F, each of whose clauses has width at most w.

The proof is by induction on S. We begin with the base case of $S = 1$. Clearly for $k' > k$, by considering the clauses of $\chi(K_{k'}, k)$ that only involve the variables for the first $k+1$ vertices of $K_{k'}$, $\chi(K_{k'}, k) \vdash_{k(k+1)} \Lambda$.

Now, consider the last inference of the tree-like McDiarmid proof and suppose that the claim is true for all strictly smaller McDiarmid proofs. If that last inference derived H from a subgraph H' then we note that $\chi(H, k) \vdash_k \chi(H', k)$ since $\chi(H', k)$ is a subformula of $\chi(H, k)$ and each of its clauses has size at most k; therefore, the size bounds follow by the inductive hypothesis for H'. Alternatively, the last inference derived H from $H_0 = H \cup \{(u, v)\}$ and $H_1 = H_{uv}$, where $(u, v) \notin H$. One of the proofs that these graphs are not k-colorable has size at most $S/2$ and the other has size at most S.

Let $T_0 = \text{Unequal}(u, v) = \bigwedge_{l \in [k]} (\neg x_{u,l} \vee \neg x_{v,l})$, $T_1 = \text{Equal}(u, v) = \bigwedge_{l \in [k]} ((\neg x_{u,l} \vee x_{v,l}) \wedge (\neg x_{v,l} \vee x_{u,l}))$, and $\text{Colored}(u) = (x_{u,1} \vee \cdots \vee x_{u,k})$. Observe that $(\text{Colored}(u) \wedge \text{Colored}(v)) \rightarrow (T_0 \vee T_1)$ is a tautology involving $2k$ variables and that for $u, v \in H$, $\text{Colored}(u)$ and $\text{Colored}(v)$ are clauses of $\chi(H, k)$.

We show that for $b = 0, 1$, $\chi(H, k) \wedge T_b \vdash_k \chi(H_b, k)$. For $b = 0$, note that $\chi(H_0, k) = \chi(H \cup \{(u, v)\}, k) = \chi(H, k) \wedge \text{Unequal}(u, v) = \chi(H, k) \wedge T_0$ and it follows trivially. For $b = 1$, observe that the only clauses of $\chi(H_{uv}, k)$ that are not already in $\chi(H, k)$ are those of the form $(\neg x_{v,l} \vee \neg x_{w,l})$, where $(\neg x_{u,l} \vee \neg x_{w,l})$ is in $\chi(H, k)$. Each such clause follows easily from $\chi(H, k) \wedge \text{Equal}(u, v)$ by resolving $(\neg x_{u,l} \vee \neg x_{w,l})$ with $(\neg x_{v,l} \vee x_{u,l})$. Therefore, $\chi(H, k) \wedge T_1 = \chi(H, k) \wedge \text{Equal}(u, v) \vdash_k \chi(H_{uv}, k) = \chi(H_1, k)$.

By the inductive hypothesis there are resolution refutations of $\chi(H_0, k)$, $\chi(H_1, k)$ such that one has width at most $w = k(k+1) + 2k \log_2 S$ and the other one has width at most $w - 2k$. Assume that the refutation of H_0 is narrower (wlog). Since $\chi(H, k) \wedge T_b \vdash_k \chi(H_b, k)$ for $b = 0, 1$ and $w \geqslant k(k+1) \geqslant 3k$, we have $\chi(H, k) \wedge T_0 \vdash_{w-2k} \Lambda$ and $\chi(H, k) \wedge T_1 \vdash_w \Lambda$.

We now come to the key point in the argument. We can convert the resolution refutation witnessing $\chi(H, k) \wedge T_0 \vdash_{w-2k} \Lambda$ into a refutation witnessing $\chi(H, k) \vdash_w T_1$. For every assignment σ to the $2k$ variables in T_0 that satisfies T_0 apply σ as a restriction to the proof. The result is a derivation $\chi(H, k)\lceil_\sigma \vdash_{w-2k} \Lambda$. By [13, Lemma 3.1] one can add back literals of $\bar{\sigma}$ as needed and apply weakening to derive $\chi(H, k) \vdash_w \bar{\sigma}$. Doing this for all such choices of σ we obtain $\chi(H, k) \vdash_w \bar{T_0}$, where $\bar{T_0}$ is the canonical CNF formula for the truth table of $\neg T_0$. Now using the clauses Colored(u) and Colored(v) from $\chi(H, k)$ together with $\bar{T_0}$ we derive T_1 which is logically implied. This last derivation requires width only $2k$. Finally, we apply the proof witnessing $\chi(H, k) \wedge T_1 \vdash_w \Lambda$. The overall width is at most w as required. □

2.4. *k-colorability of random graphs*

As proved by Achlioptas and Friedgut [1], for every integer $k \geqslant 2$, there is a function $c_k(n)$ bounded by a constant such that for $\Delta > c_k(n)$ for $G \sim \mathcal{G}(n, 2\Delta/(n-1))$, the probability G is k-colorable goes to 0 and for $\Delta < c_k(n)$, this probability goes to 1. Let $c_k^+ = \limsup_{n\to\infty} c_k(n)$ and $c_k^- = \liminf_{n\to\infty} c_k(n)$. By a result of Łuczak [34,29], both c_k^+ and c_k^- are $k \ln k + O(k \ln\ln k)$; Achlioptas and Naor [5] have recently shown even tighter results that at every density the chromatic number almost certainly takes on one of at most two values. Further, for $k = 3$ by results of Achlioptas et al. [2–4], $c_3^+ < 2.522$ and $c_3^- > 2.01$.

3. Proof of lower bounds

3.1. *Subgraph boundary size, expansion, width, and length*

We define the *k-boundary* of a graph G, denoted $\beta_k(G)$, to be the set of vertices in G of degree between 1 and $k-1$.

For a subgraph $H < G$, let $E_k(H)$ denote the conjunction of the edge (negative) clauses of $\chi(G, k)$ corresponding to the edges of H. We say that subgraph H *implies* a clause C if and only if on every truth assignment α corresponding to a total (but not necessarily proper) coloring of G the formula $E_k(H) \to C$ evaluates to true.

Lemma 2. *Let C be a clause in the variables of $\chi(G, k)$. If $H < G$ is a minimal induced subgraph of G that implies C then*

- *H has no isolated vertices*; *and*
- *$w(C) \geqslant |\beta_k(H)|$.*

Proof. First observe that if H has an isolated vertex u then $E_k(H) = E_k(H - u)$, and thus $H - u$ also implies C contradicting the assumption that H was minimal.

Now, consider a vertex v in H of degree between 1 and $k-1$. We derive the lower bound on the size of C by showing that there is some variable $x_{v,i}$ that appears in C. By the minimality of H there is some truth assignment α corresponding to a total coloring of G such that $E_k(H-v) \to C$ is false at α; i.e. α satisfies $E_k(H-v)$ but not C. Since α satisfies $E_k(H-v)$, it is proper with respect to all the edges of $H-v$. Since the degree of v is at most $k-1$, we can extend this proper coloring on $H-v$ to one that is proper on all of H by changing the color of vertex v to get a new assignment α'. Now $E_k(H)$ is satisfied by α' and since H implies C, C is satisfied by α'.

Therefore, $C(\alpha) \neq C(\alpha')$ and since α and α' differ only on the assignments to two variables, $x_{v,i}$ and $x_{v,i'}$, where $i \neq i'$ are the old and new colors of v, C must contain one of them. □

Definition. Given a clause C over the variables from $\chi(G,k)$ denote $\mu_{G,k}(C)$ to be the minimal number of vertices in an induced subgraph $H < G$ that implies C. If no such subgraph exists, let $\mu_{G,k}(C) = \infty$.

Clearly $\mu_{G,k}$ has two key properties:

Lemma 3. *Let G be a graph and $k \geqslant 2$ be an integer.*

(a) *$\mu_{G,k}(\Lambda)$ is the number of vertices in the smallest k-uncolorable subgraph of G.*
(b) *If D is a resolvent of B and C then $\mu_{G,k}(D) \leqslant \mu_{G,k}(B) + \mu_{G,k}(C)$.*

Definition. Let G be a graph. Let $s+1$ be the minimum number of vertices in a subgraph of G that is not k-colorable; if G is k-colorable then let $s=\infty$. The *subcritical k-expansion*, $e_k(G)$, of a graph G is defined to be the maximum over all t, $2 \leqslant t \leqslant s$, of the minimum k-boundary size of any induced subgraph H of G that has no isolated vertices and has between $t/2$ and t vertices.

Lemma 4. *For $k \geqslant 3$, any resolution refutation of $\chi(G,k)$ must contain a clause of width at least $e_k(G)$.*

Proof. Let $s \geqslant k \geqslant 3$ and $t \leqslant s$ be chosen as in the definition of $e_k(G)$. Let π be a resolution refutation of $\chi(G,k)$. By Lemma 3(a), $\mu_{G,k}(\Lambda) = s+1$. Further, any clause C of $\chi(G,k)$ has $\mu_{G,k}(C) \leqslant 2$. Therefore, there is a clause D in π such that $\mu_{G,k}(D) > t \geqslant 2$ and no ancestor of D has $\mu_{G,k}$ greater than t. Since $\mu_{G,k}(D) > 2$, there must be two parent clauses B and C in π such that D is the resolvent of B and C. By Lemma 3(b), at least one of these clauses, say B, must have $\mu_{G,k}$ between $t/2$ and t. If $H < G$ witnesses the value of $\mu_{G,k}(B)$ then by Lemma 2, H has no isolated vertices and $w(B) \geqslant |\beta_k(H)|$. Thus, by definition of $e_k(G)$, $w(B) \geqslant e_k(G)$ as required. □

Corollary 1. *If G is a graph and $k \geqslant 3$ is an integer then for $c = 1/(9 \ln 2)$, $Res(\chi(G,k)) \geqslant 2^{c(e_k(G)-k)^2/n}$ and $DPLL(\chi(G,k)) \geqslant 2^{e_k(G)-k}$.*

Proof. Clearly, $w(\chi(G,k)) = k$ and Lemma 4 implies that $w^*(\chi(G,k)) \geqslant e_k(G)$. Applying Proposition 1 yields the claimed results. □

We obtain a similar result for tree-like proofs in McDiarmid's proof system using Lemma 1.

Corollary 2. *If G is a graph and $k \geqslant 3$ is an integer then any tree-like proof of non-k-colorability of G in McDiarmid's proof system requires size at least* $2^{(e_k(G)-k(k+1))/2k}$.

3.2. Lower bounding subcritical k-expansion

We now prove lower bounds on $e_k(G)$ for most $G \sim \hat{\mathscr{G}}_{n,m}$. We show these bounds by first showing that such a G is almost certainly locally sparse.

We say that a graph G is (r, q)-*dense* if some subset of r vertices of G contains at least q edges of G.

Lemma 5. *Let $G \sim \hat{\mathscr{G}}_{n,m}$. For $r, q \geqslant 1$,*

$$\Pr[G \textit{ is } (r,q)\textit{-dense}] \leqslant \left(\frac{ne}{r}\right)^r \left(\frac{emr^2}{qn^2}\right)^q.$$

Proof. Let R be a set of vertices with $|R| = r$. Let $p = \binom{r}{2} / \binom{n}{2} \leqslant (r/n)^2$ denote the probability that a randomly chosen edge on n vertices is contained in R. For $G \sim \hat{\mathscr{G}}_{n,m}$, the number of edges of G contained in R has the binomial distribution, $Bin(m, p)$. The probability that at least q edges of G are contained in R is bounded above by

$$\Pr[Bin(m, p) \geqslant q] \leqslant \binom{m}{q} p^q \leqslant \left(\frac{emr^2}{qn^2}\right)^q. \tag{1}$$

Summing this over the $\binom{n}{r} \leqslant (en/r)^r$ r-subsets of the set of vertices of G we obtain

$$\Pr[G \text{ is } (r,q)\text{-dense}] \leqslant \left(\frac{ne}{r}\right)^r \left(\frac{emr^2}{qn^2}\right)^q. \qquad \square$$

Lemma 6. *For each integer $k \geqslant 3$ there is a constant C_k such that the following holds. Let m, n be integers with $m \leqslant \Delta n$ and $\Delta \geqslant 1$. If $s = C_k n/\Delta^{k/(k-2)}$ then the probability that $G \sim \hat{\mathscr{G}}_{n,m}$ contains a subgraph of size at most s that is not k-colorable is* $o(1)$ *in s.*

Proof. The probability that G contains a k-uncolorable subgraph of size at most s is the probability that there is some *minimally* k-uncolorable graph $H < G$ with $r \leqslant s$ vertices. Observe that such an H must have $r \geqslant k+1$ and have minimum degree at least k since a vertex of degree at most $k-1$ can always be colored by one of the k colors so that none of its incident edges is monochromatic. In particular, this implies that H must have average degree at least k, and thus contain at least $kr/2$ edges.

Thus, the probability that G contains such a subgraph H is at most

$$\sum_{r=k+1}^{s} \Pr[G \text{ is } (r, kr/2)\text{-dense}].$$

By Lemma 5, we have Pr[G is $(r, kr/2)$-dense] $\leqslant D(r)$ where

$$\begin{aligned} D(r) &= \left(\frac{ne}{r}\right)^r \left(\frac{2emr^2}{krn^2}\right)^{kr/2} \\ &= (ne(2em/kn^2)^{k/2} r^{(k-2)/2})^r \\ &= (Q(k,m,n) r^{(k-2)/2})^r \end{aligned}$$

for $Q(k,m,n) = ne(2em/kn^2)^{k/2}$. Now

$$\begin{aligned} \frac{D(r+1)}{D(r)} &= \frac{(Q(k,m,n)(r+1)^{(k-2)/2})^{r+1}}{(Q(k,m,n)r^{(k-2)/2})^r} \\ &= Q(k,m,n)(r+1)^{(k-2)/2}\left(\frac{r+1}{r}\right)^{r(k-2)/2} \\ &\leqslant Q(k,m,n)(e(r+1))^{(k-2)/2} \\ &= ne(2em/kn^2)^{k/2}(e(r+1))^{(k-2)/2} \\ &= (2e^2 m/kn)^{k/2}((r+1)/n)^{(k-2)/2} \\ &\leqslant (2e^2 \Delta/k)^{k/2}((r+1)/n)^{(k-2)/2} \\ &\leqslant 1/2 \end{aligned}$$

for $1 \leqslant r \leqslant C_k n/\Delta^{k/(k-2)}$, where $C_k > 0$ depends only on k. Let $s = C_k n/\Delta^{k/(k-2)}$. Therefore, the probability that G contains such an k-uncolorable subgraph is a geometric series in r and is at most twice its largest term which is less than

$$D(1) = ne(2em/kn^2)^{k/2} \leqslant e(2e\Delta/k)^{k/2}/n^{(k-2)/2} = c_k(1/s)^{(k-2)/2}$$

for some constant c_k. Thus, it is o(1) in s as required. □

Lemma 7. *For each integer* $k \geqslant 3$ *and* ε *with* $1 - 1/(k-1) > \varepsilon > 0$, *there is a constant* $c_{\varepsilon,k} > 0$ *such that the following holds. Let* m, n *be integers with* $m \leqslant \Delta n$ *and* $\Delta \geqslant 1$. *If* $t \leqslant c_{\varepsilon,k} n/\Delta^{(k-(k-1)\varepsilon)/(k-(k-1)\varepsilon-2)}$ *then the probability that* $G \sim \hat{\mathscr{G}}_{n,m}$ *contains a subgraph on* r *vertices,* $t/2 < r \leqslant t$, *that has no isolated vertices and at most* εr *vertices of degree* $< k$ *is* o(1) *in* t.

Proof. Fix $k \geqslant 3$, $\varepsilon > 0$ and m, n with $m \leqslant \Delta n$ and $G \sim \hat{\mathscr{G}}_{n,m}$. If there is a subgraph $H < G$ on r vertices that has no isolated vertices and at most εr vertices of degree $< k$, then H has at least $r - \varepsilon r$ vertices of degree at least k and the remaining vertices of degree at least 1. Therefore, H contains at least $[k(r-\varepsilon r)+\varepsilon r]/2 = r(k-(k-1)\varepsilon)/2$ edges. Thus, by Lemma 5 the probability that such an H exists with $t/2 \leqslant r \leqslant t$ is at most

$$\sum_{r=t/2}^{t} \Pr[G \text{ is } (r, r(k-(k-1)\varepsilon)/2)\text{-dense}]$$

$$\leqslant \sum_{r=t/2}^{t} \left(\frac{ne}{r}\right)^r \left(\frac{2emr^2}{r(k-(k-1)\varepsilon)n^2}\right)^{r(k-(k-1)\varepsilon)/2}$$

$$< \sum_{r=t/2}^{t} \left(\left(\frac{2e^2m}{(k-(k-1)\varepsilon)n}\right)^{(k-(k-1)\varepsilon)/2} \left(\frac{r}{n}\right)^{(k-(k-1)\varepsilon-2)/2}\right)^r$$

$$\leqslant \sum_{r=t/2}^{t} \left(\left(\frac{2e^2\Delta}{(k-(k-1)\varepsilon)}\right)^{(k-(k-1)\varepsilon)/2} \left(\frac{r}{n}\right)^{(k-(k-1)\varepsilon-2)/2}\right)^r.$$

For $t/2 \leqslant r \leqslant t$, and for some constant $c_{\varepsilon,k} > 0$, if $t \leqslant c_{\varepsilon,k} n/\Delta^{(k-(k-1)\varepsilon)/(k-(k-1)\varepsilon-2)}$, each term in the sum is at most 2^{-r} and thus the sum is less than $2^{1-t/2}$ which is $o(1)$ in t. □

Corollary 3. *For each integer $k \geqslant 3$ and ε with$1-1/(k-1) > \varepsilon > 0$, there is a constant $c'_{\varepsilon,k} > 0$ such that the following holds. Let m, n be integers with $m \leqslant \Delta n$ and $\Delta \geqslant 1$. Let $W = n/\Delta^{(k-(k-1)\varepsilon)/(k-(k-1)\varepsilon-2)}$. The probability that $G \sim \hat{\mathscr{G}}_{n,m}$ has $e_k(G) < c'_{\varepsilon,k} W$ is $o(1)$ in W.*

Proof. Let $c_{\varepsilon,k} > 0$ be the constant from Lemma 7 and C_k be the constant from Lemma 6. Let $t = \min(C_k, c_{\varepsilon,k})W$. By Lemma 6, if $G \sim \hat{\mathscr{G}}_{n,m}$ and $s = C_k n/\Delta^{k/(k-2)} \geqslant t$ then the probability that a subgraph $H < G$ of size at most s is not k-colorable is $o(1)$ in s and thus $o(1)$ in W, since s is $\Omega(1)$ in W (observe that $s/W = C_k \Delta^{2/(k-(k-1)\varepsilon-2)-2/(k-2)}$, $\Delta \geqslant 1$, and $2/(k-(k-1)\varepsilon-2) \geqslant 2/(k-2)$).

Also, by Lemma 7 the probability that $G \sim \hat{\mathscr{G}}_{n,m}$ contains a subgraph on r vertices, $t/2 < r \leqslant t$, that has no isolated vertices and at most εr vertices of degree $< k$ is $o(1)$ in t, and thus $o(1)$ in W. Thus, every induced subgraph H on r vertices with no isolated vertices and $t/2 < r \leqslant t \leqslant s$ has k-boundary of size at least $\varepsilon r \geqslant \varepsilon t/2 = \varepsilon \min(C_k, c_{\varepsilon,k})W/2$. Letting $c'_{\varepsilon,k} = \varepsilon \min(C_k, c_{\varepsilon,k})/2$ yields the lower bound on $e_k(G)$. □

3.3. Lower-bound theorems

Theorem 1. *For each integer $k \geqslant 3$ and $\varepsilon > 0$ there are constants $C_{\varepsilon,k}, C'_{\varepsilon,k} > 0$ such that if $\Delta \geqslant 1$, m, n are integers with $m \leqslant \Delta n$ and $G \sim \hat{\mathscr{G}}_{n,m}$, then with probability $1-o(1)$ in n, $Res(\chi(G,k)) \geqslant \exp(C_{\varepsilon,k} n/\Delta^{2+4/(k-2)+\varepsilon})$ and $DPLL(\chi(G,k)) \geqslant \exp(C'_{\varepsilon,k} n/\Delta^{1+2/(k-2)+\varepsilon})$.*

Proof. Let $\varepsilon' = \varepsilon(k-2)^2/[(k-1)(4+\varepsilon(k-2))]$. Clearly, $0 < \varepsilon' < (k-2)/(k-1) = 1-1/(k-1)$. By Corollary 1 the resolution complexity of $\chi(G,k)$ is at least $2^{(e_k(G)-k)^2/n}$. By Corollary

3, there is a $c'_{\varepsilon',k}$ such that with probability $1 - o(1)$ in $W = n/\Delta^{(k-(k-1)\varepsilon')/(k-(k-1)\varepsilon'-2)}$, $e_k(G) \geqslant c'_{\varepsilon',k} W$.Therefore,

$$\begin{aligned} e_k^2(G)/n &\geqslant (c'_{\varepsilon',k})^2 W^2/n \\ &= (c'_{\varepsilon',k})^2 n/\Delta^{2(1+2/(k-(k-1)\varepsilon'-2))} \\ &= (c'_{\varepsilon',k})^2 n/\Delta^{2+4/(k-2)+\varepsilon} \end{aligned}$$

by our choice of ε'. Now, if this quantity is at least 1, then clearly W is $\Omega(1)$ in n since W is larger than n by a factor of $c'_{\varepsilon',k}\Delta^{1+2/(k-(k-1)\varepsilon'-2)}$ which is $\Omega(1)$. Therefore with probability $1 - o(1)$ in n, $e_k(G)^2/n \geqslant (c'_{\varepsilon',k})^2 n/\Delta^{2+4/(k-2)+\varepsilon}$. Clearly, we can choose $C_{\varepsilon,k}$ and can absorb the $-k$ in the constant $C_{\varepsilon,k}$ to obtain the desired result for resolution. For DPLL procedures, the result follows even more directly. □

It is worth noting that we can obtain a lower bound on the size of a proof of non-k-colorability in the system considered by McDiarmid [35] that is similar to the DPLL proof size lower-bound proof size since their bound on the proof width as a function of proof size only differs by a factor of $2k$. Thus, the same bound as the DPLL bound above holds for McDiarmid's system with a slightly different value of the constant $C'_{\varepsilon,k}$.

Corollary 4. *For each integer* $k \geqslant 3$, $\varepsilon > 0$, *and* $\Delta > 0$ *there is a constant* $C'_{\varepsilon,k} > 0$ *such that*

- *if* $p = 2\Delta/(n-1)$ *and* $G \sim \mathscr{G}_{n,p}$, *then with probability* $1 - o(1)$ *in n*, $Res(\chi(G,k)) \geqslant \exp(C'_{\varepsilon,k} n/\Delta^{2+4/(k-2)+\varepsilon})$.
- *if* $m = \Delta n$ *and* $G \sim \mathscr{G}_{n,m}$, *then with probability* $1 - o(1)$ *in n*, $Res(\chi(G,k)) \geqslant \exp(C'_{\varepsilon,k} n/\Delta^{2+4/(k-2)+\varepsilon})$.

Corollary 5. *For each integer* $k \geqslant 3$, $\varepsilon > 0$, *there is a* $\delta > 0$ *such that if* $m \leqslant n^{3/2-1/k-\varepsilon}$ *and* $G \sim \mathscr{G}_{n,m}$ *or* $G \sim \hat{\mathscr{G}}_{n,m}$ *then, with probability* $1 - o(1)$ *in n*, $Res(\chi(G,k)) \geqslant 2^{n^\delta}$.

Proof. For this range of m, $\Delta \leqslant n^{(k-2)/2k-\varepsilon}$. Applying Theorem 1 with a suitably small value of ε' in place of ε yields the desired result. □

Corollary 6. *For each integer* $k \geqslant 3$, $\varepsilon > 0$, *there is a* $\delta > 0$ *such that if* $m \leqslant n^{2-2/k-\varepsilon}$ *and* $G \sim \mathscr{G}_{n,m}$ *or* $G \sim \hat{\mathscr{G}}_{n,m}$ *then, with probability* $1 - o(1)$ *in n*, $DPLL(\chi(G,k)) \geqslant 2^{n^\delta}$.

4. Upper bounds

A very simple brute-force procedure achieves a $2^{O(n/\Delta)}$ upper bound for proving non-k-colorability, based on the following observation.

Lemma 8. *Let $p = 2\Delta/(n-1)$ and $G \sim \mathcal{G}(n, p)$. Let G_r be the subgraph of G induced on the first r vertices of G. If $r - 1 > c_k^+(n-1)/\Delta$ then with probability $1 - o(1)$ in r, G_r is not k-colorable.*

Proof. Clearly, if $G \sim \mathcal{G}(m, p)$ then the induced graph $G_r \sim \mathcal{G}(r, p)$. By our assumption on r, $p(r-1)/2 > c_k^+$. The conclusion follows by the definition of c_k^+. □

Therefore, the simple algorithm that searches through all possible k-colorings of the first $c_k^+(n-1)/\Delta + 1$ vertices of a $G \sim \mathcal{G}(n, p)$ will almost certainly find a witness to the non-k-colorability of G. Such an algorithm can easily be phrased as a simple DPLL search procedure, called the *ordered DPLL procedure* in [9], that always splits on the first unset variable. In fact, [37,14] analyze essentially the same simple backtracking procedure for k-coloring, although it is not described as being based on DPLL. (In their algorithm, vertices are listed in a fixed order and all colors of a vertex compatible with previously assigned vertices are tried recursively.) For this algorithm, with $p = 2\Delta/(n-1)$, $\Delta \in o(n)$ and $G \sim \mathcal{G}(n, p)$, they show that the log of the expected number of nodes in the search tree is $k(\log k)^2 n/4\Delta$ plus lower-order terms for $pn \to \infty$. For completeness, we state and prove the following simpler version of such a theorem.

Theorem 2. *Let $k > 1$, $p = 2\Delta/(n-1)$ and $G \sim \mathcal{G}(n, p)$. With probability $1 - o(1)$ in n, the ordered DPLL procedure where all k variables associated with each vertex are numbered consecutively witnesses the fact that Res$(\chi(G, k)) \leqslant 2^{O(k \log^2 k\ n/\Delta)} n^{O(1)}$.*

Proof. By the bound of Łuczak, $c_k^+ = k \ln k + O(k \ln \ln k)$. Let $r = \max\{\log n, c_k^+ (n-1)/\Delta + 1\}$. Apply Lemma 8 to say that with probability $1 - o(1)$ in r and thus $1 - o(1)$ in n since $n \leqslant 2^r$, the induced graph G_r on the first r vertices of G is not k-colorable.

The ordered DPLL procedure will have a branch for each of k^r different k-colorings of the first r vertices of G. Although there are k boolean variables associated with each vertex, it is easy to see that the height k tree corresponding to the branches on these k variables has only k non-trivial children. Therefore, the ordered DPLL tree has size at most proportional to k^r, the number of k-colorings. Plugging in the value of r yields the desired result. □

More generally, given any upper bound r' on the number of vertices so that random graphs with density Δ almost certainly have a minimal k-uncolorable subgraph of size at most r', one obtains a naive $2^{O(r' \log n)}$ algorithm that does a brute force search for such a subgraph. Such an algorithm can easily be phrased as a resolution proof of non-k-colorability. Lemma 8 simply shows that $r' = O(n/\Delta)$.

We can, however, do much better by using a more careful, if somewhat artificial, splitting rule and which we show w.h.p. proves that $G \sim \mathcal{G}(n, \Delta/n)$ is non-k-colorable in time $\exp(O(n/\Delta^{\alpha_k}))$ where

$$\alpha_k = \frac{k-1}{k-2}. \tag{2}$$

Interestingly, this value of α_k coincides with a heuristic bound suggested by calculations of Ein-Dor and Monasson [24] who estimated the running time of DPLL on

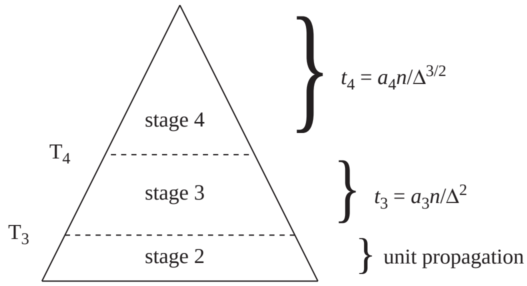

Fig. 1. Our depth-dependent splitting rule R is equivalent to a series of stages; during stage *j*, the algorithm is checking a subgraph V_j for *j*-colorability. Here $k = 4$.

$G \sim \mathscr{G}(n, \Delta/n)$ by making the (clearly false) assumption that the branches of the search tree are independent.

The splitting rule depends on the current level in the search tree, i.e., the number of vertices that are currently colored. We go through a series of stages; to test *G* for *k*-colorability, the algorithm starts in stage *k*, switches to stage $k - 1$ at a certain depth, and so on. The idea is that at stage *j*, the algorithm is currently checking a subgraph of *G* for *j*-colorability, where the vertices in this subgraph are neighbors of vertices colored at previous stages and have at most *j* allowed colors. This continues until we reach stage 2, at which point we check a subgraph for 2-colorability in polynomial time using unit propagation around an odd cycle. At each stage we use an arbitrary numbering, as in ordered DPLL, to choose among the vertices in the current subgraph. We illustrate this in Fig. 1.

We now define this splitting rule, which we call R. To introduce some notation, let *T* be the current depth, and for $1 \leqslant t \leqslant T$ let v_t be the vertex that was colored at level *t* of the tree and $c(v_t)$ be its assigned color. For $3 \leqslant i \leqslant k$ let $t_i = a_i n/\Delta^{\alpha_i}$, where α_i is given by (2) and $a_3, \ldots, a_k$ is a set of parameters we will define below. Let $T_j = \sum_{i=j}^{k} t_i$ with $T_{k+1} = 0$, and let $T_2 = n$; then we will say that we are currently in *stage j* if *j* is the largest integer such that $T < T_j$.

Given that we are in stage *j*, define a set of vertices V_j as follows. For $j < i \leqslant k$, let $S_i = \{v_t : T_{i+1} \leqslant t < T_i\}$ be the set of vertices colored during stage *i*, let c_i be the most common color among the vertices in S_i, and let $U_i = \{v \in S_i : c(v) = c_i\}$. Then, let V_j be the set of uncolored vertices v such that v has a neighbor $u_i \in U_i$ for all $j < i \leqslant k$. By construction the c_i are all distinct, so vertices in V_j have at most *j* allowed colors. We take $V_k = V$.

Finally, if $j > 2$, R splits on the vertex in V_j of smallest number. If $j = 2$, R splits on the vertex in V_2 of smallest number, and performs unit propagation whenever a vertex in V_2 exists with only one allowed color. (Note that although it does not take advantage of unit propagation until $j = 2$, up to a polynomial factor, the algorithm's running time can only improve if it performs additional unit propagations at an earlier stage.)

To do our analysis, it is convenient to recognize that DPLL with this splitting rule is equivalent to a recursive algorithm $\mathsf{A}_k(G)$ which colors $t = t_k$ vertices and then calls itself on the subgraph G' of uncolored neighbors of the vertices assigned the most common color. We illustrate this algorithm in Table 1.

Table 1
A recursive algorithm equivalent to the splitting rule R

```
Algorithm A_k(G) {
        If k = 2, check G for 2-colorability.
        Else {
                Let t = a_k n/Δ^{α_k}.
                If |G| < t, fail.
                Else, for all k^t assignments of colors to the t vertices of lowest number, do {
                        Let c be the most common color among these t vertices,
                        let U be the set of these vertices assigned color c, and
                        let G′ be the subgraph induced by U's uncolored neighbors.
                        Run A_{k−1}(G′).
                }
        }
}
```

Theorem 3. *Let $G \sim \mathscr{G}(n, p = \Delta/n)$. For all $k \geqslant 3$, there exist constants $a_3, \ldots, a_k$ and constants $b_k, c_k > 0$, and d_k such that, for all $\Delta > d_k$, DPLL with splitting rule R, or equivalently algorithm A_k, refutes $\chi(G, k)$ in time $\exp(b_k n/\Delta^{\alpha_k})n^{\mathrm{O}(1)}$ with probability at least $1 - \exp(-c_k n/\Delta^{\alpha_k})$, where $\alpha_k = (k-1)/(k-2)$. Therefore, we have*

$$DPLL(\chi(G, k)) \leqslant \exp(b_k n/\Delta^{\alpha_k})n^{\mathrm{O}(1)}$$

with probability $1 - \mathrm{o}(1)$ in n.

Our proof of Theorem 3 is similar to that of Theorem 6.1 of [9], which establishes an upper bound for DPLL on random k-SAT. The idea there is that setting a certain number of variables creates a large enough density of two-variable clauses so that an unsatisfiable 2-SAT subformula appears with high probability, which ordered DPLL with unit propagation proves is unsatisfiable in linear time. Similarly, we will show inductively that coloring the first t_k vertices yields a subgraph which A_{k-1} quickly proves is non-$(k-1)$-colorable, until $k = 2$ and we can check for 2-colorability quickly using our the smallest numbered vertex splitting rule and unit propagation. This induction works if the α_k obey a certain recurrence, yielding (2).

We first prove that if G has some simple properties, which hold with high probability whenever G has sufficiently large degree, then checking for non-2-colorability is extremely fast. (Ordered DPLL is particularly naive because it does not take advantage of the symmetry of the colors. If we used a more sophisticated splitting rule then we could easily derive a linear upper bound for all non-2-colorable graphs.)

Lemma 9. *There is a constant c_0 such that if a graph G has a non-2-colorable connected subgraph on at least $\frac{7}{8}$ of its vertices, then under a random numbering of the vertices of G, the expected time for ordered DPLL to refute $\chi(G, 2)$ is at most $c_0 n$.*

Proof. Observe that once ordered DPLL colors one vertex of a connected component of G, all other vertices of G receive the implied colors by unit propagation. Thus, as soon as ordered DPLL chooses some vertex from the large non-2-colorable connected component it reaches a contradiction and backtracks, tries the other color value, reaches a second contradiction and backtracks again, failing on that component. If such a vertex has the smallest number then a contradiction is reached immediately. However, if a vertex of some 2-colorable component is encountered previously then on backtracking past this component, ordered DPLL will recolor the first vertex of this 2-colorable component and then return to the non-2-colorable large subgraph and again determine failure before finally deriving a contradiction. Thus, if the smallest-numbered vertex of the large non-2-colorable component is numbered $b+1$ then the number of times we color first vertices in some component is at most $r=2^{b+1}$ and the cost of unit propagation on each component is Cn for some constant C. Since the large component has size at least $7n/8$ the probability that this happens is at most $8^{-b}=(2/r)^3$. The expected time is then at most $C\sum_{r=1}^{\infty}(2/r)^3 rn=(8C\sum_{r=1}^{\infty}1/r^2)n=(4C\pi^2/3)n$. □

We now prove that a random graph of sufficiently high, but constant, average degree is overwhelmingly likely to have the large non-2-colorable component called for by Lemma 9, so that our very simple DPLL procedure will determine that it is not 2-colorable in linear expected time. Note that, here and elsewhere, we have made no attempt to optimize the constants (other than α_k) that appear in the exponents.

Lemma 10. *There exist constants $c_2, d_2>0$ such that, for all $\Delta \geqslant d_2$, n sufficiently large and $G\sim\mathcal{G}(n, p=\Delta/n)$,*

$$\Pr[G \text{ contains a non-2-colorable connected component of size} \geqslant 7n/8] \geqslant 1-\mathrm{e}^{-c_2\Delta n}.$$

Proof. First, consider the probability that the largest connected component of G is of size less than $7n/8$. If this is the case then there must be some subset S of vertices with $n/16 \leqslant |S| < 15n/16$, that is, disconnected from $\bar{S}$. (To see this, consider the components $C_1,\dots,C_l$ of G in an arbitrary order, let j be the smallest integer such that $|C_1\cup\cdots\cup C_j|\geqslant n/16$ and set $S=C_1\cup\cdots\cup C_j$. Since $|C_j|\leqslant 7n/8$, $n/16\leqslant|S|<15n/16$.) For a fixed set S of this size the probability that there are no edges from S to $\bar{S}$ is at most

$$(1-p)^{|S||\bar{S}|}\leqslant \mathrm{e}^{-p(15n^2/256)}\leqslant \mathrm{e}^{-15\Delta n/256}.$$

Since there fewer than 2^n such sets S,

$$\Pr[G \text{ does not contain a connected component of size} \geqslant 7n/8]\leqslant \mathrm{e}^{-(15\Delta/256-\ln 2)n}.$$

We now show that with overwhelming probability, no subset of $m\geqslant n/2$ vertices of G can be bipartite. Fix one such subset B. Let X be the number of 2-colorings of B. There are 2^m 2-color assignments to the vertices of B. For any such assignment, the number of potential edges between vertices of the same color is at least $2\binom{\lfloor m/2\rfloor}{2}$, which is greater than $m^2/8$ for $n\geqslant 8$. Therefore,

$$\mathbf{E}[X]\leqslant 2^m(1-p)^{m^2/8}\leqslant 2^m\mathrm{e}^{-pm^2/8}=2^{n/2}\mathrm{e}^{-pn^2/32}\leqslant \mathrm{e}^{-(\Delta/32-(1/2)\ln 2)n}.$$

Since there are fewer than 2^n such sets, the probability that some such set is bipartite is at most $\mathrm{e}^{-(\Delta/32-(3/2)\ln 2)n}$.

Therefore, the overall probability that a non-2-colorable component of size at least $(\frac{7}{8})n$ fails to exist is at most

$$e^{-(15\Delta/256-\ln 2)n} + e^{-(\Delta/32-(3/2)\ln 2)n}$$

which is at most $e^{-c_2\Delta n}$ for $\Delta \geqslant d_2$ for some constants $c_2, d_2 > 0$. □

Below, we will use the following Chernoff bounds on the binomial distribution [7, Appendix A]:

$$\Pr[\mathrm{Bin}(m,q) < mq/4] \leqslant 2^{-mq/2}, \quad (3)$$

$$\Pr[\mathrm{Bin}(m,q) > Cmq] \leqslant (C/e)^{-Cmq}. \quad (4)$$

Now, to illustrate the idea and start our induction, we prove Theorem 3 in the case $k=3$.

Theorem 4. *Let $G \sim \mathcal{G}(n, p=\Delta/n)$. There exist constants $a_3, b_3, c_3 > 0$, and d_3 such that if $\Delta \geqslant d_3$, then A_3 refutes $\chi(G,3)$ in time at most $\exp(b_3 n/\Delta^2)\mathrm{O}(n)$ with probability at least $1-\exp(-c_3 n/\Delta^2)$.*

Proof. As in our definitions of R and A_3 above, let G' be the subgraph induced by the uncolored neighbors of the t colored vertices which have been assigned their most common color. We will show that w.h.p. G' satisfies the conditions of Lemma 9 for all 3^t assignments of the first t vertices, then A_3's expected running time will be at most $3^t c_0 n = \exp((a_3 \ln 3)n/\Delta^2)c_0 n$. Then, setting $b_3 = 2a_3 \ln 3$, by Markov's inequality the probability that the running time exceeds $\exp(b_3 n/\Delta^2)c_0 n$ is at most $\exp(-(a_3 \ln 3)n/\Delta^2)$.

The number n' of vertices in G' is binomially distributed as $\mathrm{Bin}(n-t, q)$ where

$$q \geqslant 1-(1-p)^{t/3} \geqslant 1-e^{-pt/3} \geqslant \frac{pt}{6} = \frac{a_3}{6\Delta}$$

since $e^{-x} \leqslant 1-x/2$ for $0 \leqslant x \leqslant 1$. Using the lower Chernoff bound (3) and choosing d_3 large enough so that $t = a_3 n/\Delta^2 < n/2$ for $\Delta \geqslant d_3$, we have

$$\Pr[n' < a_3 n/(48\Delta)] \leqslant 2^{-a_3 n/(24\Delta)}.$$

Let $a_3 = 48 d_2$, where d_2 is the constant defined by Lemma 10. Then we have

$$\Pr[n' < d_2 n/\Delta] \leqslant 2^{-2d_2 n/\Delta}. \quad (5)$$

Clearly, G' is distributed as $\mathcal{G}(n', p)$ if we condition on the value of n'. Let $\Delta' = pn'$ be the mean degree of G'; then if $n' \geqslant d_2 n/\Delta$ we have $\Delta' \geqslant d_2$. Combining Lemma 10 with (5) then gives

$$\begin{aligned}&\Pr[G' \text{ does not contain a non-2-colorable component of size } \geqslant 7n'/8]\\ &\quad \leqslant 2^{-2d_2 n/\Delta} + e^{-c_2 d_2^2 n/\Delta}\\ &\quad < \exp(-2Cn/\Delta)\end{aligned}$$

for some $C < \min(d_2 \ln 2, c_2 d_2^2/2)$. We wish to bound the probability that G' violates the condition of Lemma 9 for some assignment of the first t vertices. Choose d_3 large enough so that $(a_3 \ln 3)/d_3 < C$; then for all $\Delta \geqslant d_3$, the union bound gives a total probability of

$$3^t \exp(-2Cn/\Delta) < \exp\left(\left(\frac{a_3 \ln 3}{\Delta} - 2C\right)\frac{n}{\Delta}\right) < \exp(-Cn/\Delta).$$

Combining this with Markov's inequality as described above, the overall probability that the running time of A_3 exceeds $\exp(b_3 n/\Delta^2)c_0 n$ is at most

$$\exp(-(a_3 \ln 3)n/\Delta^2) + \exp(-Cn/\Delta) < \exp(-c_3 n/\Delta^2)$$

for all $\Delta \geqslant d_3$ and some $c_3 > a_3 \ln 3$. □

We can now prove Theorem 3 for all k.

Proof of Theorem 3. The proof works by induction on k, where each step of the induction parallels that of Theorem 4 and we use $k = 3$ as the base case. Our goal is to set a_k and d_k so that the average degree Δ' of G' is at least d_{k-1}.

First, the number n' of vertices in G' is distributed as $\mathrm{Bin}(n-t, q)$, where

$$q \geqslant 1 - (1-p)^{t/k} \geqslant 1 - \mathrm{e}^{-pt/k} \geqslant \frac{pt}{2k} = \frac{a_k}{2k\Delta^{\alpha_k - 1}}.$$

Choosing d_k large enough so that $t < n/2$ for $\Delta \geqslant d_k$ and setting $a_k = 16kd_{k-1}$, the Chernoff bounds (3), (4) (with $C = 4$) give

$$\Pr[n' < d_{k-1}n/\Delta^{\alpha_k - 1}] \leqslant 2^{-2d_{k-1}n/\Delta^{\alpha_k - 1}}, \tag{6}$$

$$\Pr[n' > 16d_{k-1}n/\Delta^{\alpha_k - 1}] \leqslant (4/\mathrm{e})^{-16d_{k-1}n/\Delta^{\alpha_k - 1}}. \tag{7}$$

Then if $n' \geqslant d_{k-1}n/\Delta^{\alpha_k - 1}$, we have $\Delta' = pn' \geqslant d_{k-1}\Delta^{2-\alpha_k} \geqslant d_{k-1}$, since $\alpha_k \leqslant 2$ and we assume w.l.o.g. $\Delta > 1$. Therefore, assuming the events of (6) and (7) hold, the running time of A_k (up to $n^{\mathrm{O}(1)}$) is

$$k^t \exp(b_{k-1}n'/(\Delta')^{\alpha_{k-1}}) \leqslant \exp\left(\left(a_k \ln k + \frac{16b_{k-1}d_{k-1}^{1-\alpha_{k-1}}}{\Delta^{\alpha_{k-1}(2-\alpha_k)-1}}\right)\frac{n}{\Delta^{\alpha_k}}\right). \tag{8}$$

Set $b_k = a_k \ln k + 16b_{k-1}d_{k-1}^{1-\alpha_{k-1}}$. Then if $\alpha_{k-1}(2-\alpha_k) - 1 = 0$, or alternately

$$\alpha_k = 2 - \frac{1}{\alpha_{k-1}}, \tag{9}$$

then the running time (8) becomes $\exp(b_k n/\Delta^{\alpha_k})$ as stated. Indeed, the solution to the recurrence (9) with the initial condition $\alpha_3 = 2$ is

$$\alpha_k = \frac{k-1}{k-2}.$$

We now bound the probability that, for some assignment of the first t vertices, A_{k-1} fails because $|G'| < t_{k-1}$ or its running time exceeds $\exp(b_k n/\Delta^{\alpha_k})$. First, note that since

$\alpha_k - 1 < \alpha_{k-1}$, for sufficiently large Δ we have $d_{k-1}n/\Delta^{\alpha_k-1} > t_{k-1}$, so the probability that $|G'| < t_{k-1}$ is bounded by (6). Then, combining (6) and (7) with our inductive assumption gives

$$\begin{aligned}&\Pr[\mathsf{A}_{k-1} \text{ fails or takes too long on } G']\\ &\quad\leqslant 2\cdot 2^{-2d_{k-1}n/\Delta^{\alpha_k-1}} + \exp\left(-\frac{c_{k-1}n'}{\Delta'^{\alpha_{k-1}}}\right)\\ &\quad\leqslant 2^{1-2d_{k-1}n/\Delta^{\alpha_k-1}} + \exp\left(-\frac{c_{k-1}d_{k-1}n}{(16d_{k-1})^{\alpha_{k-1}}\Delta^{\alpha_k}}\right)\\ &\quad< \exp(-2c_k n/\Delta^{\alpha_k})\end{aligned} \tag{10}$$

for some $c_k > c_{k-1}d_k/(16d_{k-1})^{\alpha_{k-1}}$, where we have set $d_k > c_{k-1}/((2\ln 2)(16d_{k-1})^{\alpha_{k-1}})$ so that the second term in (11) dominates for all $\Delta \geqslant d_k$. Furthermore, set c_k large enough so that $c_k > a_k \ln k$; then the union bound over the k^t assignments of the first t vertices gives

$$\begin{aligned}\Pr[\mathsf{A}_k \text{ fails}] &= k^t \exp(-2c_k n/\Delta^{\alpha_k})\\ &= \exp((a_k \ln k - 2c_k)n/\Delta^{\alpha_k})\\ &< \exp(-c_k n/\Delta^{\alpha_k})\end{aligned}$$

which completes the proof. □

Another approach, more closely analogous to [9], is to note that if $\Theta(t)$ of the initial vertices are assigned each color, then the number of their 2-color neighbors with a particular pair of allowed colors is w.h.p. $n' = \Theta(n(pt)^{k-2}) = \Theta(n/\Delta^{(\alpha_k-1)(k-2)})$. These 2-color vertices induce a graph G' of average degree $\Delta' = pn' = \Theta(\Delta^{1-(\alpha_k-1)(k-2)})$, and Lemma 10 shows that G' is w.h.p. non-2-colorable when $\Delta' \geqslant d_2$. This happens (with appropriate constants) when $\Delta' = \Theta(1)$, giving $1-(\alpha_k-1)(k-2) = 0$ and so $\alpha_k = (k-1)/(k-2)$.

However, since the colors assigned to the first t vertices are (negatively) correlated, proving that $\Theta(t)$ of them receive each of the k colors then becomes a separate problem. One possible method for this would be to use multitype branching processes as in [4].

5. Discussion of algorithms

5.1. Backtracking algorithms

In Section 4, we described the behavior of the backtracking algorithm analyzed in [37,14] on non-k-colorable graphs as a resolution refutation of the formula $\chi(G,k)$. This is very typical. Many more sophisticated backtracking-based coloring algorithms can be emulated by resolution (or by tree-like McDiarmid proofs to which our resolution-width bounds also apply). Our DPLL upper bound from Theorem 3 corresponds to one such natural algorithm which has better behavior. For example, most of the coloring algorithms in [31] were of this form.

Beigel and Eppstein also used recursive algorithms, based on extensive case analyses of local configurations, to give upper bounds for 3-coloring of $O(1.3446^n)$ [11], later improved to $O(1.3289^n)$ [12]. The case analysis is in terms of general constraint satisfaction problems (CSPs) with domain size 3 or 4 and binary constraints, of which 3-coloring is a special case where the domain size is 3 and all constraints are not equal constraints. We are not able, so far, to give strict resolution simulations for these algorithms. However, in [36] a variant of the algorithm in [11] is given which establishes the same bound of $O(1.3446^n)$ for $Res(\chi(G, 3))$. In the case of the algorithm given in [12], we can do the same for all but one case (among some two dozen).

The execution of the Beigel–Eppstein algorithms on input G may easily be described in terms of the formula $\chi(G, k)$. Consider the following "quasi-DPLL" algorithm scheme. For formula ϕ, select a set $S \subset vars(\phi)$ of variables based on some local property of ϕ, and consider a set A of (possibly partial) assignments to S. For each α in A, either α makes a clause C of ϕ false, or we make a recursive call to solve the restriction of ϕ by α. If the set A of assignments covers all assignments to S (that is, each assignment to S is an extension of some partial assignment in A), then ϕ is unsatisfiable if and only if each restricted formula is unsatisfiable. Any "quasi-DPLL" algorithm of this sort can be efficiently simulated by resolution as follows. For each α in A we may derive a clause which α makes false, and from these derived clauses plus clauses of ϕ that mention variables in S, we may construct a refutation of ϕ. The derived clauses are obtained as follows. If α makes a clause $C \in \phi$ false, we use the clause C, otherwise the recursive call to refute the restriction of ϕ by α returns a clause that α makes false. In the case where ϕ is already a restriction of the input formula by partial assignment τ, we can extract from the refutation of ϕ a clause which τ makes false, and return this clause to the parent invocation. (The refutations produced may not be strictly tree-like, but are nearly tree-like in that non-tree parts are local.)

Most of the cases in the Beigel–Eppstein algorithms are captured fairly easily by this scheme. The bounds are obtained by careful choices for sets S and A, and the corresponding number and sizes of recursive calls. However, in a small number of cases the set A of assignments does not cover all assignments to S. In these cases, the completeness arguments do not have direct analogs in resolution, and to obtain the bounds in [36] alternate handling or subcase breakdown was used. In the algorithm of [12] one case makes use of polytime bipartite matching, which is a more serious impediment to simulation by resolution. Nonetheless, it is plausible that our lower bounds can be shown to apply to both of the Beigel–Eppstein algorithms.

The connection between k-coloring algorithms and resolution-based algorithms for $\chi(G, k)$ is also borne out in practice. For example, the program used for coloring in [21] also uses graph reductions based on removing vertices, or merging vertices. These, too, can be emulated by resolutions, mostly on two clauses, but occasionally involving 3-clauses. It also does significant forward pruning by propagating statements of the type "the color of v is i or the color of w is j". These can be expressed as 2-clauses $(x_{v,i} \vee x_{w,j})$ which can be generated by resolution. In extensive head-to-head tests [21], this program on graphs and the back-jumping version of the DPLL-based tableau program **ntab-back** [20] on the associated $\chi(G, k)$ behave statistically alike on random 3-color instances. The coloring program is better for larger values of k, but this is likely due to the fact that tableau was tuned to 3-SAT instances and the fact that some features (e.g. clustering) are easier to identify knowing the graph structure.

5.2. Other complete algorithms

There are many other algorithms suitable for k-coloring or k-colorability that are not primarily based on backtracking search and therefore are not covered by our resolution bounds. Often, they are tuned for use on the random graphs we consider. However most non-backtracking coloring algorithms, such as those in [15,6,26,16] are *incomplete* in that they may miss some possible k-colorings and therefore do not produce certificates of non-k-colorability. We will only be concerned with complete algorithms.

One class is particularly interesting. At sufficiently large density, any graph is almost certainly not k-colorable, however this fact does not in itself provide a proof of such a graph's non-k-colorability. However, recently, Krivelevich and Vu [32] showed that tight concentration bounds on the polynomially computable Lovasz ϑ function of random graphs can be used to approximate the chromatic number in polynomial expected time for sufficiently large graph density Δ. Coja-Oghlan [19] has used the same general technique and sharper concentration bounds for the polynomial-time computable *vector chromatic number* $\bar{\vartheta}_{1/2}(G)$ to derive a polynomial expected time k-colorability algorithm for $k = \mathrm{o}(\sqrt{n})$ and $\Delta > ck^2$ for some constant $c > 0$. More precisely, since $\bar{\vartheta}_{1/2}(G) \leqslant \vartheta(\bar{G}) \leqslant \chi(G)$ for all graphs G, the algorithm first tests if $\bar{\vartheta}_{1/2}(G) > k$ (using semi-definite programming); if so, then this provides a proof of the non-k-colorability of G. Otherwise, the algorithm then calls a standard worst-case exponential-time k-coloring algorithm such as that of Lawler [33]. The concentration results for $\bar{\vartheta}(G)$ are such that it is exponentially unlikely that the algorithm will need to resort to the second stage and thus the algorithm has polynomial expected running time.

Clearly, based on our results, these algorithms are provably superior to resolution and backtracking for proving non-k-colorability of random graphs when the graph density is sufficiently large compared to k. These algorithms use the typical properties of a random input to try to quickly produce a certificate that the input is not k-colorable. Although such a certificate may not be guaranteed to exist, it works sufficiently frequently that it is useful. Similar ideas using spectral methods have also been employed by Goerdt et al. [28,27] for random k-SAT. It is interesting to note that in the case of k-SAT, although the best current algorithms almost certainly yield efficient certificates in a wider range of densities than does DPLL, in contrast to the situation for k-colorability it is still open whether or not they do the same when compared with general resolution.

Finally, we note that since $\chi(G, k)$ is a k-CNF formula, any complete deterministic algorithm for k-SAT is potentially relevant for k-coloring. Most such algorithms that have been analyzed are themselves resolution-based although there are exceptions such as the $(2 - 2/(k+1))^n$ worst-case time algorithm of Dantsin et al. [22] which is not competitive and does improve with improved graph density.

6. Open problems

There is a gap between the exponents of Δ in our upper and lower bounds that would be nice to close, particularly for DPLL. Our DPLL upper bound is of the form $\exp(\mathrm{O}(n/\Delta^{(k-1)/(k-2)}))$, whereas our lower bound is of the form $\exp(\Omega(n/\Delta^{k/(k-2)+\varepsilon}))$.

Classic results [25] show that for $\Delta = \omega(n^{(k-2)/k})$ a random $G \sim \mathcal{G}(n, \Delta/n)$ contains a $(k+1)$-clique with probability $1 - o(1)$. The presence of such a clique obviously yields a $O(k^2)$ size DPLL proof of non-k-colorability so we should consider densities below this bound. It is interesting that this is essentially the same range where our current lower bound yields non-trivial results. This suggests that our DPLL lower bound (except maybe for the ε) is the more likely to be the correct bound.

One obstacle for improving the DPLL upper bound to match the lower bound may be that the lower bound allows an optimal literal selection rule that may not necessarily be obtainable via an efficient DPLL algorithm. It would be interesting to see if one could obtain improved lower bounds for simple literal selection rules or show that simpler selection rules can achieve the same or better upper bounds.

Our bounds apply only above the threshold for k-colorability, but the use of $\chi(G, k)$ in k-coloring algorithms also is suitable in the k-colorable region since satisfying assignments correspond to k-colorings (unlike the ϑ-based algorithms from Section 5 which yield no information about a k-coloring when the graph is k-colorable). Jia and Moore [30] have recently shown exponential behavior of a natural greedy DPLL algorithm on easily k-colorable random graphs below the threshold: this greedy algorithm repeatedly misses constant-size contradictory subproblems. However, such behavior can easily be eliminated with a more sophisticated literal selection rule. Are there other more sophisticated algorithms where such bounds can be shown for satisfiable instances?

Acknowledgements

We would like to thank Dimitris Achlioptas, Jared Saia, and Vishal Sanwalani for helpful references and discussions.

References

[1] D. Achlioptas, E. Friedgut, A sharp threshold for k-colorability, Random Structures Algorithms 14 (1999) 63–70.

[2] D. Achlioptas, M. Molloy, The analysis of a list-coloring algorithm on a random graph, in: Proceedings 38th Annual Symposium on Foundations of Computer Science, IEEE Press, New York, October 1997, pp. 204–212.

[3] D. Achlioptas, M. Molloy, Almost all graphs with 2.522n edges are not 3-colorable, Electron. J. Combin. 6 (1) (1999) R29.

[4] D. Achlioptas, C. Moore, Almost all graphs with average degree 4 are 3-colorable, in: Proceedings of the 34th Annual ACM Symposium on Theory of Computing, Montreal, Quebec, Canada, May 2002, pp. 199–208.

[5] D. Achlioptas, A. Naor, The two possible values of the chromatic number of a random graph, in: Proceedings of the Thirty-Sixth Annual Symposium on Theory of Computing, Chicago, IL, June 2004, pp. 587–593.

[6] N. Alon, N. Kahale, A spectral technique for coloring random 3-colorable graphs, SIAM J. Comput. 26 (6) (1997) 1733–1754.

[7] N. Alon, J.H. Spencer, P. Erdös, The Probabilistic Method, Wiley, New York, 1992.

[8] D. Angluin, L. Valiant, Fast probabilistic algorithms for Hamiltonian circuits and matchings, J. Comput. System Sci. 18 (1979) 155–193.

[9] P. Beame, R. Karp, T. Pitassi, M. Saks, The efficiency of resolution and Davis–Putnam procedures, SIAM J. Comput. 31 (4) (2002) 1048–1075.
[10] P.W. Beame, Toniann Pitassi, Simplified and improved resolution lower bounds, in: Proceedings 37th Annual Symposium on Foundations of Computer Science, Burlington, VT, IEEE Press, New York, October 1996, pp. 274–282.
[11] R. Beigel, D. Eppstein, 3-coloring in time $O(1.3446^n)$: a no-MIS algorithm, in: Proceedings 36th Annual Symposium on Foundations of Computer Science, Milwaukee, WI, IEEE Press, New York, October 1995, pp. 444–452.
[12] R. Beigel, D. Eppstein, 3-coloring in time $O(1.3289^n)$. Technical Report eprint cs.DS/0004046, CoRR, June 2000.
[13] E. Ben-Sasson, A. Wigderson, Short proofs are narrow—resolution made simple, J. ACM 48 (2) (2001) 149–169.
[14] E.A. Bender, H.S. Wilf, A theoretical analysis of backtracking in the graph coloring problem, J. Algorithms 6 (2) (1985) 275–282.
[15] A. Braunstein, M. Mezard, M. Weigt, R. Zecchina, Constraint satisfaction by survey propagation. ⟨http://lanl.arxiv.org/ps/cond-mat/0212451⟩.
[16] M. Chiarandini, T. Stützle, An application of iterated local search to graph coloring, in: Proceedings of the Computational Symposium on Graph Coloring and its Generalizations, 2002, pp. 112–125.
[17] V. Chvátal, Endre Szemerédi, Many hard examples for resolution, J. ACM 35 (4) (1988) 759–768.
[18] M. Clegg, J. Edmonds, R. Impagliazzo, Using the Gröbner basis algorithm to find proofs of unsatisfiability, in: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, Philadelphia, PA, May 1996, pp. 174–183.
[19] A. Coja-Oghlan, The Lovász number of random graphs, in: Proceedings of the Seventh International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM 2003), 2003.
[20] J.M. Crawford, L.D. Auton, Experimental results on the crossover point in random 3SAT, Artificial Intelligence 81 (1996) 31–57.
[21] J. Culberson, I.P. Gent, Frozen development in graph coloring, Theoret. Comput. Sci. 265 (1–2) (2001) 227–264.
[22] E. Dantsin, A. Goerdt, E. Hirsch, R. Kannan, J. Kleinberg, C. Papadimitriou, P. Raghavan, U. Schöning, A deterministic $(2 - 2/(k + 1))^n$ algorithm for k-SAT based on local search, Theoret. Comput. Sci. 280 (2002) 69–83.
[23] M. Davis, G. Logemann, D. Loveland, A machine program for theorem proving, Comm. ACM 5 (1962) 394–397.
[24] L. Ein-Dor, R. Monasson, The dynamics of proving uncolourability of large random graphs i: symmetric colouring heuristic, J. Phys. A 36 (2003) 11055–11067.
[25] P. Erdös, A. Renyi, On the evolution of random graphs, Magyar Tud. Akad. Mat. Kat. Int. Kozl. 5 (1960) 17–61.
[26] C. Fleurent, J.A. Freland, Genetic and hybrid algorithms for graph coloring, Ann. Oper. Res. 63 (1996) 437–464.
[27] J. Friedman, A. Goerdt, Recognizing more unsatisfiable random 3-SAT instances efficiently, in: J. Orejas, P.G. Spirakis, J. van Leeuwen (Eds.), Automata, Languages, and Programming: 28th International Colloquium, Lecture Notes in Computer Science, vol. 2076, Heraklion, Crete, Springer, Berlin, July 2001, pp. 310–321.
[28] A. Goerdt, M. Krivelevich, Efficient recognition of random unsatisfiable k-SAT instances by spectral methods, in: (STACS) 2001: 18th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, vol. 2010, Dresden, Germany, Springer, Berlin, February 2001, pp. 294–304.
[29] S. Janson, T. Łuczak, A. Ruciński, Random Graphs, Wiley, New York, 2000.
[30] M. Jia, C. Moore, How much backtracking does it take to color sparse random graphs? Rigorous results on heavy tails, in: Principle and Practice of Constraint Programming - CP 2004, Tenth International Conference, Proceedings of Lecture Notes in Computer Science, vol. 3258, Toronto, Canada, 2004. Springer, pp. 742–746.
[31] D.S. Johnson, M. Trick (Eds.), Cliques, Coloring, and Satisfiability: Second Dimacs Implementation Challenge, October 11–13, 1993, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26. American Mathematical Society, Providence, RI, 1996.
[32] M. Krivelevich, V. Vu, Approximating the independence number and the chromatic number in expected polynomial time, J. Combin. Optimization 6 (2002) 143–155.

[33] E.L. Lawler, A note on the complexity of the chromatic number problem, Inform. Process. Lett. (1976) 66–67.
[34] T. Łuczak, The chromatic number of random graphs, Combinatorica 11 (1991) 45–54.
[35] C. McDiarmid, Colouring random graphs, Ann. Oper. Res. 1 (1984) 183–200.
[36] D.G. Mitchell, The resolution complexity of constraint satisfaction, Ph.D. Thesis, University of Toronto, 2002.
[37] H.S. Wilf, Backtrack: an $O(1)$ expected time algorithm for the graph coloring problem, Inform. Process. Lett. 18 (3) (1984) 119–121.