Exact and Approximative Algorithms for Colouring $G(n,p)$

Amin Coja-Oghlan* and Anusch Taraz

Humboldt-Universität zu Berlin, Institut für Informatik
Unter den Linden 6, 10099 Berlin, Germany
{coja,taraz}@informatik.hu-berlin.de

Abstract. We investigate the problem of colouring random graphs $G \in G(n,p)$ in polynomial expected time. For the case $p \leq 1.01/n$, we present an algorithm that finds an optimal colouring in linear expected time. For $p \gg \ln^6(n)/n$, we give algorithms which approximate the chromatic number within a factor of $O(\sqrt{np})$. We also obtain an $O(\sqrt{np}/\ln(np))$-approximation algorithm for the independence number. As an application, we propose an algorithm for deciding satisfiability of random $2k$-SAT formulas (with sufficiently many clauses) in polynomial expected time.

1 Introduction and Results

The problem of determining the minimum number of colours needed to colour a graph G – denoted by the $\chi(G)$, the chromatic number of G – is known to be NP-hard. In fact, Feige and Kilian [6] have shown that, unless $coRP = NP$, no polynomial time algorithm with approximation ratio less than $n^{1-\varepsilon}$ exists. Therefore the question arises whether algorithms can be designed and analyzed that perform well on *average* or *random* instances. The binomial model for such a *random graph*, usually denoted by $G(n,p)$, is defined as follows. In a graph with vertex set $[n] = \{1,\ldots,n\}$ every possible edge is present with probability p independently of all others. Here $p = p(n)$ can be a function in n.

Let $b = 1/(1-p)$. Bollobás and Luczak (cf. [3, 13]) proved that almost surely a random graph $G \in G(n,p)$ satisfies

$$\chi(G) \sim \begin{cases} \frac{n}{2\log_b n} & \text{for constant } p \\ \frac{np}{2\ln np} & \text{for } \frac{C}{n} < p = o(1) \end{cases} \qquad \alpha(G) \sim \begin{cases} 2\log_b n & \text{for constant } p \\ \frac{2\ln np}{p} & \text{for } \frac{C}{n} < p = o(1), \end{cases} \quad (1)$$

where C is a sufficiently large constant and where we say that $G(n,p)$ has a certain property A *almost surely* (a.s.), if the probability that $G \in G(n,p)$ has A tends to 1 as $n \to \infty$.

For even smaller values of p, and constant k, the situation is as follows. For any number $k \geq 3$, there are constants c_k^- and c_k^+ such that for $p \leq (1-\varepsilon)c_k^-/n$

* Research supported by the Deutsche Forschungsgemeinschaft (DFG FOR 413/1-1). An extended abstract of this paper appears at STACS 2003.

(and $p \geq (1+\varepsilon)c_k^+/n$ respectively), $G(n,p)$ almost surely is (respectively is not) k-colourable. In fact, it is conjectured that one can actually choose $c_k^- = c_k^+$ [1].

Returning to the algorithmic issue, let us consider the greedy colouring algorithm, which takes the vertices in an arbitrary order and assigns to each vertex the smallest possible colour. Grimmett and McDiarmid [11] proved that applied to $G(n, \frac{1}{2})$ the greedy algorithm a.s. uses $\sim n/\log_2 n$ colours. Though a slight improvement using randomization has been obtained by Krivelevich and Sudakov [20], to date the best known approximation ratio is asymptotically 2.

It is worth emphasizing that for inputs from $G(n,p)$ the greedy algorithm *always* has polynomial running time (in fact, linear) and achieves a 2-approximation *almost surely*. Nevertheless, it cannot *guarantee* a good approximation ratio. Indeed, there are input instances for which the approximation ratio achieved by the greedy algorithm is $\Omega(n)$, even for almost all permutations of the vertices [21]. This motivates the following question by Karp [15]:

> Is there an algorithm that for inputs from $G(n,p)$ has *expected* polynomial running time and *always* uses the minimum number of colours? (2)

Here the expected running time of an algorithm $\mathcal{A}$ is defined as $\sum_G R_{\mathcal{A}}(G)\mathrm{P}(G = G(n,p))$, where the sum runs over all graph G with vertex set $[n]$, and $R_{\mathcal{A}}(G)$ is the running time needed by $\mathcal{A}$ for input G. Our first result answers question (2) in the affirmative for the case of small edge probabilities p.

Theorem 1. *For $p \leq 1.01/n$ there exists a colouring algorithm which finds an optimal colouring and, when applied to $G(n,p)$, has linear expected running time.*

We remark that the constant 1.01 is certainly not best possible. It merely demonstrates that the algorithm can find optimal colourings even after $G(n,p)$ has passed the *phase transition* where it suddenly becomes much more complex. Variants of question (2) have been addressed by several authors, but over a different probability distribution than $G(n,p)$ (for references cf. [19]).

A new line of research has recently been initiated by Krivelevich and Vu [17], relaxing question (2) in the following way. Is there an algorithm that for inputs from $G(n,p)$ has *expected* polynomial running time and *always* approximation ratio at most r? Krivelevich and Vu [17] prove that for $n^{-1/2+\varepsilon} \leq p \leq 0.99$, there exists a colouring algorithm with approximation ratio $O(\sqrt{np}/\ln n)$ and expected polynomial running time over $G(n,p)$. Moreover they ask [17, 19] whether it is possible to find similarly good approximation algorithms for smaller values of p. Our next two results give positive answers to this question.

Theorem 2. *For $\ln(n)^6/n \ll p \leq 3/4$ there exists a colouring algorithm with approximation ratio $O(\sqrt{np})$ and polynomial expected running time over $G(n,p)$.*

Theorem 3. *Let $p \geq 1/n$. Then there exists a colouring algorithm with approximation ratio at most $3np$ and linear expected running time over $G(n,p)$.*

As an immediate by-product from our work on graph colouring, we obtain the following approximation algorithm for the independent set problem.

Theorem 4. *For $\ln(n)^6/n \ll p \leq 3/4$ there exists an algorithm that approximates the independence number within $O(\sqrt{np}/\ln(np))$ and has polynomial expected running time over $G(n,p)$.*

Krivelevich [18] has shown that for every fixed $k \geq 3$ there exists a constant $C = C(k)$ so that k-colourability can be decided in expected polynomial time over $G(n,p)$, provided that $p(n) \geq C/n$. Furthermore, he asks [19] whether it is possible to find such algorithms (i) for the case where $k = k(n)$ is a function growing in n and (ii) for any value of the edge probability $p = p(n)$. Our methods contribute to answering the above questions, even in a more general *semirandom* situation.

Semirandom graph problems have been studied e.g. in [7, 4]. In contrast to $G(n,p)$, semirandom settings allow for an adversary to change the outcome of the random experiment to a certain degree. In this way they capture a larger class of input distributions and therefore require rather robust algorithmic techniques. Here we introduce two semirandom models, $G(n,p)^+$ and $G(n,p)^-$, and give algorithms for deciding k-colourability. In the model $G(n,p)^+$, instances are created in two steps as follows:

1. First, a random graph $G_0 = G(n,p)$ is chosen.
2. Then, an adversary may *add* edges to G_0, in order to produce the input instance G.

Note that the adversary can change the vertex degrees, the independence number, the chromatic number and even the spectrum of the random graph G_0. Instances of the semirandom model $G(n,p)^-$ are created similarly, but here the adversary is only allowed to *remove* edges. Given $G_0 = G(n,p)$, let $\mathcal{I}(G_0)^+$ denote the set of all *instances over* G_0, i.e. the set of all graphs G that can be produced by adding edges to G_0. We define $\mathcal{I}(G_0)^-$ similarly. An algorithm $\mathcal{A}$ is said to run in *expected polynomial time applied to* $G(n,p)^+$ if there is a constant l such that for any map I that to each graph G_0 associates an element $I(G_0) \in \mathcal{I}(G_0)^+$, we have $\sum_{G_0} R_{\mathcal{A}}(I(G_0))P(G_0 = G(n,p)) = O(n^l)$, where $R_{\mathcal{A}}(I(G_0))$ denotes the running time of $\mathcal{A}$ on input $I(G_0)$. We define a similar notion for the model $G(n,p)^-$.

Theorem 5. *For any $k = k(n)$ and $p \leq k/(10n)$ there exists an algorithm which decides whether a graph G is k-colourable, and, when applied to $G(n,p)^-$, has linear expected running time.*

Theorem 6. *Let $k = k(n)$ and $p = p(n) \geq \max\{\ln(n)^7/n, 200k^2/n\}$. There exists an algorithm which decides whether a graph G is k-colourable, and, when applied to $G(n,p)^+$, has polynomial expected running time.*

Observe that Theorem 5 deals with the range where $G(n,p)$ almost surely is k-colourable, while Theorem 6 concerns graphs which almost surely are not k–colourable. (The threshold for k-colourability has order $(k \ln k)/n$.)

As an application of our work on the maximum independent set problem, we present an improved algorithm for deciding whether a random k-SAT formula

is satisfiable, where $k \geq 4$ is an even integer. (This problem is NP-complete in the worst case.) Let $\mathcal{F}(n,k,m)$ denote the set of all k-SAT formulas over n propositional variables $x_1,\ldots,x_n$, equipped with the uniform distribution. It is known [8] that there exists a threshold $c_k = O(1)$ such that the random formula $\mathcal{F}(n,k,m)$ in the case $m \leq (1-\varepsilon)c_k n$ is satisfiable with probability $1-o(1)$ as $n \to \infty$ (analogously, unsatisfiable for $m \geq (1+\varepsilon)c_k n$). The best known algorithmic result concerning random k-SAT, $k \geq 4$ even, is due to Goerdt and Krivelevich [10], who give a polynomial time algorithm that in the case $m \geq n^{k/2+o(1)}$ a.s. finds a certificate that $\mathcal{F}(n,k,m)$ is unsatisfiable. Replacing the spectral techniques employed in [10], we can extend this result by giving an algorithm with polynomial *expected* running time that decides satisfiability of *any* formula $F \in \mathcal{F}(n,k,m)$, $m \geq n^{k/2+o(1)}$. In fact, our algorithm can even handle *semirandom formulas*, generated according to the following model $\mathcal{F}(n,k,m)^+$.

1. A formula $F_0 = \alpha_1 \wedge \cdots \wedge \alpha_m \in \mathcal{F}(n,k,m)$ is chosen at random.
2. An adversary picks any formula F over $x_1,\ldots,x_n$ in which at least one copy of each α_i, $i = 1,\ldots,m$, occurs.

(Note that in general we cannot reconstruct F_0 from F.) Let us by $\mathcal{I}(F_0)$ denote the set of all formulas that can be obtained according to 2. above. We call $\mathcal{I}(F_0)$ the set of all *instances over* F_0. For any k-SAT formula F, let $\ell(F)$ denote the number of clauses in F. An algorithm $\mathcal{A}$ *has polynomial expected running time applied to* $\mathcal{F}(n,k,m)^+$, if there exists a constant $l > 0$ such that the expected running time of $\mathcal{A}$ is $O((n+\ell(F))^l)$.

Theorem 7. *Suppose that* $m \geq \ln(n)^7 n^{k/2}$. *There exists an algorithm* `DecideSAT` *that for any* k*-SAT formula* F *over* $x_1,\ldots,x_n$ *finds a satisfying assignment, if* F *admits any, and outputs "unsatisfiable" otherwise. Applied to* $\mathcal{F}(n,k,m)^+$, `DecideSAT` *has a polynomial expected running time.*

The remainder of the paper is organized as follows. After introducing the necessary technical terminology, we shall first prove Theorem 3 in Section 2. Section 3 deals with Theorem 1. The algorithms for Theorems 2 and 4 can be found in Section 4. Finally, in Section 5 we give the algorithms for Theorems 5 and 6 while Section 6 contains the proof of Theorem 7.

The Euler constant will be denoted by $\mathrm{e} = \exp(1)$. For a graph $G = (V,E)$ and a subset $S \subseteq V$ we let $G[S]$ be the subgraph of G induced by S. We write $H \subseteq G$ if H is a (weak) subgraph of G. Denote by $|G| = |V|$ the order of G. For a vertex v in G we let $d(v) = d_G(v)$ be the degree of v in G. Denote by $\delta(G)$ the minimal degree in G. In this paper we disregard rounding issues since they do not affect our arguments.

2 A simple $O(np)$-approximation algorithm: Theorem 3

Consider an integer $k \geq 3$. The k-*core* of a graph G is the unique subgraph $H \subseteq G$ of maximum cardinality such that $\delta(H) \geq k$. Pittel, Spencer and Wormald [23]

computed the constants γ_k which determine the precise threshold $p = \gamma_k/n$ for the appearance of the k-core in $G(n,p)$; e.g. $\gamma_3 \approx 3.35$. Although almost surely $G(n,p)$ does not contain a k-core of size, say, $\ln^{10}(n)$ at, say, $p = 1/n$, the probability for this event is not exponentially small in n, and hence such an event is not sufficiently unlikely to allow us to find the optimal colouring by brute force *on the whole graph.* Instead, the following simple algorithm will only deal with the vertices in the core.

Algorithm 8. `CoreColour`(G,k)
Input: A graph $G = (V,E)$ and an integer k.
Output: A colouring of G.

1. Let $G' := G$, set stack $S := \emptyset$.
2. While in G' there exists a vertex v with $d_{G'}(v) \le k-1$ do
   push v onto S, $G' := G' - v$.
3. Colour G' exactly in time $O(2.443^{n'})$, where $n' = |G'|$ (cf. [22]).
4. Extend this to a colouring of G: while $S \neq \emptyset$ do
   pick the top vertex v from S, assign to v the least possible colour.

Proposition 9. *For any graph G and any integer $k \ge 3$, `CoreColour`(G,k) finds a colouring of G with at most $\max\{\chi(G), k\}$ colours.*

Proof. Step 3 uses $\chi(G') \le \chi(G)$ colours. Let v be an arbitrary vertex. Since $d_{G'}(v) \le k-1$ when v was removed, Step 4 will never need a colour greater than k, since v has at most $k-1$ coloured neighbours at the moment it is coloured. □

Observe next that at the beginning of Step 3, the vertices in G' make up the k-core. Hence, to examine the expected running time of `CoreColour`, we need an upper bound on the probability that G contains a k-core of a certain size.

Lemma 10. *Let $p \ge 1/n$ and consider an integer $k \ge \mathrm{e}^2 np$. Then*

$$P(G(n,p) \text{ contains a } k\text{-core on } \nu \text{ vertices}) < \mathrm{e}^{-\nu}$$

Proof. The probability that there exists a k-core on ν vertices is bounded by the probability that there exists a weak subgraph H on ν vertices with $\mu = k\nu/2$ edges. Let $N = \binom{\nu}{2}$. Then the expected number of such subgraphs is given by

$$\binom{n}{\nu}\binom{N}{\mu}p^\mu \le \left(\frac{\mathrm{e}n}{\nu}\right)^\nu \left(\frac{\mathrm{e}\nu^2 p}{2\mu}\right)^\mu \le \left[\frac{\mathrm{e}n}{\nu}\left(\frac{\mathrm{e}\nu p}{k}\right)^{k/2}\right]^\nu \le \left[\left(\frac{\nu}{\mathrm{e}n}\right)^{\frac{k}{2}-1}\right]^\nu \le \mathrm{e}^{-\nu},$$

where the last step used $\nu \le n$ and $k \ge \mathrm{e}^2 np \ge 4$ because of $p \ge 1/n$. □

Proof of Theorem 3. We can easily check whether $\chi(G) \le 2$, and, if so, output an optimal colouring. So assume that $\chi(G) > 2$. Set $k := \mathrm{e}^2 np$ and apply `CoreColour`$(G(n,p),k)$. Proposition 9 shows that the algorithm has approximation ratio at most $k/3 = (\mathrm{e}^2/3)np$ as claimed. Its running time is linear for Step 2 and Step 4. As $2.443 < \mathrm{e}$, Lemma 10 implies that the expected running time of Step 3 can be bounded by $\sum_{\nu=0}^{n} 2.443^\nu \cdot \mathrm{P}(|G'| = \nu) = O(n)$. □

3 Finding an optimal colouring: Theorem 1

The underlying idea for the proof of Theorem 1 is basically the same as in Section 2, but we need to refine both the algorithm and the analysis a little in order to achieve an approximation ratio 1.

The basic idea of our algorithm is as follows: having checked for 2-colourability, we again peel off vertices until we get to the 3-core. For the 3-core we merely decide whether it is 3-colourable. Here we employ an exact algorithm by Beigel and Eppstein [2] which decides in time $O(1.3447^n)$ whether a given graph is 3-colourable or not. If it is 3-colourable, then we can obviously find a 3-colouring in $O(n^2 1.3447^n)$ steps, simply by trying to add edges to the graph as long as the graph remains 3-colourable. Thus in the end we have a complete 3-partite graph and can read off the colouring. If the graph is not 3-colourable, then we continue shaving off vertices until we get to the 4-core, which will most probably be substantially smaller than the 3-core. Now we optimally colour the 4-core using Lawler's algorithm.

Algorithm 11. `ExactColour`(G)
Input: A graph $G = (V, E)$
Output: A colouring of G.

1. If $\chi(G) \leq 2$ then find an optimal colouring of G and stop.
2. Let $G' := G$, set stack $S := \emptyset$.
   While there exists a vertex v with $d_{G'}(v) \leq 2$ do
   push v onto S, let $G' := G' - v$.
3. Check whether $\chi(G') \leq 3$ using Beigel and Eppstein's algorithm.
4. If yes then find a 3-colouring of G' in time $O(n'^2 1.3447^{n'})$, where $n' = |G'|$.
5. If no then run `CoreColour`$(G', 4)$.
6. Extend the obtained colouring of G' to a colouring of G as follows:
   while $S \neq \emptyset$ do
   pick the top vertex v from S, assign to v the least possible colour.

Proof of Theorem 1. By similar arguments as in Section 2, it is clear that the algorithm produces an optimal colouring.

In order to prove that the expected running time of `ExactColour` is linear, we have to estimate the probability that the k-core of $G(n,p)$ is of size ν for each $\nu \in \{1, 2, \ldots, n\}$ and $k = 3, 4$. Let $c = np$ and recall that $c \leq 1.01$. For any constant k we can bound the mentioned probability using the first moment method as follows. Given a sequence $d_1, \ldots, d_\nu$ with $\sum_{i=1}^{\nu} d_i = 2\mu$, there are at most

$$\frac{(2\mu)_\mu}{2^\mu d_1! \cdots d_\nu!} = \Theta\left(\frac{\mu^\mu 2^\mu}{\exp(\mu)\prod_{i=1}^{\nu}(d_i!)}\right) \tag{3}$$

labelled graphs on ν vertices such that the ith vertex has degree d_i. We have to count graphs with minimum degree k, hence $d_i \geq k$ for all i. Define $\lambda \geq 0$ by $\sum_{i=1}^{\nu} d_i = k\nu + 2\lambda$. Then

$$\prod_{i=1}^{\nu}(d_i!) \geq (k!)^\nu (k+1)^{2\lambda}. \tag{4}$$

Moreover, it is easy to see that the number of degree sequences $d_1, \ldots, d_\nu$ with $d_i \geq k$ and $2\mu = \sum_{i=1}^{\nu} d_i = k\nu + 2\lambda$ is at most

$$\binom{2\lambda + \nu - 1}{\nu - 1} \leq 2^{\nu+2\lambda}, \tag{5}$$

because if we let $d_i^* = d_i - k$, then $d_i^* \geq 0$ and $\sum_{i=1}^{\nu} d_i^* = 2\lambda$ and now we can generate all such sequences $(d_1^*, \ldots, d_\nu^*)$ by choosing numbers $t_1, \ldots, t_{\nu-1}$ from $\{1, \ldots, 2\lambda + \nu - 1\}$ and letting $d_i^* := t_i - t_{i-1} - 1$ (with $t_0 := 0$ and $t_\nu := 2\lambda + \nu$).

Thus, by combining (3), (4), and (5), the number of labelled graphs with ν vertices, minimum degree k and $\mu = k\nu/2 + \lambda$ edges is at most of order

$$\frac{2^{\nu+2\lambda+\mu}\mu^\mu}{\exp(\mu)(k!)^\nu(k+1)^{2\lambda}} \leq \frac{2^{\nu+2\lambda+\mu}\mu^\lambda(k\nu/2)^{k\nu/2}}{\exp(k\nu/2)(k!)^\nu(k+1)^{2\lambda}} = \frac{2^\nu(k\nu)^{k\nu/2}}{(k!)^\nu \exp(k\nu/2)} \left(\frac{8\mu}{(k+1)^2}\right)^\lambda$$

Now let $\eta = \frac{(k+1)^2}{8c} - \frac{k}{2}$. We first consider the case $\lambda \leq \eta\nu$. Here one can immediately check that

$$\frac{8\mu p}{(k+1)^2} \leq 1.$$

Thus, the expected number of induced subgraphs of order ν with minimum degree k and size $\leq k\nu/2 + \eta\nu$ in $G(n,p)$ is at most

$$\begin{aligned}
&\sum_{\lambda=0}^{\eta\nu} \binom{n}{\nu} \left(\frac{2^{2/k}k\nu}{(k!)^{2/k}\mathrm{e}}\right)^{k\nu/2} \left(\frac{8\mu}{(k+1)^2}\right)^\lambda p^{k\nu/2+\lambda} \\
&\leq \sum_{\lambda=0}^{\eta\nu} \binom{n}{\nu} \left(\frac{2^{2/k}k\nu p}{(k!)^{2/k}\mathrm{e}}\right)^{k\nu/2} \\
&\leq \eta\nu \left(\exp(H(\nu/n)) \left(\frac{2^{2/k}k(\nu/n)c}{(k!)^{2/k}\mathrm{e}}\right)^{\frac{k}{2}\cdot\frac{\nu}{n}}\right)^n,
\end{aligned} \tag{6}$$

where $H(x) = -x\ln(x) - (1-x)\ln(1-x)$ is the entropy function.

Now consider the case $\lambda \geq \eta\nu$ and let $N = \binom{\nu}{2} \leq \nu^2/2$. Then the probability that there is a k-core of order ν and size $\geq (\eta + k/2)\nu$ can be bounded as follows:

$$\begin{aligned}
\binom{n}{\nu}\binom{N}{(\eta+k/2)\nu} p^{(\eta+k/2)\nu} &\leq \exp(H(\nu/n)n) \left(\frac{\mathrm{e}p\nu}{k+2\eta}\right)^{(\eta+k/2)\nu} \\
&\leq \left(\exp(H(\nu/n)) \left(\frac{\mathrm{e}(\nu/n)c}{k+2\eta}\right)^{(\eta+k/2)\frac{\nu}{n}}\right)^n.
\end{aligned} \tag{7}$$

In order to prove that the expected running time of `ExactColour` is linear, we first consider the case $k = 3$, i.e. the 3-core. We have to show that

$$P(\text{the order of the 3-core equals } \nu) \cdot 1.3448^\nu \leq 1,$$

for all ν. By letting $x = \nu/n$ and using (6) and (7) it suffices to show that

$$H(x) + \frac{3x}{2} \ln\left(\frac{3.03 \cdot 2^{2/3} x}{6^{2/3} \mathrm{e}}\right) + x \ln(1.3448) \leq 0,$$

and (letting $\eta = 0.389$) that

$$H(x) + x\left(\frac{3}{2} + 0.389\right) \ln\left(\frac{1.01 \cdot \mathrm{e} \cdot x}{3 + 2 \cdot 0.389}\right) + x \ln(1.3448) \leq 0$$

for all $0 < x < 1$. This is easily verified, using elementary calculus. A similar argument (considering $k = 4$ in (6) and (7) and the running time of Lawler's algorithm instead of Beigel and Eppstein's) applies to Step 5 of `ExactColour`, whence we conclude that the expected running time of both Steps 3 and 5 of `ExactColour` is $O(1)$. □

We end this section by mentioning that no serious attempts have been made to optimize the constants involved. For example, an easy way to slightly enlarge the range of p for which the algorithm `ExactColour` has expected polynomial running time would be to use Eppstein's recent algorithm [5] for finding an optimal colouring instead of Lawler's – with $O(2.415^n)$ running time instead of $O(2.443^n)$. However it is clear that our analysis must fail at the latest when $p \approx 3.84/n$, as here the 3-core appears almost surely and has linear size. On the other hand it is known [1] that $G(n,p)$ remains 3-colourable a.s. at least until $p = 4.03/n$, so it would be interesting to design an algorithm that can handle the 3-core in expected polynomial time.

4 $O(\sqrt{np})$-approximation: Theorems 2 and 4

Throughout this section we shall assume that $p \gg \ln(n)^6/n$. Let $\omega = np$ and let $G \in G(n,p)$ be a random graph. Recall from Theorem 3 and Section 2, that `CoreColour` can be used as a $O(np)$-approximation algorithm for colouring $G(n,p)$ within expected polynomial time. In order to guarantee that this colouring is a good approximation, our algorithm needs an efficiently-computable lower bound on $\chi(G)$. To this end, we shall use the trivial inequality $\chi(G) \geq n/\alpha(G)$ and compute upper bounds on the independence number $\alpha(G)$.

For a set $S \subseteq V$ denote by $\overline{N}(S) := V \setminus (S \cup N(S))$ the non-neighbourhood of S. The following algorithm resembles an algorithm proposed in [17], except that it uses the Lovasz ϑ-function instead of the first eigenvalue of a suitably defined matrix. It is well-known that $\alpha(G) \leq \vartheta(G)$ for any graph G, and that $\vartheta(G)$ can be computed within polynomial time, using the ellipsoid algorithm [12]. For a thorough introduction to the ϑ-function the reader is referred to [12, 16].

Algorithm 12. `DecideMIS`(G)
Input: A graph G of order n.
Question: Is $\alpha(G) < 12\sqrt{n/p}$?
Output: Either "yes" or "no".

1. Let $a := 12(n/p)^{1/2}$ and $b := 25\ln(np)/p$.
2. Compute $\vartheta(G)$. If $\vartheta(G) < a$, then output "yes" and terminate.
3. Check whether there exists a subset S of V, $|S| = b$, such that $|\overline{N}(S)| \geq a - b$. If no such set exists, then output "yes" and terminate.
4. Check all sets of size a. If none of them is independent, then output "yes" and terminate. Otherwise, output "no".

Since Steps 3 and 4 of the above algorithm have super polynomial running time, it is clear that we have to bound the probability that we need to execute them, in other words we need to estimate expectation and concentration of $\vartheta(G(n,p))$.

Lemma 13. *Almost surely,* $\vartheta(G(n,p)) \leq 3(n/p)^{1/2}$.

Proof. The case that $p = \Omega(1)$ has previously been treated in [14]. Here we indicate how to adapt this argument for the case $\ln(n)^6/n \ll p \ll 1$. In order to bound the value of $\vartheta(G(n,p))$, we use a characterization of ϑ as an eigenvalue minimization problem (cf. [16, Sec. 6]). Given $G = G(n,p)$, let $M = (m_{ij})_{i,j \in V}$ be the symmetric $n \times n$ matrix with entries $m_{ij} = 1$, if $\{i,j\} \notin E(G)$ or $i = j$, and $m_{ij} = \frac{p-1}{p}$ otherwise. Let $\lambda_1(M) \geq \cdots \geq \lambda_n(M)$ denote the eigenvalues of M; then $\vartheta(G) \leq \lambda_1(M)$.

To bound $\lambda_1(M)$, we apply the techniques proposed in [9] to the matrix $A = (a_{ij})$, where $a_{ij} = pm_{ij}$. We have $\mathrm{E}(a_{ij}) = 0$, and $\sigma^2 = \mathrm{Var}(a_{ij}) \sim p$ $(i \neq j)$. (Since $\sigma = o(1)$, the result stated in [9] does not apply directly.) The eigenvalues of A are $\lambda_i(A) = p\lambda_i(M)$, $i = 1, \ldots, n$. Let us fix an even integer $k > 0$. We shall bound the trace of A^k, since then the estimate $\mathrm{Tr}(A^k) = \sum_{i=1}^n \lambda_i(A)^k \geq \lambda_1(A)^k$ entails a bound on $\lambda_1(A)$. As in [9], let

$$E_{n,k,\rho} = \sum_{\substack{i_0=1}}^n \cdots \sum_{i_k=1}^n \quad |\mathrm{E}(a_{i_0 i_1} \cdots a_{i_{k-1} i_k})| \\ |\{i_1, \ldots, i_k\}| = \rho$$

where $1 \leq \rho \leq k+1$, and put $E_{n,k} = \sum_{\rho=1}^{k+1} E_{n,k,\rho}$. It is straightforward to check that $E_{n,k} \geq \mathrm{Tr}(A^k) \geq \lambda_1(A)^k$. The counting argument given in [9, Sec. 3] shows that $E_{n,k,\rho} = 0$ if $\rho > 1 + k/2$, that

$$E_{n,k,1+k/2} = \frac{1}{1+k/2}\binom{k}{k/2} n(n-1)\cdots(n-k/2)\sigma^k,$$

and that $\sum_{\rho \leq k/2} E_{n,k,\rho} \leq O(k^6/n) E_{n,k,1+k/2}$. Let k be the largest even number $\leq (\sigma\sqrt{n})^{1/3}$, and put $v = 50n^{1/3}\sigma^{2/3}\ln n$. By Markov's inequality,

$$\begin{aligned} \mathrm{P}(\lambda_1(A) > 2\sigma\sqrt{n} + v) = \mathrm{P}(\lambda_1(A)^k > (2\sigma\sqrt{n} + v)^k) &\leq \frac{2E_{n,k,1+k/2}}{(2\sigma\sqrt{n}+v)^k} \\ &\leq 2n(2\sigma\sqrt{n}+v)^{-k}(2\sigma\sqrt{n})^k \leq 2n\exp\left(-\frac{kv}{2\sigma\sqrt{n}+v}\right). \end{aligned}$$

Since $p \gg \ln(n)^6/n$, we conclude that $\mathrm{P}(\lambda_1(A) > 3\sigma\sqrt{n}) \leq \exp\left(-\frac{kv}{3\sigma\sqrt{n}}\right) \leq n^{-10}$. Recalling that $\vartheta(G) \leq \lambda_1(M) = \lambda_1(A)/p$ completes the proof. □

Remark 14. The above method of bounding $\vartheta(G(n,p))$ breaks down if the edge probability p is too small. For instance, if $p = c/n$ for some constant c, then a.s. $\lambda_1(M) \gg n \geq \vartheta(G(n,p))$.

Lemma 15. *Let μ be a median of $\vartheta(G(n,p))$. Let $\xi \geq \max\{10, \sqrt{\mu}\}$. Then*

$$P(\vartheta(G(n,p)) \geq \mu + \xi) \leq 30 \exp(-\xi^2/(5\mu + 10\xi)).$$

Consequently, if $a \geq 10\sqrt{n/p}$, then

$$P(\vartheta(G(n,p)) \geq a) \leq \exp(-a/30). \tag{8}$$

Lemma 15 is proven in [4]. Its proof is based on Talagrand's inequality.

Remark 16. The algorithms in [17] are based on bounding the independence number of $G = G(n,p)$ via the eigenvalue $\lambda_1(M)$, where M is as in the proof of Lemma 13. In [17] it is show that $\lambda_1(M)$ is sufficiently sharply concentrated if $p \gg n^{-1/2}$. Our algorithm `DecideMIS` uses $\vartheta(G)$ instead of $\lambda_1(M)$, because the large deviation result in Lemma 15 also gives a sufficiently sharp bound in the sparse case $p \ll n^{-1/2}$.

We are now ready to prove that `DecideMIS` does what it is supposed to do.

Lemma 17. *For any graph G, `DecideMIS`(G) outputs the correct answer. If $p \gg \ln(n)^6/n$, then applied to $G(n,p)^+$, `DecideMIS` has polynomial expected running time. Moreover, the probability that `DecideMIS` answers "no" is at most* $\exp(-n)$.

Proof. It is straightforward to check that the answer given by `DecideMIS` is always correct. Indeed, as $\alpha(G) \leq \vartheta(G)$ this is trivially true if the algorithm terminates after Step 2, and also if it terminates after Step 4. So suppose that it terminates after Step 3 and assume for a contradiction that there does exist an independent set W of size at least a. But then consider any subset $S \subseteq W$ with $|S| = b$ and observe that $|W \setminus S| \geq a - b$ and $W \setminus S \subseteq \overline{N}(S)$.

It remains to prove that `DecideMIS` has polynomial expected running time. We make a few preliminary technical observations. First recall that $p \gg \ln^6(n)/n$ and that hence $\sqrt{np} \gg \ln^3(np)$. Thus, by inserting the definitions for a and b, it is easy to see that for any constant C and n sufficiently large

$$\frac{a}{C} - b \geq \frac{a}{C} - \ln(np)\ b = \frac{12\sqrt{np}/C - 25\ln^2(np)}{p} \geq \frac{11\sqrt{np}/C}{p} = \frac{11}{12}\frac{a}{C}. \tag{9}$$

Moreover, we have that

$$\binom{n}{a} \leq \left(\frac{en}{a}\right)^a \leq (np)^{a/2} = \exp\left(\ln(np)a/2\right), \tag{10}$$

$$\binom{n}{b} \leq \left(\frac{en}{b}\right)^b = \left(\frac{enp}{25\ln(np)}\right)^b \leq (np)^b = \exp\left(\ln(np)b\right). \tag{11}$$

Now we consider the expected running times of Steps 3 and 4. By (8) in Lemma 15, the expected time spent in Step 3 is bounded by

$$\exp(-a/30)\binom{n}{b}n \overset{(11)}{\leq} n\exp(-a/30+\ln(np)b) \overset{(9)}{=} O(n).$$

Moving on to Step 4, its expected running time is at most

$$\begin{aligned}\binom{n}{b}(1-p)^{b(a-b)}\binom{n}{a}\binom{a}{2} &\overset{(10,11)}{\leq} n^2\exp\big(\ln(np)a/2-pb(a-b)+\ln(np)b\big)\\ &\overset{(9)}{\leq} n^2\exp\big(\ln(np)a/2-25\ln(np)\frac{11}{12}a+\ln(np)b\big)\\ &\overset{(9)}{=} O(n^2).\end{aligned}$$

It remains to verify the last claim of Lemma 17. The probability that `DecideMIS` finds an independent set of size a is at most

$$\begin{aligned}\binom{n}{a}(1-p)^{\binom{a}{2}} &\overset{(10)}{\leq} \exp\big(\ln(np)a/2-pa^2/3\big)\\ &= \exp\big(n\ (-48+6\ln(np)/\sqrt{np})\ \big) \leq \exp(-n),\end{aligned}$$

as claimed. □

Having established correctness and running time of `DecideMIS`, it is straightforward to design approximation algorithms for the chromatic and the independence numbers as required by Theorems 2 and 4.

Algorithm 18. `ApproxColour`(G)
Input: A graph $G=(V,E)$.
Output: A colouring of G.

1. Run `CoreColour`$(G,10np)$. Let $\mathcal{C}$ be the resulting colouring of G.
2. Run `DecideMIS`(G). If $\alpha(G)<12\sqrt{n/p}$, then output $\mathcal{C}$ and terminate.
3. Otherwise colour G exactly in time $O(2.443^n)$ (cf. [22]).

Proof of Theorem 2. Since the expected running time of `ApproxColour`(G) is given by Lemma 17, the only thing we need to do is to check the approximation ratio achieved by `ApproxColour`(G) if it terminates after Step 2. Recall that `CoreColour`$(G,10np)$ needs at most $\max\{\chi(G),10np\}$ colours. Hence if `ApproxColour`(G) terminates after Step 2, then the approximation ratio is at most

$$\frac{10np}{\chi(G)} \leq \frac{10np\ \alpha(G)}{n} < \frac{10np\ 12\sqrt{n/p}}{n} = O(\sqrt{np}),$$

thereby proving the theorem. □

Remark 19. In [14] it is shown that for $p=\Omega(1)$ we a.s. have $\vartheta(G(n,p))=\Omega(\sqrt{n/p})$. Using similar arguments as in the proof of Lemma 13, one can extend this result to all $p \gg \ln(n)^6/n$. Therefore, one cannot hope for an approximation ratio considerably better than $O(\sqrt{np})$ using $n/\vartheta(G(n,p))$ as a lower bound on $\chi(G(n,p))$.

We conclude this section with the proof of Theorem 4 by adapting `ApproxColour` as follows. Instead of `CoreColour`, we use the greedy (colouring) algorithm and keep the largest colour class it produces. Then we run again `DecideMIS` and, finally, instead of using an exact graph colouring algorithm in Step 3, we use an exact algorithm for the maximum independent set problem.

Algorithm 20. `ApproxMIS`(G)
Input: A graph $G = (V, E)$.
Output: An independent set of G.

1. Run the greedy algorithm for graph colouring on input G. Let I be the largest resulting colour class. If $|I| < \ln(np)/(2p)$, then go to 3.
2. Run `DecideMIS`(G). If $\alpha(G) < 12\sqrt{n/p}$, then output I and terminate.
3. Otherwise, enumerate all subsets of V and output a maximum independent set.

Lemma 21. *The probability that the largest colour class produced by the greedy colouring algorithm is of size* $< \ln(np)/(2p)$ *is at most* 2^{-n}.

Sketch of proof. Use the argument given in [17] for the case $p \geq n^{\varepsilon - 1/2}$, with $\alpha_0 = \ln(np)/(2p)$ and $t = n/(2\alpha_0) = np/\ln(np)$. □
Proof of Theorem 4. Let us first prove that `ApproxMIS`(G) achieves the approximation ratio stated in the theorem. Suppose that the algorithm terminates after Step 2. Then the approximation ratio is at most

$$\frac{\alpha(G)}{|I|} < \frac{12\sqrt{n/p}}{\ln(np)/(2p)} = O(\sqrt{np}/\ln(np)).$$

If, on the other hand, the algorithm terminates after Step 3, then it has found a maximum independent set. As to the running time, the result follows from Lemma 21 together with Lemma 17. □

5 Deciding k-colourability: Theorems 5 and 6

In this section we apply the methods established so far to the problem of *deciding* k-colorability. First, let us consider the model $G(n,p)^-$.

Algorithm 22. `CoreDecide`(G)
Input: A graph $G = (V, E)$ and an integer k.

1. Run `CoreColour`(G, k). Let $\mathcal{C}$ be the resulting colouring of G.
2. If the number of colours used in $\mathcal{C}$ is at most k, output "G is k-colourable". Otherwise, output "G is not k-colourable".

Proof of Theorem 5. Let $G_0 = G(n,p)$, $p \leq k/(10n)$, and let $G \in \mathcal{I}(G_0)^-$ be the instance chosen by the adversary. Because the adversary can only remove edges, the k-core of G is contained in the k-core of G_0. Thus, by our assumption on p, the argument given in the proof of Lemma 10 shows that the expected

running time of `CoreDecide` is linear. Moreover, since `CoreColour`(G,k) uses at most $\max\{\chi(G),k\}$ colours, we must have $\chi(G)>k$ whenever the output is "not k-colourable", and thus `CoreDecide` is always correct. □

Coming to $G(n,p)^+$, let us now assume that $np \geq \max\{200k^2, \ln(n)^7\}$. In this range we almost surely have $\chi(G(n,p)) > k$.

Algorithm 23. `Decide`(G)
Input: A graph $G=(V,E)$.

1. Run `DecideMIS`(G). If $\alpha(G) < 12\sqrt{n/p}$, then output "G is not k-colourable" and terminate.
2. Compute the chromatic number of G exactly in time $O(2.443^n)$ [22].

Proof of Theorem 6. By Lemma 17, `Decide` has polynomial expected running time. If `Decide`(G) terminates after Step 1, then

$$\chi(G) \geq \frac{n}{\alpha(G)} \geq \frac{n}{12\sqrt{n/p}} = \frac{\sqrt{np}}{12} > k,$$

as claimed. □

6 An application to k-SAT: Theorem 7

Throughout this section, we assume that $k \geq 4$ is an even integer. In order to certify unsatisfiability of the random k-SAT formula $\mathcal{F}(n,k,m)$, we exploit the following connection between k-SAT and the maximum independent set problem established in [10]. Let $V = \{1,\ldots,n\}^{k/2}$. Given any k-SAT formula F over n propositional variables $x_1,\ldots,x_n$, we can define two graphs $G_F=(V,E_F)$, $G'_F=(V,E'_F)$ as follows. We let

$$E_F = \{\{v,w\} |\ x_{v_1} \vee \cdots \vee x_{v_{k/2}} \vee x_{w_1} \vee \cdots \vee x_{w_{k/2}} \text{occurs in } F\}$$

and

$$E'_F = \{\{v,w\} |\ \neg x_{v_1} \vee \cdots \vee \neg x_{v_{k/2}} \vee \neg x_{w_1} \vee \cdots \vee \neg x_{w_{k/2}} \text{occurs in } F\}.$$

Lemma 24. *[10] If F is satisfiable, then* $\max\{\alpha(G_F),\alpha(G'_F)\} \geq 2^{-k/2}n^{k/2}$.

In order to apply the methods of Section 4 to the problem of deciding whether F is satisfiable, we make use of the following lemma.

Lemma 25. *Let $F=\mathcal{F}(n,k,m)$ be a random k-SAT formula.*

1. *Let $\nu = n^{k/2}$. Conditioned on $|E_F|=\mu$, the graph G_F is uniformly distributed; i.e. $G_F = G_{\nu,\mu}$.*
2. *Let $\varepsilon > 0$. Suppose that $2^k n^{k/2} \leq m \leq n^{k-1}$. Then with probability at least $1-\exp(-\Omega(m))$ we have $|E(G_F)| \geq (1-\varepsilon)2^{-k}m$.*

Similar statements hold for G'_F.

Proof. The first part of the lemma is proven in [10]. We shall prove that with probability $\geq 1 - \exp(-\Omega(m))$ the graph G_F enjoys the following three properties. (The bounds on the probabilities of the first two events are implicit in [10].) We call a clause *all-positve* if it doesn't contain any negated variables.

1. The number $P(F)$ of all-positive clauses in F is at least $(1-\varepsilon)2^{-k}m$.
2. The number $L(F)$ of all-positive clauses of type

$$x_1 \vee \cdots \vee x_{k/2} \vee x_1 \vee \cdots \vee x_{k/2}$$

   that occur in F is $\leq \varepsilon 2^{-k}m$.
3. Let $F = \alpha_1 \wedge \cdots \wedge \alpha_m$. Given literals $l_1, \ldots, l_k$, put

$$h(l_1 \vee \cdots \vee l_k) = \{l_1 \vee \cdots \vee l_{k/2}, l_{k/2+1} \vee \cdots \vee l_k\}.$$

   Then $M(F) = |\{i \in \{1,\ldots,m\} |\ h(\alpha_i) = h(\alpha_j) \text{ for some } i \neq j\}| < \varepsilon 2^{-k}m$.

Since $|E_F| \geq P(F) - L(F) - M(F)$, the lemma is a consequence of 1.–3. above.

In order to prove that 1. holds with sufficiently high probability, note that the number of all-positive clauses is binomially distributed with expectation $2^{-k}m$. By Chernoff bounds, $\mathrm{P}(P(F) < (1-\varepsilon)2^{-k}m) \leq \exp(-\varepsilon^2 2^{-k-1}m)$. Similarly, since the number of clauses as in 2. is binomially distributed with expectation $m(2n)^{-k/2} \leq \varepsilon 2^{-k-1}m$, we conclude that

$$\mathrm{P}(L(F) > \mathrm{E}(L(F)) + \varepsilon 2^{-k-1}m) \leq \exp(-\varepsilon^2 2^{-k-2}m).$$

The probability that 3. is violated is most easily bounded using Talagrand's inequality. Indeed, we may consider $\mathcal{F}(n,k,m)$ as a product space $\Lambda_1 \times \cdots \times \Lambda_m$, where Λ_i is a random clause, $i = 1,\ldots,m$. First, let us estimate the expectation of $M(F)$. Let $F = \alpha_1 \wedge \cdots \wedge \alpha_m$. If $i \neq j$ are fixed, then the probability that $h(\alpha_i) = h(\alpha_j)$ is $2n^{-k}$. Hence, by our assumption $m \leq n^{k-1}$, $\mathrm{E}(M(F)) \leq 2m^2 n^{-k} \leq 2m/n < \varepsilon 2^{-k-2}m$. In order to apply Talagrand's inequality, we observe that if $F_1, F_2 \in \mathcal{F}(n,k,m)$ differ only in the jth clause, then $|M(F_1) - M(F_2)| \leq 2$. Furthermore, let $r > 0$. Suppose that $F = \alpha_1 \wedge \cdots \wedge \alpha_m \in \mathcal{F}(n,k,m)$ satisfies $M(F) \geq r$, and let

$$J = \{i \in \{1,\ldots,m\} |\ \text{there exists } j \neq i \text{ such that } h(\alpha_i) = h(\alpha_j)\}.$$

Then there exists $J_0 \subset J$, $|J_0| \leq r+1$, such that the following condition holds. For any $F' = \beta_1 \wedge \cdots \wedge \beta_m \in \mathcal{F}(n,k,m)$, such that $\beta_j = \alpha_j$ for all $j \in J_0$, we have $M(F') \geq r$. Put $\psi(r) = 4(r+1)$. Then Talagrand's inequality (in the version [13, p. 40]) yields

$$\mathrm{P}\Big(M(F) > 2\mathrm{E}(M(F)) + t\Big) \leq 2\exp\left(-\frac{t^2}{4\psi(2\mathrm{E}(M(F)) + t)}\right).$$

Letting $t = \varepsilon 2^{-k-1}m$ entails $\mathrm{P}(M(F) > \varepsilon 2^{-k}m) \leq \exp(-\varepsilon 2^{-k}m/100)$. □

In [10] a similar statement as Lemma 25 is established. The difference is that in order to obtain a polynomial expected running time, we need an exponentially small probability in part 2 of the lemma. Our algorithm for testing satisfiability of random k-SAT formulas can now be stated as follows.

Algorithm 26. `DecideSAT`(F)
Input: A k-SAT formula F over $x_1, \ldots, x_n$.
Output: Either a satisfying assignment of F or "unsatisfiable".

1. Construct the graphs G_F and G'_F. If `DecideMIS`(G_F) and `DecideMIS`(G'_F) (with $p = m(2n)^{-k}$) both answer "yes", then terminate with output "unsatisfiable".
2. Enumerate all 2^n assignments of $x_1, \ldots, x_n$. If an assignment $\hat{x}_1, \ldots, \hat{x}_n$ is found that satisfies F, then terminate with output $\hat{x}_1, \ldots, \hat{x}_n$. Otherwise, output "unsatisfiable".

Proof of Theorem 7. We may and shall assume that $m = \lceil \ln(n)^7 n^{k/2} \rceil$. Clearly, `DecideSAT`$(F)$ outputs "unsatisfiable" if F is unsatisfiable. If F is satisfiable, then by Lemma 24 we have $\max\{\alpha(G_F), \alpha(G'_F)\} \geq 2^{-k/2} n^{k/2}$. Since $\sqrt{\frac{n^{k/2}}{p}} \ll n^{k/2}$, either `DecideMIS`$(G_F)$ or `DecideMIS`(G'_F) will find out that

$$\max\{\alpha(G_F), \alpha(G'_F)\} > 12\sqrt{n^{k/2}/p}.$$

Hence, `DecideSAT` runs Step 2 and finds a satisfying assignment of F.

As for the running time, let F_0 denote the random k-SAT formula from which $F = \mathcal{F}(n,k,m)^+$ has been constructed. Clearly, the graphs G_F and G'_F can be constructed in polynomial time (w.r.t. $n + \ell(F)$). As the time used by Step 1 for executing `DecideMIS` is bounded by $\exp(n^{k/2})$, and because Step 2 consumes at most $\ell(F)^{O(1)} \exp(n)$ time, we may assume that $\min\{|E(G_{F_0})|, |E(G'_{F_0})|\} \geq 2^{-k-1}m$, by the second part of Lemma 25. By the first part of Lemma 25, the graphs G_F and G'_F both are have the following structure. First, a random graph $G(\nu, \mu)$ is chosen with $\nu = n^{k/2}$, $\mu = 2^{-k-1}m$. Then, an adversary adds arbitrary edges. Our choice of p implies that $\mathrm{P}(|E(G(\nu,p))| = \mu) \geq n^{-k}$. Therefore, applied to G_F and G'_F, `DecideMIS` has polynomial expected running time. Consequently, on average `DecideSAT`$(\mathcal{F}(n,k,m)^+)$ spends polynomial time executing Step 1. Since the probability that $\alpha(G(n^{k/2},p)^+) > 12(n^{k/2}/p)^{1/2}$ is $\leq \exp(-n)$, and because we can check in polynomial time whether an assignment of $x_1, \ldots, x_n$ is satisfying, the expected time spent executing Step 2 is polynomial. □

Acknowledgement. We are grateful to A. Goerdt and M. Krivelevich for helpful discussions.

References

1. Achlioptas, D., Moore, C.: Almost All Graphs of Degree 4 are 3-Colorable, Proc. Symposium on the Theory of Computing (STOC) 2002.

2. Beigel, R., Eppstein, D.: 3-coloring in time $O(1.3446^n)$: a no-MIS algorithm, Proc. 36th. IEEE Symp. Found. of Comp. Sci. (1995) 444-453
3. Bollobás, B.: Random graphs, 2nd edition, Cambridge University Press 2001
4. Coja-Oghlan, A.: Finding large independent sets in expected polynomial time. To appear in Proc. STACS 2003 (available from http://www.informatik.hu-berlin.de/~coja/)
5. Eppstein, D.: Small maximal independent sets and faster exact graph coloring. To appear in J. Graph Algorithms and Applications
6. Feige, U., Kilian, J.: Zero knowledge and the chromatic number. Proc. 11. IEEE Conf. Comput. Complexity (1996) 278–287
7. Feige, U., Kilian, J.: Heuristics for semirandom graph problems. J. Comput. and System Sci. **63** (2001) 639–671
8. Friedgut, E.: Necessary and sufficient conditions for sharp thresholds of graph properties and the k-SAT problem. J. of the AMS **12** (1999) 1017–1054
9. Füredi, Z., Komloś, J.: The eigenvalues of random symmetric matrices, Combinatorica **1** (1981) 233–241
10. Goerdt, A., Krivelevich, M.: Efficient recognition of random unsatisfiable k-SAT instances by spectral methods. Proc. 18th Int. Symp. on Theoret. Aspects of Comp. Sci. (STACS'2001) Springer LNCS **2010** 294-304.
11. Grimmett, G., McDiarmid, C.: On colouring random graphs. Math. Proc. Cam. Phil. Soc **77** (1975) 313–324
12. Grötschel, M., Lovász, L., Schrijver, A.: Geometric algorithms and combinatorial optimization. Springer 1988
13. Janson, S., Łuczak, T., Ruciński, A.: Random Graphs. Wiley 2000
14. Juhász, F.: The asymptotic behaviour of Lovász ϑ function for random graphs, Combinatorica **2** (1982) 269–270
15. Karp, R.: The probabilistic analysis of combinatorial optimization algorithms. Proc. Int. Congress of Mathematicians (1984) 1601–1609.
16. Knuth, D.: The sandwich theorem, Electron. J. Combin. **1** (1994)
17. Krivelevich, M., Vu, V.H.: Approximating the independence number and the chromatic number in expected polynomial time. J. of Combinatorial Optimization **6** (2002) 143–155
18. Krivelevich, M.: Deciding k-colorability in expected polynomial time, Information Processing Letters **81** (2002) 1–6
19. Krivelevich, M.: Coloring random graphs – an algorithmic perspective, Proc. 2nd Coll. on Mathematics and Computer Science, B. Chauvin et al. Eds., Birkhauser, Basel (2002) 175-195.
20. Krivelevich, M., Sudakov, B.: Coloring random graphs. Informat. Proc. Letters **67** (1998) 71–74
21. Kučera, L.: The greedy coloring is a bad probabilistic algorithm. J. Algorithms **12** (1991) 674–684
22. Lawler, E.L.: A note on the complexity of the chromatic number problem, Information Processing Letters **5** (1976) 66–67
23. Pittel, B., Spencer, J., Wormald, N.: Sudden emergence of a giant k-core in a random graph. JCTB **67** (1996) 111–151