Average-case complexity of backtrack search for coloring sparse random graphs

Zoltán Ádám Mann and Anikó Szajkó

This paper was published in *Journal of Computer and System Sciences* 79:(8) pp. 1287-1301, 2013.

Average-case complexity of backtrack search for coloring sparse random graphs[☆]

Zoltán Ádám Mann[a,*], Anikó Szajkó[a]

[a] Department of Computer Science and Information Theory, Budapest University of Technology and Economics, Magyar tudósok körútja 2, 1117 Budapest, Hungary

Abstract

We investigate asymptotically the expected number of steps taken by backtrack search for k-coloring random graphs $G_{n,p(n)}$ or proving non-k-colorability, where $p(n)$ is an arbitrary sequence tending to 0, and k is constant. Contrary to the case of constant p, where the expected runtime is known to be $O(1)$, we prove that here the expected runtime tends to infinity. We establish how the asymptotic behaviour of the expected number of steps depends on the sequence $p(n)$. In particular, for $p(n) = d/n$, where d is a constant, the runtime is always exponential, but it can be also polynomial if $p(n)$ decreases sufficiently slowly, e.g. for $p(n) = 1/\ln n$.

Keywords: graph coloring, average-case complexity, search tree, random graphs, backtrack

1. Introduction

Graph coloring is an important combinatorial optimization problem with many applications in engineering, such as register allocation, frequency assignment, pattern matching and scheduling [1, 2, 3]. Accordingly, graph coloring has been the subject of intensive research.

[☆] A preliminary version of this paper was presented at the 7th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications.

[*] Corresponding author. Phone: +36 20 939 8842, Fax: +36 1 463 3157.

Email addresses: zoltan.mann@gmail.com (Zoltán Ádám Mann), szajko.aniko@gmail.com (Anikó Szajkó)

One of the most important tools to mathematically investigate graph coloring is to study the coloring of random graphs. Usually, the $G_{n,p}$ random graph model is used [4], meaning that the graph has n vertices, and each pair of vertices is connected by an edge with probability p independently from each other (we will refer to p as *edge density*). Many remarkable results and mathematical methods came into existence on random graphs concerning graph coloring and many other graph-theoretic problems; see for example the extensive surveys in [5] and [6].

As a particular result of the research of the last couple of decades, the chromatic number of random graphs both with constant and varying edge density were estimated [7, 8, 9, 10, 11]. In 2004, Achlioptas and Naor [12] succeeded to almost exactly determine the chromatic number of random graphs with edge density function $p(n) = d/n$, when the size of the graph tends to infinity.

Graph coloring is NP-hard. The most widely used exact algorithm for graph coloring is the backtrack search algorithm. In this paper, we deal with a version of backtrack search that solves the #COL problem: that is, it counts the number of solutions. (For k-colorable graphs, this takes longer than merely deciding colorability, since we cannot stop after finding the first solution. However, for non-k-colorable graphs, the same amount of time is needed for solving the decision problem and the counting problem.)

Obviously, the worst-case complexity of this algorithm is exponential in the size of the graph. However, in practice, the backtrack algorithm works quite efficiently even for relatively large graphs. In fact, Wilf proved in 1984 the surprising result that the expected runtime of the backtrack algorithm is bounded even if the size of the graph tends to infinity [13]. That is, the average-case complexity of this algorithm is $O(1)$. Later, Bender and Wilf provided a more detailed analysis of the asymptotic distribution of the algorithm's runtime [14]. In our recent research, we refined the results of Bender and Wilf: with detailed examinations, we can quite precisely predict the expected runtime of the backtrack algorithm for a random graph, as a function of the number of vertices, the number of colors, and the edge density [15, 16].

However, the above results apply only to random graphs where the edge density p is constant. Note that such graphs are with high probability very dense with $\Theta(n^2)$ edges. On the other hand, sparse graphs are more common in practice [17]. To accommodate this fact in the $G_{n,p}$ model, the edge density should rather be a function $p = p(n)$ that decreases with increasing n and tends to 0 when $n \to \infty$. Therefore, in this paper, we investigate the asymptotic behavior of the expected runtime of the backtrack algorithm in cases of different such $p(n)$ functions. Previous work on coloring sparse graphs concentrated on the $p(n) = d/n$ case; the main novelty of our paper is that it applies to any $p(n)$ sequence with $p(n) \to 0$.

In order to use a machine-independent measure of complexity, we estimate the expected number of visited nodes in the algorithm's search tree.

1.1. Results

Our main results describe the asymptotic behaviour of the average-case complexity of the backtrack algorithm on $G_{n,p}$ graphs for any $p(n) \to 0$, both in a qualitative and quantitative way. The qualitative result is as follows:

Theorem 1. *Let the number of available colors k be constant, and $p = p(n)$ be any sequence between 0 and 1, tending to 0. Then, the expected number of visited nodes in the backtrack algorithm's search tree tends to infinity when $n \to \infty$.*

Although this theorem is not hard to prove, it is interesting because it is in clear contrast to Wilf's theorem [13] for constant p values: however slowly $p(n)$ tends to 0, if it does, this makes the algorithm's average-case complexity divergent.

On the other hand, as our next theorem shows, the rate by which $p(n)$ tends to 0 does have significant impact on how quickly the expected number of visited nodes in the algorithm's search tree diverges:

Theorem 2. *Let the number of available colors k be constant, and $p = p(n)$ be any sequence between 0 and 1, tending to 0. Let $\mathbb{E}(Y)$ denote the expected* *number of visited nodes in the algorithm's search tree.*

(1) If $\exists\varepsilon > 0$ *such that, for all large enough* n*,* $np(n) > k\ln k + \varepsilon$*, then*

$$\mathbb{E}(Y) = \Theta\left(\sqrt{\frac{1}{p(n)}}\exp\left(\frac{k(\ln k)^2}{2p(n)}\right)\right) = \Theta\left(\sqrt{\frac{1}{p(n)}}\cdot c^{1/p(n)}\right),$$

where $c = k^{\frac{k\ln k}{2}}$.

(2) If $\exists\varepsilon > 0$ *such that, for all large enough* n*,* $np(n) < k\ln k - \varepsilon$*, then*

$$\mathbb{E}(Y) = \Theta\left(\exp\left(\left(\ln k + \frac{n\ln(1-p(n))}{2k}\right)n\right)\right) = \Theta\left(k^n(1-p(n))^{\frac{n^2}{2k}}\right).$$

This theorem gives an almost complete quantitative characterization of the average-case complexity of the algorithm.

It should be noted that $\mathbb{E}(Y)$ is invariably exponential in the second case. This can be seen as follows: the coefficient of n in the exponent is

$$\ln k + \frac{n\ln(1-p(n))}{2k} = \ln k - \frac{np(n)}{2k}\cdot\frac{-\ln(1-p(n))}{p(n)} >$$
$$> \ln k - \left(\frac{\ln k}{2} - \frac{\varepsilon}{2k}\right)\cdot\frac{-\ln(1-p(n))}{p(n)} > \frac{\ln k}{2}$$

for all large enough n, because $\frac{-\ln(1-p(n))}{p(n)} \to 1$. To sum up: in the second case,

$$\mathbb{E}(Y) = \Omega\left(\exp\left(\frac{\ln k}{2}n\right)\right) = \Omega\left(\left(\sqrt{k}\right)^n\right).$$

In the first case, the formula can be either polynomial or super-polynomial: e.g., it is polynomial for $p(n) = 1/\ln n$, but super-polynomial for $p(n) = 1/\sqrt{n}$. That is, although the algorithm's average-case complexity is definitely divergent if $\lim_{n\to\infty} p(n) = 0$, it can still be polynomial in n, if the convergence of $p(n)$ to 0 is sufficiently slow. Actually, it can even be sub-linear, e.g. for $p(n) = 1/\ln\ln n$.

The proofs rely on the technique that we developed in [15, 16] for estimating the number of visited nodes on level t of the search tree. From here, the way to the desired theorems is largely analytical.

1.2. Paper organization

We start by describing previous, related work in Section 2. In Section 3, we introduce the necessary definitions and notations, followed by the recapitulation of our previous results in Section 4 that we will be using later on. Section 5 contains our main results: the proofs of Theorems 1 and 2. Section 6 contains a discussion on some important special cases of the theorems and how they relate to previous results in the literature. We present some numerical experiments in Section 7, and finally, Section 8 concludes the paper.

2. Previous work

Because of its importance, the study of the complexity of graph coloring started already in the early 1970s. In fact, graph coloring was one of the 21 combinatorial problems whose NP-completeness was shown by Karp in his seminal 1972 paper [18]. Afterwards, researchers' attention turned towards approximation algorithms, but it turned out quickly that approximating the chromatic number is a hard problem. An early result of Garey and Johnson showed that no polynomial-time approximation algorithm with an approximation ratio smaller than 2 can exist, unless P=NP [19]. More recently, it was shown that – under standard assumptions of complexity theory – not even an $O(n^{1-\varepsilon})$ approximation can exist for any $\varepsilon > 0$ [20, 21].

Also starting with the 1970s, different heuristic and exact algorithms were developed for the graph coloring problem (see e.g. [22, 23, 24]). The proposed exact algorithms mostly used some form of backtrack search to guarantee a complete search while also being able to prune potentially large parts of the search space.

With the availability of practical graph coloring algorithms implemented as computer programs, researchers started to gain empirical experience with graph coloring in practice [24, 25, 26, 27]. These empirical investigations lead to the discovery of some fascinating phenomena in the average-case and typical-case complexity of the backtrack algorithm for graph coloring. It turned out that, in many cases, graph coloring is actually quite easy even for quite large graphs. More precisely, graph coloring – like many hard combinatorial problems – exhibits a phase transition phenomenon with an accompanying easy-hard-easy pattern [25, 28, 29, 26]. Briefly, this means that, given k colors, for small values of the edge density (under-constrained case), almost all graphs are k-colorable. When the edge density increases, the ratio of k-colorable graphs abruptly drops from almost 1 to almost 0 (phase transition). After this critical region, almost all graphs are non-k-colorable (over-constrained case). In the under-constrained case, coloring is easy: even the simplest heuristics usually find a proper coloring [30, 24]. In the over-constrained case, it is easy for backtracking algorithms to prove uncolorability because they quickly reach contradiction [31]. The hardest instances lie in the critical region [25].

These empirical results also spawned mathematical research to explain and prove in a rigorous way the above characteristics of the average-case complexity of the backtrack algorithm for graph coloring. Wilf proved in 1984 the exciting result that the average-case complexity of the backtrack algorithm is actually $O(1)$ [13]. In order to derive this result, he considered the expected number of visited nodes of the search tree when the input graph is taken from $G_{n,p}$. Further elaborating this result, Bender and Wilf gave estimations on the asymptotic behavior of the expected number of visited nodes of the search tree [14]. In the present paper, we use the same model as Bender and Wilf. However, it should be noted that Wilf's result as well as the analysis of Bender and Wilf only apply to dense graphs with a fixed value of p.

A different approach was taken by Turner to show why graph coloring is easy for many graphs [30]. He analyzed the behavior of some simple heuristics on k-colorable graphs, and proved that they can find a coloring with high probability (*whp* for short, meaning that the probability tends to 1 as n goes to infinity). In terms of the backtrack algorithm, this means that it would find a solution whp without backtracking. Note however, that Turner's result only applies if the number of available colors is small, i.e. $k = O(\log n)$, and p is fixed.

In a similar way, the recent paper of Coja-Oghlan, Krivelevich and Vilenchik also focuses on k-colorable graphs and investigates why their coloring tends to be easy [32]. They show that all valid k-colorings lie whp in a single "cluster", agreeing on the color of most vertices. What is more important from our point of view is that they also prove that such graphs can be colored whp in polynomial time. Note that their approach works for k-colorable graphs with n vertices and $m = dn$ edges, where d is sufficiently large. (In the $G_{n,p}$ model, this would correspond to the $p \approx 2d/n$ case.)

Jia and Moore also analyzed the $p = d/n$ case, but for small values of d and with a different goal [33]. They aimed at explaining the phenomenon of heavy tails, i.e. the surprisingly high probability of extremely low or extremely high algorithm runtimes. In particular, they proved that for appropriate values of d, both the probability of 0 backtracks and the probability of an exponential number of backtracks are positive.

Because of the phenomenon of heavy-tailed runtime distributions, it was suggested in the AI community to boost practical algorithm performance by randomization and frequent restarts [34, 35]. That is, if a run of the algorithm takes long, it should be restarted in the hope that the new run will take a more lucky path in the search tree and finish sooner. In fact, this strategy works surprisingly well for many NP-hard problems, including Boolean satisfiability and other constraint satisfaction problems.

The analysis of the chromatic number of random graphs was first suggested in the seminal 1960 paper of Erdős and Rényi [4]. Subsequent work of Grimmett and McDiarmid [36], Bollobás [8], and Luczak [9], lead to an understanding of the order of magnitude of the expected chromatic number of random graphs. Through the recent work of Shamir and Spencer [11], Luczak [10], Alon and Krivelevich [7], and Achlioptas and Naor [12], we can determine almost exactly the expected chromatic number of a random graph in the limit: the expected chromatic number of a random graph is whp one of two possible values. Specifically, if k_d denotes the smallest integer k with $d < 2k \log k$, then the chromatic number of a $G_{n,d/n}$ graph is with high probability either k_d or $k_d + 1$.

Upper bounds on the chromatic number were often proven in an algorithmic way, by showing that a simple algorithm will succeed in coloring the graph with high probability. Examples include the GIC heuristic that works by determining independent sets greedily and using them as color classes [36, 37, 38], the greedy list-coloring algorithm k-GL that selects a vertex with minimum number of available colors [39], and its refinement in which ties are broken in such a way that vertices with more uncolored neighbours are selected with higher probability [40]. A possible interpretation of these results is that, for small constraint densities, the solution can be found without backtracking with positive probability [33]. In a similar way, Turner proved the No-Choice algorithm – which, after coloring a clique, colors only vertices whose color is uniquely determined – to find a coloring for almost all k-colorable graphs, if $k = O(\log n)$ and p is fixed.

Algorithmic aspects have been studied besides random graphs with constant p, also for sparse graphs with $p = p(n) = d/n$. Examples beyond the ones already mentioned include the result of Pittel and Weishaar, who proved that the greedy algorithm for coloring a random graph $G_{n,d/n}$ requires only $O(\log \log n)$ colors, and the number of used colors will be one of two possible numbers [41]. Coja-Oghlan and Taraz presented an expected-linear-time algorithm for coloring a random graph $G_{n,d/n}$ with $d \leq 1.01$ [42]. Later, Sommer proved that the algorithm's expected running time is actually linear for all $d \leq 1.33$ [43]. The algorithm of Shamir and Upfal works for graphs with mean degree $d = d(n)$ and uses not more than $d(n)/\log d(n)$ colors, which is approximately twice the chromatic number [37].

Interestingly, methods from theoretical physics (more specifically, statistical mechanics) have also been applied successfully to study the asymptotic expected performance of backtrack algorithms. After first results on the satisfiability problem [44], this machinery was also used to study the 3-coloring problem. In particular, Monasson and co-workers modeled the solution process of backtrack search with an out-of-equilibrium (multi-dimensional) surface growth problem [45, 31]. By solving the resulting partial differential equation, an estimation of the backtrack algorithm's runtime can be obtained that is fairly close to the empirical results for relatively dense graphs. Although these results are not rigorous, Monasson later developed a method based on generating functions, with which similar results were achieved in a rigorous way [46]. In particular, it was established that the expected runtime of the backtrack algorithm for 3-coloring a random graph from $G_{n,d/n}$, for large enough d, is $\exp(cn + o(n))$, where c depends only on d.

In contrast to most previous research, our focus is on graphs from $G_{n,p}$, where $p = p(n)$ is *any* sequence tending to 0. Our aim is to analyze how the asymptotic behavior of the expected number of visited nodes of the search tree depends on how quickly $p(n)$ converges to 0.

3. Preliminaries

We consider the counting version of the graph coloring problem, in which the input consists of an undirected graph $G = (V, E)$ and a number k, and the task is to count the number of possibilities for coloring the vertices of G with k colors such that adjacent vertices are not assigned the same color. The input graph is a random graph taken from $G_{n,p}$, i.e. it has n vertices and each pair of vertices is connected by an edge with probability p independently from each other. The vertices of the graph will be denoted by $v_1, \ldots, v_n$, the colors by $1, \ldots, k$. A *coloring* assigns a color to each vertex; a *partial coloring* assigns a color to some of the vertices.

The color that the (partial) coloring w assigns to vertex v is denoted by $w(v)$. If w does not assign a color to v, then $w(v)$ is undefined.

A (partial) coloring is *invalid* if there is a pair of adjacent vertices with the same color, otherwise the (partial) coloring is *valid*.

The backtrack algorithm considers partial colorings. It starts with the empty partial coloring, in which no vertex has a color. This is the root – that is, the single node[1] on level 0 – of the *complete search tree*. Level t of the complete search tree contains the k^t possible partial colorings of $v_1, \ldots, v_t$. The complete search tree, denoted by T, has $n+1$ levels $(0, 1, \ldots, n)$, the last level containing the k^n colorings of the graph. For simplicity of notation, we use $w \in T$ to denote

[1] In order to avoid misunderstandings, we use the term 'vertex' in the case of the input graph and the term 'node' in the case of the search tree.

that the partial coloring w is a node of the complete search tree. Furthermore, let T_t denote the set of partial colorings on level t of T. If $t < n$ and $w \in T_t$, then w has k children in the complete search tree: those partial colorings of $v_1, \ldots, v_{t+1}$ that assign to the first t vertices the same colors as w.

In each partial coloring w, the backtrack algorithm considers the children of w and visits only those that are valid. Invalid children are not visited, and this way, the whole subtree under an invalid child of the current node is pruned. This is correct because all nodes in such a subtree are also certainly invalid. The algorithm proceeds in a depth-first-search manner until all nodes of the search tree are visited or pruned.

T depends only on n and k, not on the specific input graph. However, the algorithm visits only a subset of the nodes of T, depending on which vertices of G are actually connected. The number of actually visited nodes of T will be used to measure the complexity of the algorithm on the given problem instance. Moreover, the number of actually visited nodes on the nth level of T yields the number of solutions, i.e. the number of valid k-colorings.

Of course, this is a simplified algorithm model. In particular, we assume that branching is performed according to a statically determined order of the vertices. This greatly simplifies the analysis of the algorithm's performance.

4. Expected number of visited nodes of the search tree

Let Y be the number of visited nodes in T, Y_t the number of visited nodes in T_t, and S the number of solutions, i.e. the number of valid k-colorings. Y, Y_t, and S are random variables, the value of which depends on the input graph.

In [16], we proved lower and upper bounds on the expected value of these quantities. Since these bounds play a vital role in deriving our current results, we repeat them here.

Proposition 3. $k^t(1-p)^{\frac{t^2-t}{2k}} \le \mathbb{E}(Y_t) \le k^t(1-p)^{\frac{t^2-kt}{2k}}$.

Proof. For $w \in T_t$, let

$$Q(w) := \big\{\{x, y\} : x, y \in \{v_1, \ldots, v_t\}, x \neq y, w(x) = w(y)\big\}$$

be the set of pairs of vertices with identical colors, and let $q(w) := |Q(w)|$. Clearly, w is valid if and only if, for all $\{x, y\} \in Q(w)$, x and y are not adjacent. It follows that the probability of w being valid is $(1-p)^{q(w)}$, and thus the expected number of visited nodes of T_t is:

$$\mathbb{E}(Y_t) = \sum_{w \in T_t} (1-p)^{q(w)}.$$

In the following, we denote by $s(w,i)$ (or simply s_i if it is clear which partial coloring is considered) the number of vertices of G that are assigned color i in the partial coloring w.

We first aim at proving the lower bound.

Since the role of the colors is symmetric, it follows that

$$\sum_{w \in T_t} q(w) \quad = \quad \sum_{w \in T_t} \sum_{i=1}^{k} \binom{s(w,i)}{2} = \sum_{i=1}^{k} \sum_{w \in T_t} \binom{s(w,i)}{2} = k \sum_{w \in T_t} \binom{s(w,1)}{2}.$$

In order to compute this sum, we should examine for how many $w \in T_t$ we have $s(w,1) = j$. In other words, how many colorings exist for the first t vertices, in which exactly j vertices receive color 1. Since the j vertices can be chosen in $\binom{t}{j}$ ways and the remaining $t-j$ vertices must receive a color from the remaining $k-1$ colors, there are $\binom{t}{j}(k-1)^{t-j}$ such partial colorings. It can be assumed that $j \geq 2$ because otherwise the contribution of color class 1 to $q(w)$ is 0. Using $\binom{j}{2}\binom{t}{j} = \binom{t}{2}\binom{t-2}{j-2}$:

$$\begin{aligned} \sum_{w \in T_t} q(w) \quad &= \quad k \sum_{j=2}^{t} \binom{j}{2}\binom{t}{j}(k-1)^{t-j} = k\binom{t}{2} \sum_{j=2}^{t} \binom{t-2}{j-2}(k-1)^{t-j} = \\ &= \quad k\binom{t}{2} \sum_{\ell=0}^{t-2} \binom{t-2}{\ell}(k-1)^{t-2-\ell}. \end{aligned}$$

Using the binomial theorem for $((k-1)+1)^{t-2}$, this can be written as

$$\sum_{w \in T_t} q(w) = k\binom{t}{2}k^{t-2} = k^{t-1}\binom{t}{2}.$$

Dividing this by $|T_t| = k^t$, we receive $\frac{1}{|T_t|}\sum_{w \in T_t} q(w) = \frac{t^2-t}{2k}$. Since $x \mapsto (1-p)^x$ is convex, thus Jensen's inequality gives

$$\frac{1}{|T_t|} \sum_{w \in T_t} (1-p)^{q(w)} \geq (1-p)^{\frac{1}{|T_t|}\sum_{w \in T_t} q(w)} = (1-p)^{\frac{t^2-t}{2k}},$$

yielding exactly the stated lower bound.

In order to prove the upper bound, we use

$$\frac{\sum_{i=1}^{k} s_i^2}{k} \geq \left(\frac{\sum_{i=1}^{k} s_i}{k} \right)^2 = \frac{t^2}{k^2},$$

thus

$$q(w) = \frac{1}{2} \left(\sum_{i=1}^{k} s_i^2 - \sum_{i=1}^{k} s_i \right) \geq \frac{1}{2} \left(\frac{t^2}{k} - t \right),$$

yielding exactly the stated upper bound. □

Since $\mathbb{E}(Y) = \sum_{t=0}^{n} \mathbb{E}(Y_t)$, and $\mathbb{E}(S) = \mathbb{E}(Y_n)$, we obtain the following bounds as a corollary of Proposition 3:

$$\mathbb{E}(Y) \geq \sum_{t=0}^{n} k^t (1-p)^{\frac{t^2-t}{2k}} \tag{1}$$

$$\mathbb{E}(Y) \leq \sum_{t=0}^{n} k^t (1-p)^{\frac{t^2-kt}{2k}} \tag{2}$$

$$k^n (1-p)^{\frac{n^2-n}{2k}} \leq \mathbb{E}(S) \leq k^n (1-p)^{\frac{n^2-kn}{2k}} \tag{3}$$

5. Asymptotic analysis

Originally, we derived the above bounds with the aim of using them in a setting where the value of p is fixed [15, 16]. However, they also apply to the case when p depends on n. In the following, we will write $p(n)$ or p_n to denote the dependence of p on n.

Our aim is to prove Theorems 1 and 2. For this purpose, we need to estimate the sums in the above inequalities (1) and (2) for large values of n. It should be noted that these are not simple series, because with growing n, not only the number of terms changes, but also the terms themselves, since p is not constant. This is why we need the following, more sophisticated method to estimate sums of this form, which is an application of Laplace's method (cf. [47, Appendix A.6]).

From inequality (1),

$$\mathbb{E}(Y) \geq \sum_{t=0}^{n} k^t \left(1-p_n\right)^{\frac{t^2-t}{2k}} = \sum_{t=0}^{n} \left(\left(1-p_n\right)^{\frac{1}{2k}}\right)^{t^2} \left(k\left(1-p_n\right)^{-\frac{1}{2k}}\right)^t. \qquad (4)$$

In this formula, $0 < \left(1-p_n\right)^{\frac{1}{2k}} < 1$ and $k\left(1-p_n\right)^{-\frac{1}{2k}} > 1$. Therefore, $\exists a_n, b_n > 0$, so that $\left(1-p_n\right)^{\frac{1}{2k}} = e^{-a_n}$ and $k\left(1-p_n\right)^{-\frac{1}{2k}} = e^{b_n}$. Introducing

$$r_n = -\ln(1-p_n),$$

we can write

$$\begin{aligned} a_n &= -\ln\left(\left(1-p_n\right)^{\frac{1}{2k}}\right) = \frac{r_n}{2k}, \\ b_n &= \ln\left(k\left(1-p_n\right)^{-\frac{1}{2k}}\right) = \ln k + \frac{r_n}{2k}. \end{aligned}$$

With this choice of a_n and b_n, the lower bound from equation (4) becomes simply

$$\mathbb{E}(Y) \geq \sum_{t=0}^{n} \exp(-a_n t^2 + b_n t).$$

In an analogous way, the upper bound (2) can be reformulated as

$$\mathbb{E}(Y) \leq \sum_{t=0}^{n} \left(\left(1-p_n\right)^{\frac{1}{2k}}\right)^{t^2} \left(k\left(1-p_n\right)^{-\frac{1}{2}}\right)^t = \sum_{t=0}^{n} \exp(-a_n t^2 + b_n' t). \qquad (5)$$

Note that a_n is the same as before, but the value of b_n' is slightly different from b_n:

$$b_n' = \ln\left(k\left(1-p_n\right)^{-\frac{1}{2}}\right) = \ln k + \frac{r_n}{2}.$$

Knowing that $\lim_{n\to\infty} p_n = 0+$, the following limits can be easily established:

$$\begin{aligned} \lim_{n\to\infty} r_n &= 0+, \\ \lim_{n\to\infty} a_n &= 0+, \\ \lim_{n\to\infty} b_n &= \ln k, \\ \lim_{n\to\infty} b_n' &= \ln k, \\ \lim_{n\to\infty} r_n/p_n &= 1. \end{aligned}$$

The following two lemmas are refinements of Lemma 3 in [14].

Lemma 4. *Let $n \in \mathbb{Z}^+$ and $a = a_n, b = b_n \in \mathbb{R}^+$ such that $2an - b > 0$. Then,*

$$\sum_{t=0}^{n} e^{-at^2} e^{bt} > \frac{1}{\sqrt{a}} e^{\frac{b^2}{4a}} \int_{-\frac{b}{2\sqrt{a}}}^{-\sqrt{a}} e^{-u^2}\, du.$$

Proof. Let $x = t - \frac{b}{2a}$, hence $-ax^2 = -at^2 + bt - \frac{b^2}{4a}$. Besides, let $u = \sqrt{a}x$, thus $u^2 = ax^2$. Accordingly:

$$\sqrt{a}e^{-\frac{b^2}{4a}} \sum_{t=0}^{n} e^{-at^2} e^{bt} = \sqrt{a} \sum_{t=0}^{n} e^{-ax^2(t)} = \sqrt{a} \sum_{x=-\frac{b}{2a}}^{-\frac{b}{2a}+n} e^{-ax^2} = \sqrt{a} \sum_{u=-\frac{b}{2\sqrt{a}}}^{-\frac{b}{2\sqrt{a}}+\sqrt{a}n} e^{-u^2}.$$

Here, x and u might denote fractions; the summation ranges over all x respectively u, for which $x = t - \frac{b}{2a}$, $u = \sqrt{a}t - \frac{b}{2\sqrt{a}}$, where t is an integer between 0 and n. Note that x goes with step 1, whereas u goes with step $\sqrt{a}$.

Since $2an - b > 0$, it follows that $-\frac{b}{2\sqrt{a}} + \sqrt{a}n > 0$. Hence, restricting the last sum to the terms where $u < 0$, and then regarding it as an upper estimation of an integral by step $\sqrt{a}$, we obtain

$$\sqrt{a}e^{-\frac{b^2}{4a}} \sum_{t=0}^{n} e^{-at^2} e^{bt} \geq \sqrt{a} \sum_{u=-\frac{b}{2\sqrt{a}}}^{0} e^{-u^2} > \int_{-\frac{b}{2\sqrt{a}}}^{-\sqrt{a}} e^{-u^2}\, du,$$

which completes the proof. (In the last inequality, we used the fact that the highest u below 0 must be in the interval $[-\sqrt{a}, 0]$. See also Figure 1. Note that we had to be careful because e^{-u^2} is not monotonous in the whole interval $[-\frac{b}{2\sqrt{a}}, -\frac{b}{2\sqrt{a}} + \sqrt{a}n]$; this is why we restricted ourselves to negative values of u.) □

Corollary 5. *Let n, a, b as in Lemma 4. Then,*

$$\sum_{t=0}^{n} e^{-at^2} e^{bt} > \frac{b}{2a} - 1.$$

Proof. As Figure 1 illustrates, the integral is higher than the area of the gray rectangle under the curve:

$$\int_{-\frac{b}{2\sqrt{a}}}^{-\sqrt{a}} e^{-u^2}\, du > \left(\frac{b}{2\sqrt{a}} - \sqrt{a}\right) e^{-\left(-\frac{b}{2\sqrt{a}}\right)^2} = \left(\frac{b}{2\sqrt{a}} - \sqrt{a}\right) e^{-\frac{b^2}{4a}},$$

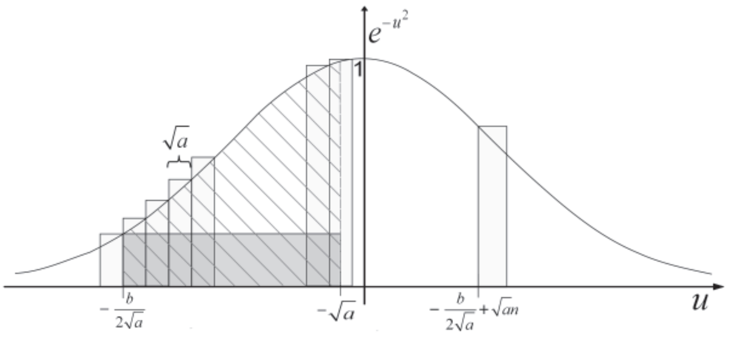

Figure 1: Lower bound in the $2an - b > 0$ case

leading exactly to the desired bound. □

Lemma 6. *Let $n \in \mathbb{Z}^+$ and $a = a_n, b' = b'_n \in \mathbb{R}^+$ such that $2an - b' > 0$. Then,*

$$\sum_{t=0}^{n} e^{-at^2} e^{b't} < \frac{1}{\sqrt{a}} e^{\frac{b'^2}{4a}} \left(\int_{-\frac{b'}{2\sqrt{a}}}^{-\frac{b'}{2\sqrt{a}}+\sqrt{a}n} e^{-u^2}\, du + \sqrt{a} \right).$$

Proof. Similarly to the proof of Lemma 4 and using its notations (but with b' instead of b), we would like to regard the received sum as a lower approximation of an integral by step $\sqrt{a}$. Again, we have $-\frac{b'}{2\sqrt{a}} + \sqrt{a}n > 0$. As can be seen in Figure 2, each negative value of u is represented with a rectangle to the right from u, whereas each positive value of u is represented with a rectangle to the left from u. This way, we get a proper lower approximation of the integral, except for the fact that there are two rectangles (the rectangle corresponding to the highest negative value of u and the rectangle corresponding to the lowest positive value of u) that overlap. The error thus made is at most $\sqrt{a} \cdot 1$. Hence,

$$\sqrt{a} e^{-\frac{b'^2}{4a}} \sum_{t=0}^{n} e^{-at^2} e^{b't} = \sqrt{a} \sum_{u=-\frac{b'}{2\sqrt{a}}}^{-\frac{b'}{2\sqrt{a}}+\sqrt{a}n} e^{-u^2} < \int_{-\frac{b'}{2\sqrt{a}}}^{-\frac{b'}{2\sqrt{a}}+\sqrt{a}n} e^{-u^2}\, du + \sqrt{a},$$

which completes the proof. □

Concerning the $2an - b \leq 0$ case, we will use the following bounds:

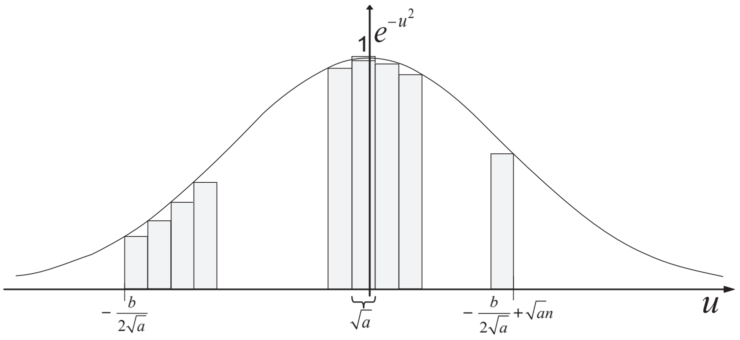

Figure 2: Upper bound in the $2an - b' > 0$ case

Lemma 7. *Let $n \in \mathbb{Z}^+$ and $a = a_n, b' = b'_n \in \mathbb{R}^+$ such that $2an - b' < 0$. Then,*

$$\sum_{t=0}^{n} \exp(-at^2 + b't) < \left(1 + \frac{2}{b' - 2an}\right) \exp(-an^2 + b'n).$$

Proof. Similarly to the proof of Lemma 6, we have

$$\sum_{t=0}^{n} \exp(-at^2 + b't) = \exp(-an^2 + b'n) + \sum_{t=0}^{n-1} \exp(-at^2 + b't) <$$

$$< \exp(-an^2 + b'n) + \frac{1}{\sqrt{a}} \exp\left(\frac{b'^2}{4a}\right) \int_{-\frac{b'}{2\sqrt{a}}}^{-\frac{b'}{2\sqrt{a}}+\sqrt{a}n} e^{-u^2}\, du. \tag{6}$$

The idea behind this is that now $-\frac{b'}{2\sqrt{a}}+\sqrt{a}n < 0$, and thus e^{-u^2} is monotonously increasing in the whole integration domain. Therefore, a member of the sum at u corresponds to a rectangle to the right from u, and thus the integration domain must have length $\sqrt{a}n$ to estimate the sum from $t = 0$ to $t = n - 1$.

Using $u_1 = -\frac{b'}{2\sqrt{a}}$ and $u_2 = -\frac{b'}{2\sqrt{a}} + \sqrt{a}n$, the integral $\int_{u_1}^{u_2} e^{-u^2} du$ can be bounded for $u_1 < u_2 < 0$ as follows:

$$\int_{u_1}^{u_2} e^{-u^2} du < \int_{u_1}^{u_2} e^{-u_2 u} du = -\frac{1}{u_2}\left(e^{-u_2^2} - e^{-u_1 u_2}\right) < -\frac{1}{u_2} e^{-u_2^2}.$$

Using the specific value for u_2, this yields

$$\int_{-\frac{b'}{2\sqrt{a}}}^{-\frac{b'}{2\sqrt{a}}+\sqrt{a}n} e^{-u^2}\, du < \frac{2\sqrt{a}}{b' - 2an} \exp\left(-\frac{b'^2}{4a} - an^2 + b'n\right).$$

Writing this back into (6) completes the proof. □

Proposition 8. *Let $n \in \mathbb{Z}^+$ and $a = a_n, b = b_n \in \mathbb{R}^+$ such that $2an - b \leq 0$. Then, $\sum_{t=0}^{n} \exp(-at^2 + bt) \geq n + 1$.*

Proof. Let $0 \leq t \leq n$. Since $2an - b \leq 0$ and $a > 0$, it follows that $b \geq 2an > an \geq at$, and thus $\exp(-at^2 + bt) = \exp(t(b - at)) \geq 1$. □

Now, all the needed machinery is in place for the proofs of the main theorems.

Proof of Theorem 1. Using Corollary 5 and Proposition 8, we obtain

$$\mathbb{E}(Y) \geq \begin{cases} \frac{b_n}{2a_n} - 1 & \text{if } 2a_n n - b_n > 0, \\ n + 1 & \text{if } 2a_n n - b_n \leq 0. \end{cases}$$

When $n \to \infty$, both lower bounds tend to infinity, which completes the proof. □

Proof of Theorem 2. Using the definition of a_n, b_n, b'_n, and r_n, we can write $2a_n n - b_n = \frac{nr_n}{k} - \ln k - \frac{r_n}{2k}$ and $2a_n n - b'_n = \frac{nr_n}{k} - \ln k - \frac{r_n}{2}$. Since $r_n \to 0$ and $r_n/p_n \to 1$, the following can be stated: in part (1), where $np_n > k \ln k + \varepsilon$, both $2a_n n - b_n$ and $2a_n n - b'_n$ will be positive for all large enough n, whereas in part (2), where $np_n < k \ln k - \varepsilon$, both $2a_n n - b_n$ and $2a_n n - b'_n$ will be negative for all large enough n.

(1) Lemma 4 can be used, yielding

$$\mathbb{E}(Y) > \frac{1}{\sqrt{a_n}} \exp\left(\frac{b_n^2}{4a_n}\right) \int_{-\frac{b_n}{2\sqrt{a_n}}}^{-\sqrt{a_n}} e^{-u^2}\, du.$$

In view of $\lim_{n\to\infty} -\frac{b_n}{2\sqrt{a_n}} = -\infty$ and $\lim_{n\to\infty} -\sqrt{a_n} = 0$,

$$\lim_{n\to\infty} \int_{-\frac{b_n}{2\sqrt{a_n}}}^{-\sqrt{a_n}} e^{-u^2}\, du = \int_{-\infty}^{0} e^{-u^2}\, du = \frac{\sqrt{\pi}}{2},$$

and thus

$$\mathbb{E}(Y) = \Omega\left(\frac{1}{\sqrt{a_n}} \exp\left(\frac{b_n^2}{4a_n}\right)\right).$$

Since $b_n > \ln k$, this can be further written as

$$\mathbb{E}(Y) = \Omega\left(\frac{1}{\sqrt{a_n}} \exp\left(\frac{(\ln k)^2}{4a_n}\right)\right) = \Omega\left(\sqrt{\frac{2k}{r_n}} \exp\left(\frac{k(\ln k)^2}{2r_n}\right)\right).$$

This is almost the desired lower bound, except that it contains r_n instead of p_n. The first occurence of r_n can be easily changed to p_n because $r_n/p_n \to 1$, and thus, for all large enough n, we have for example $r_n < 2p_n$. It is less obvious why the second occurence of r_n can be changed to p_n, as it appears in the denominator of the exponent. For this purpose, we can use the bound $r_n \leq \frac{p_n}{1-p_n}$. (This can be seen for example from Lagrange's mean value theorem and using the fact that $(-\ln(1-x))' = 1/(1-x)$ is monotonously increasing for $0 < x < 1$.) This yields

$$\mathbb{E}(Y) = \Omega\left(\sqrt{\frac{1}{p_n}}\exp\left(\frac{k(\ln k)^2}{2p_n}(1-p_n)\right)\right) = \Omega\left(\sqrt{\frac{1}{p_n}}\exp\left(\frac{k(\ln k)^2}{2p_n}\right)\right),$$

exactly as intended.

The corresponding upper bound can be obtained using Lemma 6:

$$\begin{aligned}\mathbb{E}(Y) <& \frac{1}{\sqrt{a_n}}\exp\left(\frac{b_n'^2}{4a_n}\right)\left(\int_{-\frac{b_n'}{2\sqrt{a_n}}}^{-\frac{b_n'}{2\sqrt{a_n}}+\sqrt{a_n}n} e^{-u^2}\,du + \sqrt{a_n}\right) < \\ <& \frac{1}{\sqrt{a_n}}\exp\left(\frac{b_n'^2}{4a_n}\right)\left(\int_{-\infty}^{+\infty} e^{-u^2}\,du + \sqrt{a_n}\right).\end{aligned}$$

Using that $\int_{-\infty}^{+\infty} e^{-u^2}\,du = \sqrt{\pi}$ and that $\lim_{n\to\infty}\sqrt{a_n} = 0$, we obtain

$$\mathbb{E}(Y) = O\left(\frac{1}{\sqrt{a_n}}\exp\left(\frac{b_n'^2}{4a_n}\right)\right).$$

Here, the exponent is

$$\frac{b_n'^2}{4a_n} = \frac{(\ln k + \frac{r_n}{2})^2}{4a_n} = \frac{(\ln k)^2}{4a_n} + \frac{kr_n}{8} + \frac{k\ln k}{2} = \frac{(\ln k)^2}{4a_n} + O(1),$$

and hence

$$\mathbb{E}(Y) = O\left(\frac{1}{\sqrt{a_n}}\exp\left(\frac{(\ln k)^2}{4a_n}\right)\right) = O\left(\sqrt{\frac{2k}{r_n}}\exp\left(\frac{k(\ln k)^2}{2r_n}\right)\right).$$

Using that $r_n \geq p_n$, we obtain

$$\mathbb{E}(Y) = O\left(\sqrt{\frac{1}{p_n}}\exp\left(\frac{k(\ln k)^2}{2p_n}\right)\right),$$

as intended.

(2) Here we use the trivial lower bound

$$\sum_{t=0}^{n} \exp(-a_n t^2 + b_n t) > \exp(-a_n n^2 + b_n n) > \exp(-a_n n^2 + n \ln k).$$

As upper bound, Lemma 7 yields

$$\sum_{t=0}^{n} \exp(-a_n t^2 + b'_n t) < \left(1 + \frac{2}{b'_n - 2a_n n}\right) \exp(-a_n n^2 + b'_n n).$$

It is already known that in this case $b'_n - 2a_n n > 0$. However, we need to show that this expression can even be bounded by a positive constant:

$$b'_n - 2a_n n = \ln k + \frac{r_n}{2} - \frac{n r_n}{k} \geq \ln k - \frac{n p_n}{k} \frac{r_n}{p_n} > \varepsilon'$$

for any $0 < \varepsilon' < \varepsilon/k$. This holds because $\frac{np_n}{k} < \ln k - \frac{\varepsilon}{k}$ and $r_n/p_n \to 1$. As a consequence, $1 + \frac{2}{b'_n - 2a_n n} = O(1)$ and thus $\mathbb{E}(Y) = O(\exp(-a_n n^2 + b'_n n))$. Here, $b'_n n = n \ln k + \frac{n r_n}{2} = n \ln k + \frac{n p_n}{2} \frac{r_n}{p_n} = n \ln k + O(1)$, and thus $\mathbb{E}(Y) = O(\exp(-a_n n^2 + n \ln k))$.

Together with the lower bound, we have

$$\mathbb{E}(Y) = \Theta(\exp(-a_n n^2 + n \ln k)) = \Theta\left(\exp\left(-\frac{r_n}{2k} n^2 + n \ln k\right)\right) =$$
$$= \Theta\left(\exp\left(\frac{n^2}{2k} \ln(1 - p_n) + n \ln k\right)\right) = \Theta\left((1 - p_n)^{\frac{n^2}{2k}} k^n\right).$$

□

6. Discussion

6.1. The $p_n = d/n$ case

It is interesting to investigate what Theorem 2 yields in the special case when $p_n = d/n$, where d is a positive constant (approximately the expected degree of the vertices). Obviously, $np_n > k \ln k + \varepsilon \Leftrightarrow d > k \ln k$ and $np_n < k \ln k - \varepsilon \Leftrightarrow d < k \ln k$. Let first $d > k \ln k$. Then, Theorem 2 yields $\mathbb{E}(Y) = \Theta\left(\sqrt{n} \exp\left(\frac{k(\ln k)^2}{2d} n\right)\right) = \Theta\left(\exp\left(\frac{k(\ln k)^2}{2d} n + \frac{1}{2} \ln n\right)\right)$.

In the second case ($d < k \ln k$), Theorem 2 yields $\mathbb{E}(Y) = \Theta\left(\exp\left(n \ln k - \frac{n^2 r_n}{2k}\right)\right)$. In order to obtain a formula that can be handled more easily, it would be good to replace here r_n with p_n. In general, this is not possible, but in the $p_n = d/n$ case, it is: from the Taylor expansion of $-\ln(1-x)$ it follows that $r_n = p_n + O(p_n^2) = p_n + O(1/n^2)$. Thus,

$$\mathbb{E}(Y) = \Omega\left(\exp\left(n\ln k - \frac{n^2 p_n}{2k} - O(1)\right)\right) = \Omega\left(\exp\left(n\ln k - \frac{n^2 p_n}{2k}\right)\right).$$

On the other hand, since $r_n \geq p_n$, it is obvious that $\mathbb{E}(Y) = O\left(\exp\left(n\ln k - \frac{n^2 p_n}{2k}\right)\right)$, so that we have

$$\mathbb{E}(Y) = \Theta\left(\exp\left(n\ln k - \frac{n^2 p_n}{2k}\right)\right) = \Theta\left(\exp\left(\left(\ln k - \frac{d}{2k}\right)n\right)\right).$$

To sum up:

$$\mathbb{E}(Y) = \begin{cases} \Theta\left(\exp\left(\frac{k(\ln k)^2}{2d}n + \frac{1}{2}\ln n\right)\right) & \text{if } d > k\ln k, \\ \Theta\left(\exp\left(\left(\ln k - \frac{d}{2k}\right)n\right)\right) & \text{if } d < k\ln k. \end{cases}$$

As can be seen, both expressions are exponential in n, but the behaviour is slightly different in the two cases. The transition between the two cases is quite smooth: looking at the coefficient of n in the exponent, both formulae give $\frac{1}{2}\ln k$ for $d = k\ln k$. What is more, even their derivatives with respect to d are equal at this point: $\frac{\partial}{\partial d}\frac{k(\ln k)^2}{2d}\Big|_{d=k\ln k} = -\frac{k(\ln k)^2}{2d^2}\Big|_{d=k\ln k} = -\frac{1}{2k}$ and also $\frac{\partial}{\partial d}\left(\ln k - \frac{d}{2k}\right) = -\frac{1}{2k}$.

It is interesting to relate this phenomenon to the phase transition in the geometry of the solution space, as shown recently by Achlioptas and Coja-Oghlan [48]. They proved that for $d < k\ln k$, the set of solutions builds whp a giant connected ball, whereas for $d > k\ln k$, it disintegrates into an exponential number of small components that are quite far from each other. Achlioptas and Coja-Oghlan suggest that this may be the reason why it is easy to find a solution for $d < k\ln k$, while this is not possible with any of the expected polynomial-time algorithms known today for $d > k\ln k$. It is worth noting that our results also show a transition at exactly the same point. The transition that we observe is less abrupt than the one shown by Achlioptas and Coja-Oghlan, presumably due to the following differences:

- The algorithm that we are investigating does not stop at the first found solution, but visits *all* solutions. Hence, the scattered solution space for $d > k \ln k$ is not significantly more difficult for this algorithm than the giant ball for $d < k \ln k$.

- While Achlioptas and Coja-Oghlan were focusing on the set of solutions, the algorithm that we are investigating spends significant time with partial solutions. Thus, an abrupt change in the structure of the solution space does not necessarily have a high impact on the overall search tree of our algorithm.

Nevertheless, there *is* a transition at $d = k \ln k$, and from the proof of Theorem 2 also its origins can be understood. The number of visited nodes on level t of the search tree depends on two conflicting factors: there are k^t nodes on this level of the tree, and a fraction of $(1 - p_n)^{\frac{t^2}{2k} + \Theta(t)}$ of them are visited. The first factor is increasing in t, the second decreasing. Their product starts to increase rapidly, has a maximum, and then decreases rapidly (as a bell curve). For $d < k \ln k$, the maximum would be at some $t > n$, whereas for $d > k \ln k$, the maximum is at some $t < n$. This means that for $d < k \ln k$, the number of visited nodes is exponentially increasing for all $t \leq n$, with the biggest contribution stemming from the last level, and thus even a small change in d or n alters the overall number of visited nodes of the search tree significantly. On the other hand, if $d > k \ln k$, then the maximum contribution is at some intermediate level and the contribution of the last levels is minimal; thus, changes in d or n have much lower impact on $\mathbb{E}(Y)$.

6.2. Balanced colorings

In Proposition 3, we showed that

$$k^t (1-p)^{\frac{t^2 - t}{2k}} \leq \mathbb{E}(Y_t) \leq k^t (1-p)^{\frac{t^2 - kt}{2k}},$$

which was sufficient for deriving our theorems. However, it is worth mentioning that the upper bound is tight within polynomial terms. This is due to the fact that the sum over partial colorings in T_t is dominated by *balanced partial colorings*, in which each color class has $\lceil t/k \rceil$ or $\lfloor t/k \rfloor$ vertices. For the case when t is a multiple of k, this was already shown by Achlioptas and Naor [12].

For the general case, let $t = t_1 k + t_2$, where t_1, t_2 are integers and $0 \leq t_2 \leq k-1$. In [16], we established that the number of balanced partial colorings in T_t is

$$R_0 = \binom{k}{t_2} \cdot \frac{t!}{((t_1+1)!)^{t_2}(t_1!)^{k-t_2}},$$

and their q value is

$$q_0 = t_2 \binom{t_1+1}{2} + (k - t_2)\binom{t_1}{2}.$$

Using Stirling's approximation, we obtain

$$\begin{aligned} R_0 &= \binom{k}{t_2} \cdot \frac{t!}{(t_1+1)^{t_2}(t_1!)^k} \geq \binom{k}{t_2} \cdot \frac{\sqrt{2\pi t} \cdot \frac{t^t}{e^t}}{(t_1+1)^{t_2} \cdot \left(e\sqrt{t_1} \cdot \frac{t_1^{t_1}}{e^{t_1}}\right)^k} = \\ &= \binom{k}{t_2} \cdot \frac{\sqrt{2\pi}}{e^{k+t_2}} \cdot \frac{t^{t_2+1/2}}{(t_1+1)^{t_2} \cdot t_1^{k/2}} \cdot \left(\frac{t}{t_1}\right)^{t_1 k} \geq \\ &\geq \binom{k}{t_2} \cdot \frac{\sqrt{2\pi}}{e^{k+t_2}} \cdot \frac{t^{t_2+1/2}}{(t_1+1)^{t_2} \cdot t_1^{k/2}} \cdot k^{t-t_2} = \Omega\left(\frac{1}{\alpha(t)} \cdot k^t\right), \end{aligned} \tag{7}$$

where $\alpha(t)$ is polynomial in t. Furthermore, it is easy to see that

$$q_0 - \frac{t^2 - kt}{2k} = \frac{1}{2}\left(t_2 - \frac{t_2^2}{k}\right) = O(1). \tag{8}$$

Equations (7) and (8) together yield the following lower bound:

$$\mathbb{E}(Y_t) \geq R_0 \cdot (1-p)^{q_0} = \Omega\left(\frac{1}{\alpha(t)} \cdot k^t \cdot (1-p)^{\frac{t^2-kt}{2k}}\right),$$

which is only a polynomial factor away from the upper bound of Proposition 3.

6.3. Expected number of solutions

In this section, we look at the asymptotics of the expected number of solutions, and discuss some of its consequences. It is well known that for $np_n > 2k \ln k + \varepsilon$, $\mathbb{E}(S) < c_1^n$ for some $0 < c_1 < 1$ [12]. In the $p_n = d/n$ case, this corresponds to the $d > 2k \ln k$ condition.

Applying Markov's inequality, $\lim_{n\to\infty} Pr(\exists \text{ solution}) = \lim_{n\to\infty} Pr(S \geq 1) \leq \lim_{n\to\infty} \mathbb{E}(S) = 0$. In other words, such graphs are whp non-k-colorable. As mentioned earlier, the investigated algorithm solves the counting problem in general, but for non-k-colorable graphs, the amount of computation is equal for the counting problem and the decision problem. Thus we can now conclude that, for $np_n > 2k \ln k + \varepsilon$, our results on $\mathbb{E}(Y)$ also apply to the version of the algorithm solving the decision problem.

The presented machinery can also be used to estimate $\mathbb{E}(S)$ in the $np_n < 2k \ln k - \varepsilon$ case:

Proposition 9. *Let the number of available colors k be constant, and $p = p_n$ be any sequence between 0 and 1, tending to 0. Let $\mathbb{E}(S)$ denote the expected number of k-colorings of the graph. If, for all large enough n, $np_n < 2k \ln k - \varepsilon$, then $\mathbb{E}(S) > c_2^n$ for some $1 < c_2$. (Specifically, $c_2 = \exp(\varepsilon')$, where $0 < \varepsilon' < \frac{\varepsilon}{2k}$.)*

Proof. From inequality (3),

$$\begin{aligned}\mathbb{E}(S) \geq & k^n \left(1-p_n\right)^{\frac{n^2-n}{2k}} = k^n \exp\left((\ln(1-p_n))\frac{n^2-n}{2k}\right) = \\ = & k^n \exp\left(-(1+o(1))p_n \frac{n^2-n}{2k}\right) = \left(\frac{k}{\exp\left((1+o(1))p_n\frac{n-1}{2k}\right)}\right)^n . \end{aligned} \tag{9}$$

In the exponent of the denominator, we have

$$(1+o(1))p_n\frac{n-1}{2k} < (1+o(1))\frac{np_n}{2k} < (1+o(1))\left(\ln k - \frac{\varepsilon}{2k}\right) < \ln k - \varepsilon'$$

for any $0 < \varepsilon' < \frac{\varepsilon}{2k}$. Writing this into (9) yields $\mathbb{E}(S) > c_2^n$, as stated. □

To sum up, the expected number of solutions tends exponentially to 0 for $np_n > 2k \ln k + \varepsilon$, whereas for $np_n < 2k \ln k - \varepsilon$, it tends exponentially to ∞. It should also be noted that this result is independent of the used algorithm.

In the $p_n = \frac{d}{n}$ case, if $d < 2k \ln k$, then Proposition 9 can be applied, and hence $\lim \mathbb{E}(S) = \infty$. Analyzing the $d = 2k \ln k$ case separately, by applying (3) directly:

$$\lim_{n\to\infty} \mathbb{E}(S) \geq \lim_{n\to\infty} k^n \left(1 - \frac{d}{n}\right)^{n\frac{n-1}{2k}} = \lim_{n\to\infty} \left(\frac{k}{\sqrt[2k]{e^d}}\right)^n \sqrt[2k]{e^d} = \sqrt[2k]{e^d} = k,$$

$$\lim_{n\to\infty} \mathbb{E}(S) \le \lim_{n\to\infty} k^n \left(1-\frac{d}{n}\right)^{n\frac{n-k}{2k}} = \left(\frac{k}{\sqrt[2k]{e^d}}\right)^n \sqrt{e^d} = \sqrt{e^d} = k^k.$$

Hence, $\mathbb{E}(S)$ remains finite and non-zero in this case.

It may be worth noting that this dramatic change in the behaviour of $\mathbb{E}(S)$ at $d = 2k \ln k$ does not have any impact on $\mathbb{E}(Y)$. As shown earlier, $\mathbb{E}(Y)$ has a – less dramatic – transition at $d = k \ln k$, and for $d > k \ln k$, the contribution of the last levels of the search tree to $\mathbb{E}(Y)$ is marginal.

7. Numerical examinations

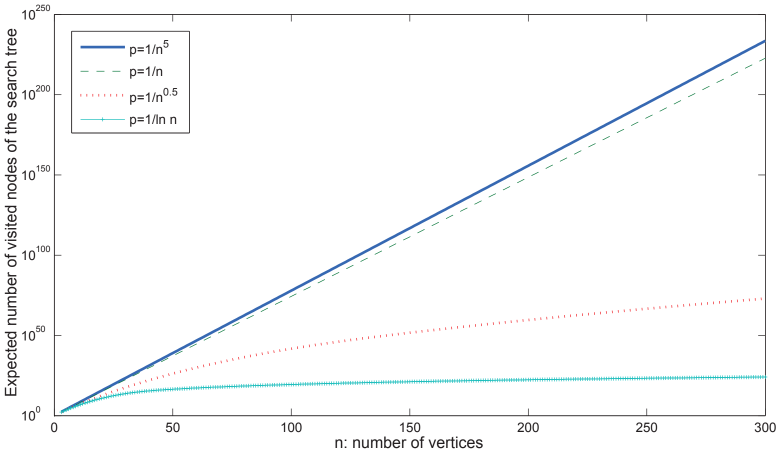

Figure 3: Expected number of visited nodes of the search tree for different edge density functions ($k = 6$).

Using the presented approach and the technique for efficiently computing $\mathbb{E}(Y)$ and $\mathbb{E}(S)$ values that we developed in [15], we can show graphically the behavior of these quantities for some representative p_n functions. See Figure 3 for the behavior of $\mathbb{E}(Y)$ and Figure 4 for the behavior of $\mathbb{E}(S)$. Please note the exponential scale on the vertical axis in both figures.

As can be seen, for $p_n = 1/n^5$ and $p_n = 1/n$, both $\mathbb{E}(Y)$ and $\mathbb{E}(S)$ tend rapidly to infinity. For $p_n = 1/n^{0.5}$, $\mathbb{E}(Y)$ grows significantly more slowly, but

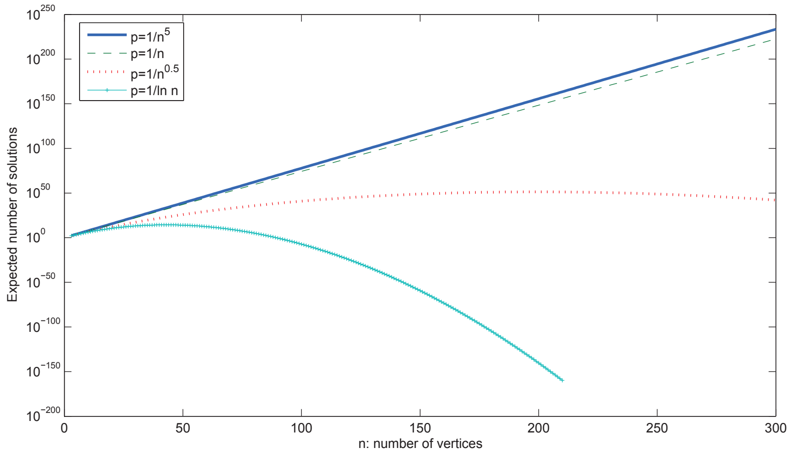

Figure 4: Expected number of solutions for different edge density functions ($k = 6$).

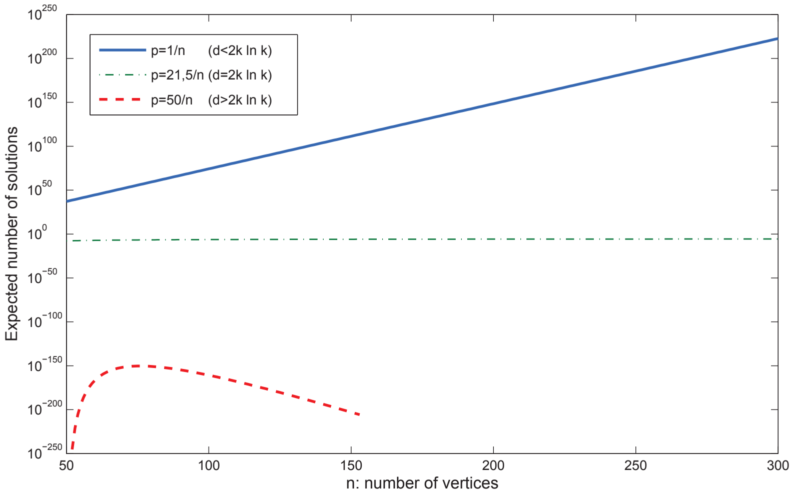

Figure 5: Expected number of solutions for different edge density functions of the form $p_n = d/n$ ($k = 6$).

as we know, still super-polynomially. $\mathbb{E}(S)$ starts as a monotonously increasing function, but has its maximum at $n = 199$, where the expected number of solutions is $2,03 \cdot 10^{51}$ and decreases afterwards. As we know, $\mathbb{E}(S)$ tends to 0 in this case, but it is interesting to note that $\mathbb{E}(S)$ is quite high for graphs with approximately 200 vertices. Finally, when $p_n = 1/\ln n$, then $\mathbb{E}(S)$ tends to 0 in a much quicker manner. Also the growth of $\mathbb{E}(Y)$ is quite moderate in this case – as we know, it is polynomial in n.

Finally, Figure 5 depicts the behavior of $\mathbb{E}(S)$ in the $p_n = d/n$ case, for different values of d. In line with the calculations, $\mathbb{E}(S)$ increases rapidly when $d < 2k \ln k$ and converges quickly to 0 when $d > 2k \ln k$. In the critical case of $d = 2k \ln k$, the value of $\mathbb{E}(S)$ stagnates.

8. Conclusion

In this paper, we analyzed the complexity of the backtrack search algorithm for coloring random graphs from $G_{n,p}$. Our main focus was on estimating the expected number of visited nodes in the algorithm's search tree. In contrast to most previous research, our results apply to any p_n sequence with $\lim p_n = 0$. In particular, we proved that, for all such sequences, the average-case complexity of the algorithm goes to infinity. This is in contrast with the case of fixed p, where the average-case complexity of the algorithm is known to be $O(1)$. We also established how quickly the average-case complexity increases for different p_n sequences, and we showed examples where it is polynomial respectively exponential. Finally, we estimated with the same method the expected number of valid k-colorings, and showed that, apart from a narrow critical region, it quickly goes to either 0 or infinity. Our analytical results were supplemented by corresponding numerical experiments.

The most important question that remains open is how the presented method can be transferred to a variant of the algorithm that solves the decision version of the problem and is also realistic for k-colorable graphs.

Acknowledgments

This work was partially supported by the Hungarian National Research Fund and the National Office for Research and Technology (Grant Nr. OTKA 67651), as well as the grant TÁMOP - 4.2.2.B-10/1–2010-0009. The work of Zoltán Ádám Mann was partially supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

[1] P. Briggs, K. D. Cooper, L. Torczon, Improvements to graph coloring register allocation, ACM Transactions on Programming Languages and Systems 16 (3) (1994) 428–455.

[2] N. K. Mehta, The application of a graph coloring method to an examination scheduling problem, Interfaces 11 (5) (1981) 57–65.

[3] Z. Mann, A. Orbán, Optimization problems in system-level synthesis, in: 3rd Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, 2003, pp. 222–231.

[4] P. Erdős, A. Rényi, On the evolution of random graphs, Magyar Tud. Akad. Mat. Kutató Int. Közl. 5 (1960) 17–61.

[5] B. Bollobás, Random Graphs, 2nd Edition, Cambridge University Press, Cambridge, 2001.

[6] S. Janson, T. Luczak, A. Rucinski, Random Graphs, Wiley, New York, 2000.

[7] N. Alon, M. Krivelevich, The concentration of the chromatic number of random graphs, Combinatorica 17 (3) (1997) 303–313.

[8] B. Bollobás, The chromatic number of random graphs, Combinatorica 8 (1) (1988) 49–55.

[9] T. Luczak, The chromatic number of random graphs, Combinatorica 11 (1) (1991) 45–54.

[10] T. Luczak, A note on the sharp concentration of the chromatic number of random graphs, Combinatorica 11 (3) (1991) 295–297.

[11] E. Shamir, J. Spencer, Sharp concentration of the chromatic number on random graphs $G_{n,p}$, Combinatorica 7 (1) (1987) 121–129.

[12] D. Achlioptas, A. Naor, The two possible values of the chromatic number of a random graph, in: 36th ACM Symposium on Theory of Computing (STOC '04), 2004, pp. 587–593.

[13] H. S. Wilf, Backtrack: an O(1) expected time algorithm for the graph coloring problem, Information Processing Letters 18 (1984) 119–121.

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

[15] Z. Mann, A. Szajkó, Determining the expected runtime of exact graph coloring, in: Mini-conference on Applied Theoretical Computer Science (MATCOS), published in the Proceedings of the 13th International Multiconference, Information Society - IS, Volume A, 2010, pp. 389–392.

[16] Z. Mann, A. Szajkó, Improved bounds on the complexity of graph coloring, in: Advances in the Theory of Computing (AITC 2010), published in the Proceedings of the 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, IEEE Computer Society, 2010, pp. 347–354.

[17] Z. Mann, A. Orbán, V. Farkas, Evaluating the Kernighan-Lin heuristic for hardware/software partitioning, International Journal of Applied Mathematics and Computer Science 17 (2) (2007) 249–267.

[18] R. M. Karp, Reducibility among combinatorial problems, in: R. E. Miller, J. W. Thatcher (Eds.), Complexity of computer computations, Plenum, 1972, pp. 85–103.

[19] M. R. Garey, D. S. Johnson, The complexity of near-optimal graph coloring, Journal of the ACM 23 (1976) 43–49.

[20] U. Feige, J. Kilian, Zero knowledge and the chromatic number, Journal of Computer and System Sciences 57 (1998) 187–199.

[21] D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, Theory of Computing 3 (2007) 103–128.

[22] J. R. Brown, Chromatic scheduling and the chromatic number problem, Management Science 19 (4) (1972) 456–463.

[23] D. W. Matula, G. Marble, J. D. Isaacson, Graph coloring algorithms, in: R. C. Read (Ed.), Graph Theory and Computing, Academic Press, 1972, pp. 109–122.

[24] D. Brélaz, New methods to color the vertices of a graph, Communications of the ACM 22 (4) (1979) 251–256.

[25] P. Cheeseman, B. Kanefsky, W. M. Taylor, Where the really hard problems are, in: 12th International Joint Conference on Artificial Intelligence (IJCAI '91), 1991, pp. 331–337.

[26] J. Culberson, I. Gent, Frozen development in graph coloring, Theoretical Computer Science 265 (1-2) (2001) 227–264.

[27] T. Szép, Z. Mann, Graph coloring: the more colors, the better?, in: Proceedings of the 11th IEEE International Symposium on Computational Intelligence and Informatics, 2010, pp. 119–124.

[28] T. Hogg, C. P. Williams, The hardest constraint problems: A double phase transition, Artificial Intelligence 69 (1-2) (1994) 359–377.

[29] T. Hogg, Refining the phase transition in combinatorial search, Artificial Intelligence 81 (1-2) (1996) 127 – 154.

[30] J. S. Turner, Almost all k-colorable graphs are easy to color, Journal of Algorithms 9 (1) (1988) 63–82.

[31] R. Monasson, On the analysis of backtrack procedures for the coloring of random graphs, in: E. Ben-Naim, H. Frauenfelder, Z. Toroczkai (Eds.), Complex Networks, Springer, 2004, pp. 235–254.

[32] A. Coja-Oghlan, M. Krivelevich, D. Vilenchik, Why almost all k-colorable graphs are easy to color, Theory of Computing Systems 46 (3) (2010) 523–565.

[33] H. Jia, C. Moore, How much backtracking does it take to color random graphs? Rigorous results on heavy tails, in: Principles and Practice of Constraint Programming (CP 2004), 2004, pp. 742–746.

[34] C. Gomes, B. Selman, N. Crato, H. Kautz, Heavy-tailed phenomena in satisfiability and constraint satisfaction problems, Journal of Automated Reasoning 24 (1-2) (2000) 67–100.

[35] C. Gomes, B. Selman, H. Kautz, Boosting combinatorial search through randomization, in: Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), 1998, pp. 431–437.

[36] G. R. Grimmett, C. J. H. McDiarmid, On colouring random graphs, Mathematical Proceedings of the Cambridge Philosophical Society 77 (2) (1975) 313–324.

[37] E. Shamir, E. Upfal, Sequential and distributed graph coloring algorithms with performance analysis in random graph spaces, Journal of Algorithms 5 (1984) 488–501.

[38] W. F. de la Vega, On the chromatic number of sparse random graphs, in: B. Bollobás (Ed.), Graph Theory and Combinatorics, Academic Press, 1984, pp. 321–328.

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

[40] D. Achlioptas, C. Moore, Almost all graphs with average degree 4 are 3-colorable, Journal of Computer and System Sciences 67 (2003) 441–471.

[41] B. Pittel, R. S. Weishaar, On-line coloring of sparse random graphs and random trees, Journal of Algorithms 23 (1997) 195–205.

[42] A. Coja-Oghlan, A. Taraz, Exact and approximative algorithms for coloring G(n,p), Random Structures and Algorithms 24 (3) (2004) 259–278.

[43] C. Sommer, A note on coloring sparse random graphs, Discrete Mathematics 50 (2009) 3381–3384.

[44] S. Cocco, R. Monasson, Trajectories in phase diagrams, growth processes and computational complexity: how search algorithms solve the 3-satisfiability problem, Phys. Rev. Lett. 86 (2001) 1654.

[45] L. Ein-Dor, R. Monasson, The dynamics of proving uncolourability of large random graphs. I. Symmetric colouring heuristic, Journal of Physics A: Mathematical and General 36 (2003) 11055–11067.

[46] R. Monasson, A generating function method for the average-case analysis of DPLL, in: Proceedings of APPROX-RANDOM '05, 2005, pp. 402–413.

[47] C. Moore, S. Mertens, The Nature of Computation, Oxford University Press, 2011.

[48] D. Achlioptas, A. Coja-Oghlan, Algorithmic barriers from phase transitions, in: 49th Annual IEEE Symposium on Foundations of Computer Science, 2008, pp. 793–802.