# ON THE TRACTABILITY OF COLORING SEMIRANDOM GRAPHS

JULIA BÖTTCHER AND DAN VILENCHIK

Abstract. As part of the efforts put in understanding the intricacies of the $k$-colorability problem, different distributions over $k$-colorable graphs were analyzed. While the problem is notoriously hard (not even reasonably approximable) in the worst case, the average case (with respect to such distributions) often turns out to be "easy". Semi-random models mediate between these two extremes and are more suitable to imitate "real-life" instances than purely random models. In this work we consider semi-random variants of the planted $k$-colorability distribution. This continues a line of research pursued by Coja-Oghlan [7] and by Krivelevich and Vilenchik [20]. Our aim is to study a more general semi-random framework than suggested there. On the one hand we show that the algorithmic techniques developed in [20] extend to our more general semi-random setting; on the other hand we give a hardness result, proving that a closely related semi-random model is intractable. Thus, we provide some indication about which properties of the input distribution make the $k$-colorability problem hard.

**keywords:** graph algorithms, average case analysis, semi-random models, k-coloring

## 1. Introduction and Definitions

A (proper) *$k$-coloring* of a graph $G = (V, E)$ is a partition $V_1 \dot\cup \ldots \dot\cup V_k = V$ of its vertex set such that $u \in V_i$ and $v \in V_j$ with $i \neq j$ for every edge $uv \in E$. We will also denote this coloring by simply writing $V_1, \ldots, V_k$. The *chromatic number* $\chi(G)$ of $G$ is the minimal $k$ such that there is a $k$-coloring for $G$.

In the graph $k$-coloring problem we are asked to produce a $k$-coloring of a $k$-colorable graph $G$ for given $k$ and $G$. This problem is of course well known to be NP-hard for $k \geq 3$, and although good approximation algorithms are at hand for several NP-hard problems, this is not the case for the $k$-coloring problem. In fact, so far all known approximation algorithms for this problem use as many as $n^{\alpha(k)}$ colors in general, where $n$ is the number of vertices of the input graph and $\alpha(k)$ depends on $k$ only [2, 3, 16, 19]. In addition, Dinur, Mossel, and Regev [8] recently gave some evidence that an approximation of the $k$-coloring problem within a constant factor is unlikely. This is not surprising, since (turning to the case where $k$ is not necessarily fixed) almost a decade ago Feige and Kilian [9] proved that no polynomial time algorithm approximates the chromatic number $\chi(G)$ within a factor of $n^{1-\epsilon}$ for all input graphs $G$.

**Average Case Analysis.** The wide range of worst-case NP-hardness and inapproximability results for problems in graph theory motivates the study of heuristics that give "useful" answers for a "typical" subset of the problem instances, where "useful" and "typical" are often not well defined. One way of evaluating and comparing heuristics is by running them on a collection of input graphs ("benchmarks"), and checking which heuristic usually performs better. Though empirical results of this type are sometimes informative, we seek more rigorous measures of evaluating heuristics. One possibility of rigorously modeling typical instances is to use a random distribution over the set of possible inputs and design algorithms that work *with high probability* (*whp* for short), i.e. with probability tending to 1 as $n$ goes to infinity, with respect to this distribution. In many settings the random graph $\mathcal{G}_{n,p}$, generated by including each of the $\binom{n}{2}$ possible edges on $n$ vertices with probability $p = p(n)$ independently, is used for this purpose. However, $\mathcal{G}_{n,p}$ is not suitable for the study of $k$-colorable graphs (when thinking of $k$ as constant but allowing for an arbitrarily large average degree $np$): for most values of $p$ the chromatic number of $\mathcal{G}_{n,p}$ is known to be roughly $np/(2\ln(np))$ [5, 22].

*Date*: December 6, 2007.
The first was supported by a Minerva grant and DFG grant TA 309/2-1.

To enable the average case study of graphs with a prescribed chromatic number $k$, Kučera [21] suggested the following model for generating random $k$-colorable graphs, denoted by $\mathcal{G}_{n,p,k}$. First randomly partition the vertex set $V = \{1, ..., n\}$ into $k$ classes $V_1, ..., V_k$ of size $n/k$ each. Then for every $i, j \in [k]$ with $i \neq j$, independently include every edge connecting a vertex in $V_i$ with a vertex in $V_j$ with probability $p = p(n, k)$; this type of random graph model is sometimes called a *planted model*. The aim is then to develop an efficient algorithm that solves the corresponding search problem (finding the *planted coloring* $V_1, ..., V_k$ in our case). For $\mathcal{G}_{n,p,k}$ it turned out that this problem is comparably complicated in the *sparse regime*, i.e. when the average degree $np$ does not tend to infinity with $n$ (but does possibly depend on $k$). In 1997 Alon and Kahale established the following result.

**Theorem 1** (Alon & Kahale [1])**.** *There exists a polynomial time algorithm that* whp *finds a $k$-coloring of $\mathcal{G}_{n,p,k}$ with $np \geq d_0k^2$ where $d_0$ is a sufficiently large constant.*

The proof of this theorem is constructive in the sense that an algorithm, based on spectral techniques, is described. Similar results for the same or different planted models followed (see [17, 6, 11, 12, 13], to mention just a few).

**Semi-random Models.** The main drawback of random models is that they may not capture the space of "useful" instances we have in mind: the various results on random graph models (see e.g. [18]) show that instances generated by such models typically have a very special structure, which will probably not reflect "real-world" examples. Further, there is the temptation of over-exploiting the statistical properties of random graphs (eigenvalue structure, vertex degrees, etc.) and designing algorithms that perform well on a specific distribution but fail completely when the distribution is slightly changed. Since it is desirable to have algorithms that are robust in this sense, semi-random models were introduced. In the semi-random setting, first a random instance $I_0$ is generated. Then a computationally unlimited adversary may further change $I_0$. Of course these modifications cannot be arbitrary (otherwise the adversary can transform any $I_0$ into a worst-case instance) and therefore the operations the adversary is allowed to perform on $I_0$ are usually restricted. By blending random and adversarial decisions, semi-random models intermediate between the overly pessimistic worst-case and the idealized purely random case. As such, they often serve as a driving force towards designing more natural and efficient algorithms (e.g., introducing semi-definite programming not only as an important tool in approximation algorithms but rather as part of heuristics that solve typical and adversarial instances [10, 11]).

The following semi-random variant $\mathcal{G}^*_{n,p,k}$ of $\mathcal{G}_{n,p,k}$ was suggested by Blum and Spencer [4]. First, a graph $G_0 = \mathcal{G}_{n,p,k}$ is generated. Then, an adversary is allowed to add arbitrary $V_i - V_j$ edges for $i \neq j$. The algorithm of Alon and Kahale fails on $\mathcal{G}^*_{n,p,k}$ (even for super-constants values of $np$) because the adversary can completely jumble the spectrum of the graph $G_0$. However, Blum and Spencer [4] described a polynomial time algorithm that works *whp* for $\mathcal{G}^*_{n,p,k}$ when $np \geq n^{\alpha_k}$, $\alpha_k \geq 2/5$. Feige and Kilian [10] improved upon this result, giving an SDP-based algorithm that $k$-colors $\mathcal{G}^*_{n,p,k}$ for $np \geq c(1+\epsilon)k \ln n$. Coja-Oghlan [7] gave a simpler SDP-based heuristic that $k$-colors $\mathcal{G}^*_{n,p,k}$ for $np \geq c(1+\epsilon)k \ln n$. Improving upon a hardness result of Feige and Kilian [10] he moreover shows the following theorem.

**Theorem 2** (Coja-Oghlan [7])**.** *It is NP-hard to $k$-color $\mathcal{G}^*_{n,p,k}$ for $np \leq (1-\varepsilon)\frac{k}{2} \ln \frac{n}{k}$ unless NP is contained in RP.*

For constant values of $np$ this means that the coloring problem on $\mathcal{G}^*_{n,p,k}$ is as difficult as in the worst-case world; accordingly, this model is not suitable to study the sparse regime of $\mathcal{G}_{n,p,k}$. In the next section we will address the question of how to deal with this difficulty.

## 2. Our Contribution

In this work we focus on the sparse regime of $\mathcal{G}_{n,p,k}$, i.e. we suggest a semi-random framework for the case $np = O(1)$. As discussed in the last section, $\mathcal{G}^*_{n,p,k}$ is tractable for $np = \Omega(\log n)$ but gets hard for $np = O(1)$. It follows that for defining our model we have to limit the power of the adversary.

Let $\mathcal{A}$ be some vertex property. We define $\mathcal{G}^{\mathcal{A}}_{n,p,k}$ to be the semi-random graph model where first a random graph $G_0 = \mathcal{G}_{n,p,k}$ is generated and then an adversary may add edges between vertices with property $\mathcal{A}$ as long as they respect the planted coloring of $G_0$. We will usually abstractly define a set $A$ (where $A$ may depend on the choice of $G_0$) as the set of all vertices with property $\mathcal{A}$ and then simply write $\mathcal{G}^{A}_{n,p,k}$ instead of $\mathcal{G}^{\mathcal{A}}_{n,p,k}$. This defines a family of semi-random models, where taking $A = V$ gives $\mathcal{G}^{*}_{n,p,k}$ and setting $A = \emptyset$ is simply $\mathcal{G}_{n,p,k}$.

In [20], $A$ was a carefully chosen set of vertices that depends on $G_0$ and typically contains most vertices, let us call it $H$ (in this paper we will call this set a super-core, see Definition 3 below). The authors of [20] illustrate that the algorithm of Alon and Kahale fails on the semi-random graph model $\mathcal{G}^{H}_{n,p,k}$ (for all values of $np$). They also prove, however, that an appropriate adaptation of this algorithm works *whp* on $\mathcal{G}^{H}_{n,p,k}$ for $np \geq d_0k^2$ where $d_0$ is some sufficiently large constant. Thus, the $k$-coloring problem remains tractable on $\mathcal{G}^{H}_{n,p,k}$ in the sparse setting.

The main disadvantage of the set $H$ is that its definition depends on $np$. Furthermore, the semi-random model $\mathcal{G}^{H}_{n,p,k}$ imposes some arguably unnatural restrictions on the adversary. For example, the adversary cannot add edges that connect vertices of very high degree. However one may expect such vertices to be the ones most easily colored by an algorithm. Extending the work of [20] we suggest a more natural and general semi-random model, which does not depend on $np$ and eases the restrictions. Before stating our results we need some more definitions.

**Definition 3.** *$J_c$ is called a $c$-**core** of a $k$-colorable graph $G_0$ (w.r.t a $k$-coloring $V_1, \ldots, V_k$ of $G_0$) if every $v \in J_c \cap V_i$ has at least $c$ neighbors in $J_c \cap V_j$ for every $i \neq j$. $H$ is called a **super-core** of a $k$-colorable graph $G_0$ (w.r.t a $k$-coloring $V_1, \ldots, V_k$ of $G_0$) if every $v \in H \cap V_i$ has $(1 \pm 0.01)np/k$ neighbors in $H \cap V_j$ for every $i \neq j$. When there is no danger of confusion we may also omit the coloring we refer to.*

By definition if $J_c$ and $J'_c$ are $c$-cores of a graph $G_0$ (w.r.t. the same coloring) then also $J_c \cup J'_c$ is a $c$-core of $G_0$. Therefore, we may speak of a unique maximal $c$-core. Hence from now on, when referring to *the $c$-core of a graph* we mean this maximal one (notice however, that a maximal super-core $H$ need not be unique). Observe further that $J_d \subseteq J_c$ if $d \geq c$. This explains why we concentrate on the case $c = 3$ in the following; the case $c = 2$ seems to be more complicated, see the discussion in Section 8. As we will see later, the 3-core of $\mathcal{G}_{n,p,k}$ typically contains all but a tiny (though still constant) fraction of the vertices (see Lemma 8 in Section 3). This property will be crucial in the proof of our first result, and also shows that the adversary can jumble a large portion of the input graph.

**Theorem 4.** *Let $J_3$ denote the 3-core of a graph. There exists a polynomial time algorithm that* whp *$k$-colors $\mathcal{G}^{J_3}_{n,p,k}$ when $np \geq d_0k^2$, $d_0$ a sufficiently large constant.*

A result of similar flavor was proven in [20], with the 3-core $J_3$ replaced by a super-core. As we mentioned already, considering $J_3$ is more natural in different respects. First of all it is not difficult to see that it contains every super-core. In addition, $J_3$ is unique and does not depend on $np$.

For proving Theorem 4, we will show in Section 6 that the coloring algorithm used in [20] continues to work when giving the adversary more freedom, i.e. allowing it to add edges to the graph induced by the vertices of $J_3$ instead of a the vertices of the super-core. For this purpose we first need to investigate the properties of typical instances from $\mathcal{G}_{n,p,k}$. This is done in Section 3. Our analysis then relies on a decomposition of $J_3$ that is centered around a super-core $H$ contained in $J_3$. This decomposition is introduced in Section 4. In Section 5 we use this decomposition to show that semidefinite programming can typically be used to color $J_3$ which will turn out to be important for the correctness of the algorithm. Thus, the main contribution of this paper lies in new analytical insights and in suggesting a less strict semi-random framework for sparse $k$-colorable graphs.

A natural question is which of the structural properties of the 3-core are essential for the existence of a polynomial time coloring algorithm as asserted by Theorem 4. One characteristic of a $c$-core is that it rules out vertices of degree smaller than $c(k-1)$. The following theorem shows however that replacing the 3-core by *all vertices of degree at least* $c'$ results in a semi-random graph model that can *not* be $k$-colored in polynomial time *whp* in the sparse regime for every $c'$. This suggests that it is essential in the definition of a $c$-core that its vertices have neighbors in every color class other than their own; otherwise the problem becomes untractable.

**Theorem 5.** *For all $4 \le k$ and $c$ the following is true. Let $F_c$ denote the set of vertices that have degree at least $c$. Unless $NP \subseteq RP$ there is no polynomial time algorithm that $k$-colors $\mathcal{G}^{F_c}_{n,p,k}$* whp *when $np = O(1)$ and $c = O(1)$.*

The proof of this theorem is given in Section 7 and uses techniques developed in [7, 10]. The following question however remains open.

**Question 6.** *Let $U_c$ be the set of vertices such that $v \in U_c \cap V_i$ has at least $c$ neighbors in $V_j$ for every $j \neq i$ (*not necessarily in $U_c$*). Does there exist a polynomial time algorithm that* whp *$k$-colors $\mathcal{G}^{F_c}_{n,p,k}$, for some constant $c$ (in the regime $np = O(1)$)?*

Further discussion related to this problem is provided in Section 8.

## 3. Basic Properties of $\mathcal{G}_{n,p,k}$

In this section we are interested in examining the typical structure of graphs in $\mathcal{G}_{n,p,k}$. The techniques applied here are standard and most of the properties we present (or variants of them) are well-known. For several proofs we will therefore refer the reader to other sources from the area.

Our first lemma states that *whp* a graph from $\mathcal{G}_{n,p,k}$ has no small dense subgraphs. The *density* $d(F)$ of a graph $F = (V, E)$ is the average degree of $F$, i.e. $d(F) := 2|E|/|V|$.

**Lemma 7.** *Let $G_0$ be a random graph in $\mathcal{G}_{n,p,k}$, and let $\delta \le (10np)^{-3}$. There* whp *exists* no *subgraph $F$ of $G_0$ with $|V(F)| \le \delta n$ and density $d(F) \ge 3$.*

This lemma is proved using a simple first moment calculation. Details omitted. It is not possible to replace $d(F) \ge 3$ by $d(F) \ge 2$ in Lemma 7, which is not a mere weakness of the proof technique, but a true obstacle: a random graph with $np > 1$ contains *whp* "short" cycles (see [18], Chapter 5). This fact actually translates to the difficulty in (immediately) replacing $J_3$ with $J_2$ in Theorem 4.

We will now examine the structure of $\mathcal{G}_{n,p,k}$ in more detail and investigate some properties of cores and super-cores. We will show that a super-core *whp* covers most vertices of $\mathcal{G}_{n,p,k}$ and is contained in *any* 3-core (Lemma 8). The remaining vertices consist of connected components of size $O(\log_k n)$ only (Lemma 10).

**Lemma 8.** *Let $G_0$ be a graph from $\mathcal{G}_{n,p,k}$ with $np \ge d_0 k^2$, $d_0$ a sufficiently large constant (independent of $k$). Let $J_3$ be the 3-core and $H$ a super-core of $G$ both w.r.t. the planted $k$-coloring $V_1, \dots, V_k$. Then,* whp

(*a*) $|H_i|/|V_i| \ge 1 - \exp\big(-\Omega(np/k)\big)$, *and*
(*b*) $H \subseteq J_3$

*where $H_i = H \cap V_i$.*

*Proof.* (Outline) Part (*a*) was proven in [20] (Lemma 6 (*a*)). For part (*b*) note that for $c \le 0.99np/k$ a super-core is in particular a $c$-core and thus also a 3-core. By the uniqueness and maximality of the 3-core the claim follows. □

**Corollary 9.** *Let $H$ be as in Lemma 8. Then $|H| \ge n - k \cdot \exp\big(-\Omega(np/k)\big)\frac{n}{k} \ge (1 - (10np)^{-3})n$.*

**Lemma 10.** *Let $G_0 = \mathcal{G}_{n,p,k}$ with $np \ge d_0 k^2$, $d_0$ a sufficiently large constant. Let $J_3$ be the 3-core of $G$ w.r.t. the planted coloring, and let $G[V \setminus J_3]$ be the graph induced by the non-core vertices. Then* whp *the largest connected component in $G_0[V \setminus J_3]$ is of size $O(\log_k n)$.*

*Proof.* In [1] it is proven that if $H$ is a super-core of $G$ then $G[V \setminus H]$ satisfies the required property. Since $H \subseteq J_3$ it follows immediately that also $G[V \setminus J_3]$ satisfies the property. □

## 4. A decomposition of the 3-core

In order to use the structural properties established in the previous section we need to take a closer look at the 3-core $J_3$ of $\mathcal{G}_{n,p,k}$. For this purpose we introduce a decomposition of $J_3$ "centered" around some maximum super-core $H$ contained in $J_3$. We will start by defining the notion of safe vertices in $J_3$. Lemma 12 then demonstrates that this notion indeed provides a decomposition of $J_3$.

**Definition 11.** *(safe vertices) Let $G$ be a graph from $\mathcal{G}_{n,p,k}$ with super-core $H$ contained in a 3-core $J_3$. A vertex $v$ in $J_3$ is called* 0-safe *(w.r.t. $H$) if it belongs to $H$ and* $i$-safe *(w.r.t. $H$) if in every color class (other than its own) $v$ has at least one neighbor which is $j$-safe with $j < i$. We say that $v$ is* safe *(w.r.t. $H$), if it is $i$-safe for some $i$. Otherwise $v$ is* unsafe.

**Lemma 12.** *Let $G_0 = \mathcal{G}_{n,p,k}$, $np \geq d_0 k^2$ where $d_0$ is a sufficiently large constant. Further, let $H$ be a super-core contained in the 3-core $J_3$ of $G_0$. Then* whp *every vertex in $J_3$ is safe w.r.t. $H$.*

*Proof.* By contradiction, let $U \subseteq J_3$ be the set of vertices which are unsafe. Now, consider some vertex $v$ in $U$. By definition there is a color class $V_j$ such that $v$ has no safe neighbor in $V_j$. Since $v \in J_3$ it follows that $v$ has at least 3 unsafe neighbors in $V_j$. Thus $G[U]$ has minimum degree at least 3. Further, $|U| \leq |J_3 \setminus H| \leq (1-(10np)^{-3})n$ (Corollary 9); this however contradicts Lemma 7. □

We conclude this section with the observation that Lemma 12 remains true for $\mathcal{G}^{J_3}_{n,p,k}$ instead of $\mathcal{G}_{n,p,k}$.

**Observation 13.** *Let $G_0$ be a graph from $\mathcal{G}_{n,p,k}$ and assume that we constructed a decomposition of its core $J_3$ into sets $S_i$ of $i$-safe vertices (w.r.t. some super-core $H$). Now, observe that if we construct a semi-random graph $G$ from $G_0$ we only add edges. Therefore vertices of $S_i$ are also $i'$-safe in $G$ with $i' \leq i$.*

## 5. Using semidefinite programming for coloring the 3-core of $\mathcal{G}_{n,p,k}$

In this section we prove that the planted coloring of the 3-core $J_3$ of $\mathcal{G}_{n,p,k}$ can be typically determined in polynomial time. The algorithmic tool that we will use in order to establish this results is the following semidefinite programming relaxation for the Max $k$-Cut problem suggested by Frieze and Jerrum [14].

$$SDP_k(G) := \max \sum_{(u,v)\in E} \frac{k-1}{k}(1-\langle x_u, x_v\rangle)$$
$$\text{s.t. } \forall\, u,v \in V,\ \langle x_u, x_v\rangle \geq -\frac{1}{k-1},$$

where the maximum is taken over all families $(x_v)_{v\in V}$ of unit vectors in $\mathbb{R}^{|V|}$. A *solution* of $SDP_k(G)$ is a family $(x_v)_{v\in V}$ for which the maximum is attained. A non-empty vertex set $A = \{v \in V : x_v = x\}$ for some fixed $x \in \mathbb{R}^{|V|}$ is also called a *color class* of this solution. Restricting the choice of the vectors $x_v$ to the vertices of an equilateral $(k-1)$-dimensional simplex $S$ on the unit sphere in $\mathbb{R}^{|V|}$ shows that the optimal value of $SDP_k(G)$ gives an upper bound on the size of a maximum $k$-cut $C$ in $G$: simply assign one vertex of $S$ to each of the classes of $C$. For a $k$-colorable graph $G = (V,E)$ we clearly have $|C| = |E|$ and because $SDP_k(G) \leq |E|$ by definition we get that the optimal value of $SDP_k(G) = |E|$. This implies the following observation.

**Observation 14.** *For a $k$-colorable graph $G = (V,E)$ a solution $(x_v)_{v\in V}$ of $SDP_k(G)$ satisfies $\langle x_u, x_v\rangle = -\frac{1}{k-1}$ for all $uv \in E$.*

Since $SDP_k$ is a semidefinite program, its optimal value (and a corresponding solution) can be computed within any desired precision in time polynomial in $n, k$ and the encoding of the tolerated numerical error (e.g. using the ellipsoid method [15, 19]).

**Definition 15.** *Let $G=(V,E)$ be a graph that admits a $k$-coloring $V_1,\dots,V_k$ A family $(x_v)_{v\in V}$ of vectors indexed by the vertices of $G$ is called* integral on the set $A\subseteq V$ *(w.r.t. this $k$-coloring), if for all $i,j\in[k]$ with $i\neq j$ the following holds. For all $s,s'\in V_i\cap A$ and $t\in V_j\cap A$ we have $v_s=v_{s'}$ and $\langle x_s,x_t\rangle=-\frac{1}{k-1}$. Moreover, we say that $(x_v)_{v\in V}$ is* non-degenerate on $A$ *if $(x_v)_{v\in A}$ spans at least $k$ color classes.*

Note, that a solution of $SDP_k$ that is integral on a vertex set $A$ allows us to immediately construct a $k$-coloring of $A$: simply group the vertices according to the color classes of the solution. Therefore it is desirable to identify such sets $A$. The following lemma from [20] states that a solution of $SDP_k$ is typically integral on the super-core when the input graphs are distributed according to $G^*_{n,p,k}$. The proof of this lemma is rather technical and based on ideas from [7].

**Lemma 16** (Krivilevich & Vilenchik [20])**.** *Let $G=\mathcal{G}^*_{n,p,k}$ with $np\geq d_0k^2$, $d_0$ a sufficiently large constant, and let $H$ be a super-core of the underlying graph $G_0$ taken from $\mathcal{G}_{n,p,k}$. Then* whp *a solution $(x_v)_{v\in V}$ of $SDP_k(G)$ is integral on $H$.*

Our next step is to prove that Lemma 16 is also true for the 3-core $J_3$ of $\mathcal{G}^*_{n,p,k}$. For this we will make use of the decomposition of $J_3$ into $i$-safe vertices suggested in Section 3 (see Definition 11).

**Lemma 17.** *Let $G=\mathcal{G}^*_{n,p,k}$, $np\geq d_0k^2$, $d_0$ a sufficiently large constant and let $J_3$ be the 3-core of $G$. Then* whp *a solution $(x_v)_{v\in V}$ of $SDP_k(G)$ is integral on $J_3$.*

*Proof.* Let $H$ be the super-core of the underlying random graph $G_0$ of $G$, and let $V_1,\dots,V_k$ be the planted $k$-coloring. Further, denote by $S_j$ the set of $j$-safe vertices in $J_3$ (w.r.t. $H$) and let $J_3^i:=\cup_{j\leq i}S_j$. We will proof by induction on $i$ that $(x_v)_{v\in V}$ is integral and non-degenerate on $J_3^i$. Since by Lemma 12 and Observation 13 each vertex in $J_3$ is safe (w.r.t. $H$) the claim follows.

The base case is given by Lemma 16 as $J_3^0=H$. Lemma 8 states that $J_3^0\cap V_j\neq\emptyset$ for all $j\in[k]$ and therefore $(x_v)_{v\in V}$ is non-degenerate on $J_3^0$. For the inductive step assume the claim is true for $i-1$, i.e. that $(x_v)_{v\in V}$ is integral and non-degenerate on $J_3^{i-1}$. This means that there are vectors $y_1,\dots,y_k$ fulfilling $\langle y_j,y_{j'}\rangle=-\frac{1}{k-1}$ for all $j,j'\in[k]$ with $j\neq j'$ such that $x_s=y_j$ whenever $s\in J_3^{i-1}\cap V_j$. Trivially, $(x_v)_{v\in V}$ is also non-degenerate on $J_3^i$. Moreover, we have

$$\big\langle\, y_1+\cdots+y_k\,,\,y_1+\cdots+y_k\,\big\rangle=\sum_{j=1}^{k}\langle y_j,y_j\rangle+\sum_{j,j'\in[k],j\neq j'}\langle y_j,y_{j'}\rangle=k-k(k-1)\frac{1}{k-1}=0$$

which implies $y_k=-(y_1+\cdots+y_{k-1})$. Now consider an $i$-safe vertex $v\in J_3$. Without loss of generality we assume that $v\in V_k$. By definition $v$ has neighbors $v_1,\dots,v_{k-1}$ with $v_j\in J_3^{i-1}\cap V_j$ and by the induction hypothesis we have $x_{v_j}=y_k$. We need to prove that this implies $x_v=v_k$. From the optimality of $(x_v)_{v\in V}$ for $SDP_k(G)$ and Observation 14 it follows that $\langle y_j,x_v\rangle=-\frac{1}{k-1}$ for all $1\leq j\leq k-1$. Similarly as above we conclude

$$\big\langle\, y_1+y_2+\ldots+y_{k-1}+x_v\,,\,y_1+y_2+\ldots+y_{k-1}+x_v\,\big\rangle=0$$

and thus $x_v=-(y_1+\cdots+y_{k-1})=y_k$ which completes the proof of the inductive step. □

Observe that, since $\mathcal{G}^{J_3}_{n,p,k}$ is a special case of $\mathcal{G}^*_{n,p,k}$, the lemmas of this section remain true if $\mathcal{G}^*_{n,p,k}$ is replaced by $\mathcal{G}^{J_3}_{n,p,k}$.

## 6. Proof of Theorem 4

For proving Theorem 4 and coloring $\mathcal{G}^{J_3}_{n,p,k}$ we use the following algorithm (a variant of) which was described in [20]. A vertex $v$ of a partially colored graph $G$ is called *suspicious* if it has less than 3 neighbors in some color other than its own (w.r.t. a given, not necessarily proper, coloring of the graph).

**Algorithm 1**: Color($\mathcal{G}_{n,p,k}^{J_3}$)

1 Compute an optimal solution $(x_v)_{v \in V}$ for $SDP_k(G)$ ;
  **foreach** $i \in [k]$ **do**
2     Color the vertices of the $i$-biggest color class of $(x_v)_{v \in V}$ with $i$ ;
3 Iteratively, while there exists a suspicious vertex, uncolor it ;
  Let $U$ be the set of all uncolored vertices ;
4 **if** some component of $G[U]$ has size $C_0 \log_k n$ **then** give up ;
5 Exhaustively color the components of $G[U]$ ;

As we already mentioned, the optimal solution to $SDP_k(G)$ can be computed (up to any numerical precision) in polynomial time. In the exhaustive search (Step 5) we use a brute force algorithm to $k$-color the components of $G[U]$. Since Step 4 guarantees that all these components are of size at most $C_0 \log_k n$, the exhaustive search consumes no more than $n \cdot k^{C_0 \log_k n} = n^{C_0+1}$ steps, where $C_0$ is the implicit constant in Lemma 10. All other steps of the algorithm are clearly also polynomial. Thus it remains to prove that typically Algorithm 1 finds a legal $k$-coloring of the input graph when the graph is sampled according to $\mathcal{G}_{n,p,k}^{J_3}$, $np \geq d_0 k^2$, $d_0$ some sufficiently large constant. In the analysis that follows we assume hat $G$ is a typical graph from $\mathcal{G}_{n,p,k}^{J_3}$, i.e. $G$ satisfies the properties studied in Section 3. As before, let $J_3$ denote the 3-core of $G$ and $H$ the super-core of the underlying graph $G_0$ from $\mathcal{G}_{n,p,k}$.

**Claim 18.** *The SDP step of the algorithm (Step 1 and Step 2 of Algorithm 1) colors $J_3$ according to the planted coloring.*

*Proof.* By Lemma 16 a solution $(x_v)_{v \in V}$ of $SDP_k$ is integral on $J_3$. Moreover, by Lemma 8 $(a)$, $|J_3 \cap V_i| \geq |H \cap V_i| \geq (1 - \exp(-C_0 \cdot np/k))\frac{n}{k}$ for all $i \in [k]$ and therefore $H \cap V_i$ is contained in one of the $k$ biggest color classes of $(x_v)_{v \in V}$ for every $i$. It follows that all vertices of $J_3 \cap V_i$ are colored (w.l.o.g.) by color $i$ when Step 2 ends. □

Moreover, by Lemma 8 $(a)$, $|J_3 \cap V_i| \geq |H \cap V_i| \geq (1 - \exp(-C_0 \cdot np/k))\frac{n}{k}$ for all $i \in [k]$ and therefore the color class of $(x_v)_{v \in V}$ containing $H \cap V_i$ is among the $k$ biggest color classes of $(x_v)_{v \in V}$. It follows that in all vertices of $H \cap V_i$ are colored by $i$ in Step 2 (up to a permutation of the colors). Observe also, that the SDP step produces a legal partial $k$-coloring of $G$ by the definition of $SDP_k$. Next, we will show that the uncoloring step (Step 3 of Algorithm 1) produces a partial coloring that coincides with the planted coloring and, moreover, leaves all vertices of the super-core colored. For this we use ideas from [1].

**Claim 19.** *After the uncoloring step (Step 3 of Algorithm 1), all colored vertices are colored according to the planted coloring and $J_3 \cap U = \emptyset$.*

*Proof.* We will first prove the second claim. Assume to the contrary, that there is some first vertex $v$ of $J_3$ that gets uncolored. Without loss of generality, let $v \in V_k$. However, by definition, $v$ has at least 3 neighbors in $J_3 \cap V_i$ for each $i \in [k-1]$, which are all colored correctly by Claim 18. Thus $v$ is not suspicious and does therefore not get uncolored, a contradiction.

To see the first claim, let $W$ be the set of vertices not colored according to the planted coloring (but not uncolored) when the uncoloring step ends, and assume without loss of generality that there is some vertex $v \in W \cap V_1$ colored $k$. Since $v$ did not get uncolored, it has at least 3 neighbors colored with 1 and from $v \in V_1$ it follows that all these vertices are contained in $W$ as well; therefore, $G[W]$ has density at least 3. Further, $W \cap J_3 = \emptyset$ by Claim 18 and the first part of this proof; since the adversary cannot add any edges to $W$, also the density of $G_0[W]$ is bounded from below by 3. Moreover, $|W| \leq n - |J_3| \leq n - |H| \leq (10np)^{-3}n$ (Corollary 9), contradicting Lemma 7. □

Claim 19 and Lemma 10 imply that all components of $U$ are of size at most $C_0 \log_k n$. Therefore Algorithm 1 typically does not fail in Step 4. By Claim 19 the partial coloring can be extended to a coloring of the whole graph. It follows that the exhaustive search (Step 5 of Algorithm 1) eventually assigns a color to every vertex, and thus we conclude that Algorithm 1 produces a $k$-coloring of $G$ which verifies Theorem 4.

## 7. Proof of Theorem 5

In the proof of Theorem 5 we follow the strategy of [7]. We will need the following easy lemma.

**Lemma 20.** *For all $4 \le k$ and $c = O(1)$ there is an $\varepsilon$ such that the following is true. Let $G_0$ be a graph from $G_{n,p,k}$ with planted coloring $V_1, \ldots, V_k$ and such that $np = O(1)$.* Whp *each color class $V_i$ with $i \in [3]$ has at least $\varepsilon n/k$ vertices that have no neighbors in $V_1 \cup V_2 \cup V_3$ and at least $c$ neighbors in $V_4 \cup \cdots \cup V_k$.*

*Proof.* Let $d := np$ and choose $\varepsilon$ such that $\exp(-4d/k) \ge \sqrt{\varepsilon}/2$ and $(d/4c)^c \exp(-2d/5) \ge \sqrt{\varepsilon}/2$. The probability that a vertex $v \in V_i$ for $i \in [3]$ has no neighbor in $V_1 \cup V_2 \cup V_3$ is at least

$$(1-p)^{2n/k} \ge \exp\left(-2p\frac{2n}{k}\right) = \exp(-4d/k) \ge \sqrt{\varepsilon}/2.$$

On the other hand, the probability that $v$ has at least $c$ neighbors in $V_4 \cup \ldots V_k$ is greater than

$$\begin{aligned}\binom{n-3\frac{n}{k}}{c} p^c (1-p)^{n-\frac{3n}{k}-c} \ge \binom{\frac{n}{4}}{c} p^c (1-p)^{\frac{n}{5}} &\ge \left(\frac{np}{4c}\right)^c \exp\Big(-2p\frac{n}{5}\Big) \\ &\ge (d/4c)^c \exp(-2d/5) \ge \sqrt{\varepsilon}/2.\end{aligned}$$

We conclude that the probability that $v$ has both required conditions is at least $\varepsilon/2$. Therefore there are at least $(n/k)(\varepsilon/2)$ such vertices $v$ in $V_i$ in expectation. By the Chernoff bound we have concentration around the expected value and it follows that *whp* there are at least $\varepsilon n/k$ such vertices in $V_i$. □

*Proof of Theorem 5.* Let $\varepsilon$ be the constant guaranteed by Lemma 20 for $k$ and $c$ and assume that $\mathcal{G}^{F_c}_{n,p,k}$ can be $k$-colored *whp* in polynomial time by an algorithm $\mathcal{A}$. We will show that we can use $\mathcal{A}$ to solve 3-colorability in randomized polynomial time and thus derive a contradiction.

Let $H$ be an arbitrary 3-colorable graph with color classes of size $\varepsilon\frac{n}{k}$ and let $G_0$ be a graph from $\mathcal{G}_{n,p,k}$. Our adversary then pursues the following strategy. First it finds a $k$-coloring of $G_0$ with color classes $V1\dot\cup \ldots \dot\cup V_k$ of equal size and with vertex sets $W_i \subseteq V_i$ with $i \in [3]$ of size $\varepsilon\frac{n}{k}$ such that each $v \in W_i$ has no neighbors in $V_1 \cup V_2 \cup V_3$ and at least $c$ neighbors in $V_4 \cup \cdots \cup V_k$. Such a coloring exists *whp* by Lemma 20 (if not, we give up on finding a 3-coloring for $H$). Then, the adversary inserts all edges $uv$ between $V_i$ and $V_j$ with $i \ne j$ and $v \notin W_1 \cup W_2 \cup W_3$. Furthermore, it permutes the vertices of each color class of $H$ randomly and then inserts the edges of $H$ into $W_1 \cup W_2 \cup W_3$ by mapping color class $i$ of $H$ to $W_i$. The resulting graph $G$ is a graph from $\mathcal{G}^{F_c}_{n,p,k}$. Therefore $\mathcal{A}$ finds a $k$-coloring of $G$ *whp* .

The key observation now is that in the generation of $G$ above two vertices $u \in W_i$ and $v \in W_j$ with $i \ne j$ are indistinguishable in the sense that exchanging $u$ and $v$ in $G$ does not result in a different graph $G$. Thus the distribution of $G$ is equivalent to the following distribution (that can be generated in polynomial time): Given $H$ let $G'$ be a graph on $n$ vertices with color classes $V'_1\dot\cup \ldots \dot\cup V'_k$ of equal size. Select vertex sets $W'_i \subseteq V'_i$ with $i \in [3]$ of size $(\frac{n}{k})^\varepsilon$ and embed $H$ randomly on $W'_1 \cup W'_2 \cup W'_3$ (without taking care of the coloring of $H$). Then, again, insert all edges $uv$ between $V'_i$ and $V'_j$ with $i \ne j$ and $v \notin W'_1 \cup W'_2 \cup W'_3$. It follows that we can use $\mathcal{A}$ to find a $k$-coloring of $G'$ *whp* and thus a 3-coloring of $H$ which gives the desired contradiction. □

## 8. Discussion

In this work we investigate the tractability of semi-random distributions over $k$-colorable graphs where the random part has constant average degree. The main contribution of this work is to extend the machinery currently available for coloring graphs of this type to a more general and natural semi-random framework. Several questions remain open. Recall in this connection that we are interested in the sparse case, as otherwise already the model $\mathcal{G}^*_{n,p,k}$ is tractable [7].

One question is whether one can replace the 3-core $J_3$ by the set $U_c$ of all vertices having degree at least $c$ for some constant $c$ in every color class other than their own. This was already mentioned in Section 2, Question 6 (note the difference from $\mathcal{G}^{J_3}_{n,p,k}$). This question somewhat mediates between our Theorems 4 and 5. Our analysis breaks down when replacing $J_3$ with $U_3$ in Theorem 4 because it heavily relies on the decomposition of $J_3$ around some super-core $H$ of the underlying graph from $\mathcal{G}_{n,p,k}$ suggested in Section 4. Although (as in the case of $J_3$) the set $U_3$ contains every super-core, there need not necessarily be a corresponding decomposition for $U_3$ since we do not have control over the neighbors of $v \in U_3$ *inside* $U_3$. At this point it is not clear if the $k$-coloring problem for this semi-random model is tractable at all.

Another problem is to consider $\mathcal{G}^{J_2}_{n,p,k}$ where $J_2$ denotes the 2-core of the underlying random graph $G_0 = \mathcal{G}_{n,p,k}$. Our analysis also breaks down in this case. As mentioned in Section 3 there is no analogue of Lemma 7 claiming that there are no small subgraphs of density at least 2 in $\mathcal{G}_{n,p,k}$. As a consequence the decomposition of the 3-core introduced in Section 4 does not carry over to the 2-core. As in the previous question it is not clear if the $k$-coloring problem for $\mathcal{G}^{J_2}_{n,p,k}$ is tractable or not.

We feel that solving the above questions should contribute to a better understanding of the nature of the $k$-colorability problem, and hopefully will lead to the development of new algorithmic techniques and new approaches in the analysis of algorithms solving this problem.

## References


1. N. Alon and N. Kahale, *A spectral technique for coloring random 3-colorable graphs*, SIAM J. Comput. **26** (1997), no. 6, 1733–1748.
2. S. Arora, E. Chlamtac, and M. Charikar, *New approximation guarantee for chromatic number*, Proc. 38th ACM Sympos. Theory Comput. (STOC), 2006, pp. 215–224.
3. A. Blum and D. Karger, *An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs*, Inform. Process. Lett. **61** (1997), no. 1, 49–53.
4. A. Blum and J. Spencer, *Coloring random and semi-random $k$-colorable graphs*, J. Algorithms **19** (1995), no. 2, 204–234.
5. Bella Bollobás, *The chromatic number of random graphs*, Combinatorica **8** (1988), no. 1, 49–55.
6. J. Böttcher, *Coloring sparse random $k$-colorable graphs in polynomial expected time*, Proc. of the 30th Internat. Sympos. Math. Found. Comput. Sci. (MFCS), 2005, pp. 156–167.
7. A. Coja-Oghlan, *Coloring semirandom graphs*, Combin. Probab. Comput. **16** (2007), 515–552.
8. I. Dinur, E. Mossel, and O. Regev, *Conditional hardness for approximate coloring*, Proc. 38th ACM Sympos. Theory Comput. (STOC), 2006, pp. 344–353.
9. U. Feige and J. Kilian, *Zero knowledge and the chromatic number*, J. Comput. System Sci. **57** (1998), no. 2, 187–199.
10. ———, *Heuristics for semirandom graph problems*, J. Comput. System Sci. **63** (2001), no. 4, 639–671.
11. U. Feige and R. Krauthgamer, *Finding and certifying a large hidden clique in a semirandom graph*, Random Structures Algorithms **16** (2000), no. 2, 195–208.
12. U. Feige and D. Vilenchik, *A local search algorithm for 3SAT*, Tech. report, Weizmann Institute, 2004.
13. A. Flaxman, *A spectral technique for random satisfiable 3CNF formulas*, Proc. 14th ACM-SIAM Symp. on Discrete Algorithms, 2003, pp. 357–363.
14. A. Frieze and M. Jerrum, *Improved approximation algorithms for MAX $k$-CUT and MAX BISECTION*, Algorithmica **18** (1997), no. 1, 67–81.
15. M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Springer, 1993.
16. E. Halperin, R. Nathaniel, and U. Zwick, *Coloring $k$-colorable graphs using relatively small palettes*, J. Algorithms **45** (2002), no. 1, 72–90.
17. C. Hui and A. Frieze, *Coloring bipartite hypergraphs*, Proc. of the 5th Internat. Conf. Integer Program. Comb. Optim. (IPCO), 1996, pp. 345–358.
18. S. Janson, T. Łuczak, and A. Rucinski, *Random graphs*, Wiley-Interscience, 2000.
19. D. Karger, R. Motwani, and M. Sudan, *Approximate graph coloring by semidefinite programming*, J. ACM **45** (1998), no. 2, 246–265.
20. M. Krivelevich and D. Vilenchik, *Semirandom models as benchmarks for coloring algorithms*, Third Workshop on Analytic Algorithmics and Combinatorics (ANALCO), 2006, pp. 211–221.
21. L. Kučera, *Expected behavior of graph colouring algorithms*, Proc. of the 1977 Internat. Conf. Fundam. of Comput. Theory (FCT), 1977, pp. 447–451.
22. T. Łuczak, *The chromatic number of random graphs*, Combinatorica **11** (1991), no. 1, 45–54.