Semirandom Models as Benchmarks for Coloring Algorithms [*][†]

Michael Krivelevich [‡] Dan Vilenchik [§]

Abstract

Semirandom models generate problem instances by blending random and adversarial decisions, thus intermediating between the worst-case assumptions that may be overly pessimistic in many situations, and the easy pure random case. In the $G_{n,p,k}$ random graph model, the n vertices are partitioned into k color classes each of size n/k. Then, every edge connecting two different color classes is included with probability $p = p(n)$. In the semirandom variant, $G^*_{n,p,k}$, an adversary may add edges as long as the planted coloring is respected. Feige and Killian prove that unless $NP \subseteq BPP$, no polynomial time algorithm works **whp** when $np < (1-\epsilon)\ln n$, in particular when np is constant. Therefore, it seems like $G^*_{n,p,k}$ is not an interesting benchmark for polynomial time algorithms designed to work **whp** on *sparse* instances (np a constant). We suggest two new criteria, using semirandom models, to serve as benchmarks for such algorithms. We also suggest two new coloring heuristics and compare them with the coloring heuristics suggested by Alon and Kahale 1997 and by Böttcher 2005. We prove that in some explicit sense both our heuristics are preferable to the latter.

1 Introduction and Results

Introduction. A *k-coloring* f of a graph $G = (V, E)$ is a mapping from the set of vertices V to $\{1, 2, ..., k\}$. f is a *legal coloring* of G if for every edge $(u, v) \in E$, $f(u) \neq f(v)$. In the graph coloring problem we are given a graph $G = (V, E)$ and are asked to produce a legal k-coloring f with a minimal possible k. Such k is called the chromatic number, commonly denoted by $\chi(G)$. For a broad view of the coloring problem the reader is referred to [19].

The plethora of worst-case NP-hardness results for problems in graph theory motivates the study of heuristics that give "useful" answers for "typical" subset of the problem instances, where "useful" and "typical" are usually 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 gives better results. Though empirical results are sometimes informative, we seek more rigorous measures of evaluating heuristics. A rigorous candidate for the analog of a "useful" answer is the notion of approximation, where the goal of the heuristic is to provide with a solution which is guaranteed to be within small distance from an optimal one. Although approximation algorithms are known for several NP-hard problems, the coloring problem is not amongst them. In particular, Feige and Kilian [10] prove that no polynomial time algorithm approximates $\chi(G)$ within a factor of $n^{1-\epsilon}$ for all input graphs G, unless ZPP=NP.

When very little can be done in the worst case, comparing heuristics' behavior on "average" instances comes in mind. One possibility of rigourously modeling "average" instances is to use random models. A good candidate can be the well known random graph model $G_{n,p}$ introduced by Erdös and Rényi. A random graph G in $G_{n,p}$ consists of n vertices, and each of $\binom{n}{2}$ possible edges is included w.p. $p = p(n)$ independently of the other. Bollobás [5] and Luczak [24] calculated the probable value of $\chi(G_{n,p})$ to be **whp** [1] approximately $np/(2\ln(np))$ for $p \in [C_0/n, \log^{-7} n]$ ([5] actually extends to $p \leq 0.99$ with a somewhat different expression for the chromatic number). Observe that the chromatic number is typically rather high (roughly comparable with the expected degree). In order to consider graphs with a smaller chromatic number, Kučera [23] suggested a model for generating random k-colorable graphs, denoted throughout by $G_{n,p,k}$. First, randomly partition the vertex set $V = \{1, ..., n\}$ into k classes $V_1, ..., V_k$, of size n/k each. Then, for every $i \neq j$, include every possible edge connecting a vertex in V_i with a vertex in V_j (abbreviated $V_i - V_j$ edges) with probability $p = p(n)$. This model is the analog of the planted clique, planted bisection, and planted SAT distributions, studied e.g. in [2], [11], [13], [14].

The Semirandom Model. The main drawback of random models is that they may simply not capture the space of "useful" problems. The instances generated using random models are extremely unstructured (see [16] for example), which probably does not reflect the real-world examples. Further, there is the temptation

[*] Supported in part by USA-Israel BSF Grant 2002-133, and by Grant 526/05 from the Israel Science Foundation.

[†] Part of the second author's PhD thesis prepared at Tel Aviv University under the supervision of Prof. Michael Krivelevich.

[‡] School Of Mathematical Sciences, Sackler Faculty of Exact Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel.

[§] School of Computer Science, Sackler Faculty of Exact Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel.

[1] Writing **whp** we mean with probability tending to 1 as n goes to infinity.

of over-exploiting the statistical properties of the random graph (eigenvalues structure, vertex degrees, etc) and design algorithms that perform well on a specific distribution but fail completely when slightly changing the distribution to a more realistic one (as many such graph properties no longer possess the "clean" and manageable behavior they have in the random setting).

To capture this notion of robustness desired from an algorithm, semirandom models are introduced. In the semirandom setting, first a random instance is generated. Next, an adversary may change the instance further. These modifications cannot be arbitrary, or the adversary can remake the graph into a worst-case instance. Put differently, semirandom models generate problem instances by blending random and adversarial decisions, thus intermediating between the worst-case assumptions, which may be overly pessimistic in many situations, and the easy pure 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 [9], [11]. [13] present a simpler and more natural algorithm for a semi-random planted 3SAT distribution, compared with [14] for the random setting).

The following semirandom variant of $G_{n,p,k}$ was suggested by Blum and Spencer [4], and is denoted throughout by $G^*_{n,p,k}$. First, a graph $G_0 = G_{n,p,k}$ is generated (throughout, we use G_0 to denote a random graph sampled according to $G_{n,p,k}$, or the underlying random part of a semirandom instance. The meaning will be clear from the context). Next, an adversary is allowed to add $V_i - V_j$ edges for $i \neq j$.

Related Work. [22] suggests an $O(\sqrt{np}/\log n)$-approximation algorithm for the chromatic number of graphs on n vertices. They prove that over $G_{n,p}$, $p \in [n^{-\frac{1}{2}+\epsilon}, \frac{3}{4}]$, the algorithm runs in expected polynomial time [2]. [8] extends the latter to $p \geq C/n$, C a sufficiently large constant. In this work we focus on heuristics that find a correct solution for "almost all" instances. Alon and Kahale [1] suggest a polynomial time algorithm based on spectral techniques that **whp** finds a k-coloring of $G_{n,p,k}$ with $np \geq C_0 k^2$, C_0 a sufficiently large constant. Combining techniques from [1] and [7], [6] suggests an expected polynomial time algorithm for $G_{n,p,k}$ based on SDP (semi-definite programming). Both [1] and [6] fail in $G^*_{n,p,k}$, as the adversary may foil many statistics of the random graph which both algorithms heavily rely on (eigenvalues structure, vertex degrees, etc). Blum and Spencer [4] present a heuristic that k-colors $G^*_{n,p,k}$ **whp** for a constant k, and $np \geq n^{\alpha_k}$, $\alpha_k \geq 2/5$. Feige and Kilian [9] improve upon this result, giving an SDP-based algorithm that k-colors $G^*_{n,p,k}$ for $np \geq c(1+\epsilon)k \ln n$. They also provide with a hardness result, proving that unless $NP \subseteq BPP$, it is hard[3] to k-color $G^*_{n,p,k}$ for $np < (1-\epsilon)\ln n$. The proof of the hardness result is based on the existence **whp** of isolated vertices when $np < (1-\epsilon)\ln n$, more details ahead. Coja-Oghlan [7] gives a simpler SDP-based heuristic that k-colors $G^*_{n,p,k}$ for $np \geq c(1+\epsilon)k \ln n$, and also provides a certificate for the optimality of the coloring. [7] improves upon the hardness result of [9], proving that unless $NP \subseteq RP$, it is hard to k-color $G^*_{n,p,k}$ for $np \leq (1-\epsilon)\frac{k}{2}\ln(n/k)$.

Our Results. In this work, we focus on the *sparse* case, namely when np is constant (that may depend on k). Semirandom distributions of sparse instances need to be addressed with more delicacy. For example, in $G^*_{n,p,k}$, np a constant, **whp** there will be a constant fraction of isolated vertices in G_0 on which the adversary (if not restricted otherwise) can plant a worst-case instance, turning the problem hard. This is the basic idea behind the hardness proofs in [7], [9]. Consequently, $G^*_{n,p,k}$ is not an interesting benchmark in our (sparse) setting . Therefore, different criteria for evaluating the robustness of algorithms in the sparse case are in due.

The first to consider semirandom models for sparse settings were [13] and [25], in the context of the planted SAT distribution. In this paper we discuss two alternatives for the coloring problem, which prove quite useful. One possibility is to further limit the adversary and to require that the algorithm succeeds **whp** when spending polynomial time. To this end we introduce the semirandom model $G^H_{n,p,k}$. The only difference between $G^*_{n,p,k}$ and $G^H_{n,p,k}$ is that in the latter, the adversary is allowed to add only edges whose both endpoints belong to a certain set $H \subseteq V$ (which will be rigorously defined and analyzed in the sequel). Another possibility is to require that the algorithm finds a solution **whp** over $G^*_{n,p,k}$. This time however it will not be realistic to require that the algorithm spends only polynomial time (the aforementioned hardness results), rather the algorithm is allowed to spend as much time as needed to guarantee a solution **whp**. The preferable heuristic in this case is the one guaranteeing a solution **whp** while spending as little time as possible.

Building upon [1] and [7] we present two new coloring heuristics, COLOR and COLOR2. Using the afore-

[2] An algorithm $\mathcal{A}$ with running time $t_{\mathcal{A}}(I)$ on an input instance I, has *expected polynomial running time* over a distribution $\mathcal{D}$ on the inputs, if $\sum_I t_{\mathcal{A}}(I) \cdot Pr_{\mathcal{D}}[I]$ is polynomial.

[3] By "hard" we mean there exists no polynomial time algorithm that solves the problem **whp**

mentioned criteria, we compare COLOR, COLOR2, [1], and [6]. We prove that in some exact sense, COLOR is the most robust heuristic of the four, and COLOR2 is preferable to [1], [6]. As a byproduct, we identify the weakest links in all four algorithms, which yields a deeper algorithmic understanding, thus serving as yet another motivation for using semirandom models. Formally, we prove:

THEOREM 1.1. *Let $np \geq C_0k^2$ for some sufficiently large constant C_0 and a constant k, then there is an algorithm COLOR that* ***whp*** *k-colors $G_{n,p,k}$ in polynomial time.*

THEOREM 1.2. *In the setting of Theorem 1.1, the algorithm COLOR finds* ***whp*** *a k-coloring in polynomial time for the semirandom distribution $G^H_{n,p,k}$ defined in Section 4.*

THEOREM 1.3. *In the setting of Theorem 1.1, the algorithm COLOR finds* ***whp*** *a k-coloring for $G^*_{n,p,k}$ in time $(1+\alpha)^n$, where $\alpha = \exp\{-\Omega(np/k)\}$.*

Theorem 1.1 is already proven in [1], and an even stronger version of it is proven in [6]. However, we show that [1] fails to meet the requirements of Theorems 1.2 and 1.3, and that at least the analysis given in [6] fails Theorem 1.2, and in Theorem 1.3, α should be adjusted to $\Omega((np/k)^{-0.5})$. As for COLOR2,

THEOREM 1.4. *In the setting of Theorem 1.1, COLOR2 finds* ***whp*** *a k-coloring of $G_{n,p,k}$ in polynomial time.*

THEOREM 1.5. *In the setting of Theorem 1.2, the algorithm COLOR2 finds* ***whp*** *a k-coloring of $G^H_{n,p,k}$ in polynomial time.*

However COLOR2 fails to meet the requirements of Theorem 1.3. The proofs of Theorems 1.4 and 1.5 are not given fully and only an outline is sketched.

The rest of the paper is structured as follows. In Section 2 we present both algorithms, COLOR and COLOR2. In Section 3 we analyze COLOR in the random setting of $G_{n,p,k}$. In Section 4, we describe in details the semirandom variant $G^H_{n,p,k}$, and prove Theorems 1.2 and 1.3. We also compare all four algorithms using $G^H_{n,p,k}$ and $G^*_{n,p,k}$ as benchmarks. In Section 5 we discuss some possibly interesting topics for future research.

2 The Algorithms

In this section we present both algorithms, COLOR and COLOR2. As the underlying ideas are common to all four algorithms, and since we compare their performances, we start by giving a short description of [1] and [6]. In the description, we make remarks as to the aforementioned set H, which correspond to the analysis part. The reader at this stage may disregard these notes, as we do not assume familiarity with the analysis in [1]. The scheme in both algorithms is basically the same, we start with [1]. First, using a spectral technique, a $(1-\epsilon)$-approximation of the planted coloring is obtained **whp** (ϵ is a small constant, the probability is taken over $G_{n,p,k}$) . Next, a recoloring procedure is applied to reach an even closer distance from the planted coloring (and in particular, **whp** the set H is now colored correctly). Then, a careful uncoloring step guarantees that **whp** only correctly colored vertices remain colored (in particular, H remains colored). Finally, using exhaustive search on the graph induced by the uncolored vertices, the partial coloring is completed **whp** to a legal one. Since this graph breaks down **whp** to connected components of size $O(\log_k n)$, the latter can be carried out successfully while spending polynomial time. [6] runs in expected polynomial time over $G_{n,p,k}$. To achieve this, some modifications to [1] are done, amongst which, the spectral step is replaced with SDP, and possible mistakes in the recoloring and uncoloring steps are corrected using a careful exhaustive search.

Notation. For a graph G, we let $V(G)$ denote its set of vertices and $E(G)$ the set of edges. For a set $U \subseteq V$, $G[U]$ denotes the subgraph of G induced by the vertices of U. For a vertex v, $N(v)$ denotes its set of neighbors in G. For $A, B \subseteq V$ we let $e(A, B)$ be the number of edges connecting a vertex from A and a vertex from B in G. For two vertices $u, v \in V(G)$, $G + (u, v)$ denotes the graph G with the additional edge (u, v). For two graphs G_1, G_2, $G1 \setminus G2$ denotes $G1$ with all the edges of G_2 removed from it. The scalar product of two vectors $x, y \in \mathbb{R}^n$ is denoted $\langle x, y\rangle$.

Motivation. The algorithm COLOR consists of two steps. First (lines 1-12), using SDP, one obtains a partial coloring of the graph, which satisfies (a) it coincides with the planted coloring and (b) **whp** the partial coloring colors all but a small fraction of the vertices (in particular, the set H is colored). Next (lines 13-14), as done in Alon-Kahale, the partial coloring is completed to a legal one using exhaustive search on the set of uncolored vertices. In the setting of Theorems 1.1 and 1.2, the subgraph induced by the uncolored vertices breaks down **whp** to connected components of size at most $O(\log_k n)$, allowing the exhaustive search to remain polynomial. COLOR2 is more similar to [1] and [6]. Using SDP, a partial coloring is obtained. However, using different analysis than [6], one can show that the partial coloring is such that the recoloring step can be skipped, and one can immediately approach the uncoloring procedure. Finally, using exhaustive search, a legal coloring is **whp** found.

As the first step in COLOR and COLOR2 is based on the SDP of the max h-cut problem, suggested by Frieze and Jerrum [15], we start by presenting it.

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

where the max is taken over all families $(x_v)_{v\in V}$ of unit vectors in $\mathbb{R}^{|V|}$. If $h \geq 2$ is an integer, then SDP_h is a relaxation of the MAX h-CUT problem. Since SDP_h is a semidefinite program, its optimal value can be computed up to an arbitrary high precision $\epsilon > 0$, in time polynomial in $|V|, h, \log\frac{1}{\epsilon}$ (e.g. using the Ellipsoid algorithm [17], [20]).

If G is k-colorable, then $SDP_k(G) = |E(G)|$ (since $|E(G)| = \text{max-k-cut}(G) \leq SDP_k(G) \leq |E(G)|$) . Moreover, if G is a subgraph of G', then $SDP_h(G) \leq SDP_h(G')$. The following Lemma is the key to understanding and analyzing both algorithms.

Lemma 2.1. *Let $G = G^*_{n,p,k}$, $np \geq C_0k^2$. **Whp** there exists a set $V_0 \subseteq V$ of vertices, such that:*
(a) Let $V_0^{(i)} = V_0 \cap V_i$, then $|V_0^{(i)}|/|V_i| \geq 1 - \exp\{-\Omega(np/k)\}$.
(b) For every $i \in \{1, 2, ..., k\}$, for every $u^*, v^* \in V_0^{(i)}$, and for every $h \leq k$, $SDP_h(G + (u^*, v^*)) \leq |E(G)| - \Omega(\frac{n^2p}{hk})(k-h)$. In particular, for $h = k$, $SDP_k(G + (u^*, v^*)) = |E(G)|$.

In the sequel (Lemma 3.1), we explicitly identify such a set V_0, proving the lemma.
Another simple observation is that for any $u \in V_i$, $v \in V_j$ s.t. $i \neq j$, it holds that $SDP_k(G + (u, v)) = SDP_k(G) + 1$. This observation, combined with Lemma 2.1, are the motivation behind steps 5-10 of COLOR. As we shall prove in the next section, in steps 1-12, a set meeting the requirements of V_0 is colored according to the planted coloring **whp**. Further, $G[V \setminus V_0]$ breaks down **whp** to connected components of size $O(\log_k n)$.

Remark 2.1. Instead of iteratively picking up the v_i's (line 4), one can go over all $\binom{n}{k}$ possibilities for $v_1, ..., v_k$. Another option is to choose all the v_i's in one step (and possibly amplify the success probability by repeating the execution).

Algorithm 1 : COLOR(G, k)

```
1:  Compute the value of $SDP_k(G)$ up to precision of 0.05.
2:  Let $W_i = \emptyset$ for $i = 1, 2, ..., k$.
3:  for $i = 1$ to $k$ do
4:      Let $W = \bigcup_{j=1}^{i-1} W_j$, pick $v_i \in V \setminus W$ u.a.r and set $W_i = \{v_i\}$.
5:      for all $u \in V \setminus W$ and non-edges $(v_i, u) \notin E$ do
6:          Compute $SDP_k(G + (u, v_i))$ up to precision of 0.05.
7:          if $|SDP_k(G + (v_i, u)) - SDP_k(G)| \leq 0.1$ then
8:              $W_i \leftarrow W_i \cup \{u\}$
9:          end if
10:     end for
11: end for
12: Color $W_i$ with color $i$, and let $U = V \setminus \bigcup_{i=1}^{k} W_i$
13: Find the connected components in $G[U]$.
14: In every component separately, use exhaustive search to complete the partial coloring of $\bigcup_{i=1}^{k} W_i$ to a legal $k$-coloring of $G$.
```

Remark 2.2. The algorithm receives k as a parameter. However, this is done only to simplify the description. If one could efficiently calculate a lower bound r on $\chi(G)$, then a simple way to circumvent the problem of not knowing k is to run the algorithm with $k = r, r+1, ..., n$ (of course the trivial bound $r = 1$ suffices, but one can do better). Such a non trivial lower bound can be calculated via SDP (in fact for $G^*_{n,p,k}$ the value $\chi(G)$ itself can be calculated **whp**, Lemma 2.3 ahead). To obtain this result and as motivation for the algorithm COLOR2, we need the following discussion.

Definition 2.1. *Let $G = (V, E)$ be a graph, $|V| = n$. We say that a family of n unit vectors $(x_v)_{v\in V}$ in $\mathbb{R}^n$ is a* rigid vector k-coloring *of G, if for every $u, v \in V$, $\langle x_u, x_v\rangle \geq -1/(k-1)$ and if $(u, v) \in E$ then $\langle x_u, x_v\rangle = -1/(k-1)$.*

$\bar{\vartheta}_2(G)$ commonly denotes the minimal real $k > 1$ such that G admits a rigid vector k-coloring. Since $\bar{\vartheta}_2(G)$ can be stated as a semidefinite program, it can be computed up to an arbitrary precision in polynomial time. The definition of $\bar{\vartheta}_2(G)$ in terms of vector coloring is related to the work of Karger, Motwani and Sudan [20].

Lemma 2.2. *For every graph G, $\bar{\vartheta}_2(G) \leq \chi(G)$.*

Proof. Let $\chi(G) = c \leq n$. One can find c unit vectors $\{v_1, ..., v_c\}$ in $\mathbb{R}^n$ s.t. $\langle v_i, v_j\rangle = -1/(c-1)$ for every $1 \leq i \neq j \leq h$ (for a proof, see for example [21] Claim 2.2). Consider a c-coloring of G, and assign v_i to all vertices of color i. Clearly $\{v_1, ..., v_c\}$ is a rigid vector c-coloring of G, proving the Lemma.

Lemma 2.3. *For $G = G^*_{n,p,k}$, **whp** $\bar{\vartheta}_2(G) = \chi(G) = k$.*

Proof. It is enough to show that $\bar{\vartheta}_2(G) = k$ (since $\chi(G) \leq k$ by definition, and $\bar{\vartheta}_2(G) \leq \chi(G)$ by the previous Lemma). By contradiction, assume that $\bar{\vartheta}_2(G) = h < k$. Let $(x_v)_{v \in V}$ be a rigid vector h-coloring of G. $(x_v)_{v \in V}$ is also a feasible solution to SDP_h, therefore, $SDP_h(G) \geq |E(G)|$. On the other hand, by Lemma 2.1, **whp** $SDP_h(G) < |E(G)|$ for $h < k$.

Therefore, when the input is $G = G^*_{n,p,k}$, we can **whp** calculate k. Using the terminology of [7], we call a rigid vector k-coloring $(x_v)_{v \in V}$ of $G = G^*_{n,p,k}$ *integral* w.r.t. the planted k-coloring of G, if there are k vectors $(x^*_i)_{i=1,\dots,k}$ s.t. $x_v = x^*_i$ for all $v \in V_i$, and $\langle x^*_i, x^*_j \rangle = -\frac{1}{k-1}$ for $i \neq j$. [7] proves that for $np \geq \Omega(k \ln n)$, the rigid vector k-coloring is **whp** integral, therefore the planted coloring of the graph can be easily reconstructed. This is not necessarily true in the sparse case. However, in the sparse case, a rigid vector k-coloring of G is integral on V_0. Formally,

Proposition 2.1. *Let $(x_v)_{v \in V}$ be a rigid vector k-coloring of $G = G^*_{n,p,k}$.* **Whp** *for every i, and every $s, t \in V_0^{(i)}$, $x_s = x_t$.*

Proof. Observe that $(x_v)_{v \in V}$ is also a feasible solution to $SDP_k(G + (s,t))$, though not necessarily the optimal one. Thus,

$$E(G) = \sum_{(u,v) \in E(G)} \frac{k-1}{k}(1 - \langle x_u, x_v \rangle) \leq$$

$$\sum_{(u,v) \in E(G)} \frac{k-1}{k}(1 - \langle x_u, x_v \rangle) + \frac{k-1}{k}(1 - \langle x_s, x_t \rangle)$$

$$\leq SDP_k(G + (s,t)) \underbrace{\leq}_{\text{Lemma 2.1}} |E(G)|$$

We conclude that $\frac{k-1}{k}(1 - \langle x_s, x_t \rangle) = 0$, or equivalently that $x_s = x_t$. Since **whp** there exist $V_0^{(i)} - V_0^{(j)}$ edges for every i, j, the second requirement, $\langle x^*_i, x^*_j \rangle = -\frac{1}{k-1}$, holds by the definition of a rigid vector k-coring.

Remark 2.3. Since the SDP of the rigid vector k-coloring is not solved exactly (rather up to some predefined precision) there are some extra technical issues to take care of. For example, two vertices that should have received the same vector assignment x^*_i may have received in fact two different, though very close, vectors. As such issues are only technical in nature, details are omitted.

3 The Random Setting

In this section we prove Theorem 1.1, and sketch the analysis of COLOR2. The key to proving Theorem 1.1 is to show that **whp**, in steps 1-12, a huge fraction of the vertices is colored correctly, and the remaining vertices break into connected components of size at most $\log_k(n)$. To this end, we introduce a set of vertices $H \subseteq V$ similar to the one described in [1]. Let us briefly define H (a similar description can also be found in [1]). First, let H_0 be the set of vertices having at most $1.01np/k$ neighbors in G_0 in every color class (other than their own). Consider the subgraph $G_0[H_0]$ (the subgraph induced by the vertices of H_0) and set $i = 0$. While there exists a vertex $v_i \in H_i$ that has less than $0.99np/k$ neighbors in $G_0[H_i]$ in some color class (other than its own), define H_{i+1} to be $H_i \setminus \{v\}$ and increment i by 1. Let H be the remaining set of vertices when the iterative procedure stops.

Algorithm 2 : COLOR2(G, k)

```
1: Compute a rigid vector $k$-coloring of G, $\{x^*_i\}$.
2: Group the vertices in $V$ according to the vectors
   $\{x^*_i\}$ assigned to them by the rigid vector $k$-coloring.
3: Let $W_i \subseteq V$ be the set of vertices assigned with the
   vector $x^*_i$.
4: Color the $k$ largest $W_i$'s (w.l.o.g $i = 1, ..., k$) with $k$
   different colors (one for every set).
5: while $\exists i, j \in [1..k], v \in V$ s.t. $v \in W_i$ and $v$ has less
   than $0.98np/k$ neighbors colored $j \neq i$ do
6:     Uncolor $v$.
7: end while
8: Let $U$ be the set of uncolored vertices. Find the
   connected components in $G[U]$.
9: In every component separately, use exhaustive
   search to complete the partial coloring of $G[V \setminus U]$
   to a legal $k$-coloring of $G$.
```

Lemma 3.1. ***Whp****, the set H satisfies the requirements to the set V_0 in Lemma 2.1.*

Lemma 3.1 proves Lemma 2.1. The proof of requirement (a) in Lemma 2.1 is given in [1] Lemma 2.7 (for $k = 3$). The proof of requirement (b) follows closely the one given in [7] Lemma 10, while using results from [1] and [12]. The proof of Lemma 3.1 is highly technical in nature, thus it is deferred to the appendix. In the remainder of the section, it would be convenient for the reader to think of V_0 as H.

Recall that $V_0^{(i)}$ is the set of vertices in V_0 that belong to color class V_i. We start by proving that if v_i happens to fall in $V_0^{(i)}$, then **whp** most of the color class V_i is recovered. Formally,

Lemma 3.2. *If v_i, chosen in line 4, belongs to $V_0^{(i)}$, then* ***whp*** *the corresponding W_i recovered in steps 5-8 satisfies $V_0^{(i)} \subseteq W_i \subseteq V_i$.*

Proof. First we prove that $W_i \subseteq V_i$ w.p. 1. By contradiction, suppose that in some step of the `for` iteration in lines 5-8, some $u \in V_j, j \neq i$ was included in W_i. Then the condition in line 7 held. However, it holds that $SDP_k(G + (v_i, u)) = SDP_k(G) + 1$ for $u \notin V_i$. Choosing the precision of the SDP solution to be high enough (0.05 in our case), we get a contradiction. $V_0^{(i)} \subseteq W_i$ holds **whp** by Lemma 2.1, requirement (b) and again choosing a sufficiently high precision for the SDP.

It therefore remains to prove that with rather high (we shall prove constant) probability, the vertices $v_1, v_2, ..., v_k$ chosen in line 4, satisfy $v_i \in V_0^{(i)}$. This, combined with Lemma 3.2, proves that **whp** most of the vertices (and in particular the set V_0) are colored correctly when line 13 begins to execute. Formally,

Lemma 3.3. *With probability* $1 - \exp\{\Omega(np/k)\}$*, all* k *vectors* $v_1, v_2, ..., v_k$*, chosen in line 4, satisfy w.l.o.g.* $v_i \in V_0^{(i)}$.

Proof. For starters, consider the first iteration where v_1 is chosen. Assume that Lemma 2.1 indeed holds. By requirement (a) of Lemma 2.1, v_1 belongs to some $V_0^{(i)}$ w.p. $1 - \exp\{\Omega(np/k)\}$, assume w.l.o.g. $i = 1$. Now assume that $v_1, v_2, ..., v_{i-1}$ were all chosen to be in $V_0^{(1)}, V_0^{(2)}, ..., V_0^{(i-1)}$ respectively, and ask what is the probability that $v_i \in V_0^{(i)}$. When picking v_i, the bad events could be either that $v_i \in V_j$ for some $j \leq i-1$ (by Lemma 3.2 and the assumption on the v_j's, v_i will then belong to $V_j \setminus V_0^{(j)}$), or that $v_i \in V_i \setminus V_0^{(i)}$. The number of bad vertices is then at most $n \cdot \exp\{-\Omega(np/k)\}$. The total number of remaining vertices to choose from is at least n/k (since in every iteration at most one color class is colored). Therefore, the probability of a bad event happening is bounded by

$$\frac{n \cdot \exp\{-\Omega(np/k)\}}{n/k} = k \cdot \exp\{-\Omega(np/k)\} =$$

$$= \exp\{-\Omega(np/k)\}.$$

The last equality is due to $np \geq C_0 k^2$. Let A_i be the event $v_i \in V_0^{(i)}$. Then,

$$\begin{aligned} &Pr[\bigwedge_{i=1}^{k} A_i] = Pr[A_1] \cdot Pr[A_2|A_1] \cdots Pr[A_k| \bigwedge_{i=1}^{k-1} A_i] \\ &\geq (1 - \exp\{-\Omega(np/k)\})^k \simeq 1 - k \cdot \exp\{-\Omega(np/k)\} \\ &= 1 - \exp\{-\Omega(np/k)\}. \end{aligned}$$

Since Lemma 2.1 holds w.p. $1 - o(1)$, the lemma follows.

Remark 3.1. One can ensure that Lemma 3.3 occurs with probability 1 by going over all possible $\binom{n}{k}$ choices for $v_1, ..., v_k$ and run COLOR for each such choice (skipping line 4).

Proposition 3.1. *H is colored when the exhaustive search begins with probability* $1 - \exp\{-\Omega(np/k)\}$*. Further, (with probability 1) the partial coloring induced by the W_i's (line 12) coincides with the planted coloring (up to renaming of color classes).*

Proposition 3.1 follows readily from Lemmas 3.1, 3.2 and 3.3.

Proposition 3.2. ***Whp**, the largest connected component in* $G[V \setminus H]$ *is of size at most* $\log_k(n)$.

The proof of this Lemma is given in [1] Proposition 2.11, for $k = 3$. It generalizes easily for any constant k. The intuition behind the proof comes from a well known result concerning $G_{n,p}$. If $np < 1$, then **whp** the largest connected component in $G_{n,p}$ is of size at most $O(\log n)$ (see e.g. [3] for a complete discussion). Now consider a random subgraph of size αn of $G_{n,p}$, this subgraph is exactly $G_{\hat{n},\alpha p}$, where $\hat{n} = \alpha n$. If $\alpha np < 1$, **whp** the largest connected component in $G_{\hat{n},\alpha p}$ is of size $O(\log \hat{n})$. In our case, **whp** $|V \setminus H| \cdot p < 1$ but unfortunately $V \setminus H$ is not a truly random subset of vertices. Therefore, the proof is burdened with more technicalities.

Combining Propositions 3.1 and 3.2, **whp** the exhaustive search consumes polynomial time, and succeeds in completing the partial coloring (line 12), to a legal coloring of G (since at least the partial coloring can be completed to the planted one). Theorem 1.1 then follows.

Let us now sketch the proof of Theorem 1.4. Proposition 2.1 and requirement (a) in Lemma 2.1 imply that the set H is set correctly before the uncoloring procedure begins. By the definition of H, and the fact that the coloring of H coincides with the planted coloring, it follows that H survives the uncoloring procedure. By expansion properties of $G_{n,p,k}$, it holds that **whp** every vertex that survived the uncoloring is colored according to the planted coloring (Lemma 2.9 in [1]). This combined with Proposition 3.2, allow the exhaustive search to succeed **whp** in completing the partial coloring to a legal one in polynomial time.

4 The Semirandom Setting

Let us recall the definition of the semirandom model $G^*_{n,p,k}$. First a random graph $G_0 = G_{n,p,k}$ is generated in the aforementioned way. Let $V_1, V_2, ..., V_k$ be the planted color classes. Next an adversary may add edges connecting a vertex in V_i with a vertex in V_j for $i \neq j$.

Improving upon [9], [7] proves that unless $NP \subseteq RP$, there is no polynomial time algorithm that **whp** k-colors $G^*_{n,p,k}$ when $np \leq (1-\epsilon)\frac{k}{2}\ln(n/k)$. In our case np is constant, therefore it is not realistic to expect COLOR to work **whp** in polynomial time over $G^*_{n,p,k}$. Let $A \subseteq V$ be an arbitrary set of vertices, denote by $G^A_{n,p,k}$ the following semirandom model. As in $G^*_{n,p,k}$, first G_0 is generated. Next, an adversary may add $V_i - V_j$ edges, only this time both endpoints belong to A. Observe that $G^V_{n,p,k} = G^*_{n,p,k}$ and $G^{\emptyset}_{n,p,k} = G_{n,p,k}$. Therefore, the choice of A in $G^A_{n,p,k}$ allows us to regulate the hardness of the resulting distribution. In this work we consider $A = H$. Note that H is in some sense a random set which depends on G_0. Arguably, $G^H_{n,p,k}$ is not the most natural semirandom model to consider. In particular, the fact that H depends on G_0 is not a desirable property. However, as we show shortly, it already suffices to separate COLOR and COLOR2 from [1] and [6], identifying two weakest links in the latter.

4.1 Proof of Theorem 1.2

Proof. Note that Lemmas 3.1, 3.2 and 3.3 are valid in $G^*_{n,p,k}$ to begin with, therefore remain valid in the more restricted $G^H_{n,p,k}$. Hence, Proposition 3.1 is also valid in $G^H_{n,p,k}$. Regarding Proposition 3.2, its proof for $G_{n,p,k}$ relies on the random properties of $G_0[V \setminus H]$. As the adversary is not allowed to add edges to $G_0[V \setminus H]$, it remains valid in $G^H_{n,p,k}$. To summarize, the analysis given in Section 3 remains valid for $G^H_{n,p,k}$ as well. Thus, Theorem 1.2 follows.

As for Theorem 1.5, Lemmas 2.1, 3.1, and Proposition 2.1 assume $G = G^*_{n,p,k}$. Proposition 3.2 is also valid in $G^H_{n,p,k}$. The only issue to address is the correctness of the uncoloring procedure. As the expansion properties of $G_0[V \setminus H]$ remain valid in $G^H_{n,p,k}$, the assertions regarding the uncoloring procedure carry through.

Let us now compare the four algorithms, [1], [6], COLOR and COLOR2, using $G^H_{n,p,k}$ as a benchmark. [1] fails already in the spectral step since **whp** H contains almost all vertices allowing the adversary to jumble with the spectra of the graph (emptying the eigenvectors, in particular the last two used in [1], from meaningful information regarding the planted coloring). One can prove that the SDP approximation (replacing the spectral one), used in [6], still provides a $(1-\epsilon)$-approximation in $G^*_{n,p,k}$ (though the issue of semirandomness is not addressed in [6]). However, the analysis of the recoloring step employed in [6] relies on random properties of $G_{n,p,k}$ which need not hold in $G^H_{n,p,k}$. Therefore, the analysis in [6] doesn't show that the algorithm finds a legal coloring **whp** in *polynomial* time. On the other hand, both COLOR and COLOR2 find **whp** a legal coloring in polynomial time when the graph G is sampled according to $G^H_{n,p,k}$. The two weakest links - the spectral technique, and the recoloring procedure - are not used in the latter. However, this model doesn't suffice to separate COLOR from COLOR2. This will be done promptly.

4.2 Proof of Theorem 1.3

Proof. Lemmas 3.1, 3.2 and 3.3 are valid in $G^*_{n,p,k}$, hence Proposition 3.1 is also valid in $G^*_{n,p,k}$. However, Proposition 3.2 need not necessarily hold. Therefore, based on the above analysis, it need not hold that the exhaustive search can end up successfully while spending polynomial time. However, requirement (a) in Lemma 2.1 and Lemma 3.1 imply that **whp** $|U| \leq \exp\{-\Omega(np/k)\}n$ (where U is the set of uncolored vertices). Thus, the exhaustive search consumes at most

$$k^{|U|} = k^{\exp\{-\Omega(np/k)\}n} \leq (1+\exp\{-\Omega(np/k)\})^n$$

steps. Theorem 1.3 then follows.

Now let us compare [1], [6], COLOR and COLOR2 in this setting. [1] fails for at least the aforementioned reason. The analysis of the uncoloring procedure employed by COLOR2 (as well as by [1] and [6]) breaks in $G^*_{n,p,k}$, failing to show that all vertices surviving the uncoloring are colored according to the planted coloring. Therefore, it might be the case that COLOR2 will not be able to produce a legal coloring regardless of the amount of time it is given (since the graph $G[U]$ may no longer be k-colorable when taking into account the constraints imposed by the coloring of $V \setminus U$). As part of the augmentations of [1] towards becoming an expected polynomial time algorithm, [6] employs a recovery step (which is basically a careful exhaustive search), which eventually sets correctly the ϵn vertices possibly missed by the SDP approximation. At that stage a legal coloring is found, but not necessarily beforehand (at least the analysis fails to show that). The analysis in [6] requires $\epsilon = \Omega((np/k)^{-0.5})$. Therefore, the time guaranteed by the current analysis in [6] to find a legal coloring is at least $(1+\Omega((np/k)^{-0.5}))^n$. This is of course much larger than the time COLOR spends.

5 Discussion

Semirandom models serve in many cases as useful benchmarks for evaluating the robustness of algorithms designed to work **whp** for random structures. However, when considering sparse random distributions, it might be the case that the interesting and natural adversaries render the problem hard (see for example [7], [9]). Therefore, one cannot expect them to serve as useful benchmarks when inspecting *polynomial* time algorithms. In this work, we suggest two alternatives to address the issue, and as a case study apply them to [1], [6], COLOR, and COLOR2. The first alternative is to further restrict the adversary, which translates to $G^H_{n,p,k}$ in our setting. Using $G^H_{n,p,k}$, $np \geq C_0k^2$, as a benchmark, we were able to claim that in some explicit sense COLOR and COLOR2 are more robust than [1] and [6]. However, $G^H_{n,p,k}$ doesn't separate COLOR from COLOR2.

The second alternative is to consider the natural semirandom distribution $G^*_{n,p,k}$, allow the algorithms to use as much time as needed to find a solution **whp** in the semirandom setting, and then compare the running times (which now may be superpolynomial). In our setting, for $G^*_{n,p,k}$, $np \geq C_0k^2$, [1] and COLOR2 fail to produce a solution (at least their analysis fails to show it), regardless of the amount of time spent trying to find one. In contrast, COLOR and [6] find a legal coloring **whp** in time $(1+b)^n$, where $b > 0$ is a constant depending on np/k. The exponent base guaranteed by COLOR is much smaller than the one guaranteed by [6] (b decreases exponentially in np/k rather than polynomially).

As we already mentioned, though useful, $G^H_{n,p,k}$ is not the most natural model to consider. An interesting question for further research is to come up with a more natural semirandom model for $np \geq C(k)$, $C(k)$ a constant, while keeping the model interesting, in the sense that one can expect a polynomial time algorithm to solve it **whp**. Then, one naturally asks for a polynomial time algorithm that works **whp** in that model. Another interesting question is to characterize the maximal set A of vertices s.t. $G^A_{n,p,k}$ can be solved in polynomial time **whp**.

References

[1] N. Alon and N. Kahale. *A spectral technique for coloring random* 3*-colorable graphs.* SIAM J. Comput., 26 (1997), pp. 1733–1748.

[2] N. Alon, M. Krivelevich, and B. Sudakov. *Finding a large hidden clique in a random graph.* Random Structures and Algorithms, 13 (1998), pp. 457–466.

[3] N. Alon and J. H. Spencer. *The probabilistic method.* Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience [John Wiley & Sons], New York, second edition, 2000.

[4] A. Blum and J. Spencer. *Coloring random and semirandom k-colorable graphs.* J. of Algorithms, 19 (1995), pp. 204–234.

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

[6] J. Böttcher. *Coloring sparse random k-colorable graphs in polynomial expected time.* In Proc. 30th International Symp. on Mathematical Foundations of Computer Science. Lecture Notes in Comput. Sci. 3618 (2005), pp. 156–167.

[7] A. Coja-Oghlan. *Coloring semirandom graphs optimally.* In Proc. 31st International Colloquium on Automata, Languages, and Programming, pp. 383–395, 2004.

[8] A. Coja-Oghlan. *The Lovász number of random graphs.* Combin. Probab. Comput. 14 (2005) pp. 439–465.

[9] U. Feige and J. Kilian. *Heuristics for semirandom graph problems.* J. Comput. and Syst. Sci., 63 (2001), pp. 639–671.

[10] U. Feige and J. Kilian. *Zero knowledge and the chromatic number.* J. Comput. and Syst. Sci., 57 (1998), pp. 187–199.

[11] U. Feige and R. Krauthgamer. *Finding and certifying a large hidden clique in a semirandom graph.* Random Structures and Algorithms, 16 (2000), pp. 195–208.

[12] U. Feige and E. Ofek. *Spectral techniques applied to sparse random graphs.* Random Structures and Algorithms, 27 (2000), pp 251–275.

[13] U. Feige and D. Vilenchik. *A local search algorithm for 3SAT.* Technical report, The Weizmann Institute of Science, 2004.

[14] A. Flaxman. *A spectral technique for random satisfiable 3CNF formulas.* In Proc. 14th ACM-SIAM Symp. on Discrete Algorithms, pp. 357–363, 2003.

[15] A. Frieze, M. Jerrum. *Improved approximation algorithms for MAX k-CUT and MAX BISECTION.* Algorithmica, 18 (1997), pp. 67–81.

[16] A. Frieze and C. McDiarmid. *Algorithmic theory of random graphs.* Random Structures and Algorithms, 10 (1997), pp. 5–42.

[17] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization.* Algorithms and Combinatorics (2). Springer-Verlag, Berlin, second edition, 1993.

[18] C. Helmberg. *Semidefinite programming.* European Journal of Operational Research, 137 (2002), pp. 461–482.

[19] T. R. Jensen and B. Toft. *Graph coloring problems.* Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience [John Wiley & Sons], New York, 1995.

[20] D. Karger, R. Motwani, and M. Sudan. *Approximate graph coloring by semidefinite programming.* J. of the ACM, 45 (1998), pp. 246–265.

[21] M. Krivelevich. *Deciding k-colorability in expected polynomial time.* Info. Process. Letters, 81 (2002), pp. 1–6.

[22] M. Krivelevich and V. H. Vu. *Approximating the independence number and the chromatic number in expected polynomial time.* J. Comb. Optim., 6 (2002), pp. 143–155.

[23] L. Kučera. *Expected behavior of graph coloring algorithms.* Proc. Fundamentals of Computation Theory, 56 (1977), pp. 447–451.

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

[25] D. Vilenchik. *Finding a satisfying assignment for semi-random satisfiable* 3*CNF formulas.* Master's thesis, The Weizmann Institute of Science, Rehovot, Israel, January 2004.

A Proof of Lemma 3.1

To prove Lemma 3.1, one has to prove that both requirements of Lemma 2.1 are met by H. The proof of requirement (a) is given in [1] Lemma 2.7, for $k = 3$, and readily generalizes to any constant k. The proof of requirement (b) follows closely the one given in [7] Lemma 10, while combining results from [1] and [12]. In this work we only prove requirement (b).

Notation. For a vector $x \in \mathbb{R}^n$, let $\|x\|$ be the ℓ_2-norm of x. We let $\mathbf{1}_n \in \mathbb{R}^n$ be the vector whose all entries equal 1, $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ be the unit matrix, and $\mathbf{J}_n \in \mathbb{R}^{n\times n}$ the matrix whose all entries are 1 (the subscript n is omitted when it is clear from context). By $diag(x)$ we denote the $n \times n$ diagonal matrix whose entries are x. For two matrices $A, B \in \mathbb{R}^{n\times n}$, the notation $A \leq B$ means $(B - A)$ is a positive semidefinite matrix. For a graph G, let $A(G)$ denote the adjacency matrix associated with the graph. Let $L(G)$ be the Laplacian of G , namely, $L(G) = diag(A\mathbf{1}) - A(G)$.

Recall that given $G = G^*_{n,p,k}$, $G_0[H]$ stands for the random part of G induced by the vertices of H. The key to proving Lemma 3.1 is the following claim:

Proposition A.1. ***Whp**, If $u^* \in V_i \cap H$, $v^* \in V_j \cap H$, $i \neq j$, then $SDP_h(G_0[H] + (u^*, v^*)) \leq |E(G_0[H])| - \Omega(\frac{n^2p}{hk})(k - h)$.*

Corollary A.1. *For $G = G^*_{n,p,k}$, u^*, v^* as in Proposition A.1, **whp** $SDP_h(G + (u^*, v^*)) \leq |E(G)| - \Omega(\frac{n^2p}{hk})(k - h)$.*

Proof.

$$SDP_h(G + (u^*, v^*)) \leq SDP_h(G_0[H] + (u^*, v^*))$$
$$+ SDP_h(G \setminus G_0[H]) \leq |E(G_0[H])| - \Omega(\frac{n^2p}{hk})(k - h)$$
$$+ |E(G \setminus G_0[H])| = |E(G)| - \Omega(\frac{n^2p}{hk})(k - h).$$

In the first inequality, we used the fact that for every G_1, G_2 s.t. G_1 is a subgraph of G_2, it holds that $SDP_h(G_2) \leq SDP_h(G_1) + SDP_h(G_2 \setminus G_1)$. In the second inequality, we used Proposition A.1 and the fact that for every graph G, $SDP_h(G) \leq |E(G)|$.

Corollary A.1 proves that H meets requirement (b) of Lemma 2.1, and finishes the proof of Lemma 3.1.

It remains to prove Proposition A.1, namely to bound $SDP_h(G_0[H] + (u^*, v^*))$ accordingly. As noted in [7], SDP duality is often a convenient tool to bound the value of a maximization problem. Given a graph G, one can bound the value of $SDP_h(G)$ using its dual problem $DSDP_h(G)$:

$$DSDP_h(G) = \min_Y \frac{h-1}{2h}\sum_{i=1}^n y_{ii} - \frac{1}{2h}\sum_{i\neq j} y_{ij}$$
$$\text{s.t. } Y = (y_{ij})_{i,j=1..n} \in \mathbb{R}^{n\times n},$$
$$L(G) \leq Y,\ y_{ij} \leq 0 \text{ for } i \neq j.$$

$SDP_h(G) \leq DSDP_h(G)$ by weak SDP duality (see for example [18]). In particular, we present a solution Y to $DSDP_h(G_0[H] + (u^*, v^*))$, which is **whp** feasible and whose value is $|E(G_0[H])| - \Omega(\frac{n^2p}{hk})(k - h)$. Since the dual is a minimization problem, Proposition A.1 follows.

Our last task is to provide with the aforementioned solution Y. Recall that $V_1, ..., V_k$ are the planted coloring classes of G_0. Let $W_i = V_i \cap H$, and $m = |H|$. Recall that $u^*, v^* \in W_i$ for some i. Assume some order on V, and let $i(W_a)$ denote the i'th vertex in color class W_a, let d_v be the degree of v in $G_0[H] + (u^*, v^*)$, and $d_v^{(i)}$ be $|N(v) \cap W_i|$. Moreover, we let

$$p_{ij}^{(ab)} = \frac{d^{(b)}_{i(W_a)} \cdot d^{(a)}_{j(W_b)}}{e(W_a, W_b)},\ p_{min} = \min_{i,j,a\neq b} p_{ij}^{(ab)},$$

$$d_{min} = p_{min} \cdot \min_a |W_a|.$$

Observe that $p_{ij}^{(ab)}$ has "probability units", and therefore can be viewed as an estimate for the probability of an edge between $i(W_a)$ and $j(W_b)$ given their degrees and $e(W_a, W_b)$. Following the same logic, d_{min} is the lowest expected degree of a vertex $v \in W_a$ in color class $W_b, b \neq a$.

We are ready to define the solution Y. For $1 \leq a, b \leq k$ define $|W_a| \times |W_b|$ matrices Y'_{ab} as follows:

$$Y'_{aa} = 0 \text{ for } a = 1, .., k.$$
$$\text{For } a \neq b,\ , Y'_{ab} = \left[\frac{d_{min}}{|W_b|} - p_{ij}^{(ab)}\right]_{i=1,...,|W_a|, j=1,...,|W_b|}$$

Let Y' be the $m \times m$ matrix whose blocks are the Y'_{ab}'s. Let $y' = (d_v + d_{min})_{v\in H} \in \mathbb{R}^m$. Finally, we let $Y = Y' + diag(y')$.

Proposition A.2. *The value of the solution Y is **whp** $|E(G_0[H])| - \Omega(\frac{n^2p}{hk})(k - h)$.*

Proof. First observe that $\sum_{j=1}^{|W_b|} d^{(a)}_{j(W_b)} = e(W_a, W_b)$.

Therefore, for $a \neq b$,

$$Y'_{ab}\mathbf{1} = \left[d_{min} - \sum_{j=1}^{|W_b|} \frac{d^{(b)}_{i(W_a)} \cdot d^{(a)}_{j(W_b)}}{e(W_a, W_b)} \right]_{1 \leq i \leq |W_a|} = \left[d_{min} - d^{(b)}_{i(W_a)} \right]_{1 \leq i \leq |W_a|}.$$

Further, observe that

$$\sum_{i \neq j} y_{ij} = \sum_{a \neq b} \langle Y'_{ab}\mathbf{1}, \mathbf{1} \rangle = \sum_{a \neq b} |W_a| \cdot d_{min} - e(W_a, W_b)$$
$$= (k-1)md_{min} - 2|E(G_0[H])|.$$
$$\sum_{i=1}^{m} y_{ii} = \langle y', \mathbf{1} \rangle = 2|E(G_0[H])| + md_{min}.$$

Therefore,

$$DSPD_h(G_0[H] + (u^*, v^*)) \leq \frac{h-1}{2h} \sum_{i=1}^{n} y_{ii} - \frac{1}{2h} \sum_{i \neq j} y_{i,j}$$
$$= |E(G_0[H])| - \frac{md_{min}}{2h}(k-h).$$

By the definition of H, $p_{min} = \Omega(p)$. Further, **whp** $|W_a| = \Omega(n/k)$ for every $a = 1, ..., k$, (requirement (a) in Lemma 2.1) and hence $m = |H| \geq (1 - \exp\{-np/k\})n = \Omega(n)$. It therefore follows that $d_{min} \geq \Omega(np/k)$. Plugging the values of d_{min} and m in the above expression, the claim follows.

Proposition A.3. *The matrix Y defined above is **whp** a feasible solution to the semidefinite programm $DSDP_h(G_0[H] + (u^*, v^*))$.*

Proof. Indeed, Y is a real symmetric matrix. Further, the definition of d_{min} ensures that every off diagonal y_{ij} obeys $y_{ij} \leq 0$. It remains to verify that $L(G_0[H] + (u^*, v^*)) \leq Y$, or equivalently, that $Y - L(G_0[H] + (u^*, v^*))$ is a positive semidefinite matrix (abbreviated **psd**). Let $A = A(G_0[H])$, $L = L(G_0[H])$, $L^+ = L(G_0[H] + (u^*, v^*))$, and $B = L^+ - L$. It is easy to see that $Y - L^+ = d_{min}\mathbf{I} - (B - A - Y')$. Therefore, it suffices to prove that the largest eigenvalue of $(B - A - Y')$ is at most d_{min} (since then, all eigenvalues of $Y - L^+$ are non-negative, and this is equivalent to being **psd**). The following lemma completes the proof of Proposition A.3. Proposition A.1 then follows from Propositions A.2, A.3, and SDP weak duality.

Lemma A.1. *The largest eigenvalue of $B - A - Y'$ is **whp** at most d_{min}.*

Proof. (Lemma A.1) Our proof strategy is as follows. We identify a subspace $K \subseteq \mathbb{R}^m$ spanned by a subset of the eigenvectors of $(B - A - Y')$. K has two useful properties: (a) it contains no eigenvector whose corresponding eigenvalue is greater than d_{min} and (b) **whp**, $|\langle (B - A - Y')x, x \rangle| < d_{min}$ for every unit vector $x \perp K$. Let λ be an eigenvalue of $(B - A - Y')$, and v_λ its corresponding eigenvector (assume all eigenvectors are mutually perpendicular, and normalized to unit vectors, e.g. via the Graham-Schmidt procedure). If $v_\lambda \in K$, then by property (a), $\lambda \leq d_{min}$. Otherwise, $v_\lambda \perp K$, and by property (b), $|\langle (B - A - Y')v_\lambda, v_\lambda \rangle| = |\lambda| < d_{min}$.

Let us start by presenting the subspace K. For $a = 1, ..., k$, we let $\mathbf{1}_{W_a} \in \mathbb{R}^m$ denote the vector whose entries are 1 if the entry corresponds to a vertex in W_a and 0 otherwise. Let δ_{v,W_a} be 1 if $v \in W_a$ and 0 otherwise. Similar to to the proof of Proposition A.2,

$$Y'\mathbf{1}_{W_a} = [(1 - \delta_{v,W_a})(d_{min} - e(v, W_a))]_{v \in H}$$

Further,

$$A\mathbf{1}_{W_a} = [e(v, W_a)]_{v \in H} \text{ and } B\mathbf{1}_{W_a} = 0.$$

Therefore,

$$(B - A - Y')\mathbf{1}_{W_a} = [-(1 - \delta_{v,W_a})d_{min}]$$

Finally, let $\xi^{(a,b)} = \mathbf{1}_{W_a} - \mathbf{1}_{W_b} \in \mathbb{R}^m$. By the above,

$$(B - A - Y')\xi^{(a,b)} = d_{min}\xi^{(a,b)} \text{ for } a \neq b,$$

$$(B - A - Y')\mathbf{1} = -(k-1)d_{min}\mathbf{1}.$$

Let K be the vector space spanned by the $\xi^{(a,b)}$'s and $\mathbf{1}$. Property (a) stated above follows from the latter. It remains to prove that property (b) holds.

Proposition A.4. *$|\langle (B - A - Y')x, x \rangle| < d_{min}$, for all unit vectors $x \perp K$.*

Before proving the Proposition we make the following observations regarding some spectral properties of A and Y'. Similar to Y'_{ab}, A_{ab} denotes the minor of A corresponding to W_a (rows) and W_b (columns).

Lemma A.2. *Let $G_0 = G_{n,p,k}$, Let $A = A(G_0[H])$ be the adjacency matrix of the subgraph of G_0 induced by the vertices of H. Then the following holds **whp**:*
(1) For all unit vectors $x \perp K$, $|\langle Ax, x \rangle| \leq O(\sqrt{np})$.
(2) For all $a, b \in \{1, ..., k\}$ and all $\mathbf{1} \perp x \in \mathbb{R}^{|W_a|}$, $|\langle A_{ab}\mathbf{1}, x \rangle| \leq \|\mathbf{1}_{W_a}\| O(\sqrt{np/k})$

The proof of Lemma A.2 is given in parts in [1] and [12]. We only point out the differences. Observe that $\mathbf{1}_{W_a} \in K$ for every $a = 1, ..., k$ (using linear combinations of the

$\xi^{(a,b)}$'s and $\mathbf{1}$). Therefore, the proof of (1) is essentially the one given in [1], Lemma 3.2. The only difference is that we consider the set H and [1] consider the set of vertices, call it W, whose degree in every color class doesn't exceed $5np/k$. The two properties of W used in [1] are $|W| \geq (1 - \exp\{-np/k\})n$ and the bound on the degree of vertices in W. Both properties hold for H as well **whp**. The proof of (2) is essentially the one given in [12], Lemma 3.2. Again, the only properties of W used in the proof are its size, and the bound on the degree of its vertices ([12] prove that property 2 holds with probability $1-O(\exp\{-np/k\})$, however, it can be shown that it holds with probability $1 - o(1)$ by using stronger methods than Markov's inequality).

Lemma A.3. *For every unit vector $x \perp K$, **whp** $|\langle Y'x, x\rangle| \leq O(\sqrt{np})$*

Proof. Consider the following $|W_a| \times |W_b|$ matrix Z_{ab}: $Z_{aa} = 0$, $Z_{ab} = \frac{1}{|W_b|}\mathbf{J} - Y'_{ab}$. Let Z be the $m \times m$ matrix whose blocks are the Z_{ab}'s. Since $x \perp \mathbf{1}_{W_a}$ for every a, $\langle Y'x, x\rangle = -\langle Zx, x\rangle$. Therefore it suffices to estimate $|\langle Zx, x\rangle|$. Let $\xi \in \mathbb{R}^{|W_b|}, \eta \in \mathbb{R}^{|W_a|}$ be two vectors perpendicular to $\mathbf{1}$.

$$e(W_a, W_b)|\langle Z_{ab}\xi, \eta\rangle| = \left\langle \left[\sum_{j=1}^{|W_b|} d^{(b)}_{i(W_a)} d^{(a)}_{j(W_b)} \xi_j\right]_{1\leq i\leq |W_a|}, \eta \right\rangle = \left\langle \left[d^{(b)}_{i(W_a)} \langle A_{ba}\mathbf{1}, \xi\rangle\right]_{1\leq i \leq |W_a|}, \eta\right\rangle = \langle A_{ba}\mathbf{1}, \xi\rangle \sum_{i=1}^{|W_a|} d^{(b)}_{i(W_a)}\eta_i = \langle A_{ba}\mathbf{1}, \xi\rangle\langle A_{ab}\mathbf{1}, \eta\rangle$$

Putting everything together,

$$|\langle Z_{ab}\xi, \eta\rangle| \leq |\langle A_{ba}\mathbf{1}, \xi\rangle| \cdot |\langle A_{ab}\mathbf{1}, \eta\rangle| / e(W_a, W_b) = O(np/k)\sqrt{|W_a|}\sqrt{|W_b|}/\Omega(p(n/k)^2) = O(1) \text{ (*)}$$

The last equality is due to Lemma A.2 (2), the properties of H (in particular, $e(W_a, W_b) = \Omega(p(n/k)^2)$), and $|W_a| = O(n/k)$ for every a.
We are ready to bound $|\langle Zx, x\rangle|$. For $x \perp K \in \mathbb{R}^m$, we let $x_a \in \mathbb{R}^{|W_a|}$ denote the entries of x corresponding to W_a.

$$|\langle Zx, x\rangle| = \left|\sum_{a,b}\langle Z_{ab}x_a, x_b\rangle\right| \leq \sum_{a,b}\|x_a\| \cdot \|x_b\| \cdot \left|\langle Z_{ab}\frac{x_a}{\|x_a\|}, \frac{x_b}{\|x_b\|}\rangle\right| \underbrace{\leq}_{(*)} \sum_{a,b}\|x_a\| \cdot \|x_b\| \cdot O(1) \leq O(1)\left(\sum_a \|x_a\|\right)^2 \leq O(k) \leq O(\sqrt{np})$$

The last equality is by the choice of p.

We are finally ready to prove property (b) of the vector space K (which was introduced at the beginning of this section), concluding the proof of Lemma A.1. It is readily seen that for every unit vector x, $|\langle Bx, x\rangle| \leq 2$. For a unit vector $x \perp K$,

$$|\langle (B - A - Y')x, x\rangle| \leq |\langle Bx, x\rangle| + |\langle Ax, x\rangle| + |\langle Y'x, x\rangle| \underbrace{\leq}_{\text{Lemmas A.2, A.3}} O(\sqrt{np})$$

On the other hand, $d_{min} = \Theta(np/k)$ by the properties of H. Demanding $O(\sqrt{np}) \leq \Theta(np/k)$ (or equivalently, $C_0k^2 \leq np$ for a suitable choice of a constant C_0), Lemma A.1 then follows.