Approximation Algorithms for Partially Colorable Graphs

Suprovat Ghoshal
Indian Institute of Science, Bangalore, India
suprovat@iisc.ac.in

Anand Louis
Indian Institute of Science, Bangalore, India
anandl@iisc.ac.in

Rahul Raychaudhury
Indian Institute of Science, Bangalore, India
rahulr@iisc.ac.in

Abstract

Graph coloring problems are a central topic of study in the theory of algorithms. We study the problem of partially coloring *partially colorable graphs*. For $\alpha \leq 1$ and $k \in \mathbb{Z}^+$, we say that a graph $G = (V, E)$ is α-partially k-colorable, if there exists a subset $S \subset V$ of cardinality $|S| \geq \alpha|V|$ such that the graph induced on S is k-colorable. Partial k-colorability is a more robust structural property of a graph than k-colorability. For graphs that arise in practice, partial k-colorability might be a better notion to use than k-colorability, since data arising in practice often contains various forms of noise.

We give a polynomial time algorithm that takes as input a $(1-\epsilon)$-partially 3-colorable graph G and a constant $\gamma \in [\epsilon, 1/10]$, and colors a $(1-\epsilon/\gamma)$ fraction of the vertices using $\tilde{O}\left(n^{0.25+O(\gamma^{1/2})}\right)$ colors. We also study natural semi-random families of instances of partially 3-colorable graphs and partially 2-colorable graphs, and give stronger bi-criteria approximation guarantees for these family of instances.

2012 ACM Subject Classification Mathematics of computing → Approximation algorithms

Keywords and phrases Approximation Algorithms, Vertex Coloring, Semi-random Models

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2019.28

Category APPROX

Related Version A full version of the paper is available at `https://arxiv.org/abs/1908.11631`.

Funding *Anand Louis*: Supported in part by SERB Award ECR/2017/003296.

Acknowledgements The first author thanks Pasin Manurangsi for pointing him to the Odd Cycle Transversal problem.

1 Introduction

Graph coloring problems are a central topic of study in the theory of algorithms [33, 17, 4, 20]. An undirected graph $G = (V, E)$ is said to be k-colorable if there exists an assignment of colors $f : V \to [k]$ such that $f(u) \neq f(v)$ for each $\{u, v\} \in E$. For a graph G, the minimum value of k for which it is k-colorable is called its chromatic number. Computing a 3-coloring of a 3-colorable graph is a fundamental NP-hard problem. Efficiently computing a coloring of a 3-colorable graph which only uses a few colors is a major open problem in the study of algorithms. The current best known algorithm colors a 3-colorable graph on n vertices using $O(n^{0.199})$ colors [20]. We study the problem of coloring partially colorable graphs.

▶ **Definition 1.** *An undirected graph $G = (V, E)$ is defined to be α-partially k-colorable, denoted by α-PkC , if there exists a subset $V_{\text{good}} \subset V$ such that $|V_{\text{good}}| \geq \alpha |V|$ and the graph induced on V_{good} is k-colorable. We will call such a set V_{good} the set of* good *vertices, and $V_{\text{bad}} \stackrel{\text{def}}{=} V \setminus V_{\text{good}}$ the set of* bad *vertices.*

We remark that for a given graph the partitioning of the vertex set V into V_{good} and V_{bad} may not be unique. In such cases, the claims we make in this paper will hold for any such fixed partition.

It is well known that for a fixed k, the problem of determining whether a given graph is k-colorable is an NP-hard problem [18]. Therefore, determining whether a graph belongs to 1-PkC is an NP-hard problem, and hence, computing the largest value of α for which a graph belongs to α-PkC is also an NP-hard problem.

Note that a graph that is $(1-\epsilon)$-partially 3-colorable can have chromatic number as large as $|V_{\text{bad}}| = \epsilon n$. Therefore, the notion of the chromatic number of the graph does not capture the structural property (3-colorability) satisfied by most of the graph. Partial k-colorability is a more robust stuctural property than k-colorability. Therefore, for graphs that arise in practice, partial k-colorability might be a better notion to use than k-colorability, since data arising in practice often contains various forms of noise; the notion of bad vertices can be used to capture some types of noisy vertices in the graph.

Other notions of partial k-coloring

Another related notion of partial coloring is the following.

▶ **Definition 2.** *An undirected graph $G = (V, E)$ is defined to be α-partially k-colorable, if there exists a coloring of the vertices $f : V \to [k]$ such that for at least $\alpha |E|$ edges $\{ u, v \}$, $f(u) \neq f(v)$.*

This definition, which asks that the coloring should "satisfy" at least α fraction of the edges, can be viewed as the *edge* version of partial k-colorability, whereas Definition 1 can be viewed as the *vertex* version of partial k-colorability. For a fixed constant k, computing the maximum value of α for which the input graph satisfies Definition 2 can be formulated as a Max-2-CSP with alphabet size k; approximation algorithms for Max-2-CSPs have been extensively studied in the literature [29, 30, 7] etc. Therefore, we focus our attention on Definition 1.

1.1 Our Results

We give an efficient (bi-criteria) approximation algorithm for coloring partially 3-colorable graphs.

▶ **Theorem 3.** *There exists a polynomial time algorithm that takes as input a $(1-\epsilon)$-P3C graph $G = (V, E)$ and any fixed choice of $\gamma \in [\epsilon, 1/100]$, and produces a set $S \subset V$ such that $|S| \leq (3\epsilon/\gamma)|V|$ and a coloring of $V \setminus S$ using $\tilde{O}(n^{0.25+O(\gamma^{1/2})})$ colors*[1].

We point out that the above theorem gives a bi-criteria approximation guarantee which exhibits the tradeoff between the size of the set S, and the number of colors used to color the remaining graph $G[V \setminus S]$. In particular, setting $\gamma = \sqrt{\epsilon}$ in the above theorem gives

[1] $\tilde{O}(\cdot)$ hides factors polylogarithmic in n.

us the following guarantee. Given a $(1-\epsilon)$-P3C graph, one can color $(1-\sqrt{\epsilon})$-fraction of its vertices using $\tilde{O}(n^{0.25+\epsilon^{1/4}})$-colors. Using similar techniques we can give an efficient approximation algorithm for the partial 2-coloring setting as well. For completeness, we formally state the result below[2] :

▶ **Proposition 4.** *There exists a polynomial time algorithm that takes as input a $(1-\epsilon)$-P2C graph $G=(V,E)$ and any fixed choice of $\gamma \in [\epsilon, 1/100]$, and produces a set $S \subset V$ such that $|S| \le (\epsilon/\gamma)|V|$ and a coloring of $V \setminus S$ using $\tilde{O}(n^{C\gamma})$ colors, for some constant $C > 0$.*

The proof of the above proposition can be found in the full version and uses exactly the same techniques as Theorem 3. We also study a semi-random family of partially colorable graphs α-PkC$^{\mathcal{R}}(n,p)$, which we define as follows.

▶ **Definition 5.** *An instance of α-PkC$^{\mathcal{R}}(n,p)$ is generated as follows.*

1. *Let V be a set of n vertices. Arbitrarily partition V into sets V_{good} and V_{bad} such that $|V_{\text{good}}| \ge \alpha n$.*
2. *Add edges between an arbitrary number of arbitrarily chosen pairs of vertices in V_{good} such that the graph induced on V_{good} is k-colorable.*
3. *Add edges between an arbitrary number of arbitrarily chosen pairs of vertices in V_{bad}.*
4. *Between each pair of vertices in $V_{\text{good}} \times V_{\text{bad}}$, independently add an edge with probability p. We call this set of edges E_0.*
5. *Add arbitrary number of edges between pairs of vertices of $V_{\text{good}} \times V_{\text{bad}}$. We call this set of edges E_1.*

Output the resulting graph.

In the study of approximation algorithms for NP-hard problems, there have been many works studying algorithms random and semi-random instances of various problems [11, 15, 21, 24, 25]. Random and semi-random instances are often good models for instances arising in practice; designing algorithms specifically for such instances, whose performance guarantee is significantly better than guarantees for general instances, could have more applications in practice. Moreover, from a theoretical perspective, designing algorithms for semi-random instances helps us to better understand what aspects of a problem make it intractable. We study our semi-random model α-PkC$^{\mathcal{R}}(n,p)$ for the same reasons. The following is our main result.

▶ **Theorem 6.** *Suppose there exists an efficient algorithm which colors a 3-colorable graph using n^{θ} colors. Then the following holds for all choices of $\epsilon = \Omega(\log n/n)$ and $p \ge (\epsilon\theta^{-2})^{O(\theta)}$. There exists a polynomial time algorithm that takes as input a graph G sampled from $(1-\epsilon)$-P3C$^{\mathcal{R}}(n,p)$ and produces a set S such that $|S| = O\left(\epsilon\theta^{-2}np^{-(O(1/\theta))}\right)$ and a coloring of $V \setminus S$ using at most n^{θ} colors with high probability. Moreover, the algorithm runs in time $n^{O(1/\theta)}\text{poly}(n)$.*

In particular, instantiating the above theorem with the algorithm from [20], w.h.p., we can color $(1-O(\epsilon))n$ fraction of vertices with $\tilde{O}(n^{0.199})$-colors. We also study the partial 2-coloring problem in the semi-random setting. Our guarantees for this setting are as follows:

▶ **Theorem 7.** *Let $\epsilon = \Omega(\log n/n)$ and $p > \sqrt{\epsilon}$. Then, there exists a polynomial time algorithm that takes as input a graph G sampled from $(1-\epsilon)$-P2C$^{\mathcal{R}}(n,p)$, and with high probability, produces a set $S \subseteq V$ such that $|S| = O\left(\epsilon np^{-2}\right)$ and the induced subgraph on the remaining vertices $G[V \setminus S]$ is 2-colorable.*

---

[2] We implicitly use the algorithm in the degree reduction step of the algorithm from Theorem 3. See Claim 18 for details.

In particular, in the above theorem the number of vertices removed is bounded by $O(\epsilon n)$ which is stronger than the best known bound of $O(\sqrt{\log n}.\epsilon n)$ [1] in the adversarial setting.

1.2 Related Work

3-colorable graphs. There is extensive literature on algorithms for coloring 3-colorable graphs. Wigderson [33] gave a simple combinatorial algorithm that used $O(n^{\frac{1}{2}})$ colors. Blum [9] improved the number of colors used to $\tilde{O}(n^{\frac{3}{8}})$. These algorithms used purely combinatorial techniques. Karger, Motwani and Sudan [17] used semidefinite programming to develop an algorithm, which when balanced with Wigderson's technique [33] used $\tilde{O}(n^{\frac{1}{4}})$ colors. Blum and Karger [10] improved the number of colors used to $\tilde{O}(n^{\frac{3}{14}})$ by combining the techniques used in [9] and [17]. Arora, Chlamtac and Charikar [3] got the bound down to $\tilde{O}(\Delta^{0.21111})$ using techniques from the ARV algorithm [5], which was further improved by Chlamtac [12] to $\tilde{O}(n^{0.2072})$ using SDP hierarchies. Using new combinatorial techniques, Kawarabayashi and Thorup improved the approximation bound to $\tilde{O}(n^{0.2049})$ in [19]. Subsequently, by combining their techniques with [12], they were able to give a approximation of $\tilde{O}(n^{0.19996})$ [20], which is the current state of the art.

Partially 2-colorable graphs. The partial 2-coloring problem, better known as Odd Cycle Transversal (OCT) in the literature, has also been studied extensively. Formally, the setting here is as follows. We are given a $(1-\epsilon)$-partially 2-colorable graph $G = (V, E)$ and the objective is to find a set S of minimum size such that $G[V \setminus S]$ is 2-colorable (i.e., odd cycle free). Yannakakis first showed that it is NP-Complete in [34]. Later, Khot and Bansal [6] showed that OCT is hard to approximate to any constant factor, assuming the Unique Games Conjecture. From the algorithmic side, via a reduction through the Min2CNF Deletion problem, [16] gave a $O(\log n)$ approximation for the problem. This was later improved to $O(\sqrt{\log n})$ by [1] by using techniques from the Arora-Rao-Vazirani [5] algorithm for sparsest cut. This problem has also been studied under the lens of parameterized complexity. In [31], Reed et al. showed that OCT is fixed parameter tractable when parameterized by the number of bad vertices, following which a sequence of works [6, 28, 23] gave algorithms with improved running times.

Partially 3-colorable graphs. In contrast to the 3-colorable setting, there has been very little work on coloring partially 3-colorable graph. The paper which is closest to our setting is by Kumar, Louis and Tulsiani [22], which also addresses the partial 3-coloring problem, albeit in a more restrictive setting. Assuming that the $(1-\epsilon)$-partially 3-colorable graph has threshold rank r and the 3-coloring on the good vertices satisfies certain psuedorandomness properties, they give an algorithm which 3-colors $1 - O(\gamma + \epsilon)$ fraction of vertices in time $(r.n)^{O(r)}$.

Graph problems in Semi-random Models. The semi-random model used in this paper is similar to semi-random models which have been considered for the Max-Independent Set problem [11] [15] [32] [26]. Semi-random models offer a natural way of understanding the complexity of problems in settings which are less restrictive than worst case complexity, but are still far from being average case. While semi-random models were first introduced for studying graph coloring in [11], it has also subsequently been used to study several other fundamental problems such as Unique Games [21], Graph Partitioning [24], Clustering [25], to name a few.The problem of coloring 3-colorable graphs has also been studied in average-case and planted models. Alon and Kahale [2] gave an efficient algorithm that finds an exact

3-Coloring of a random 3-Colorable graph with high probability. David and Fiege [13] studied the complexity of finding a planted random/adversarial 3-coloring for both adversarial and random host graphs.

1.3 Discussion and Proof Overview

Adversarial Model. The key component in most approximation algorithms for 3-coloring involves solving a SDP relaxation of the 3-coloring problem, and followed by a randomized rounding procedure for coloring the graph. The standard SDP relaxation for 3-coloring is the following which was introduced in [17]:

▶ **SDP 8** (Exact 3-Coloring SDP).

$$\begin{aligned} &\textit{minimize} && 0 \\ &\textit{subject to} && v_i \cdot v_j \leq -\tfrac{1}{2} && \forall \{i,j\} \in E \\ & && \|v_i\|^2 = 1 && \forall i \in V \end{aligned}$$

SDP 8 doesn't optimize any objective function, it finds a feasible solution which satisfies all the constraints of SDP. The intended solution to the above SDP is as follows. Let $\sigma : V \to \{1,2,3\}$ be any legal coloring of G. Furthermore, let $u_1, u_2, u_3 \in \mathbb{R}^2$ be any three unit vectors satisfying $\langle u_i, u_j \rangle = -1/2$ for every $i, j \in \{1,2,3\}, i \neq j$. We identify the vector u_i with the color i, and assign $v_j = u_{\sigma(j)}$ for every $j \in V$. It can be easily verified that this is a feasible solution to the above SDP. As is usual, while the SDP in general may not return the above vector coloring, one can round a feasible vector coloring to color the graph using not too many colors [17]. The approximation guarantee is usually of the form Δ^c (for some $c \in (0,1)$), where Δ is the maximum degree of the graph.

Since in general, one cannot hope to have a degree bound on the graph, the above step is usually preceded by a *degree reduction* sub-routine. Note that if a graph is 3-colorable (more generally k-colorable), then the graph induced on the neighbours of any vertex v is 2-colorable (more generally $k-1$ colorable). Since a 2-colorable graph can be colored with 2 colors efficiently, the graph induced on any vertex and its neighbours can be colored efficiently with 3 colors. Therefore, fixing a threshold Δ, this procedure iteratively removes vertices (and their neighbours) having degree larger than Δ from the graph while coloring them with few colors, and terminates when maximum degree of the remaining graph is at most Δ. In particular, if the degree reduction step uses $f(n,\Delta)$ colors, then the total number of colors used by the algorithm is at most $f(n,\Delta) + \Delta^c$. Then one can optimize the choice of Δ for giving the best possible approximation guarantee. This degree reduction approach and its variants, first studied by Wigderson [33], has been subsequently used in almost all known approximation algorithms for graph coloring

In translating the above template to the setting of partially 3-colorable graphs, we face several immediate challenges. SDP 8 is guaranteed to return a feasible solution only for 3-colorable graphs, it might be infeasible if the graph is not 3-colorable. If we could compute the set of good vertices then we could use SDP 8 only on the set of good vertices. However, in general, the problem of identifying the set of good vertices is NP-hard (Fact 27). Finally, the preprocessing steps for degree reduction rely heavily on the combinatorial structural properties of the neighborhood of vertices in *exactly* 3-colorable graphs, which, in general, may not be satisfied by *partially* 3-colorable graphs.

Our approach is to begin with an SDP relaxation that tries to solve both problems together: identifying the set of bad vertices, and coloring the set of good vertices. We introduce variables $w_1, w_2, \ldots, w_n$ where the i^{th} variable w_i is meant to indicate if vertex

i is bad. Additionally, for every edge $(i, j) \in E$, we introduce slack variables z_{ij} which are meant to indicate if at least one of the vertices i, j is bad. Using the slack variables we relax the edge constraints as $\langle v_i, v_j \rangle \le -1/2 + (3/2)z_{ij}$. Finally, we connect the edge indicator variables with vertex indicator variables using constraints of the form $z_{ij} \le w_i + w_j$. Since we want the set of bad vertices to be small, our objective function will be to minimize $\sum_{i \in V} w_i$. Our SDP relaxation is the following.

▶ **SDP 9** (Partial 3-Coloring SDP).

$$
\begin{array}{lll}
\textit{minimize} & \sum_{i \in V} w_i & \\
\textit{subject to} & \langle v_i, v_j \rangle \le -\frac{1}{2} + \frac{3}{2} z_{ij} & \forall \{i, j\} \in E \\
& z_{ij} \le w_i + w_j & \forall \{i, j\} \in E \\
& 0 \le z_{ij} \le 1 & \forall \{i, j\} \in E \\
& 0 \le w_i \le 1 & \forall i \in V \\
& \|v_i\|^2 = 1 & \forall i \in V
\end{array}
$$

Since the optimal "integer solution" forms a feasible solution to the SDP relaxation, it is easy to show that for a $(1 - \epsilon)$-partially 3-colorable graph, the optimal of the above SDP is at most ϵn. Therefore by Markov's inequality, we get that for a large fraction of $i \in [n]$, the w_i variables are small. Let $V' \subset V$ be the set of vertices with small w_i. Since $|V \setminus V'| = O(\epsilon n)$, we can focus on coloring the induced subgraph $G' = G[V']$. G' has the following nice property: *for every edge (i, j) in G', the corresponding edge constraint is approximately satisfied i.e., $\langle v_i, v_j \rangle \le -1/2 + o_\epsilon(1)$, where the second term goes to 0 as ϵ goes to 0.* We call such graphs as being approximately vector 3-colorable (See Definition 11 for a formal description). We use this property crucially in designing our preprocessing step.

We observe that the neighborhood of any vertex in an approximately vector 3-colorable graph is approximately vector 2-colorable. Furthermore, we show that approximately vector 2-colorable graphs are *short odd cycle* free. Graphs having this property are known to have large independent sets which can be found efficiently [27]. Thus one can find such large independent sets recursively to color the neighborhood of large degree vertices using a small number of colors.

For the randomized rounding step, we observe that hyperplane rounding based procedures are naturally robust to small perturbations, and the arguments for analyzing the guarantees of such procedures hold even when the edge constraints are approximately satisfied. In particular, we can use known randomized rounding algorithm as is, while adapting the analysis to account for the edge constraints being satisfied approximately.

Semi-random model. While the guarantees of our algorithm from the adversarial setting also apply to the semi-random instances, here we seek to achieve the best known approximation bounds for exactly 3-colorable graphs. We begin by describing two distinct classes of instances which illustrate the technical challenges in designing such an algorithm.

In this setting, the adversary is free to choose $G[V_{\text{bad}}]$ in a way such that it is noisy and has large chromatic number (e.g, graphs sampled from Erdos Renyi random model). For such instances, it is easy to see that the only way an algorithm can have good approximation guarantees is when it can eliminate a significant fraction of from V_{bad}. Then, for a start, one can hope to address this setting by first using a preprocessing step that deletes V_{bad} and then running the best possible approximation algorithm on the graph induced on the remaining vertices.

On the other hand, the adversary can also choose $G[V_{\text{bad}}]$ in a way so that it is *structurally indistinguishable* from the good subgraph $G[V_{\text{good}}]$. For instance, suppose the good subgraph $G[V_{\text{good}}]$ is a randomly sampled unbalanced bipartite graph, where the smaller side (which we call V_S) has size at most ϵn. Then the adversary can choose V_{bad} to be an independent set, in which case the entire graph is 3-colorable. In particular, it is information theoretically impossible to distinguish the set V_S from V_{bad}, since they are both independent sets and the edges incident on them are identically distributed. While the instances constructed here make it difficult to identify V_{good}, they are also naturally easy instances for us. In particular, these instances are also $(1-\epsilon)$-partially 2-colorable, and one can use tools for coloring partially 2-colorable graphs to color these instances with small number of colors.

However, the two cases above clearly do not cover the full range of instances that we can encounter in our model. Therefore, we need a way to relax the above two characterizations which allows for a seamless transition from one class of instances to other. It turns out that we can robustly characterize both classes of instances by the number of *vertex disjoint short odd cycles* present in the graph. Informally, if the number of short odd cycles is large, then with high probability, they will show up in the neighborhood of the bad vertices, and therefore this can be used to identify and eliminate V_{bad}. We can then simply run the best known approximation algorithm on the remaining induced graph $G[V_{\text{good}}]$. On the other hand, if the number of short odd cycles is small, by eliminating a small fraction of vertices, we can make the graph short odd cycle free. Finally, as discussed in the adversarial model setting, such graphs can be colored efficiently using a small number of colors by recursively finding large independent sets [27].

2 Preliminaries

We introduce some notation used frequently in this paper. Throughout the paper, for a $(1-\epsilon)$-partially 3-colorable graph $G = (V, E)$, we will write $V = V_{\text{good}} \uplus V_{\text{bad}}$ where V_{good} and V_{bad} are the set of good vertices and bad vertices as defined in Definition 1. For a subset $V' \subseteq V$, we use $G[V']$ to denote the subgraph induced on the set of vertices V'. For a subgraph $G' \subseteq G$, we shall use $\text{vert}(G')$ to denote the vertex set of G'. Additionally, for any vertex $i \in \text{vert}(G')$, we use $N_{G'}(i)$ denote the set of neighbors of i in the graph G'. We use $\mathbb{1}(\cdot)$ to denote the indicator function, and $\tilde{O}(\cdot)$ to hide terms which are polylogarithmic in the number of vertices.

Approximate Vector Coloring

We begin by recalling the notion of vector coloring of a graph which was introduced in [17].

▶ **Definition 10** (Vector Coloring). *Given a positive integer $k \in \mathbb{N}$, we say that a graph $G = (V, E)$ is k-vector colorable if there exists unit vectors $v_1, v_2, \ldots, v_n \in \mathbb{R}^d$ for some $d \in \mathbb{N}$ which satisfy*

$$\langle v_i, v_j \rangle \leq -\frac{1}{k-1} \qquad \forall \{ i, j \} \in E.$$

We will use the notion of *approximate vector colorings* of a graph, which we define as follows.

▶ **Definition 11** (Approximate Vector Coloring). *Given a positive integer $k \in \mathbb{N}$ and a $\gamma > 0$, we say that a graph $G = (V, E)$ is (k, γ)-vector colorable if there exists unit vectors $v_1, v_2, \ldots, v_n \in \mathbb{R}^d$ for some $d \in \mathbb{N}$ which satisfy*

$$\langle v_i, v_j \rangle \leq -\frac{1}{k-1} + \gamma \qquad \forall \{ i, j \} \in E.$$

Observe that a graph that $(k,0)$ vector colorable is vector-k-colorable. We now state a couple of lemmas which illustrate some useful properties of approximate vector colorings. In [17], it was observed that the vector chromatic number of sub-graph induced on the neighborhood of a vertex is strictly less than the vector chromatic number of the actual graph. In the following lemma, we observe that this property can be extended to approximate vector colorings as well.

▶ **Lemma 12.** *Let $G = (V,E)$ be $(3,\gamma)$-vector colorable, for some $0 < \gamma < 1/10$. Then for any vertex $i \in V$, the graph induced on $N(i)$ is $(2,4\gamma)$-vector colorable.*

The next lemma says that approximately vector 2-colorable graphs cannot contain short odd cycles.

▶ **Lemma 13.** *Let $G = (V,E)$ be a $(2,\gamma)$-vector colorable, where $\gamma \le 1/16$. Then G does not contain odd cycles of length at most $1/8\sqrt{\gamma}$.*

The proofs of the two lemmas above can be found in Appendix B.

Coloring graphs without short odd cycles

A key combinatorial tool used in our paper is the following Ramsey theoretic result which says that graphs without short odd cycles contain large independent sets which can be found efficiently.

▶ **Lemma 14** ([27])**.** *There exists a constant $\epsilon_0 \in (0,1)$ such that for every choice of $0 < \epsilon < \epsilon_0$ the following holds. Let $G = (V,E)$ be a graph without odd cycles of length at most $1/\epsilon$. Then, G contains an independent set of size at least $|V|^{1-2\epsilon}$. Furthermore, there exists a polynomial time algorithm which finds such an independent set.*

Consequently, given a graph without short odd cycles, one can color it efficiently using a small number of colors, as stated in the following corollary.

▶ **Corollary 15.** *There exists a constant $\epsilon_0 \in (0,1)$ for which the following holds. Given a graph $G = (V,E)$ which does not contain odd cycles of length at most $1/\epsilon$ where $\epsilon < \epsilon_0$, there exists a polynomial time algorithm which can compute a coloring of G using $\tilde{O}(n^{2\epsilon})$ colors.*

Establishing the above corollary using Lemma 14 is straightforward, and just uses the fact that one can keep removing large independent sets in the graph using Lemma 14, and recurse on the remaining vertices. For the sake of completeness, we include a proof in Appendix C.

3 Approximation algorithm for General Setting

In this section, we prove our approximation guarantees in the adversarial model, as formally stated in the following theorem:

▶ **Theorem 16** (Theorem 3 restated)**.** *There exists a polynomial time algorithm that takes as input a $(1-\epsilon)$-P3C graph $G = (V,E)$ and any fixed choice of $\gamma \in [\epsilon, 1/100]$, and produces a set $S \subset V$ such that $|S| \le (3\epsilon/\gamma)\,|V|$ and a coloring of $V \setminus S$ using $\tilde{O}(n^{0.25+O(\gamma^{1/2})})$ colors.*

The algorithm for the above theorem is described in Algorithm 1. In the following subsections, we prove the correctness of the above algorithm. The proof of Theorem 3 can broken down into the analysis of steps (i),(ii) and (iii) of the Partial-3-Coloring algorithm. Broadly, we show the following: In step (i), we show that the optimal of the SDP-P3C is

Algorithm 1 Partial-3-Coloring.

Set $\Delta = n^{3/4}$;
Solve the Partial-3-Coloring SDP (SDP-P3C):

$$\begin{aligned}
\text{minimize} \quad & \sum_{i \in V} w_i \\
\text{subject to} \quad & \langle v_i, v_j \rangle \le -\frac{1}{2} + \frac{3}{2} z_{ij} && \forall \{i,j\} \in E \\
& z_{ij} \le w_i + w_j && \forall \{i,j\} \in E \\
& 0 \le z_{ij} \le 1 && \forall \{i,j\} \in E \\
& 0 \le w_i \le 1 && \forall i \in V \\
& \|v_i\|^2 = 1 && \forall i \in V
\end{aligned}$$

(i) Thresholding:
Let $S \leftarrow \{i \in V | w_i \ge \gamma/3\}$;
Let $G' \leftarrow G[V \setminus S]$ be the graph obtained after deleting S;
(ii) Coloring Large Degree vertices:
while $\exists i \in G'$ *such that* $\deg_{G'}(i) \ge \Delta$ **do**
    Color $G'[\{i\} \cup N_{G'}(i)]$ using $\tilde{O}(n^{C\sqrt{\gamma}})$ colors using the algorithm guaranteed by Corollary 15;
    Remove $\{i\} \cup N_{G'}(i)$ from G';
end
(iii) Coloring Low Degree vertices:
Use *randomized rounding* from Theorem 19 to color the remaining vertices in G';

small (i.e., at most ϵn), therefore by averaging, the fraction of large w vertices is small. Furthermore, the graph induced on the surviving vertices must satisfy the edge constraints from the SDP with small slack γ, and therefore must be approximately vector 3-colorable. As is usual in coloring algorithms, we first iteratively color large degree (i.e., $\ge \Delta$) vertices and their neighborhoods using small number of colors until the graph has degree bounded by Δ (Claim 18). Finally, the remaining graph is also approximately vector 3-colorable, and has degree bounded by Δ. Therefore, using a hyperplane based randomized rounding procedure to iteratively find large independent sets in G', we can give a $\tilde{O}(\Delta^{1/3+O(\sqrt{\gamma})})$ coloring of the remaining vertices (Theorem 19). In the following subsection, we formally prove the steps described above.

To begin with, we first show that the thresholding step throws away at most a small fraction of vertices.

▷ Claim 17 (Removing Large Slack Vertices). Let $S \subset V$ be as constructed in the thresholding step. Then $|S| \le 3\epsilon n/\gamma$.

We defer the proof of the above claim to Appendix A. From the above claim, the graph $G' = G[V \setminus S]$ induced on the remaining vertices satisfies the following properties:

1. The graph G' contains at least $(1 - 3\epsilon/\gamma)n$ vertices.
2. The graph G' is $(3, \gamma)$-vector colorable. In particular, the vectors $(v_i)_{i \in V \setminus S}$ themselves are a $(3, \gamma)$-vector coloring of G'.

The second point shall be used crucially in the analysis of the remaining two steps. The next claim bounds the number of colors used while coloring the large degree vertices in step (ii).

▷ Claim 18 (Degree Reduction). In step (ii), over all the iterations of the while loop, the algorithm uses at most $(n/\Delta)\tilde{O}\left(n^{C\sqrt{\gamma}}\right)$ colors, where $C > 0$ is a constant.

Proof. Fix any vertex $i \in G'$, and let $\tilde{G}_i = G'[N(i)]$ the graph induced on the neighborhood of vertex i. Since the graph G' is $(3, \gamma)$-vector colorable, using Lemma 12 we know that $\tilde{G}_i$ is $(2, 4\gamma)$-vector colorable. Furthermore, from Lemma 13, we know that G' does not contain odd cycles of length at most $1/(8\sqrt{4\gamma})$. Therefore, we can use Corollary 15 to obtain a $\tilde{O}(n^{C\sqrt{\gamma}})$ coloring of $\tilde{G}_i \cup \{i\}$. Finally, note that each iteration of the for loop removes and colors at least $\Delta + 1$ vertices of the graph. Therefore, the total number of iterations of the for loop is bounded by n/Δ. Since in each such iteration we can color the vertex and its neighborhood using $n^{C\sqrt{\gamma}}$ number of colors, the claim follows. ◁

After steps (i) and (ii), we are left with the graph $G' = (V', E')$ which is $(3, \gamma)$-vector colorable graph and has degree at most Δ. In particular, for every edge $(i, j) \in E'$, the corresponding vectors satisfy $\langle v_i, v_j \rangle \leq -\frac{1}{2} + \gamma$. Since the independent set based rounding technique [17] [3] for coloring vector 3-colorable graphs is *robust*, we can still use it to round the vector coloring of approximately 3-colorable graphs with similar guarantees, as formally stated in the following theorem.

▶ **Theorem 19.** *Let $G = (V, E)$ be a graph with maximum degree Δ which is $(3, \alpha)$-vector colorable. Then there exists an efficient randomized algorithm that can color it using $O\left((\ln \Delta)^{1/2} \Delta^{\frac{\frac{3}{4}+\alpha-\alpha^2}{(\frac{3}{2}-\alpha)^2}} \ln n\right)$ colors.*

In particular, if $\alpha \leq 1/10$, then the algorithm uses at most $\tilde{O}\left((\ln \Delta)^{1/2} \Delta^{\frac{1}{3}+10\alpha}\right)$, where $\tilde{O}$ hides polylogarithmic factors in n.

The proof of the above theorem is an extension of the proofs from [17, 3] to the setting of approximately vector 3-colorable graphs. Due to space constraints, we skip the proof here and provide it in the full version. Here for simplicity assume that $\gamma \leq 1/10$. Instantiating the above theorem with $G = G'$ and $\alpha = \gamma$, we get that G' is colored using $\tilde{O}(\Delta^{1/3+10\gamma})$ colors. Overall, the algorithm throws away at most $2\epsilon/\gamma$ fraction of vertices in step (i). Furthermore, it uses a total of $\tilde{O}\left((n/\Delta)n^{O(\sqrt{\gamma})} + \Delta^{1/3+10\gamma}\right)$ colors in steps (ii) and (iii) respectively. Setting $\Delta = n^{3/4}$ in the previous expression, we get that the algorithm uses at most $\tilde{O}(n^{1/4+O(\sqrt{\gamma})})$ colors. This concludes the analysis of the Partial-3-Coloring algorithm and the proof of Theorem 3.

4 Algorithm for Semi-random instances

In this section, we prove Theorem 6, which we again state here for convenience.

▶ **Theorem 20** (Theorem 6 restated). *Suppose there exists an efficient algorithm which colors a 3-colorable graph using n^θ colors. Then the following holds for all choices of $\epsilon = \Omega(\log n/n)$ and $p \geq (\epsilon\theta^{-2})^{O(\theta)}$. There exists a polynomial time algorithm that takes as input a graph G sampled from $(1-\epsilon)$-$\mathcal{P}3\mathcal{C}^{\mathcal{R}}(n, p)$ and produces a set S such that $|S| = O\left(\epsilon\theta^{-2} n p^{-(O(1/\theta))}\right)$ and a coloring of $V \setminus S$ using at most n^θ colors with high probability. Moreover, the algorithm runs in time $n^{O(1/\theta)}\mathrm{poly}(n)$.*

Algorithm 2 P3C-Random.

Let $\mathcal{A}$ be the algorithm which can color 3-colorable graphs using n^{θ} colors;
Set $\delta = \theta/10$;

$\{$*Many short odd cycles*$\}$:
for *every vertex* $v \in V$ **do**
  Let $G_v := G[N_G(v)]$ the subgraph induced by the neighborhood of G;
  Greedily construct a maximal set $\mathcal{C}_v$ of vertex disjoint odd cycles of length at most $1/\delta$ in G_v;
end
Construct set $S \leftarrow \{v \in V : |\mathcal{C}_v| \geq 2\epsilon n\}$;
Let $G_0 \leftarrow G[V \setminus S]$ be the graph obtained after deleting S;
Let σ_1 be the coloring of $V \setminus S$ obtained by running algorithm $\mathcal{A}$ on G_0. Let L denote the number of colors used by the algorithm;

$\{$*Few short odd cycles*$\}$:
Compute a maximal set $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ of vertex disjoint odd cycles in G of length at most $1/\delta$ using greedy algorithm;
Let $V' = V \setminus \left(\bigcup_{i \in [m]} \mathrm{vert}(C_i)\right)$;
Use the algorithm guaranteed by Corollary 15 to give a $\tilde{O}\left(n^{4\delta}\right)$ coloring σ_2 of $G[V']$;

$\{$*Output best coloring*$\}$:
if $|S| \leq \epsilon n$ *and* $L \leq n^{\theta}$ **then**
  Output coloring σ_1 of $V \setminus S$
end
else
  Output coloring σ_2 of V';
end

We begin by describing the algorithm for the semi-random setting:

The algorithm proceeds case wise depending on whether there exists many vertex disjoint short odd cycles in G. If it does, then since V_{bad} is small, $G[V_{\mathrm{good}}]$ must also contain many vertex disjoint odd cycles. We show that these short cycles will show up in the neighborhood of the bad vertices with high probability, which can be used to identify them. On removing these vertices, we will be left with a 3-colorable graph. On the other hand, if the number of short odd cycles is small, we can remove them. The remaining graph will still contain most of the vertices and will be short odd cycle free. We can then use Lemma 14 to recover large independent sets. Finally, since the odd cycles we consider are of length at most $1/\delta$, we can work with a *maximal* set of vertex disjoint odd cycles, instead of the largest cardinality set of vertex disjoint odd cycles, while only losing a factor of $1/\delta$ in our analysis.

4.1 Correctness of the P3C-Random algorithm

Let $\mathcal{C}^* = \{C_1^*, C_2^*, \ldots, C_{m^*}^*\}$ be a *fixed largest cardinality set of vertex disjoint odd cycles* of length at most $1/\delta$ in $G[V_{\mathrm{good}}]$. *In particular,* $\mathcal{C}^*$ *and consequently* m^*, *does not depend on the realization of the random and adversarial edges (i.e., the* E_0 *and* E_1 *edges) between* V_{good} *and* V_{bad}. We break our analysis into two cases depending on whether m^* is small or large.

Case (i): $m^* > 4\epsilon n/(\delta p^{1/\delta})$. For ease of exposition, we say that an odd cycle C in graph G is *good* if it consists of only good vertices, otherwise we call it *bad*. The first claim shows the set $\mathcal{C}_v$ must be small for good vertices.

▷ Claim 21. For every good vertex $v \in V$, we have $|\mathcal{C}_v| \leq \epsilon n$.

Proof. Fix a good vertex $v \in V_{\text{good}}$. We claim that a good cycle C can never appear in the neighborhood of a good vertex. For contradiction, let C be a good odd cycle appearing in the neighborhood of v. Let $\tilde{G} = G\Big[\text{vert}(C) \cup \{v\}\Big]$ be the subgraph induced on the vertex v and the vertices from cycle C. Since $\tilde{G} \subseteq G[V_{\text{good}}]$, the subgraph $\tilde{G}$ is also 3-colorable. Hence, the neighborhood of v in the induced subgraph $\tilde{G}$ must be 2-colorable, and therefore it cannot contain odd cycles, and in particular C. This gives us the contradiction.

Hence, any odd cycle which appears in the neighborhood $N_G(v)$ must be bad. Since the number of bad vertices is bounded by ϵn, and the cycles in $\mathcal{C}_v$ are vertex disjoint, the claim follows. ◁

On the other hand, with high probability, we show that $|\mathcal{C}_v|$ is large for all the bad vertices.

▷ Claim 22. With probability at least $1 - e^{-O(\epsilon n)}$, every vertex $v \in V_{\text{bad}}$ satisfies $|\mathcal{C}_v| \geq 2\epsilon n$.

Proof. Consider the subgraph $G'(V, E_0)$ consisting of edges from E_0 (i.e., the randomly distributed set of edges). Fix a bad vertex $v \in V_{\text{bad}}$, and let $G_v = G[N_G(v)]$ denote the subgraph induced by the neighborhood of v. We shall first give a high probability lower bound on the number of odd cycles from $\mathcal{C}^*$ which can appear in $N_G(v)$. Recall that $|\mathcal{C}^*| = m^*$. We also point out again that the choice of $\mathcal{C}^*$ is not affected by the choice of E_0 and E_1 edges, and can be fixed ahead.

For every $i \in [m^*]$, we define $Z_i := \mathbb{1}\Big(\text{vert}(C_i^*) \subseteq N_{G'}(v)\Big)$ to be the indicator random variable that the i^{th} cycle appears in the neighborhood of vertex v in the graph G'. Note that these random variables depend only on the realization of the E_0 edges. Then we have

$$\begin{aligned} \mathbb{E}_G[Z_i] \geq \Pr_{E_0}\Big[\text{vert}(C_i^*) \subseteq N_G(v)\Big] &\geq \Pr_{E_0}\Big[\text{vert}(C_i^*) \subseteq N_{G'}(v)\Big] \\ &= \Pr_{E_0}\Big[\forall j \in \text{vert}(C_i^*), j \in N_{G'}(v)\Big] \\ &\geq p^{|C_i^*|} \geq p^{1/\delta} \end{aligned}$$

Here the last step uses the fact that any cycle $C_i^* \in \mathcal{C}^*$ has length at most $1/\delta$. It follows that

$$\mathbb{E}_G\left[\sum_{i \in [m^*]} Z_i\right] = \sum_{i \in [m^*]} \mathbb{E}_G[Z_i] \geq m^* p^{1/\delta} \geq (4\epsilon/\delta)n \tag{1}$$

Furthermore, since the cycles $C_1^*, C_2^*, \ldots, C_{m^*}^*$ are vertex disjoint, the corresponding random variables $Z_1, Z_2, \ldots, Z_{m^*}$ are also independent. Therefore using Chernoff bound we get that

$$\Pr_G\left[\sum_{i \in [m^*]} Z_i < (2\epsilon/\delta)n\right] \leq \Pr_G\left[\sum_{i \in [m^*]} Z_i < \frac{1}{2}\mathbb{E}\Big[\sum_{i \in [m^*]} Z_i\Big]\right] \leq e^{-\epsilon n/4\delta} \tag{2}$$

Now let $\mathcal{C}_v^* = \{C_i^* : i \in [m^*], Z_i = 1\}$ be the set of cycles from $\mathcal{C}^*$ which appear in the neighborhood of v in graph G due to the E_0 edges. Furthermore, let $\widetilde{\mathcal{C}}_v$ be a *largest cardinality* set of vertex disjoint odd cycles of length at most $1/\delta$ in G_v (which contains edges from both E_0 and E_1). Then by definition we have $|\widetilde{\mathcal{C}}_v| \geq |\mathcal{C}_v^*|$. On the other hand, by construction, the set $\mathcal{C}_v$ is a *maximal set* of such vertex disjoint odd cycles in G_v, and therefore, it must be a δ-approximation to the largest cardinality set $\widetilde{\mathcal{C}}_v$ i.e., $|\mathcal{C}_v| \geq \delta|\widetilde{\mathcal{C}}_v|$ (see Proposition 26). Therefore using Equation 2, with probability at least $1 - e^{-\epsilon n/4\delta}$ we have

$$|\mathcal{C}_v| \geq \delta|\widetilde{\mathcal{C}}_v| \geq \delta|\mathcal{C}_v^*| \geq 2\epsilon n$$

Hence, for any fixed vertex $v \in V_{\text{bad}}$, w.h.p. we have $|\mathcal{C}_v| \geq 2\epsilon n$. Therefore, by a union bound and using the lower bound on ϵ, we get that $\Pr_G\left[\exists v \in V_{\text{bad}} : |\mathcal{C}_v| < 2\epsilon n\right] \leq \epsilon n e^{-\epsilon n/4\delta} \leq n e^{-\epsilon n/8\delta}$. ◁

Combining the two claims above, it follows that w.h.p. the set $(V \setminus S)$ must exactly be the set of good vertices, and therefore $G[V \setminus S]$ must be 3-colorable. Hence algorithm $\mathcal{A}$ will give a n^θ coloring of $G[V \setminus S]$.

Case (ii): $m^* \leq 4\epsilon n/(\delta p^{1/\delta})$**.** Let $\mathcal{C} = \mathcal{C}_{\text{good}} \uplus \mathcal{C}_{\text{bad}}$ be the partition of $\mathcal{C}$ into the set of good and bad cycles respectively. Then, since $\mathcal{C}_{\text{good}}$ is a set of vertex disjoint odd cycles of length at most $1/\delta$ in $G[V_{\text{good}}]$, it follows that $|\mathcal{C}_{\text{good}}| \leq |\mathcal{C}^*| \leq 4\epsilon n/(\delta p^{1/\delta})$. Furthermore, by arguments similar to the proof of Claim 21, we have $|\mathcal{C}_{\text{bad}}| \leq \epsilon n$. Therefore, combining the two bounds, we have $|\mathcal{C}| \leq 5\epsilon n/(\delta p^{1/\delta})$. Since every cycle $C \in \mathcal{C}$ contains at most $1/\delta$ vertices, the total number of vertices thrown away at this step is at most $5\epsilon n/(\delta^2 p^{1/\delta})$. Furthermore, using the *maximality* of $\mathcal{C}$, we know that the induced subgraph $G' = G[V']$ must be free of odd cycles of length at most $1/\delta$. Therefore, using Corollary 15, we can color G' using $\tilde{O}(n^{2\delta})$ colors. This concludes the analysis of case (ii).

Putting Things Together. If case (i) holds, then w.h.p., in the *Many short odd cycles* block of the algorithm, the set S constructed is identical to V_{bad}, in which case the algorithm $\mathcal{A}$ will find a n^θ-coloring of $G[V \setminus S] = G[V_{\text{good}}]$. In particular, this implies that the conditions of the "if" block will be satisfied and the algorithm will return a n^θ-coloring of $(1-\epsilon)n$ vertices.

On the other hand, if case (ii) holds, we know that $m \leq 5\epsilon n/(p^{1/\delta}\delta)$, and the *Few short odd cycles* block deletes at most $5\epsilon n/(p^{1/\delta}\delta^2)$ vertices, and colors the remaining vertices using $\tilde{O}(n^{2\delta})$ colors. Then the *else* block of the algorithm will return a $\tilde{O}(n^{2\delta})$ coloring of $\left(1 - 5\epsilon n/(p^{1/\delta}\delta^2)\right)n$ vertices. Since the else block is evaluated only when the conditions of the *if* block are not satisfied, it follows that in this case, the algorithm will throw away at most $\max\left(\epsilon n, 5\epsilon n/(\delta^2 p^{1/\delta})\right) = O(\epsilon n/\delta^2 p^{1/\delta})$ vertices, and color the remaining graph with at most $\max\left(n^\theta, \tilde{O}(n^{2\delta})\right) = n^\theta$ colors.

Combining the two above cases gives us Theorem 6.

5 Conclusion

In this work we consider the problem of coloring partial 3-colorable graphs in adversarial and semi-random settings. In the adversarial setting, we give an efficient approximation algorithm which can color $(1 - O(\epsilon^c))$-fraction of vertices using $\tilde{O}(n^{0.25+\epsilon^{c'}})$ colors. On the other hand, the best known approximation guarantees for 3-colorable graphs is $n^{0.199}$ [20]. An obvious open question here is to achieve analogous approximation bounds for partially 3-colorable graphs as well.

One direct way to improve on our approximation bounds in the adversarial setting is through the use of more efficient degree reduction mechanisms as typically done in the exact 3-coloring setting [10],[19, 20] using combinatorial techniques like Blum's coloring tools [8]. However, these tools rely on fragile combinatorial properties present in 3-colorable graphs (e.g. two vertices whose common neighborhood is not an independent set must have the same color in any legal coloring), and as such, it is not obvious how to extend these techniques to the setting of partially 3-colorable graphs.

In the semi-random model, we show how any efficient algorithm for exact 3-coloring that uses n^θ colors can be leveraged to obtain an efficient algorithm in this setting which uses the same number of colors with high probability and also does not remove too many vertices. An obvious next step would be to see if similar results can also be obtained for partially k-colorable graphs with $k > 3$. Another interesting question would be to see if one can design efficient approximation algorithms with similar guarantees, where the adversary can also delete the randomly sampled edges.

References

1 Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. O(sqrtlog n)) approximation algorithms for min UnCut, min 2CNF deletion, and directed cut problems. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 573–581. ACM, 2005.

2 Noga Alon and Nabil Kahale. A spectral technique for coloring random 3-colorable graphs. *SIAM Journal on Computing*, 26(6):1733–1748, 1997.

3 Sanjeev Arora and Eden Chlamtac. New approximation guarantee for chromatic number. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 215–224. ACM, 2006.

4 Sanjeev Arora and Rong Ge. New Tools for Graph Coloring. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 14th International Workshop, APPROX 2011, and 15th International Workshop, RANDOM 2011, Princeton, NJ, USA, August 17-19, 2011. Proceedings*, pages 1–12, 2011.

5 Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009.

6 Nikhil Bansal and Subhash Khot. Optimal long code test with one free bit. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 453–462. IEEE, 2009.

7 Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding Semidefinite Programming Hierarchies via Global Correlation. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 472–481, 2011.

8 Avrim Blum. Some tools for approximate 3-coloring. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 554–562. IEEE, 1990.

9 Avrim Blum. New Approximation Algorithms for Graph Coloring. *J. ACM*, 41(3):470–516, 1994. doi:10.1145/176584.176586.

10 Avrim Blum and David Karger. An algorithm for 3-colorable graphs. *Information Processing Letters*, 61(1):49–53, 1997. doi:10.1016/s0020-0190(96)00190-1.

11 Avrim Blum and Joel Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19(2):204–234, 1995.

12 Eden Chlamtac. Approximation Algorithms Using Hierarchies of Semidefinite Programming Relaxations. *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS07)*, 2007. doi:10.1109/focs.2007.72.

13 Roee David and Uriel Feige. On the effect of randomness on planted 3-coloring models. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 77–90. ACM, 2016.

14 Reinhard Diestel. Graph Theory, volume 173 of. *Graduate texts in mathematics*, page 7, 2012.

15 Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *Journal of Computer and System Sciences*, 63(4):639–671, 2001.

16 Naveen Garg, Vijay V Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.

17 David Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM (JACM)*, 45(2):246–265, 1998.

18 Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

19 Ken-Ichi Kawarabayashi and Mikkel Thorup. Combinatorial Coloring of 3-Colorable Graphs. *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, 2012. doi:10.1109/focs.2012.16.

20 Ken-Ichi Kawarabayashi and Mikkel Thorup. Coloring 3-Colorable Graphs with Less than $n^{1/5}$ Colors. *Journal of the ACM*, 64(1):1–23, 2017. doi:10.1145/3001582.

21 Alexandra Kolla, Konstantin Makarychev, and Yury Makarychev. How to play unique games against a semi-random adversary: Study of semi-random models of unique games. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 443–452. IEEE, 2011.

22 Akash Kumar, Anand Louis, and Madhur Tulsiani. Finding Pseudorandom Colorings of Pseudorandom Graphs. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2017, December 11-15, 2017, Kanpur, India*, pages 37:1–37:12, 2017.

23 Daniel Lokshtanov, NS Narayanaswamy, Venkatesh Raman, MS Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Transactions on Algorithms (TALG)*, 11(2):15, 2014.

24 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 367–384, 2012.

25 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Algorithms for Semi-random Correlation Clustering. *CoRR*, abs/1406.5667, 2014. arXiv:1406.5667.

26 Theo McKenzie, Hermish Mehta, and Luca Trevisan. A New Algorithm for the Robust Semi-random Independent Set Problem. *CoRR*, abs/1808.03633, 2018. arXiv:1808.03633.

27 Burkhard Monien and Ewald Speckenmeyer. Ramsey numbers and an approximation algorithm for the vertex cover problem. *Acta Informatica*, 22(1):115–123, 1985.

28 NS Narayanaswamy, Venkatesh Raman, MS Ramanujan, and Saket Saurabh. LP can be a cure for parameterized problems. In *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)*, volume 14, pages 338–349. LIPIcs, 2012.

29 Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 245–254, 2008.

30 Prasad Raghavendra and David Steurer. How to Round Any CSP. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 586–594, 2009.

31 Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

32 Jacob Steinhardt. Does robustness imply tractability? A lower bound for planted clique in the semi-random model. *arXiv preprint*, 2017. arXiv:1704.05120.

33 Avi Wigderson. Improving the performance guarantee for approximate graph coloring. *Journal of the ACM*, 30(4):729–735, January 1983. doi:10.1145/2157.2158.
34 Mihalis Yannakakis. Node-and edge-deletion NP-complete problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 253–264. ACM, 1978.

A Proof of Claim 17

We begin by showing that the optimal of SDP-P3C is at most ϵn. Let $V = V_{\rm good} \cup V_{\rm bad}$ be any partition of the vertex sets into good and bad vertices such that (a) $G[V_{\rm good}]$ is 3-colorable and (b) $|V_{\rm bad}| \leq \epsilon n$. Using this partition we now construct a 2-dimensional feasible solution $(\widehat{v}, \widehat{w}, \widehat{z})$ to SDP-P3C as follows. We set the $\widehat{w}_i$ and $\widehat{z}_{ij}$ variables as

$$\widehat{w}_i = \begin{cases} 0, & \text{if } i \in V_{\rm good} \\ 1, & \text{otherwise} \end{cases} \qquad \text{and} \qquad \widehat{z}_{ij} = \begin{cases} 0, & \text{if } i,j \in V_{\rm good} \\ 1, & \text{otherwise} \end{cases}$$

Furthermore, we set $\{\widehat{v}_i\}_{i \in V_{\rm good}}$ be a vector 3-coloring of $G[V_{\rm good}]$, and for every $i \in V_{\rm bad}$ we set $\widehat{v}_i = [1 \quad 0]$. We quickly verify that the $\widehat{v}, \widehat{w}$ and the $\widehat{z}$ variables constructed as above form a feasible solution to the SDP. By construction, for every $i \in V$ we have $\widehat{w}_i \in [0,1]$ and $\|\widehat{v}_i\|^2 = 1$, and for every edge $(i,j) \in E$ we have $z_{ij} \in [0,1]$. Furthermore, for any edge (i,j) we also have

$$\widehat{z}_{ij} = \mathbb{1}\Big(\{i \in V_{\rm bad}\} \vee \{j \in V_{\rm bad}\} \Big) \leq \mathbb{1}\big(\{i \in V_{\rm bad}\} \big) + \mathbb{1}\big(\{j \in V_{\rm bad}\} \big) = \widehat{w}_i + \widehat{w}_j$$

All that remains to verify is that the variables also satisfy the approximate vector coloring constraints. We look at two cases: if $i,j \in V_{\rm good}$, then $\widehat{v}_i, \widehat{v}_j$ come from the vector 3-coloring of $G[V_{\rm good}]$ and therefore they satisfy $\langle \widehat{v}_i, \widehat{v}_j \rangle \leq -\frac{1}{2} \leq -\frac{1}{2} + \widehat{z}_{ij}$. On the other hand if $i \in V_{\rm bad}$ or $j \in V_{\rm bad}$ then by construction we have $\widehat{z}_{ij} = 1$, and therefore $\langle \widehat{v}_i, \widehat{v}_j \rangle \leq \|\widehat{v}_i\| \|\widehat{v}_j\| = 1 = -\frac{1}{2} + \frac{3}{2}\widehat{z}_{ij}$.

Therefore, we have established that $(\widehat{z}, \widehat{w}, \widehat{v})$ are a feasible solution for SDP-P3C. Since by construction $\widehat{w}_i = \mathbb{1}\{i \in V_{\rm good}\}$, and the $|V_{\rm bad}| \leq \epsilon n$, it follows that the SDP optimal $\sum_{i \in V} w_i$ is at most $\sum_{i \in V} \widehat{w}_i \leq \epsilon n$. Therefore, using Markov's inequality, we get

$$|S| = n \cdot \Pr_{i \sim V} \Big[w_i \geq \gamma/3 \Big] \leq n \cdot \frac{3 \sum_{i \in V} w_i}{n\gamma} = \frac{3\epsilon n}{\gamma}$$

B Auxiliary Lemmas

In this section we give the proofs of Lemmas 12 and 13.

B.1 Proof of Lemma 12

The proof of this lemma follows along the lines of Lemma 4.3 from [17], which says that subgraphs induced by neighborhoods of vertices in vector 3-colorable graphs are vector 2-colorable. Without loss of generality, let $N_G(i) = \{1, 2, \dots, r\}$ and let $\{v_1, v_2, \dots, v_r\}$ be the set of vectors which are a $(3, \gamma)$-vector coloring of $N_G(i)$. For every $j \in [r]$, we can write $v_j = v_j^{\|} + v_j^{\perp}$ where $v_j^{\|}$ and $v_j^{\perp}$ are the projections of v_j along v_i and $(\mathrm{span}(v_i))^{\perp}$ respectively. Finally, for every $j \in [r]$ we define $\tilde{v}_j := v_j^{\perp} / \|v_j^{\perp}\|$ to be unit vector given by the projection of v_j on the subspace $(\mathrm{span}(v_i))^{\perp}$. It can be easily verified that $\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_r$ is a $(2, 4\gamma)$-vector coloring of the graph induced on $N(v)$. To see this, fix any $j \in V$. By construction, we have $\|v_j^{\|}\| = |\langle v_i, v_j \rangle| \geq \frac{1}{2} - \gamma$, and therefore $\|v_j^{\perp}\| = \sqrt{1 - \|v_j^{\|}\|^2} \leq \sqrt{\frac{3}{4} + \gamma - \gamma^2}$. Therefore for any $j, j' \in [r]$ such that $(j, j') \in E$, using the orthonormal decomposition of v_j and $v_{j'}$ we have

$$\begin{aligned}\langle \tilde{v}_j, \tilde{v}_{j'} \rangle = \left\langle \frac{v_j^{\perp}}{\|v_j^{\perp}\|}, \frac{v_{j'}^{\perp}}{\|v_{j'}^{\perp}\|} \right\rangle &= \frac{1}{\|v_j^{\perp}\|\|v_{j'}^{\perp}\|}\Big(\langle v_j, v_{j'}\rangle - \langle v_j^{\|}, v_{j'}^{\|}\rangle\Big) \\ &= \frac{1}{\|v_j^{\perp}\|\|v_{j'}^{\perp}\|}\Big(\langle v_j, v_{j'}\rangle - \langle v_i, v_j\rangle\langle v_i, v_{j'}\rangle\Big) \\ &\leq \frac{1}{\Big(\frac{3}{4}+\gamma-\gamma^2\Big)}\Big(-1/2+\gamma-\Big(\frac{1}{2}-\gamma\Big)^2\Big) \\ &\leq -1+4\gamma\end{aligned}$$

Since the above holds for any pair of vertices $j, j' \in [r]$ which forms an edge, the claim follows.

B.2 Proof of Lemma 13

Let $v_1, v_2, \ldots, v_n$ be the $(2,\gamma)$-vector coloring of G. For contradiction, let C be an odd cycle in G of length $r \leq 1/(8\sqrt{\gamma})$. Without loss of generality, let $C = \{1, 2, \ldots, r\}$, such that for every $i \in [r]$, the pair $\{i, (i \mod r) + 1\}$ forms an edge. Let $r = 2k+1$. Now for any $i \in [r]$, we have $-1 \leq \langle v_i, v_{i+1}\rangle \leq -1+\gamma$. Since v_i, v_{i+1} are unit vectors, we have

$$\|v_i + v_{i+1}\|^2 = \|v_i\|^2 + \|v_{i+1}\|^2 + 2\langle v_i, v_{i+1}\rangle \leq 2\gamma \tag{3}$$

which implies that $\|v_i+v_{i+1}\| \leq 2\sqrt{\gamma}$ i.e, any consecutive pair of vectors are *almost anti-podal*. Then, for any $i \in [r]$ we also get that

$$\|v_i - v_{i+2}\| \leq \|v_i + v_{i+1}\| + \|v_{i+1} + v_{i+2}\| \leq 4\sqrt{\gamma} \tag{4}$$

We shall now use the above observations to arrive at a contradiction. From the upper bound on r, we have $k \leq (r-1)/2 \leq 1/(16\sqrt{\gamma})$, and hence using Eq. 4 we get that

$$\|v_1 - v_r\| \leq \sum_{j=0}^{k-1} \|v_{1+2j} - v_{1+2(j+1)}\| \leq 4k\sqrt{\gamma} < 1/4 \tag{5}$$

But on the other hand, since v_1, v_r are consecutive vertices in the cycles C, we also have $\langle v_1, v_r\rangle \leq -1+\gamma$ which implies that $\|v_1 - v_r\| \geq \sqrt{4-4\gamma} > 1$, which give us the contradiction.

C Proof of Corollary 15

Consider Algorithm IndSetColoring for coloring by iteratively finding large independent sets.

Algorithm 3 IndSetColoring.

```
Input: Graph G = (V, E)
Initialize t ← 1 and G_1 ← G;
while G_t ≠ φ do
    Let I_t be the independent set from Lemma 14 instantiated with G_t;
    Set G_{t+1} ← G_t \ I_t;
    Update t ← t + 1;
end
Output coloring I_1 ⊎ I_2 ⊎ · · · ⊎ I_t;
```

In the above algorithm, we use Lemma 14 to iteratively remove independent sets $I_1, I_2, \ldots, I_t$, where each independent set forms a color class. Let $G_t = G[V \setminus (I_1 \cup I_2 \cup \cdots I_t)]$ denote the graph on the surviving vertices after t iterations. We claim that in every $T = n^{2\epsilon}$ applications of Lemma 14 at least a constant fraction of vertices are removed, i.e., for any iteration t, we have $|\mathrm{Vert}(G_{t+T})| \leq (1 - 1/2^{1-2\epsilon})|\mathrm{Vert}(G_t)|$.

This can be shown as follows. Let $n_t = |\mathrm{Vert}(G_t)|$ denote the number of vertices in graph G_t. Then, we can assume that $|\mathrm{vert}(G_{t+T})| > n_t/2$ (otherwise we are done). Then, in T iterations the number of vertices removed can be lower bounded by

$$\sum_{j=1}^{T} |I_{j+T}| \geq \sum_{j=1}^{T} |\mathrm{Vert}(G_{t+j})|^{1-2\epsilon} \geq n^{2\epsilon}(n_t/2)^{1-2\epsilon} \geq n_t/2^{(1-2\epsilon)} \tag{6}$$

where the first inequality follows from the guarantee of Lemma 14. Therefore, in $\tilde{O}(n^{2\epsilon})$ iterations, all the vertices will be accounted for.

D Partial 2-Coloring in the Semi-random model

In this section, we give an efficient approximation algorithm for partial 2-coloring problem in the semi-random model with tighter guarantees. The following theorem formally states our guarantees for this setting.

▶ **Theorem 23** (Theorem 7 restated). *Let $\epsilon = \Omega(\log n/n)$ and $p > \sqrt{\epsilon}$. Then, there exists a polynomial time algorithm that takes as input a graph G sampled from $(1-\epsilon)$-$\mathsf{P2C}^{\mathcal{R}}(n,p)$, and with high probability, produces a set $S \subseteq V$ such that $|S| = O\left(\epsilon n p^{-2}\right)$ and the induced subgraph on the remaining vertices $G[V \setminus S]$ is 2-colorable.*

The algorithm for the above theorem (described as Algorithm 4) is quite similar to P3C-Random algorithm, but overall, the algorithm and its analysis are much simpler. We begin by describing the algorithm.

Algorithm 4 P2C-Random.

For every vertex $v \in V$, compute a greedy triangle count as follows:
for $v \in V$ **do**
    Let $G_v = G[N_G(v)]$ be the graph induced on the neighborhood of v;
    Construct a maximal matching $T(v)$ in G_v using greedy algorithm;
    Set $t(v) \leftarrow |T(v)|$;
end
Let $S \leftarrow \{v \in V : t(v) \geq 2\epsilon n\}$;
Let $G_0 = G[V \setminus S]$;
Let $G_1 \subseteq G$ be the independent set obtained using the 2-factor approximation for Vertex Cover on G;
if $|\mathrm{vert}(G_0)| \geq |\mathrm{vert}(G_1)|$ *and* G_0 *is bipartite* **then**
    Output bipartite graph G_0;
end
else
    Output independent set G_1;
end

The key difference here is that the algorithm uses triangles as forbidden subgraphs for identifying bad vertices instead of neighborhoods with short odd cycles. As before, the algorithm broadly addresses two cases depending on the size of the maximum matching in $G[V_{\text{good}}]$. Suppose the subgraph $G[V_{\text{good}}]$ contains a linear sized matching M. Then, for every bad vertex $v \in V_{\text{bad}}$, with high probability, at least one of the matching edges from M will appear in the neighborhood of v, which together will form a triangle, which can then be used to identify the bad vertices. On the other hand, if the size of maximum matching in $G[V_{\text{good}}]$ is small, then the subgraph $G[V_{\text{good}}]$ and consequently G must admit a small sized vertex cover. Therefore, using the greedy approximation algorithm for vertex cover, we can find a small sized vertex cover, whose complement must be a large independent set (which is 1-colorable).

D.1 Proof of Theorem 7

Let $M \subseteq G[V_{\text{good}}]$ be a *fixed matching of maximum size* in $G[V_{\text{good}}]$, and let $m^* := |M|$ denote the size of the maximum matching. *We point out that the matching M^* is not affected by the realization of edges between V_{good} and V_{bad} (i.e, the E_0 and E_1 edges).* As before, we break the analysis into two cases depending on whether m^* is small or large.

Case (i): $m^* \geq (8\epsilon/p^2)n$**.** This case is similar to case (i) of the proof of Theorem 6. We begin by stating and proving two lemmas which say that the greedy triangle count $t(v)$ is small for all the good vertices, and large for all the bad vertices.

▶ **Lemma 24.** *For every good vertex $v \in V_{\text{good}}$, we have $t(v) \leq \epsilon n$*

Proof. Fix a good vertex $v \in V_{\text{good}}$, and let $T(v)$ be a set of edges as constructed in the algorithm. Observe that every edge $(a, b) \in T(v)$ along with vertex v induces a triangle in G. Furthermore, since $G[V_{\text{good}}]$ is bipartite (and hence triangle free), any triangle $T \subseteq G$ must contain at least one bad vertex. Therefore, as the vertex v is good, every edge $e \in T(v)$ must contain at least one bad vertex. Finally, we observe that the edges in $T(v)$ are vertex disjoint, and there are at most ϵn bad vertices, which together implies that $t(v) = |T(v)| \leq \epsilon n$. ◀

▶ **Lemma 25.** *With probability at least $1 - e^{-O(\epsilon n)}$, for every vertex $v \in V_{\text{bad}}$, we have $t(v) \geq 2\epsilon n$.*

Proof. Let G' be the subgraph on G consisting of edges from E_0 (i.e,. the randomly sampled set of edges). Recall that $M = \{(a_i, b_i)\}_{i \in [m^*]} \subseteq G[V_{\text{good}}]$ is the fixed maximum matching in $G[V_{\text{good}}]$ of size m^*. Let $Z_i := \mathbb{1}\Big(\{a_i, b_i \in N_{G'}(v)\}\Big)$ be the indicator variable for the event that a_i, b_i are neighbors of v in the graph G'. Then,

$$\mathbb{E}_{G'}\left[\sum_{i \in [m^*]} Z_i\right] = \sum_{i \in [m^*]} \Pr_{G'}\left[\{a_i, b_i \in N_{G'}(v)\}\right] = m^* p^2 \geq 8\epsilon n \tag{7}$$

Furthermore, since the edges in M are vertex disjoint, the random variables $Z_1, \ldots, Z_{m^*}$ are independent and identical. Therefore using Chernoff bound we get

$$\Pr_{G'}\left[\sum_{i \in [m^*]} Z_i \leq 4\epsilon n\right] \leq \Pr_{G'}\left[\sum_{i \in [m^*]} Z_i \leq \frac{1}{2}\mathbb{E}\sum_{i \in [m^*]} Z_i\right] \leq e^{-O(\epsilon n)} \tag{8}$$

Let $M_v = \{(a_i, b_i) : i \in [m^*], Z_i = 1\}$ be the set of matching edges from M^* appearing in the neighborhood of v in the graph G'. Furthermore, let $\tilde{M}_v$ be a maximum matching in the subgraph $G_V := G[N_G(v)]$ induced on the neighborhood of v (which contains both E_0 and E_1

edges). Then, by definition we have $|\tilde{M}_v| \geq |M_v|$. On the other hand, by construction, the set $T(v)$ is a maximal matching in the induced subgraph G_v. Since a maximal matching is a 2-approximation to the maximum matching, it follows that $|T(v)| \geq |\tilde{M}_v|/2 \geq |M_v|/2 \geq 2\epsilon n$.

Therefore, for a fixed bad vertex $v \in V_{\text{bad}}$, with probability at least $1 - e^{-O(\epsilon n)}$, we have $t(v) \geq 2\epsilon n$. The claim now follows by taking a union bound over all vertices $v \in V_{\text{bad}}$. ◀

Therefore, combining Lemmas 24 and 25, we know that with probability at least $1 - e^{-O(\epsilon n)}$, we have $t(v) \leq \epsilon n$ if and only if $v \in V_{\text{good}}$. Conditioned on this event, the set S must exactly be the set of bad vertices, in which case $G[V \setminus S] = G[V_{\text{good}}]$ is bipartite.

Case (ii): $m^* \leq (8\epsilon/p^2)n$. Since the size of maximum matching in $G[V_{\text{good}}]$ is at most $(8\epsilon/p^2)n$, and $G[V_{\text{good}}]$ is bipartite, by König's theorem (Theorem 2.1.1 [14]), it follows that the minimum vertex cover of $G[V_{\text{good}}]$ has size at most $(8\epsilon/p^2)n$. Then G has a vertex cover of size at most $(8\epsilon/p^2)n + \epsilon n \leq (10\epsilon/p^2)n$. Therefore, the greedy approximation algorithm for vertex cover returns a vertex cover S' of size at most $(20\epsilon/p^2)n$, and consequently, $V \setminus S'$ will be an independent set of size at least $(1 - (20\epsilon/p^2))n$.

Putting things together. In case (i), the algorithm throws away at most ϵn vertices and returns a 2-colorable graph, with probability at least $1 - e^{-O(\epsilon n)}$. In case (ii), the algorithm throws away at most $O(\epsilon/p^2)n$ vertices, and returns an indpendent set. Combining the two cases gives us the guarantees for Theorem 7.

E Maximal and Maximum Short Odd Cycle sets

▶ **Proposition 26.** *For any graph $G := (V, E)$, and parameter $\delta \in (0,1)$ the following holds. Let $\mathcal{C}$ be a maximal set of vertex disjoint odd cycles in G of length at most $1/\delta$, and let $\tilde{\mathcal{C}}$ be a set of largest cardinality of vertex disjoint odd cycles in G of length at most $1/\delta$. Then $|\mathcal{C}| \geq \delta|\tilde{\mathcal{C}}|$.*

Proof. Since $\mathcal{C}$ is a maximal set of vertex disjoint odd cycles of length at most $1/\delta$, for every odd cycle $\tilde{C} \in \tilde{\mathcal{C}}$, there exists an odd cycle $C \in \mathcal{C}$ such that $C \cap \tilde{C} \neq \emptyset$ i.e,. $\tilde{C}$ is hit by C. Now we observe that (i) the cycles in $\mathcal{C}$ are vertex disjoint and (ii) each cycle $C \in \mathcal{C}$ has size at most $1/\delta$. Hence, it follows that any cycle $C \in \mathcal{C}$ hits at most $1/\delta$ cycles in $\tilde{\mathcal{C}}$. Since every cycle in $\tilde{\mathcal{C}}$ is hit by some cycle in $\mathcal{C}$, we must have $|\mathcal{C}| \geq \frac{|\tilde{\mathcal{C}}|}{1/\delta} = \delta|\tilde{\mathcal{C}}|$. ◀

F Identifying the set of Good Vertices is NP-hard

▶ **Fact 27.** *For all $k \in \mathbb{N}$, given a graph α-partially k-colorable graph $G = (V, E)$ it is NP-Hard to identify a set $V_{\text{good}} \subset V$ of size at least αn such that $G[V_{\text{good}}]$ is k-colorable*

Proof. For $\alpha = 1 - 1/2n$, this is exactly the k-Coloring problem which is NP-Hard [18]. ◀