



# On Finding Randomly Planted Cliques in Arbitrary Graphs

Francesco Agrimonti ✉ 🏠 

Parma, Italy

Marco Bressan ✉ 🏠 

Department of Computer Science, University of Milan, Italy

Tommaso d’Orsi ✉ 🏠 

Department of Computing Sciences, Bocconi University, Milan, Italy

---

## Abstract

We study a planted clique model introduced by Feige [18] where a complete graph of size  $c \cdot n$  is planted uniformly at random in an arbitrary  $n$ -vertex graph. We give a simple deterministic algorithm that, in almost linear time, recovers a clique of size  $(c/3)^{O(1/c)} \cdot n$  as long as the original graph has maximum degree at most  $(1-p)n$  for some fixed  $p > 0$ . The proof hinges on showing that the degrees of the final graph are correlated with the planted clique, in a way similar to (but more intricate than) the classical  $G(n, 1/2) + K_{\sqrt{n}}$  planted clique model. Our algorithm suggests a separation from the worst-case model, where, assuming the Unique Games Conjecture, no polynomial algorithm can find cliques of size  $\Omega(n)$  for every fixed  $c > 0$ , even if the input graph has maximum degree  $(1-p)n$ . Our techniques extend beyond the planted clique model. For example, when the planted graph is a balanced biclique, we recover a balanced biclique of size larger than the best guarantees known for the worst case.

**2012 ACM Subject Classification** Theory of computation → Computational complexity and cryptography; Mathematics of computing → Discrete mathematics; Mathematics of computing → Approximation algorithms; Mathematics of computing → Graph algorithms

**Keywords and phrases** Computational Complexity, Planted Clique, Semi-random, Unique Games Conjecture, Approximation Algorithms

**Digital Object Identifier** 10.4230/LIPIcs.APPROX/RANDOM.2025.11

**Category** APPROX

**Related Version** *Previous Version:* <https://arxiv.org/abs/2505.06725>

## 1 Introduction

Finding large cliques in a graph is a notoriously hard problem. Its decision version was among the first problems shown to be NP-complete [31]. In fact, it turns out that for any  $\varepsilon > 0$  it is NP-hard to find a clique of size  $n^\varepsilon$  even in graphs containing cliques of size  $n^{1-\varepsilon}$  [26, 39, 32].

A large body of work [14, 25, 2, 30, 17, 9, 29] focused on designing polynomial time algorithms to find large cliques given an  $n$ -vertex graph containing a clique of size  $cn$ . When  $c < 1/\log n$ , the best algorithm known only returns a clique of size  $\tilde{O}(\log(n)^3)$  [17]. For larger values of  $c$  it is possible to find a clique of size  $O(cn)^{O(c)}$ , which is of order  $n^{\Omega(1)}$  when the largest clique in the graph contains a constant fraction of the vertices [9, 2]. The current algorithmic landscape further suggests a phase-transition phenomenon around  $c = \frac{1}{2}$ . For sufficiently small  $\varepsilon > 0$  and  $c = \frac{1}{2} - \varepsilon$  there exists an algorithm finding a clique of size  $n^{1-O(\varepsilon)}$  [30]. Instead, when  $c = \frac{1}{2} + \varepsilon$ , one can efficiently find a complete graph of size  $2\varepsilon n$  via a reduction to the classical 2-approximation algorithm for vertex cover. Finally, finding a clique of size  $\Omega(\varepsilon n)$  for  $c = \frac{1}{2} - \varepsilon$  was shown to be UGC-hard in [33, 7].



© Francesco Agrimonti, Marco Bressan, and Tommaso d’Orsi;  
licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025).

Editors: Alina Ene and Eshan Chattopadhyay; Article No. 11; pp. 11:1–11:18



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

■ **Table 1** Performance of state-of-the-art efficient algorithms for CLIQUE when a clique of size  $cn$  exists in the graph (note that  $c$  can depend on  $n$ ).

Regime	Output clique	References
$c \geq \frac{1}{2} + \varepsilon$	$2\epsilon n$	[14]
$c \geq \Omega(1)$	$(cn)^{\Omega(c)}$	[2]
$c \geq 1/\log n$	$\Omega\left(\frac{n^c}{c}\right)$	[9, 25]
any $c > 0$	$\Omega\left(\frac{\log^3(cn)}{\log^2 \log(cn)}\right)$	[17]

Given the grim worst-case picture, a substantial body of work has focused on designing algorithms that perform well under structural or distributional assumptions on the input graph. One research direction has investigated CLIQUE and related problems on graphs satisfying expansion or colorability properties [4, 15, 34, 6]. Another line of work has explored planted average case models [31, 27, 35, 3, 20, 19, 13, 22]. In the planted clique model, the input graph is generated by sampling a graph from the Erdős-Rényi distribution  $\text{ER}(n, \frac{1}{2})$  and then embedding a clique of size  $cn$  by fully connecting a randomly chosen subset of vertices. Here, basic semidefinite programming relaxations [20, 21], as well a simple rounding of the second smallest eigenvector of the Laplacian [3], are known to efficiently recover the planted clique whenever  $c \geq 1/\sqrt{n}$ . Lower bounds against restricted computational models further provide evidence that these algorithmic guarantees may be tight [23, 8].

In an effort to bridge the worst-case settings and the average-case settings, Feige and Kilian [19] introduced a semi-random model in which the above planted clique instance is further perturbed by: (i) arbitrarily removing edges between the planted clique  $K$  and the remainder of the graph  $G \setminus K$ , and (ii) arbitrarily modifying the subgraph induced by  $G \setminus K$ . The randomness of this model lies in the cut  $(K, G \setminus K)$  which separates the clique from the rest of the graph. A flurry of works [12, 37, 10] led to an algorithm that, leveraging the randomness of this cut, can recover a planted clique of size  $n^{\frac{1}{2}+\varepsilon}$  in time  $n^{O(1/\varepsilon)}$ . This picture suggests that, from a computational perspective, this semi-random model may be closer to the planted average case model than to worst-case graphs. (Information theoretically the semi-random model differs drastically from the planted clique model [38].)

To better understand the role of randomness in the CLIQUE problem, Feige [18] proposed another model in which a clique is randomly planted in an *arbitrary* graph, and asked what approximation guarantees are efficiently achievable in this setting. In comparison to the aforementioned semi-random case, here the randomness only affects the location of the clique but not the topology of the rest of the graph.

Investigating this model is the main focus of this paper. We provide a first positive answer to Feige’s question, showing that a surprisingly simple deterministic algorithm achieves significantly stronger guarantees than those known for the worst-case settings, for a wide range of parameters. Our results suggest that this model may sit in between the average case and the worst case regimes.

## 1.1 Results

To present our contributions we first formally state our random planting model. In fact, as our results extend beyond CLIQUE, the model we state is a generalization of the one in [18].

► **Definition 1** (Random planting in arbitrary graphs). *Let  $G$  and  $H$  be graphs with  $|V(H)| \leq |V(G)|$ .  $\mathcal{G}(G, H)$  describes the following distribution over graphs:*

1. *Sample a random uniform injective mapping  $\phi : V(H) \rightarrow V(G)$ .*
2. *Return  $\hat{G}$  with  $V(\hat{G}) = V(G)$  and  $E(\hat{G}) = E(G) \cup \{\{\phi(u), \phi(u')\} : \{u, u'\} \in E(H)\}$ .*

When  $H$  is the  $cn$ -sized complete graph  $K_{cn}$ , Definition 1 corresponds to the planted clique model of [18]. In this specific setting we obtain the following result.

► **Theorem 2** (Simplified version). *There exists a deterministic algorithm  $\mathcal{A}$  with the following guarantees. For every  $c \in (0, 1)$  and every  $n$ -vertex graph  $G$ , if  $\hat{G} \sim \mathcal{G}(G, K_{cn})$  then  $\mathcal{A}(\hat{G})$  with probability at least  $1 - \frac{1}{n^{\frac{1}{2}}}$  returns a clique of size at least:*

$$\frac{n}{5} \cdot \left(\frac{c}{3}\right)^{\frac{4}{c} \log \frac{2}{p}}$$

where  $p = 1 - \frac{\Delta}{n}$  and  $\Delta$  is the maximum degree of  $G$ . Moreover  $\mathcal{A}$  runs in time  $\tilde{O}(\|\hat{G}\|)$ .

To appreciate the guarantees of Theorem 2 consider the setting  $p = \Omega(1)$ , so that  $\Delta = (1-p)n$  is bounded away from  $n$ . In this case, for every fixed  $c > 0$  the algorithm of Theorem 2 finds with high probability a clique of size  $\Omega(n)$ . In the worst case, however, this is not possible unless the Unique Games Conjecture fails. More precisely, assuming UGC, no polynomial-time algorithm can find a clique of size  $\varepsilon \cdot n$  even when one of size  $(\frac{1}{2} - \varepsilon) \cdot n$  exists [33, 7]; indeed, state-of-the-art algorithms [2, 9, 25] are only known to return cliques of size  $n^{O(c)}$ . By adding  $n^{\frac{p}{1-p}}$  isolated vertices to the graph, it also follows that under UGC one cannot efficiently find a clique of size  $\varepsilon \cdot n$  even when one of size  $(\frac{1-p}{2} - \varepsilon) \cdot n$  exists and the input graph has degree  $\Delta \leq (1-p)n$ , as in the statement of Theorem 2. Thus, unless UGC fails, we cannot expect Theorem 2 to hold in the worst case. We remark that Theorem 2 also guarantees to recover cliques of size  $n^{\Omega(1)}$  for  $c \geq \Omega\left(\frac{\log \log n}{\log n}\right)$ , a regime in which worst-case algorithms are only known to find cliques of size  $\text{poly} \log(n)$ . Finally, as one can expect, the failure probability can be actually made smaller than  $n^{-a}$  for any desired  $a \geq 1$ ; see the full formal version of Theorem 2 in Section 5.

Note that the performance of our algorithm deteriorates as  $p$  approaches 0; that is, as the maximum degree approaches  $n$ . While it remains an open question whether some assumption on the degree is inherently necessary, we provide some preliminary evidence in Theorem 5, see Section 2 and Section 7.

Our results extend beyond the case where the planted graph is a complete graph. To illustrate this, we also consider the BALANCED BICLIQUE problem, where the goal is to find a largest complete balanced bipartite subgraph. The BALANCED BICLIQUE problem has a long history [24, 28, 1] and a strong connection to CLIQUE [11]. Assuming the Small Set Expansion Hypothesis, there is no polynomial-time algorithm that can find a balanced biclique within a factor  $n^{1-\varepsilon}$  of the optimum for every  $\varepsilon > 0$ , unless  $\text{NP} \subseteq \text{BPP}$  [36]. Remarkably, in the worst case, the bicliques that existing algorithms are known to return are significantly smaller than the complete graphs found in the context of CLIQUE. In fact, the best algorithm known [11] works through a reduction to CLIQUE which constructs an instance with a complete graph of size  $O(c^2 \cdot n)$  from a BALANCED BICLIQUE instance with a biclique of size  $c \cdot n$ .

In comparison, under Definition 1, we obtain the following guarantees.

► **Theorem 3** (Simplified version). *There exists a deterministic polynomial-time algorithm  $\mathcal{A}$  with the following guarantees. For every  $c \in (0, 1)$  and every  $n$ -vertex graph  $G$ , if  $\hat{G} \sim \mathcal{G}(G, K_{\frac{cn}{2}, \frac{cn}{2}})$  then  $\mathcal{A}(\hat{G})$  with probability at least  $1 - \frac{1}{n^2}$  returns a balanced biclique of size at least:*

$$\frac{c}{48} \cdot 2\sqrt{\frac{c \log n}{2}}.$$

Moreover  $\mathcal{A}(\hat{G})$  runs in time  $\tilde{O}(\|\hat{G}\|)$ .

The main point of Theorem 3 is again the difference with the worst case bounds. In the worst case, existing algorithms are known to find a biclique of size  $(\log n)^{\omega(1)}$  only if there exists one of size  $c \cdot n \geq \omega\left(\frac{\log \log n}{\sqrt{\log n}}\right) \cdot n$  in the input graph. In contrast, Theorem 3 states that in typical instances from  $\mathcal{G}(G, K_{\frac{cn}{2}, \frac{cn}{2}})$  we can efficiently find a biclique of size  $(\log n)^{\omega(1)}$  whenever there exists one of size  $c \cdot n \geq \omega\left(\frac{(\log \log n)^2}{\log n}\right) \cdot n$ ; that is, for value of  $c$  up to  $\frac{\log \log n}{\sqrt{\log n}}$  times smaller than for the worst case. Furthermore, unlike the bounds of Theorem 2, the ones of Theorem 3 are insensitive to the structure of  $G$ , and in particular to its maximum degree.

## 2 Techniques

This section gives an intuitive description of our techniques, using the planted clique problem as a running example. Let  $G$  be an arbitrary  $n$ -vertex graph, and let  $\hat{G} \sim \mathcal{G}(G, K_{cn})$ . For simplicity, we suppose that  $c > 0$  and  $p > 0$  are fixed constants, and that  $G$  has maximum degree  $\Delta \leq (1 - p)n$ . Let  $\phi : V(K_{cn}) \rightarrow V(G)$  be the injective mapping sampled in the process of constructing  $\hat{G}$ . Because we have almost no knowledge of the global structure of  $G$ , it appears difficult to recover the planted clique via the topology of  $\hat{G}$  without running into any of the barriers observed in worst-case instances.

On the other hand, since the clique is planted randomly, we can expect certain basic statistics to change in a convenient and somewhat predictable way between  $G$  and  $\hat{G}$ . Our approach focuses on perhaps the simplest such statistic – the degree profile – guided by the intuition that vertices with higher degree in  $\hat{G}$  are more likely to belong to the planted clique than those with lower degree. For notational convenience we use the degree in the complement graph, which we call *slack*. To be precise, for a vertex  $v \in V(G)$ , the slack of  $v$  in  $G$  is  $s_v = (n - 1) - d_v$  where  $d_v$  is the degree of  $v$  in  $G$ . In the same way we define the slack of  $v$  in  $\hat{G}$  as  $\hat{s}_v = (n - 1) - \hat{d}_v$ , where  $\hat{d}_v$  is the degree of  $v$  in  $\hat{G}$ .

To formalize the intuition above, suppose  $G$  contains a subset  $V' \subseteq V$  such that (i) the vertices of  $V'$  have approximately the same slack, in the sense that if  $s := \min_{v \in V'} s_v$ , then any  $v \in V'$  satisfies

$$s_v \left(1 - \frac{c}{2}\right) < s,$$

and (ii) the set  $V_{<s}(G)$  of vertices in  $G$  with slack smaller than  $s$  has size at most, say,  $\frac{c}{10} \cdot |V'|$ . Because the map  $\phi$  is chosen uniformly at random, we expect a  $c$  fraction of  $V'$  will be in the image of  $\phi$ . Furthermore, every  $v \in V'$  in the image of  $\phi$  acquires  $cs_v$  new neighbors in expectation, which by (i) gives:

$$\mathbb{E}_\phi [\hat{s}_v] \leq s_v \cdot (1 - c) < s.$$

In fact, as long as  $|V'|$  and  $s$  are large enough (roughly  $\Omega(c^{-1} \log n)$ ), by standard concentration bounds at least  $\frac{c}{2}|V'|$  vertices of  $V'$  will be in the image of  $\phi$ , and all those vertices  $v$  will satisfy  $\hat{s}_v < s$ . Under these circumstances, by (ii) we conclude that, in  $\hat{G}$ , the set  $|V_{<s}(\hat{G})|$

of vertices having slack smaller than  $s$  has size at least  $\frac{c}{10}|V'|$ , and moreover a fraction at least  $\frac{1/2}{1/2+1/10} > 0.8$  of those vertices form a clique. We can then immediately recover a clique of size  $\Omega(c|V'|)$  via the standard reduction to vertex cover applied to the subgraph of  $\hat{G}$  induced by  $|V_{<s}(\hat{G})|$ .

The above discussion suggests our intuition is correct whenever a sufficiently large set satisfying (i) and (ii) exists.<sup>1</sup> While arbitrary graphs may not contain such a set, it turns out that the only obstacle towards the existence of a linear size set  $V'$  is the presence of a large set of vertices of slack strictly smaller than  $s$ . Choosing  $V'$  so that  $s \leq p \cdot n$  we deduce that such a  $V'$  must exist.

► **Remark 4.** The above reasoning works beyond the parameters regime of our example and, in fact, does not require the planted graph to be a clique. In the context of BALANCED BICLIQUE the existence of a large set with slack  $< s$  makes the problem easier. Therefore, we are able to drop the assumption on the maximum degree in  $G$ .

We complement the intuition above with a lower bound on the performance of degree profiling. Essentially this states that, if we have no guarantees on the maximum degree of  $G$ , then the degree profile of  $\hat{G} \sim \mathcal{G}(G, K_{cn})$  is uncorrelated with  $K_{cn}$ . Formally:

► **Theorem 5.** *For every  $c \in (0, \frac{1}{2})$  and  $n \geq 3$  there exists an  $n$ -vertex graph  $G$  such that  $\hat{G} \sim \mathcal{G}(G, K_{cn})$  satisfies what follows with probability at least  $1 - \frac{1}{n}$ . For every ordering  $v_1, \dots, v_n$  of the vertices of  $\hat{G}$  by nonincreasing degree, and for every  $j \in [n]$ , the largest clique in the induced subgraph  $\hat{G}[\{v_1, \dots, v_j\}]$  has size at most*

$$O\left(\frac{\sqrt{n \ln n}}{c} + cj\right). \quad (1)$$

To appreciate Theorem 5 let  $t = \Omega(c^{-2}\sqrt{n \ln n})$ . Then the theorem says that, if one takes the first  $t$  vertices of  $\hat{G}$  in order of degree, the largest clique therein has size  $O(ct)$  with high probability. In other words, for all  $t$  not too small compared to  $n$ , the  $t$  vertices of highest degree have roughly the same clique density of the entire graph. This suggests that, using degree statistics alone, one has little hope to find cliques larger than  $\tilde{O}(\sqrt{n})$  even for constant  $c$ . Note that there is no contradiction with the upper bounds of Theorem 2: those bounds become trivial for large  $\Delta$ , and the graph behind the proof of Theorem 5 has indeed a large  $\Delta$ .

### 3 Preliminaries

Let  $G$  be a graph. We let  $V(G)$  be its set of vertices and  $E(G)$  its set of edges. For  $V' \subseteq V(G)$  we let  $G[V']$  be the subgraph induced by  $V'$ . We often use  $n = |V(G)|$ . We let  $\|G\| = |V(G)| + |E(G)|$ . For  $v \in V(G)$ , let  $d_v$  be its degree and  $s_v := n - 1 - d_v$  its slack. For  $V' \subseteq V(G)$ , let  $s_{V'} := \min_{v \in V'} s_v$ . We write  $V_{<s}(G) := \{v \in V(G) \mid s_v < s\}$ . We do not specify the graph when the context is clear and we define  $V_{<s}(\hat{G})$  as  $\hat{V}_{<s}$ . We let  $K_n$  be the complete graph of size  $n$  and  $K_{a,b}$  be the biclique with sides of size  $a$  and  $b$ . We let  $[n] := \{1, \dots, n\}$ ,  $\log = \log_2$  and  $\ln = \log_e$ .

The computational model is the standard RAM model with words of logarithmic size. Unless otherwise stated, all our graphs are given as adjacency list. By performing a  $O(n)$  preprocessing we henceforth assume the adjacency lists are sorted, so that one can perform binary search and check the existence of any given edge in time  $O(\log n)$ .

<sup>1</sup> We remark that our algorithm does not need to find this set.

## 11:6 On Finding Randomly Planted Cliques in Arbitrary Graphs

The following theorem says that, for every fixed  $c > \frac{1}{2}$ , one can efficiently find a clique of size  $\Omega(n)$  in an  $n$ -vertex graph that contains one of size  $cn$ .

► **Theorem 6.** *There exists an algorithm, DENSECLIQUEFINDER, with the following guarantees. For every  $\varepsilon > 0$ , if  $\mathcal{A}$  is given in input an  $n$ -vertex graph  $G = (V, E)$  that contains a clique of size  $(\frac{1}{2} + \varepsilon)n$ , then  $\mathcal{A}$  finds in deterministic  $\tilde{O}(n^2)$ -time a clique of size  $2\varepsilon n$ .*

The proof is folklore – take the complement of  $G$ , find a 2-approximation of the smallest vertex cover through a maximal matching, and return its complement. See also [14].

### 4 Slackness profile and densification

In this section we prove our structural results on the degree and slackness profile of graphs from Definition 1. We start with a definition.

► **Definition 7** (Bulging set). *Let  $G = (V, E)$  be a graph and  $\alpha, \beta > 0$ . A set  $U \subseteq V$  is  $(\alpha, \beta)$ -bulging if:*

1.  $s_v < \frac{s_U}{1-\beta}$  for all  $v \in U$ .
2.  $|V_{< s_U}| < \frac{1}{\alpha}|U|$ .

The next statement characterizes the existence of  $(\alpha, \beta)$ -bulging sets in *any* graph based on the value of  $\alpha$  and  $\beta$  and the slackness of its vertices.

► **Lemma 8.** *Let  $G = (V, E)$  be an  $n$ -vertex graph. Then for every  $\beta \in (0, 1/2)$ ,  $\alpha \geq 2$ , and  $s \in \mathbb{R}_{>0}$  at least one of the following facts holds:*

- (i)  $|V_{< s}| \geq \frac{n}{\alpha^{2+\frac{1}{\beta} \log \frac{n}{s}}}$ .
- (ii)  $G$  contains an  $(\alpha, \beta)$ -bulging set  $U$  such that  $|U| \geq \frac{n}{\alpha^{1+\frac{1}{\beta} \log \frac{n}{s}}}$  and  $s_U \geq s$ .

**Proof.** Let  $\eta = \frac{\beta}{1-\beta} > 0$  and  $h = \lceil \log_{1+\eta} \frac{n}{s} \rceil$ . We define a partition of  $V$  into  $h+1$  possibly empty sets, as follows:

$$V_0 := V_{< s} \tag{2}$$

$$V_j := V_{< s(1+\eta)^j} \setminus V_{< s(1+\eta)^{j-1}} = \{v \in V \mid s(1+\eta)^{j-1} \leq s_v < s(1+\eta)^j\} \quad j \in [h] \tag{3}$$

It is immediate to see that this is indeed a partition of  $V$ , since  $0 \leq s_v < n$  for every  $v \in V$ . We prove the statement by contradiction. Suppose (ii) does not hold. Then it must be that for  $j \geq 1$ :

$$|V_j| < \frac{n}{\alpha^{2+\frac{1}{\beta} \log \frac{n}{s}}} \cdot \alpha^j, \tag{4}$$

Indeed, if this was not the case, then one can check that for the smallest  $j \in [h]$  violating Equation (4) the set  $V_j$  would be  $(\alpha, \beta)$ -bulging, and moreover every vertex in  $V_j$  would have slack at least  $s$  (since  $j \geq 1$ ). Suppose further (i) is not verified. Then as the sets  $V_j$  form a partition of  $V$ , and as  $\alpha \geq 2$ ,

$$n = \sum_{j=0}^h |V_j| < \frac{n}{\alpha^{2+\frac{1}{\beta} \log \frac{n}{s}}} \cdot \sum_{j=0}^h \alpha^j < \frac{n}{\alpha^{2+\frac{1}{\beta} \log \frac{n}{s}}} \cdot \alpha^{h+1} \tag{5}$$

Now observe that

$$h \leq 1 + \frac{\log \frac{n}{s}}{\log(1+\eta)} \leq 1 + \frac{1+\eta}{\eta} \log \frac{n}{s} = 1 + \frac{1}{\beta} \log \frac{n}{s} \tag{6}$$

where we used the facts that  $\log(1+x) \geq \frac{x}{1+x}$  for all  $x \geq 0$ , and that  $\frac{\eta}{1+\eta} = \beta$ . Substituting this bound in Equation (5) yields the absurd  $n < n$ . Thus at least one among (i) and (ii) holds.  $\blacktriangleleft$

Our next key result states that the subgraph of  $\hat{G}$  induced by the set of vertices  $v$  with slack  $\hat{s}_v < s_U$ , where  $U \subseteq V$  is the bulging set that exists in  $G$  for Lemma 8, will contain a large number of vertices of  $H$  with high probability.

**► Lemma 9 (Densification Lemma).** *Let  $G$  be an  $n$ -vertex graph,  $\alpha \geq 2$  and  $c \in (0, 1)$ . Let  $H$  be a regular graph with  $|V(H)| \leq n$  and minimum degree at least  $cn \geq 10$ . Let  $U$  be an  $(\alpha, \frac{c}{2})$ -bulging set of  $G$  with  $\min\{s_U, |U|\} \geq \frac{12+29a \ln n}{c}$  for some  $a \geq 1$ . Finally, let  $\hat{G} \sim \mathcal{G}(G, H)$ , and let  $\hat{H}$  be the image of  $H$  in  $\hat{G}$ . Then, with probability at least  $1 - n^{-a}$  the set  $\hat{V}_{< s_U}$  satisfies:*

- (i)  $|\hat{V}_{< s_U} \cap \hat{H}| > \frac{c}{2} \cdot |U|$ .
- (ii)  $|\hat{V}_{< s_U} \cap \hat{H}| > \frac{c\alpha}{2} \cdot |\hat{V}_{< s_U} \setminus \hat{H}|$ .

**Proof.** First, we claim that  $\hat{H} \cap U \subseteq \hat{V}_{< s_U}$  with high probability. Consider any  $v \in U$ , and note that  $v \notin \hat{V}_{< s_U}$  means  $\hat{s}_v \geq s_U$ . Now, if  $v \in \hat{H}$ , then  $\hat{s}_v = s_v - X$ , where  $X = \sum_{i=1}^{s_v} X_i$  is the sum of non-positively correlated Bernoulli random variables of parameter  $c' = c - \frac{1}{n}$ . The event  $\hat{s}_v \geq s_U$  is therefore the event  $X \leq s_v - s_U$ ; since  $s_v - s_U \leq \frac{c}{2}s_v$ , as  $v \in U$  and  $U$  is  $(\alpha, \frac{c}{2})$ -bulging, this implies the event  $X \leq \frac{c}{2}s_v$ . Now, as  $cn \geq 10$ , then  $c' \geq \frac{9}{10}c$ , and  $\frac{c}{2}s_v \leq \frac{5}{9}c's_v$ . Since moreover  $\mathbb{E}X = c's_v$ , we conclude that  $\hat{s}_v \geq s_U$  implies the event  $X \leq (1 - 4/9)\mathbb{E}X$ . We then use Lemma 16 with  $\varepsilon = 4/9$ . To this end note that:

$$\mathbb{E}X = c's_v \geq \frac{9}{10}cs_U > 10 + 26a \ln n \geq 13(\ln 2 + (1+a) \ln n) = 13 \ln(2n^{a+1}) \quad (7)$$

Therefore:

$$\Pr[v \notin \hat{V}_{< s_U}] = \Pr[\hat{s}_v \geq s_U] \leq \Pr[X \leq (1 - 4/9)\mathbb{E}X] \leq e^{-\frac{(4/9)^2}{2+4/9}\mathbb{E}X} < e^{-\frac{\mathbb{E}X}{13}} < \frac{1}{2n^{a+1}} \quad (8)$$

By a union bound over all  $v \in U$  we conclude that  $\Pr[\hat{H} \cap U \not\subseteq \hat{V}_{< s_U}] \leq \frac{1}{2}n^{-a}$ .

Next, consider  $|\hat{H} \cap U|$ . Note that  $|\hat{H} \cap U| = X$  where again  $X = \sum_{i=1}^{|U|} X_i$  is the sum of non-positively correlated Bernoulli random variables of parameter  $c$ . Using again Lemma 16 with  $\varepsilon = 4/9$ , and noting as done above that  $\mathbb{E}X = c|U| > 13 \ln(2n^{a+1})$ , we obtain:

$$\Pr[|\hat{H} \cap U| \leq \frac{c}{2}|U|] = \Pr[X \leq (1 - 4/9)\mathbb{E}X] < \frac{1}{2n^{a+1}} < \frac{1}{2}n^{-a} \quad (9)$$

Finally, let  $S := \hat{V}_{< s_U}$ . The bounds above show that, with probability at least  $1 - n^{-a}$ , we have  $U \cap \hat{H} \subseteq S \cap \hat{H}$  and  $|U \cap \hat{H}| > \frac{c}{2}|U|$ , which implies  $|S \cap \hat{H}| > \frac{c}{2}|U|$ , that is, (i).

Moreover  $S \setminus \hat{H} \subseteq \hat{V}_{< s_U}$ , which implies  $|S \setminus \hat{H}| < \frac{1}{\alpha}|U|$  as  $U$  is  $(\alpha, \frac{c}{2})$ -bulging. Together with (i) we conclude that:

$$\frac{|S \cap \hat{H}|}{|S \setminus \hat{H}|} > \frac{c\alpha}{2} \quad (10)$$

which proves (ii).  $\blacktriangleleft$



## 5 Application to CLIQUE

In this section we prove Theorem 2, which we restate in a fully formal way and with more general probabilistic guarantees.

► **Theorem 10.** *There exists a deterministic algorithm  $\mathcal{A}$  with the following guarantees. Fix any  $a \geq 1$ . Let  $c := c(n) \in \omega\left(\frac{1}{\log n}\right)$ , and define:*

$$K(n, c, p) := \frac{n}{5} \cdot \left(\frac{c}{3}\right)^{2 + \frac{2}{c} \log \frac{2}{p}}.$$

*Then for every  $n$  large enough and every  $n$ -vertex graph  $G$  what follows holds. Letting  $p = 1 - \frac{\Delta}{n}$  where  $\Delta$  is the maximum degree of  $G$ , if  $K(n, c, p) \geq 1 + 2a \ln n$ , then  $\mathcal{A}$  on input  $\hat{G} \sim \mathcal{G}(G, K_{cn})$  returns a clique of  $\hat{G}$  whose size is at least  $K(n, c, p)$  with probability at least  $1 - n^{-a}$ . Moreover  $\mathcal{A}(\hat{G})$  runs in time  $\tilde{O}(\|\hat{G}\|)$  for every input graph  $\hat{G}$ .*

**Proof.** We start by proving that Algorithm 1 runs in time  $\tilde{O}(n^3)$  and guarantees a clique of size  $\frac{7}{5}K(n, c, p)$  with the prescribed probability. We then show how to lower the running time to  $\tilde{O}(\|\hat{G}\|)$  while reducing the clique size to  $K(n, c, p)$ .

■ **Algorithm 1** CLIQUEFINDER( $\hat{G}$ ).

---

```

1:  $S \leftarrow \emptyset$ 
2:  $v_1, \dots, v_n \leftarrow$  vertices of  $\hat{G}$  in nonincreasing order of degree
3: for  $1 \leq i \leq n$ : do
4:    $T \leftarrow \text{DENSECLIQUEFINDER}\left(\hat{G}[\{v_1, \dots, v_i\}]\right)$ 
5:   if  $|T| > |S|$ : then
6:      $S \leftarrow T$ 
7:   end if
8: end for
9: return  $S$ 

```

---

The inequalities we are going to claim assume  $n$  is indeed sufficiently large (formally, larger than some  $n_0$  that may depend on  $a$ ). To begin with, observe that if  $c \leq \frac{1}{\log n}$  or  $p \leq n^{-1/2}$  then  $K(n, c, p) \leq 1$  and therefore our algorithm certainly satisfies the bound of Theorem 10. Indeed, if  $c \leq \frac{1}{\log n}$  then the second multiplicative term in the expression of  $K(n, c, p)$  satisfies:

$$\left(\frac{c}{3}\right)^{2 + \frac{2}{c} \log \frac{2}{p}} \leq \left(\frac{1}{3 \log n}\right)^{2 + 2 \log n} < 9^{-\log n} \leq \frac{5}{n} \quad (11)$$

If instead  $p \leq n^{-1/2}$  then the same term satisfies:

$$\left(\frac{c}{3}\right)^{2 + \frac{2}{c} \log \frac{2}{p}} \leq \left(\frac{1}{3}\right)^{2 \log \sqrt{n}} = 3^{-\log n} \leq \frac{5}{n} \quad (12)$$

Thus we may assume  $cp \geq \frac{1}{\sqrt{n} \log n}$ , and therefore:

$$cp \geq \frac{13 + 29a \ln n}{n} \geq \frac{12 + 29a \ln n}{n} + \frac{c}{n} \quad (13)$$



Now let  $s = pn - 1$ . Then  $s = (n - 1) - \Delta$ ; hence all vertices of  $G$  have slack at least  $s$ , and therefore  $|V_{<s}| = 0$ . Now apply Lemma 8 with  $\alpha = \frac{3}{c}$  and  $\beta = \frac{c}{2}$ . Note that item (i) fails, thus item (ii) holds. Therefore  $G$  contains a  $(\frac{3}{c}, \frac{c}{2})$ -bulging set  $U$  such that:

$$|U| \geq \frac{n}{(\frac{3}{c})^{1+\frac{2}{c}\log \frac{n}{s}}} = n \cdot \left(\frac{c}{3}\right)^{1+\frac{2}{c}\log \frac{n}{s}} \geq n \cdot \left(\frac{c}{3}\right)^{1+\frac{2}{c}\log \frac{2}{p}} \geq \frac{15}{c} \cdot K(n, c, p) \quad (14)$$

where we used the fact that  $p \geq \frac{1}{\sqrt{n}}$  and that  $n$  is large enough to obtain that  $\frac{n}{s} \leq \frac{2}{p}$ . Thus, when  $K(n, c, p) \geq 1 + 2a \ln n$  we have  $|U| \geq \frac{12+29a \ln n}{c}$ .

Moreover  $s_U \geq s = pn$ ; using Equation (13) this yields  $s_U \geq \frac{12+29a \ln n}{c}$ , too. We can then apply Lemma 9. It follows that, with probability at least  $1 - n^{-a}$ , we have  $\frac{|\hat{V}_{<s_U} \cap H|}{|\hat{V}_{<s_U} \setminus H|} > \frac{c\alpha}{2} > \frac{3}{2}$ .

We deduce that  $G[\hat{V}_{<s_U}]$  contains a clique whose density is at least:

$$\frac{|\hat{V}_{<s_U} \cap H|}{|\hat{V}_{<s_U}|} = \frac{|\hat{V}_{<s_U} \cap H|}{|\hat{V}_{<s_U} \cap H| + |\hat{V}_{<s_U} \setminus H|} > \frac{|\hat{V}_{<s_U} \cap H|}{(\frac{2}{3} + 1)|\hat{V}_{<s_U} \cap H|} = \frac{1}{2} + \frac{1}{10}. \quad (15)$$

With the same probability we have simultaneously that  $|\hat{V}_{<s_U}| \geq \frac{c}{2}|U| > 7 \cdot K(n, c, p)$ . Now consider the invocation of DENSECLIQUEFINDER on  $G[\hat{V}_{<s_U}]$ . By Theorem 6, that invocation finds a clique of size at least:

$$7 \cdot K(n, c, p) \cdot \left(2 \cdot \frac{1}{10}\right) = \frac{7}{5} \cdot K(n, c, p) \quad (16)$$

Next, we bring the running time in  $\tilde{O}(n^2)$  while ensuring an output clique of size  $K(n, c, p)$ . To this end, change the loop at line 3 so as to iterate only over  $i$  in the form  $i = (1 + \eta)^j$  for some  $\eta > 0$ . For the smallest  $i = (1 + \eta)^j$  such that  $V_{<s_U} \subseteq \{v_1, \dots, v_i\}$  the subgraph  $\hat{G}[v_1, \dots, v_i]$  will then have clique density  $\frac{\frac{1}{2} + \frac{1}{10}}{1 + \eta}$  and will contain  $\hat{V}_{<s_U}$  plus at most  $\eta|\hat{V}_{<s_U}|$  other vertices. By choosing  $\eta > 0$  sufficiently small one can then ensure that DENSECLIQUEFINDER when ran on  $\hat{G}[v_1, \dots, v_i]$  returns a clique of size at least  $K(n, c, p)$ . The total number of iterations is in  $O(\log_{1+\eta} n) = O(\log n)$ , and by Theorem 6 every iteration takes time  $\tilde{O}(n^2)$ , giving a total time of  $\tilde{O}(n^2)$  too.

To finally bring the running time in  $\tilde{O}(\|\hat{G}\|)$ , upon receiving  $\hat{G}$  we check whether  $\|\hat{G}\| \leq (n^{\frac{1}{\log n}})$ . If that is the case then  $c \leq \frac{1}{\log n}$  and  $K(n, c, p) \leq 1$  as shown above; in this case we return any vertex of  $\hat{G}$ . Otherwise, we run the algorithm above. In both cases the bounds are satisfied and the running time is in  $\tilde{O}(\|\hat{G}\|)$ . ◀

## 6 Application to BALANCED BICLIQUE

In this section we restate and prove a more formal and general version of Theorem 3:

► **Theorem 11.** *There exists a deterministic polynomial-time algorithm  $\mathcal{A}$  with the following guarantees. Fix any  $a \geq 1$ . Let  $c := c(n) \in \omega\left(\frac{1}{\log n}\right)$ . For every  $n$  large enough and every  $n$ -vertex graph  $G$ , when given  $\hat{G} \sim \mathcal{G}\left(G, K_{\frac{cn}{2}, \frac{cn}{2}}\right)$  in input,  $\mathcal{A}$  with probability at least  $1 - n^{-a}$  returns a balanced biclique of size at least:*

$$\frac{c}{48} \cdot 2\sqrt{\frac{c \log n}{2}}.$$

Moreover  $\mathcal{A}(\hat{G})$  runs in time  $\tilde{O}(\|\hat{G}\|)$  for every input graph  $\hat{G}$ .

## 11:10 On Finding Randomly Planted Cliques in Arbitrary Graphs

The algorithm behind the theorem, Algorithm 2, is based on the following intuition. Observe that our main technical result, Lemma 8, essentially says that every graph  $G$  contains either (i) a large number of vertices of small slack (and thus large degree), or (ii) a large bulging set. If (i) holds, then we can hope to find a large biclique by just intersecting the neighborhoods of those vertices (namely, of the  $k$  vertices with largest degree for some suitable value of  $k$ ). If (ii) holds, then we can hope to find a large biclique by exploiting the “densification” phenomenon used by our clique algorithm (see Section 5). The structure of the algorithm follows this intuition, with a first phase that finds a large biclique if (i) holds and a second phase that finds a large biclique if (ii) holds.

■ **Algorithm 2** BALANCEDBICLIQUEFINDER( $\hat{G}$ ).

---

```

1:  $v_1, \dots, v_n \leftarrow$  vertices of  $\hat{G}$  in non-increasing order of degree
2:  $L, R \leftarrow \emptyset$ 
3: for  $1 \leq i \leq n$ : do ▷ Phase 1
4:    $L' \leftarrow \{v_1, \dots, v_i\}$ 
5:    $R' \leftarrow \bigcap_{v \in L'} N_v$ 
6:   if  $\min\{|L'|, |R'|\} > \min\{|L|, |R|\}$  then
7:      $L, R \leftarrow L', R'$ 
8:   end if
9: end for
10: for  $1 \leq i < j \leq n$ : do ▷ Phase 2
11:    $L', R' \leftarrow \text{BICLIQUEEXTRACTOR}(\hat{G}, \{v_i, \dots, v_j\})$ 
12:   if  $\min\{|L'|, |R'|\} > \min\{|L|, |R|\}$  then
13:      $L, R \leftarrow L', R'$ 
14:   end if
15: end for
16: return  $L, R$ 

```

---

Before delving into the proof, we need a certain subroutine that “extracts” a large balanced biclique of a graph  $G$  when given a subset  $S$  of vertices of some (larger) balanced biclique of  $G$ . This is the subroutine BICLIQUEEXTRACTOR appearing at line 11 of Algorithm 2, and it plays a role similar to the one played by DENSECLIQUEFINDER in the case of clique.

► **Lemma 12.** *There exists a deterministic algorithm  $\mathcal{A}$  with the following guarantees. Let  $G = (V, E)$  be an  $n$ -vertex graph containing a balanced biclique with sides  $A$  and  $B$ . Given in input  $G$  and  $S \subseteq A \cup B$ , algorithm  $\mathcal{A}$  returns a biclique of  $G$  with sides  $L, R$  such that  $\min(|L|, |R|) \geq \frac{|S|}{3}$ . The running time of  $\mathcal{A}$  is  $O(|S| \cdot n \log n)$ .*

**Proof.** We prove that Algorithm 3 satisfies the statement.

■ **Algorithm 3** BICLIQUEEXTRACTOR( $G, S$ ).

---

```

1: compute the complement  $\bar{G}[S]$  of  $G[S]$ 
2: compute  $G_1, \dots, G_r$ , the connected components of  $\bar{G}[S]$  in nonincreasing order of vertex size
3: if  $|V(G_1)| > \frac{|S|}{3}$  then
4:   return  $L = V(G_1)$  and  $R = \bigcap_{u \in V(G_1)} N_G(u)$ 
5: else
6:   compute the smallest  $i \in [r]$  such that  $\sum_{j=1}^i |V(G_j)| > \frac{|S|}{3}$ 
7:   return  $L = \bigcup_{j=1}^i V(G_j)$  and  $R = \bigcup_{j=i+1}^r V(G_j)$ 
8: end if

```

---

First, let us prove that the algorithm returns sets  $L, R$  that are sides of a complete biclique and such that  $\min(|L|, |R|) > \frac{|S|}{3}$ . We begin by noting the following crucial fact: for each  $i = 1, \dots, r$  we have  $V(G_i) \subseteq A$  or  $V(G_i) \subseteq B$ . Suppose, in fact, that there exist  $u \in V(G_i) \cap A$  and  $v \in V(G_i) \cap B$ . Since  $G_i$  is connected, along any path from  $u$  to  $v$  in  $G_i$  there must exist an edge whose vertices belong to  $A$  and  $B$ , respectively. Without loss of generality we can thus assume that  $u$  and  $v$  are such vertices. By definition of  $G_i$  this means that  $u$  and  $v$  are not adjacent in  $G$ , a contradiction. Now we distinguish the two cases on which the algorithm branches.

1.  $|V(G_1)| > \frac{|S|}{3}$ .

In this case, as  $V(G_1) \subseteq A$  or  $V(G_1) \subseteq B$ , by construction of  $R$  we have  $R \supseteq B$  or  $R \supseteq A$ . Therefore:

$$|R| \geq |A| = \frac{|A \cup B|}{2} \geq \frac{|S|}{2} \geq \frac{|S|}{3} \quad (17)$$

Hence,  $\min(|L|, |R|) > \frac{|S|}{3}$ . Moreover note that  $L, R$  are sides of a biclique by construction of  $R$ .

2.  $|V(G_1)| \leq \frac{|S|}{3}$ . Then by the ordering of  $G_1, \dots, G_r$  we have  $|V(G_i)| \leq \frac{|S|}{3}$  for all  $i = 1, \dots, r$ . Note how this implies that the index  $i$  computed by the algorithm satisfies:

$$\left| \bigcup_{j=1}^{i-1} V(G_j) \right| \leq \frac{|S|}{3} \quad (18)$$

Therefore:

$$\frac{|S|}{3} < \left| \bigcup_{j=1}^i V(G_j) \right| = \left| \bigcup_{j=1}^{i-1} V(G_j) \right| + |V(G_i)| \leq \frac{|S|}{3} + \frac{|S|}{3} = \frac{2|S|}{3} \quad (19)$$

This implies again  $\min(|L|, |R|) \geq \frac{|S|}{3}$ . Moreover  $L, R$  form again the sides of a biclique; this is because  $G_1, \dots, G_r$  are connected components of  $\bar{G}[S]$ , hence in  $G$  all edges are present between  $V(G_j)$  and  $V(G_{j'})$  for every distinct  $j, j'$ .

We now analyze the running time of the algorithm. Computing  $\bar{G}[S]$  takes time  $O(|S|^2 \log n)$  by checking for each of the edges in the sorted adjacency lists of  $G$ . Computing and sorting the connected components  $G_1, \dots, G_r$  takes time  $O(|S|^2 + |S| \log |S|)$ . The case  $|V(G_1)| > \frac{|S|}{3}$  requires time  $O(|V(G_1)| \cdot n) = O(|S| \cdot n)$  if the intersection of the neighborhoods is done using a bitmap indexed by  $V(G)$ . The case  $|V(G_1)| \leq \frac{|S|}{3}$  takes time  $O(|S|)$ . We conclude that the algorithm runs in time  $O(|S|^2 \log n + |S| \cdot n) = O(|S| \cdot n \log n)$ . ◀

We are now ready to prove Theorem 11.

**Proof of Theorem 11.** We prove the biclique size guarantees and the running time bounds separately.

**Guarantees.** Let  $\beta = \frac{c}{2}$ , and define:

$$f(\alpha, n, s) := \frac{n}{\alpha^{2 + \frac{1}{\beta} \log \frac{n}{s}}} \quad (20)$$

We begin by showing that, whenever  $s \leq \frac{n}{f(\alpha, n, s)}$  and  $\alpha \geq \max(2, f(\alpha, n, s))$ ,

1. If item (i) of Lemma 8 holds, then the first phase of Algorithm 1 finds a biclique with at least  $\lfloor \frac{2}{3} \cdot f(\alpha, n, s) \rfloor$  vertices per side.

## 11:12 On Finding Randomly Planted Cliques in Arbitrary Graphs

2. If item (ii) of Lemma 8 holds, then the second phase of Algorithm 1 finds with high probability a biclique with at least  $\frac{c}{6} \cdot f(\alpha, n, s)$  vertices per side.

Since by Lemma 8 itself at least one of items (i) and (ii) holds, the algorithm finds with high probability a balanced biclique of size  $\Omega(c \cdot f(\alpha, n, s))$ . We will then choose  $\alpha, s$  that satisfy the constraints above while (roughly) maximizing  $f(\alpha, n, s)$ .

We prove 1. To ease the notation let  $f = f(\alpha, n, s)$ . If item (i) of Lemma 8 holds, then  $|V_{<s}| \geq f$ , hence in  $G$  (and thus in  $\hat{G}$ ) there are at least  $f$  vertices of slack smaller than  $s \leq n/f$ . By a simple counting argument, any  $k$  of those vertices have at least  $n - k(s + 1)$  neighbors in common. Choosing  $k = \lfloor \frac{2f}{3} \rfloor$ , and using the fact that  $s \leq n/f$ , the common neighbors are at least  $n - \frac{2f}{3}(n/f + 1) = \frac{n-2f}{3}$ . Now observe that, since  $\alpha \geq 2$ , then  $n \geq 4f$ , hence  $\frac{n-2f}{3} \geq \frac{2f}{3}$ . We conclude that the loop at line 3 of Algorithm 2 eventually returns a biclique whose smallest side has at least  $\lfloor \frac{2f}{3} \rfloor$  vertices.

We prove 2. If item (ii) of Lemma 8 holds, then  $G$  contains an  $(\alpha, c/2)$ -bulging set  $U$  of size at least  $\alpha f$ . Let  $S = \hat{V}_{<s|U}$ . Leveraging Lemma 9 through the same arguments used in the proof of Theorem 10, as long as  $\alpha f$  and  $s$  are in  $\Omega(c^{-1}a \log n)$  and sufficiently large, with probability at least  $1 - n^{-a}$  we have  $|S \cap H| > \frac{c}{2}|U|$  and  $|S \cap H| > \frac{c\alpha}{2}|S \setminus H|$ , where  $H$  is the set of vertices of the planted biclique. Now consider any ordering of  $S$  (in particular the one given by the degrees). If  $|S \setminus H| = \emptyset$ , then the ordering itself is a sequence of elements of  $H$  of length  $|S| = |S \cap H| > \frac{c}{2}|U| \geq \frac{c\alpha f}{2}$ . If instead  $|S \setminus H| \neq \emptyset$ , as  $|S \cap H| > \frac{c\alpha}{2}|S \setminus H|$  the pigeonhole principle implies that the ordering contains a contiguous sequence of vertices of  $H$  of length at least  $\frac{c\alpha}{2}$ , and therefore are least  $\frac{cf}{2}$  as we are assuming  $\alpha \geq f$ . We conclude that in any case the ordering of  $S$  contains a contiguous sequence of vertices of  $H$  of length  $\min\left(\frac{cf}{2}, \frac{c\alpha f}{2}\right) = \frac{cf}{2}$ . By construction Algorithm 2 eventually runs BICLIQUEEXTRACTOR on that sequence and thus, by Lemma 12, finds a biclique with at least  $\frac{cf}{6}$  vertices per side.

It remains to choose suitable values of  $\alpha, s$  so as to approximately maximize  $f$  subject to the constraints  $s \leq \frac{n}{f(\alpha, n, s)}$  and  $\alpha \geq \max(2, f(\alpha, n, s))$ . The argument above then yields with probability  $1 - n^{-a}$  a biclique with  $\Omega(cf(\alpha, n, s))$  vertices per side. We set:

$$s = \frac{n}{f} \quad (21)$$

$$\alpha = f \quad (22)$$

This yields the equation:

$$\alpha = f(\alpha, n, s) = \frac{n}{\alpha^{2+\frac{1}{\beta} \log \frac{n}{s}}} = \frac{n}{\alpha^{2+\frac{1}{\beta} \log \alpha}} \quad (23)$$

Recalling that  $\beta = c/2$ , rearranging, and taking logarithms yields:

$$\frac{2}{c} \log^2 \alpha + 3 \log \alpha - \log n = 0. \quad (24)$$

Solving for  $\log \alpha$  gives:

$$\log \alpha = \frac{-3 + \sqrt{9 + \frac{8 \log n}{c}}}{\frac{4}{c}} = \sqrt{\frac{9}{16}c^2 + \frac{c}{2} \log n} - \frac{3}{4}c > \sqrt{\frac{c}{2} \log n} - 1 \quad (25)$$

We conclude that:

$$\alpha > 2\sqrt{\frac{c}{2} \log n} - 1 \quad (26)$$

Notice that by definition  $s, \alpha$  satisfy the constraint  $\alpha \geq \max(2, f(\alpha, n, s))$  as long as  $\alpha = f \geq 2$ . Since we are assuming that  $c \in \omega\left(\frac{1}{\log n}\right)$ , Equation (26) guarantees that  $f = \alpha \geq 2$  holds for large enough  $n$ .

Finally, the lower bound above on the size of each side of the biclique is thus:

$$\frac{cf}{6} = \frac{c\alpha}{6} > \frac{c}{12} \cdot 2\sqrt{\frac{c}{2} \log n} \quad (27)$$

**Running time.** We describe a variant of BALANCEDBICLIQUEFINDER that runs in time  $\tilde{O}(\|\hat{G}\|)$  and finds a biclique of size at least  $1/4$ -th of that of the original algorithm. As a first thing, we compute  $|E(\hat{G})|$ . If  $|E(\hat{G})| < (\frac{cn}{2})^2$ , then necessarily  $c \leq \frac{1}{\log n}$ , and the bound of Theorem 11 is smaller than 1. In this case we immediately return any edge of  $G$ , satisfying the bounds. If instead  $|E(\hat{G})| \geq (\frac{cn}{2})^2$  then we run the  $\tilde{O}(n^2)$ -time variant of Algorithm 2 described below. This makes the running time in  $\tilde{O}(\|\hat{G}\|)$  in every case. As a byproduct, the lower bound on the biclique size will shrink by a factor  $\frac{4}{5}$ .

The variant of Algorithm 2 is as follows. First, observe that the loop of line 3 can be implemented in  $\tilde{O}(n^2)$  total time by computing  $R'$  incrementally (this can be done either via a bitmap or using binary search over the sorted adjacency lists). For the loop at line 10, we reduce the running time by coarsening. Instead of iterating over all  $1 \leq i < j \leq n$ , for each  $h = 1, \dots, \lceil \log n \rceil$  we iterate over all subsequences  $v_i, \dots, v_j$  with  $i = k2^h$  and  $j = k2^h + k - 1$ , for  $k = 0, 1, 2, \dots$ . Clearly, for every contiguous subsequence  $S$  of  $v_1, \dots, v_n$ , we will iterate over some subsequence  $S' \subseteq S$  with  $|S'| \geq |S|/4$ . The bound on the size of the biclique thus decreases by a factor of 4. The running time can be easily bounded by noting that, for every  $h = 1, 2, \dots$ , the total cost of invoking BICLIQUEEXTRACTOR on all the subsequences of size  $2^i$  is in  $\tilde{O}(n^2)$  by Lemma 12. As the loop iterates over  $O(\log n)$  values of  $i$ , we conclude that the second phase takes  $\tilde{O}(n^2)$  time overall.  $\blacktriangleleft$

## 7 A lower bound on densification

In this section we prove Theorem 5. This shows that, whenever  $c < 1/2$ , there exist arbitrarily large graphs  $G$  such that the high degree profiles of typical instances from  $\mathcal{G}(G, K_{cn})$  are essentially uncorrelated with the planted clique.

Throughout the section, for a graph  $G$  we let  $\kappa(G)$  be the size of the largest clique in  $G$ . We start by defining a graph  $H$  that has between one and two vertex for every degree (or, equivalently, every slack) from 1 to  $n - 1$ . Let  $H = (V, E)$  where  $V = [n]$  for  $n \geq 3$ , and

$$E = \left\{ \{u, v\} : u, v \in V, u \neq v, u + v \leq n + 1 \right\}. \quad (28)$$

Note that  $N_H(u) = [1, n - u + 1] \setminus \{u\}$ ; hence

$$s_u = \begin{cases} u - 1 & u \leq \frac{n+1}{2} \\ u - 2 & u > \frac{n+1}{2} \end{cases} \quad (29)$$

This implies that, for every  $0 \leq s \leq n - 1$ ,

$$V_{\leq s} \in [s + 1, s + 2]. \quad (30)$$

The graph  $G$  of Theorem 5 is a perturbation of  $H$  as given by the next result.

► **Lemma 13.** *Let  $G$  be an  $n$ -vertex graph, let  $\eta \in [0, 1]$ , and let  $G'$  be obtained from  $G$  by deleting each edge independently with probability  $\eta$ . For every  $a > 1$ , with probability at least  $1 - 2n^{1-a}$ :*

1.  $\kappa(G') < 2a \frac{\ln n}{\eta} + 1$ .
2.  $|V_{\leq s}| \leq |V'_{\leq s'}|$  for all  $s \geq 0$ , where  $s' = s + \eta n + \sqrt{an \ln n}$ .

## 11:14 On Finding Randomly Planted Cliques in Arbitrary Graphs

**Proof.** *Item 1.* Fix  $U \subseteq V$  on  $k \geq 2a \frac{\ln n}{\eta} + 1$  vertices. Then:

$$\Pr[G'[U] \text{ is a clique}] \leq (1 - \eta)^{\binom{k}{2}} < e^{-\eta \binom{k}{2}} = e^{-k \cdot \eta \frac{k-1}{2}} \leq e^{-ka \ln n} = n^{-ak}. \quad (31)$$

Taking a union bound over all  $U$  yields  $\Pr[\kappa(G') \geq k] < n^{(1-a)k} \leq n^{1-a}$ .

*Item 2.* Fix  $u \in V$ . Then  $s'_u = s_u + \sum_{i=1}^{d_u} X_i$ , where the  $X_i$  are independent Bernoulli random variables with parameter  $\eta$ . By Hoeffding's inequality, for every  $t \geq 0$ ,

$$\Pr[s'_u \geq s_u + \eta n + t] \leq \Pr[s'_u \geq s_u + \eta d_u + t] \leq e^{-\frac{t^2}{d_u}} < e^{-\frac{t^2}{n}} \quad (32)$$

For  $t = \sqrt{an \ln n}$  we obtain  $\Pr[s'_u \geq s_u + \eta n + \sqrt{an \ln n}] \leq n^{-a}$ . This implies that, for every  $s \geq 0$ , every  $v \in V_{\leq s}$  satisfies  $v \in V'_{\leq s'}$  with probability  $1 - n^{-a}$ , where  $s' = s + \eta n + \sqrt{an \ln n}$ . By a union bound we conclude that, with probability  $1 - n^{1-a}$ , we have  $|V'_{\leq s'}| \geq |V_{\leq s}|$  for all  $s \geq 0$ .  $\blacktriangleleft$

As a corollary we get the graph  $G$  used in the proof of Theorem 5:

► **Corollary 14.** *For every  $\eta \in [0, 1]$  and every  $n \geq 3$  there exists an  $n$ -vertex graph  $G$  such that:*

1.  $\kappa(G) \leq 4 \frac{\ln n}{\eta} + 1$ .
2.  $s - \eta n - \sqrt{2n \ln n} \leq |V_{\leq s}| \leq s + 2$  for all  $s \geq 0$ .

**Proof.** Apply Lemma 13 to the graph  $H$  defined above for  $a = 2$ , noting that  $1 - 2n^{1-a} > 0$ .  $\blacktriangleleft$

The next result bounds the number of vertices of the planted clique that end up having a certain slack in  $\hat{G}$ .

► **Lemma 15.** *Let  $c \in (0, 1)$ , let  $G$  be any graph, and let  $\hat{G} \sim \mathcal{G}(G, K_{cn})$ . With probability at least  $1 - \frac{3}{n}$  we have simultaneously for all  $s \geq 0$ :*

$$c \cdot |V_{\leq s}| - \sqrt{2n \ln n} \leq |K \cap \hat{V}_{\leq s}| \leq c \cdot |V_{\leq s^*}| + \sqrt{2n \ln n}. \quad (33)$$

where  $s^* = \frac{s + \sqrt{2n \ln n}}{1 - c}$ .

**Proof.** *Lower bound.* Note that  $|K \cap \hat{V}_{\leq s}| \geq |K \cap V_{\leq s}|$ , and  $|K \cap V_{\leq s}| = \sum_{i=1}^{|V_{\leq s}|} X_i$  where the  $X_i$  are non-positively correlated Bernoulli random variables of parameter  $c$ . By Hoeffding's inequality, then, the probability that the lower bound of the claim fails is at most  $\frac{1}{n^2}$  for any given  $s \geq 0$ . By a union bound, thus, the lower bound holds fails for some  $s$  with probability at most  $\frac{1}{n}$ .

*Upper bound.* Let  $v \notin V_{\leq s^*}$ , so  $s_v > s^*$ . Note that  $\hat{s}_v = s_v - \sum_{i=1}^{s_v} X_i$ , with the  $X_i$  non-positively correlated Bernoulli random variables of parameter  $c$ . Therefore  $\mathbb{E}[\hat{s}_v] = (1 - c)s_v$ , and:

$$s = (1 - c)s^* - \sqrt{2n \ln 2n} < (1 - c)s_v - \sqrt{2n \ln n} = \mathbb{E}[\hat{s}_v] - \sqrt{2n \ln 2n} \quad (34)$$

By Hoeffding's inequality we then get  $\Pr[\hat{s}_v \leq s] \leq \frac{1}{n^2}$ . By a union bound this implies that, with probability at least  $1 - \frac{1}{n}$ ,

$$K \cap \hat{V}_{\leq s} \subseteq K \cap V_{\leq s^*} \quad \forall s = 1, \dots, n - 1 \quad (35)$$

Consider then  $|K \cap V_{\leq s^*}|$ . Note that this is a sum of  $|V_{\leq s^*}|$  non-positively correlated Bernoulli random variables of parameter  $c$ . Another application of Hoeffding's inequality yields with probability at least  $1 - \frac{1}{n^2}$ :

$$|K \cap V_{\leq s^*}| \leq c \cdot |V_{\leq s^*}| + \sqrt{2n \ln n} \quad (36)$$

A final union bound over all  $s \geq 0$  and the three events above concludes the proof.  $\blacktriangleleft$

We are now ready to prove Theorem 5.

**Proof of Theorem 5.** Let  $\eta = a \cdot c^{-1} \sqrt{\frac{\ln n}{n}}$  for some  $a > 0$  to be defined. Let  $G$  be the corresponding graph given by Corollary 14, and let  $\hat{G} \sim \mathcal{G}(G, K_{cn})$ . We begin by observing that it is sufficient to prove Theorem 5 for the case  $\{v_1, \dots, v_j\} = \hat{V}_{\leq s}$  for some  $s \geq 0$ .

Consider indeed any ordering  $v_1, \dots, v_n$  of the vertices of  $\hat{G}$  by nonincreasing degree. Observe that for every  $j = 1, \dots, n$  there exists  $s \geq 0$  and  $\hat{S} \subseteq \hat{V}_{\leq s} \setminus \hat{V}_{\leq s-1}$  such that

$$\{v_1, \dots, v_j\} = \hat{V}_{\leq s-1} \cup \hat{S} \quad (37)$$

Now suppose the bound of Lemma 15 holds. We claim that  $|\hat{S}| \leq (2 + a) \frac{\sqrt{n \ln n}}{c}$ . Indeed:

$$|\hat{S}| \leq |\hat{V}_{\leq s}| \setminus |\hat{V}_{\leq s-1}| \quad (38)$$

$$\leq |\hat{V}_{\leq s}| \setminus |V_{\leq s-1}| \quad V_{\leq s-1} \subseteq \hat{V}_{\leq s-1} \quad (39)$$

$$\leq \left( c \cdot \frac{s + \sqrt{2n \ln n}}{1 - c} + \sqrt{2n \ln n} \right) - (s - 1 - \eta n) \quad \text{Lemma 15 and Corollary 14} \quad (40)$$

$$= \left( c \cdot \frac{s + \sqrt{2n \ln n}}{1 - c} + \sqrt{2n \ln n} \right) - \left( s - 1 - a \frac{\sqrt{n \ln n}}{c} \right) \quad \text{definition of } \eta \quad (41)$$

$$\leq (2 + a) \frac{\sqrt{2n \ln n}}{c} \quad (42)$$

where in the last inequality we used  $c \leq \frac{1}{2}$ . Now notice that the upper bound of Theorem 5 has an  $O\left(\frac{\sqrt{2n \ln n}}{c}\right)$  additive term. Therefore, as said, it is sufficient to prove the theorem for the case  $\{v_1, \dots, v_j\} = \hat{V}_{\leq s}$  for some  $s \geq 0$ .

Consider then any  $0 \leq s \leq n - 1$ . If  $|\hat{V}_{\leq s}| \leq a \cdot c^{-1} \sqrt{n \ln n}$  then Equation (1) is trivially true. Suppose then  $|\hat{V}_{\leq s}| > a \cdot c^{-1} \sqrt{n \ln n}$ . We have:

$$|K \cap \hat{V}_{\leq s}| \leq c \cdot |V_{\leq s^*}| + \sqrt{2n \ln n} \quad \text{Lemma 15} \quad (43)$$

$$\leq c \cdot (s^* + 2) + \sqrt{2n \ln n} \quad \text{item 2 of Corollary 14} \quad (44)$$

$$= O\left(cs + \sqrt{n \ln n}\right) \quad \text{definition of } s^* \text{ and } c \leq \frac{1}{2} \quad (45)$$

By item 1 of Corollary 14, and since  $V_{\leq s} \subseteq \hat{V}_{\leq s}$ , we have  $s \leq |\hat{V}_{\leq s}|$ . As  $|\hat{V}_{\leq s}| > a \cdot c^{-1} \sqrt{n \ln n}$ , we have  $\sqrt{n \ln n} < \frac{c}{a} |\hat{V}_{\leq s}|$ . Plugging these bounds in the inequality above gives  $|K \cap \hat{V}_{\leq s}| = O(c |\hat{V}_{\leq s}|)$ . To conclude, observe that:

$$\kappa\left(\hat{G}[\hat{V}_{\leq s}]\right) \leq \kappa(G) + |K \cap \hat{V}_{\leq s}| \quad (46)$$

and that  $\kappa(G) \leq \frac{\ln n}{\eta} = a c \sqrt{n \ln n}$  by Corollary 14 and our choice of  $\eta$ . Together with our bound on  $|K \cap \hat{V}_{\leq s}|$  this gives the claim.  $\blacktriangleleft$



## 8 Conclusions and Future Work

In this paper, we considered Feige’s semi-random planted clique model, where the input graph is obtained by planting uniformly at random a clique of size  $c \cdot n$  in an arbitrary  $n$ -vertex graph. We presented a simple, deterministic, almost-linear-time algorithm that successfully recovers a clique of size  $(c/3)^{O(1/c)} \cdot n$  under the mild assumption that the original graph has maximum degree at most  $(1-p)n$  for some constant  $p > 0$ . This result suggests a separation from worst-case scenarios: assuming the Unique Games Conjecture, no polynomial-time algorithm can recover cliques of size  $\Omega(n)$  in general graphs, even when the maximum degree bound holds. Our technique also extends to the case of planting balanced bicliques. Overall, our results show that even the limited randomness provided by planting a clique or biclique in an arbitrary graph can be successfully harnessed by rather simple algorithms.

We leave open three main questions. The first one had already been posed by Feige in the original formulation of the problem: is it possible to prove any hardness result under any complexity assumption? The main obstacle seems to be the presence of randomness in the formulation of the problem, which makes the design of suitable reductions a non-trivial task. The second one concerns removing the degree constraint: is it possible to retain guarantees similar to our, but without any constraint on the maximum degree of the original graph? If this the case, the results in Section 7 suggest that different techniques are required. The third question is whether our results can be extended or generalized to other kinds of planted subgraphs, beyond cliques and balanced bicliques. This may include dense or multipartite graphs, potentially shedding light on how randomly planting a subgraph alone affects the complexity of this kind of problems.

---

## References

- 1 Noga Alon. The algorithmic aspects of the regularity lemma. In *Proceedings., 33rd Annual Symposium on Foundations of Computer Science*, pages 473–481, October 1992. doi:10.1109/SFCS.1992.267804.
- 2 Noga Alon and Nabil Kahale. Approximating the independence number via the  $\vartheta$ -function. *Mathematical Programming*, 80:253–264, February 1998. doi:10.1007/BF01581168.
- 3 Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Structures & Algorithms*, 13(3-4):457–466, 1998. doi:10.1002/(SICI)1098-2418(199810/12)13:3/4<457::AID-RSA14>3.0.CO;2-W.
- 4 Sanjeev Arora and Rong Ge. New tools for graph coloring. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 1–12, 2011. doi:10.1007/978-3-642-22935-0\_1.
- 5 Anne Auger and Benjamin Doerr. *Theory of Randomized Search Heuristics*. WORLD SCIENTIFIC, 2011. doi:10.1142/7438.
- 6 Mitali Bafna, Jun-Ting Hsieh, and Pravesh K. Kothari. Rounding large independent sets on expanders. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC ’25*, pages 631–642, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3717823.3718137.
- 7 Nikhil Bansal and Subhash Khot. Optimal long code test with one free bit. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS ’09*, pages 453–462, USA, 2009. IEEE Computer Society. doi:10.1109/FOCS.2009.23.
- 8 Boaz Barak, Samuel B. Hopkins, Jonathan Kelner, Pravesh Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. In *Proceedings - 57th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2016*, pages 428–437, United States, 2016. IEEE Computer Society. doi:10.1109/FOCS.2016.53.
- 9 Ravi Boppana and Magnús Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT Numerical Mathematics*, 32:13–25, January 2006. doi:10.1007/3-540-52846-6\_74.

- 10 Rares Darius Buhai, Pravesh K. Kothari, and David Steurer. Algorithms approaching the threshold for semi-random planted clique. In *STOC 2023 - Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1918–1926, 2023. doi:10.1145/3564246.3585184.
- 11 Parinya Chalermsook, Wanchote Po Jiamjitrak, and Ly Orgo. On finding balanced bicliques via matchings. In *Graph-Theoretic Concepts in Computer Science: 46th International Workshop, WG 2020, Leeds, UK, June 24–26, 2020, Revised Selected Papers*, pages 238–247, Berlin, Heidelberg, 2020. Springer-Verlag. doi:10.1007/978-3-030-60440-0\_19.
- 12 Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*, pages 47–60, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3055399.3055491.
- 13 Amin Coja-Oghlan. Finding large independent sets in polynomial expected time. *Combinatorics, Probability and Computing*, 15(5):731–751, 2006. doi:10.1017/S0963548306007553.
- 14 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.
- 15 Roei 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, STOC '16*, pages 77–90, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2897518.2897561.
- 16 Devdatt Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, October 2009. doi:10.1017/CB09780511581274.
- 17 Uriel Feige. Approximating maximum clique by removing subgraphs. *SIAM J. Discrete Math.*, 18:219–225, October 2004. doi:10.1137/S089548010240415X.
- 18 Uriel Feige. Introduction to semirandom models. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, pages 189–211. Cambridge University Press, United Kingdom, January 2021. doi:10.1017/9781108637435.013.
- 19 Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *J. Comput. Syst. Sci.*, 63(4):639–671, December 2001. doi:10.1006/jcss.2001.1773.
- 20 Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures & Algorithms*, 16(2):195–208, 2000. doi:10.1002/(SICI)1098-2418(200003)16:2<195::AID-RSA5>3.0.CO;2-A.
- 21 Uriel Feige and Robert Krauthgamer. The probable value of the lovász-schrijver relaxations for maximum independent set. *SIAM Journal on Computing*, 32(2):345–370, 2003. doi:10.1137/S009753970240118X.
- 22 Uriel Feige and Eran Ofek. Finding a maximum independent set in a sparse random graph. *SIAM Journal on Discrete Mathematics*, 22(2):693–718, 2008. doi:10.1137/060661090.
- 23 Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh S. Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *J. ACM*, 64(2), April 2017. doi:10.1145/3046674.
- 24 Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990.
- 25 Magnús M. Halldórsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45(1):19–23, 1993. doi:10.1016/0020-0190(93)90246-6.
- 26 Johan Håstad. Clique is hard to approximate within  $n^{1-\epsilon}$ . In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 627–636, 1996. doi:10.1109/SFCS.1996.548522.
- 27 Mark Jerrum. Large cliques elude the metropolis process. *Random Structures & Algorithms*, 3(4):347–359, 1992. doi:10.1002/rsa.3240030402.
- 28 David S. Johnson. The NP-completeness column: An ongoing guide. *J. Algorithms*, 8(5):438–448, 1987. doi:10.1016/0196-6774(87)90021-6.
- 29 George Karakostas. A better approximation ratio for the vertex cover problem. *ACM Trans. Algorithms*, 5(4), November 2009. doi:10.1145/1597036.1597045.

- 30 David Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, March 1998. doi:10.1145/274787.274791.
- 31 Richard Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 40:85–103, January 1972. doi:10.1007/978-3-540-68279-0\_8.
- 32 S. Khot. Improved inapproximability results for maxclique, chromatic number and approximate graph coloring. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 600–609, 2001. doi:10.1109/SFCS.2001.959936.
- 33 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within  $2 - \epsilon$ . *Journal of Computer and System Sciences*, 74(3):335–349, 2008. Computational Complexity 2003. doi:10.1016/j.jcss.2007.06.019.
- 34 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)*, volume 93 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:12, Dagstuhl, Germany, 2018. doi:10.4230/LIPIcs.FSTTCS.2017.37.
- 35 Luděk Kučera. Expected complexity of graph partitioning problems. *Discrete Appl. Math.*, 57(2–3):193–212, 1995. doi:10.1016/0166-218X(94)00103-K.
- 36 Pasin Manurangsi. Inapproximability of maximum biclique problems, minimum k-cut and densest at-least-k-subgraph from the small set expansion hypothesis. *Algorithms*, 11(1), 2018. doi:10.3390/a11010010.
- 37 Theo McKenzie, Hermish Mehta, and Luca Trevisan. A new algorithm for the robust semi-random independent set problem. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 738–746, 2020. doi:10.1137/1.9781611975994.45.
- 38 Jacob Steinhardt. Does robustness imply tractability? a lower bound for planted clique in the semi-random model. *arXiv e-prints*, 2017. doi:10.48550/arXiv.1704.05120.
- 39 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 681–690, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1132516.1132612.

## A

 Concentration inequalities

The following bounds can be found in [5] or derived from [16]. Let  $X_1, \dots, X_n$  be binary random variables. We say that  $X_1, \dots, X_n$  are non-positively correlated if for all  $I \subseteq \{1, \dots, n\}$ :

$$\Pr(\forall i \in I \mid X_i = 0) \leq \prod_{i \in I} \Pr(X_i = 0) \quad (47)$$

and

$$\Pr(\forall i \in I \mid X_i = 1) \leq \prod_{i \in I} \Pr(X_i = 1). \quad (48)$$

Then:

► **Lemma 16.** *Let  $X_1, \dots, X_n$  be independent or, more generally, non-positively correlated binary random variables. Let  $a_1, \dots, a_n \in [0, 1]$  and  $X = \sum_{i=1}^n a_i X_i$ . Then, for any  $\epsilon > 0$ , we have:*

$$\Pr(X \leq (1 - \epsilon)\mathbb{E}[X]) \leq e^{-\frac{\epsilon^2}{2}\mathbb{E}[X]} \quad (49)$$

and

$$\Pr(X \geq (1 + \epsilon)\mathbb{E}[X]) \leq e^{-\frac{\epsilon^2}{2+\epsilon}\mathbb{E}[X]} \quad (50)$$