Finding a Maximum Independent Set in a Sparse Random Graph

Uriel Feige * Eran Ofek †

December 23, 2007

Abstract

We consider the problem of finding a maximum independent set in a random graph. The random graph G, which contains n vertices, is modelled as follows. Every edge is included independently with probability $\frac{d}{n}$, where d is some sufficiently large constant. Thereafter, for some constant α, a subset I of αn vertices is chosen at random, and all edges within this subset are removed. In this model, the planted independent set I is a good approximation for the maximum independent set I_{max}, but both $I \setminus I_{max}$ and $I_{max} \setminus I$ are likely to be nonempty. We present a polynomial time algorithms that with high probability (over the random choice of random graph G, and without being given the planted independent set I) finds the maximum independent set in G when $\alpha \geq \sqrt{\frac{c_0}{d}}$, where c_0 is some sufficiently large constant independent of d.

1 Introduction

Let $G = (V, E)$ be a graph. An independent set I is a subset of vertices which contains no edges. The problem of finding a maximum size independent set in a graph is a fundamental problem in Computer Science and it was among the first problems shown to be NP-hard [18]. Moreover, Hastad shows [15] that for any $\epsilon > 0$ there is no $n^{1-\epsilon}$ approximation algorithm for the maximum independent set problem unless NP=ZPP. The best approximation ratio currently known for maximum independent set [7] is $O(n(\log\log n)^2/(\log n)^3)$.

In light of the above mentioned negative results, we may try to design a heuristic that performs well on typical instances. Karp [17] proposed trying to find a maximum independent set in a random graph. However, even this problem appears to be beyond the capabilities of current algorithms. For example, let $G_{n,1/2}$ denote the random graph on n vertices obtained by choosing randomly and independently each possible edge with probability $1/2$. The size of the maximum independent set in a random $G_{n,1/2}$ graph is almost surely $2(1 + o(1)) \log_2 n$. A simple greedy algorithm almost surely finds an independent set of size $\log_2 n$ [14]. However, there is no known polynomial time algorithm that almost surely finds an independent set of size $(1 + \epsilon) \log_2 n$ (for any $\epsilon > 0$).

To further simplify the problem, Jerrum [16] and Kucera [19] proposed a *planted model* $G_{n,1/2,k}$ in which a random graph $G_{n,1/2}$ is chosen and then a clique of size k is randomly placed in the graph. (A clique in a graph G is an independent set in the edge complement of G, and hence all algorithmic results that apply to one of the problems apply to the other.) Alon, Krivelevich and Sudakov [2] gave an algorithm based on spectral techniques that almost surely finds the planted clique for $k = \Omega(\sqrt{n})$. More generally, one may extend the range of parameters of the above model by planting an independent set in $G_{n,p}$, where p need not be equal to $1/2$, and may also depend on n. The $G_{n,p,\alpha}$ model is as follows: n vertices are partitioned at random into two sets of vertices, I of size αn and C of size $(1 - \alpha)n$. No edges are placed within the set I, thus making it an independent set. Every other possible edge (with at least one endpoint not in I) is added independently at random with probability p. The goal of the algorithm, given the input G (but without being given the partition into I and C) is to find a maximum independent set. Intuitively, as α becomes

*Department of Computer Science and Applied Mathematics, the Weizmann Institute, Rehovot 76100, Israel. uriel.feige@weizmann.ac.il

†Schema, Herzlia 46905, Israel. eran.ofek@gmail.com

smaller the size of the planted independent is closer to the probable size of the maximum independent set in $G_{n,p}$ and the problem becomes harder.

We consider values of p as small as d/n where d is a large enough constant. A difficulty that arises in this sparse regime (i.e. when d is constant) is that the planted independent set I is not likely to be a maximum independent set. Moreover, with high probability I is not contained in a maximum independent set of G. For example, there are expected to be $e^{-d}n$ vertices in C of degree one. It is very likely that two (or more) such vertices $v, w \in C$ will have the same neighbor, and that it will be some vertex $u \in I$. This implies that every maximum independent set will contain v, w and not u, and thus I contains vertices that are not contained in the maximum independent set.

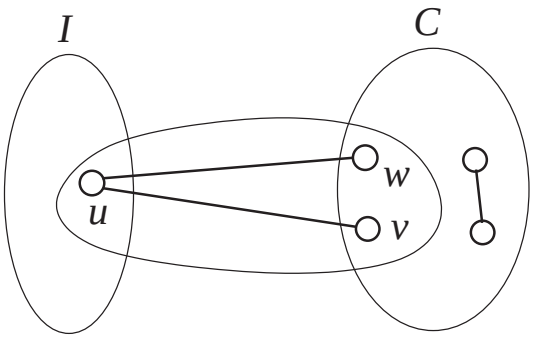

Figure 1: The vertex $u \in I$ is not contained in any maximum independent set because no other edges touch v, w.

A similar argument shows that there are expected to be $e^{-\Omega(d)}n$ isolated edges. This implies that there will be an exponential number of maximum independent sets.

1.1 Our result

Our main result is summarized in the following theorem.

Theorem 1.1. *There is a polynomial time algorithm that almost surely finds the maximum independent set of a graph G selected at random from the distribution $G_{n,\frac{d}{n},\alpha}$, when $d > d_0$ and $\alpha \geq \sqrt{c_0/d}$ (d_0, c_0 are some universal constants). The parameter d can also be an increasing function of n.*

The bulk of the paper is devoted to proving the above theorem. To simplify the presentation of the proof it will be convenient to assume upper bounds on d and on α, namely $d < n^{1/40}$ and $\alpha < 1/3$. These upper bounds do not limit the generality of our results, because when d or α exceed these uppers bounds the proof becomes simpler (see later discussions in the paper).

1.2 Related work

For $p = 1/2$, $\alpha = \Omega(1/\sqrt{n})$, Alon, Krivelevich and Sudakov [2] gave an efficient spectral algorithm which almost surely finds the planted independent set. For the above mentioned parameters the planted independent set is likely to be the unique maximum independent set.

A few papers deal with *semi-random* models which extend the planted model by enabling a mixture of random and adversarial decisions. Feige and Kilian [8] considered the following model: a random $G_{n,p,\alpha}$ graph is chosen, then an adversary may add arbitrarily many edges between I and C, and make arbitrary changes (adding or removing edges) inside C. For any constant $\alpha > 0$ they give a heuristic that almost surely outputs a list of independent sets containing the planted independent set, whenever $p > (1+\epsilon)\ln n/\alpha n$ (for any $\epsilon > 0$). The planted independent set may not be the only independent set of size αn since the adversary has full control over the edges inside C. Possibly, this makes the task of finding the planted independent set

harder. In [9] Feige and Krauthgamer considered a less adversarial semi-random model in which an adversary is allowed to add edges to a random $G_{n,\frac{1}{2},\frac{1}{\sqrt{n}}}$ graph. Their algorithm almost surely extracts the planted independent set and certifies its optimality. Coja-Oghlan [5] considered similar semi-random variants of random $G_{n,p,\sqrt{c/pn}}$ graphs, with $\ln(n)^2/n < p < 0.99$ (and c a sufficiently large constant). He shows among other things that in most such graphs a maximum independent set can be found in polynomial time.

Extracting from the above discussion the known previous results on planted independent set models (and ignoring issues such as handling semirandom instances), we see that the same tradeoff between d and α as claimed in the current paper (namely, $\alpha \geq \sqrt{c_0/d}$) was previously established for the case $d = n/2$ (first in [2], and then again in [9]), and then extended all the way down to $d = \ln^2 n$ in [5]. The condition $\alpha \geq \sqrt{c_0/d}$ is needed mainly to ensure that the effect of planting the independent set I shows up in the spectrum of the adjacency matrix of the graph. For the previous range of parameters, the planted independent set is likely to be the maximum independent set in the graph. In this paper we address the case that d is a (sufficiently large) constant, which introduces difficulties not present in earlier work, mainly due to the fact that the planted independent set is likely not to be the maximum independent set.

Heuristics for optimization problems different than max independent set will be discussed in the following section.

1.2.1 Technique and outline of the algorithm

Our algorithm builds on ideas from the algorithm of Alon and Kahale [1], which was used for recovering a planted 3-coloring in a random graph. The algorithm that we propose has four phases, and is sketched below. This overall structure is similar to the structure of the algorithm of [1], but phases 3 and 4 include ingredients that are different than those of [1]. These differences are discussed towards the end of this section.

1. Get a coarse approximation I', C' of I, C with $|C \triangle C'| + |I \triangle I'| < \frac{1}{60}|I|$. This phase is based on spectral techniques.

2. Reduce the error in the approximation by using an iterative procedure based on the number of neighbors that each vertex has in I'. The error term $|C \triangle C'| + |I \triangle I'|$ is reduced to at most n/d^{18}.

3. Commit to placing certain vertices of I' in the final independent set. This is done by moving vertices from I' and C' to OUT (a new set), and committing to placing the remaining part of I' in the final independent set. The choice of vertices to move to OUT depends on their degree into I'. When this process ends, I' is an independent set, every vertex of C' has at least 4 edges to I' and no vertex of I' has edges to OUT. Using the fact that sparse random graphs (almost surely) have no small dense sets, it will be shown that $I' \subseteq I_{max}$ and also that OUT is rather small.

4. Extend the independent set I' optimally using the vertices of OUT. This is done by finding a maximum independent set among the vertices of OUT and adding it to I'. The structure of OUT will be simple enough so that a maximum independent set can be efficiently found. The proof is based to a large extent on the fact that OUT is small. If vertices of OUT were chosen independently at random from the input graph G, then it would have been easy to show that all connected components in OUT are very small (say, of size $O(\log n)$), and then a maximum independent set in OUT can be found in time polynomial in n (even by exhaustive search). However, OUT is a result of a deterministic process applied to G, which makes the analysis of its structure considerably more difficult.

The technique of [1] was implemented successfully on various problems in the planted model: planted hypergraph coloring, planted 3-SAT, planted 4-NAE, min-bisection (by Chen and Frieze [4], Flaxman [12], Goerdt and Lanka [13], Coja-Oghlan [6] respectively).

Perhaps the work closest in nature to the work in the current paper is that of Amin Coja-Oghlan [6] on finding a bisection in a sparse random graph. Both in our work and in that of [6], one is dealing with an optimization problem, and the density of the input graph is such that the planted solution is not an optimal solution. The algorithm for bisection in [6] is based on spectral techniques, and has the advantage that it

provides a certificate showing that the solution that it finds is indeed optimal. We do not address the issue of certification in this paper. In [6] the random instance is generated as follows. The vertices of the graph are partitioned into two classes of equal size randomly. Then the edges are inserted: edges inside the two classes with probability p' and edges crossing the partition with probability p independently. Intuitively, as $p'-p$ becomes smaller, the problem becomes harder. Denote by $d_1 = np'/2, d_2 = np/2$ the expected degree of a vertex into its own class and into the other class respectively. The algorithm in [6] is proven to succeed (almost surely) whenever $d_1 - d_2 \geq \sqrt{c_0 d_1 \log d_1}$. In our independent set model the problem becomes harder as αd becomes smaller. If we denote by $\tilde{d}_1 = d, \tilde{d}_2 = (1-\alpha)d$ the expected degrees of a vertex in C and I respectively, then our algorithm (almost surely) succeeds whenever $\tilde{d}_1 - \tilde{d}_2 = \alpha\tilde{d}_1 \geq \sqrt{c_0\tilde{d}_1}$. We remark that a significant source for complications in our algorithm and its analysis comes from a tightening of the parameters, saving a $\sqrt{\log \tilde{d}}$ factor in the size of α. A preliminary version of this work (see [11]) presents a simpler algorithm but requires $\alpha\tilde{d}_1 \geq \sqrt{c_0\tilde{d}_1 \log \tilde{d}_1}$

An important difference between planted models for independent set and those for other problems such as 3-coloring and min-bisection is that in our case the planted classes I, C are not symmetric. The lack of symmetry between I and C makes some of the ideas used for the more symmetric problems insufficient. In the approach of [1], a vertex is removed from its current color class and placed in OUT if its degree into some other current color class is very different than what one would typically expect to see between the two color classes. This procedure is shown to "clean" every color class C from all vertices that should have been from a different color class, but were wrongly assigned to class C in phase 2 of the algorithm. (The argument proving this goes as follows. Every vertex remaining in the wrong color class by the end of phase 3 must have many neighbors that are wrongly assigned themselves. Thus the set of wrongly assigned vertices induces a small subgraph with large edge density. But G itself does not have any such subgraphs, and hence by the end of phase 3 it must be the case that all wrongly assigned vertices were moved into OUT.) It turns out that this approach works well when classes are of similar nature (such as color classes, or two sides of a bisection), but does not seem to suffice in our case where I' is supposed to be an independent set whereas C' is not. Specifically, the set I' might still contain wrongly assigned vertices, and might not be a subset of a maximum independent set in the graph. Under these circumstances, phase 4 will not result in a maximum independent set. Our solution to this problem involves the following aspects, not present in previous work. In phase 3 we remove from I' every vertex that has even one edge connecting it to OUT. This adds more vertices to OUT, and makes the connected components in OUT larger. We do not attempt to analyze the maximum size of connected components in OUT, which is a key ingredient in previous approaches. Instead, we analyze the 2-core of OUT and show that the 2-core has no large components. Then, in phase 4, we use dynamic programming to find a maximum independent set in OUT, and use the special structure of OUT to show that the algorithm runs in polynomial time.

1.3 Notation

Let $G = (V, E)$ and let $U \subset V$. The subgraph of G induced by the vertices of U is denoted by $G[U]$. When the set of edges used is clear from the context, we will use $\deg(v)_U$ to denote the degree of a vertex v into a set $U \subset V$. To specify exactly the set of edges used, we use $\deg^E(v)_U$ which is the degree of a vertex v into a set U induced by the set of edges E. We use $\Gamma(U)$ to denote the vertex neighborhood of $U \subset V$ (excluding U). The parameter d (specifying the expected degree in the random graph G) is assumed to be sufficiently large, and some of the inequalities that we shall derive implicitly use this assumption, without stating it explicitly. The term *with high probability* (w.h.p.) is used to denote a sequence of probabilities that tends to 1 as n tends to infinity.

2 The Algorithm

The first step of the algorithm is tailored to deal with the case that α is small. When α is large (in fact, $\alpha > \sqrt{100 \log d/d}$ suffices), the combination of step 1 and step 2 can be replaced by a simpler step based on

partitioning the vertices according to their degrees (see Section 3.2 for details). Hence we shall assume here that $\alpha \leq 1/3$, and this will somewhat simplify the presentation of the algorithm and its analysis. Furthermore, we assume for simplicity that the values of the parameters α, d are given as input to the algorithm. This assumption can be avoided by enumerating all possible values of α, d (actually we need only a good approximations of α, d).

Algorithm $FindIS(V, E)$

1. Let A' be the adjacency matrix of the graph induced by removing from G all vertices of degree $> 5d$. Compute the eigenvector of the most negative eigenvalue of A', denoted by $v_{n'}$. Sort the vertices of V' in order of the value of the corresponding entry in $v_{n'}$, breaking ties arbitrarily. Let I_1 be either the first αn vertices in this sorted order or the last αn vertices. (To choose among the above two options, find a maximal matching in each of the two possible induced graphs $G(I_1)$, and pick the option for which the maximal matching found is smaller.) Set $C_1 = V \setminus I_1$.
2. Set $C_2^0 = C_1, I_2^0 = I_1$. Iterate $j = 1, 2, .., \log n$:
   for every vertex v: if $\deg(v)_{I_2^{j-1}} < \alpha d/2$ then $v \in I_2^j$, otherwise $v \in C_2^j$.
3. (a) Set $I_3 = I_2^{\log n}$, $C_3 = C_2^{\log n}$, $OUT_3 = \emptyset$.
   (b) For every edge (u, v) such that both u, v are in I_3, move u, v to OUT_3.
   (c) A vertex $v \in C_3$ is *removable* if $\deg(v)_{I_3} < 4$.
   Iteratively: find a removable vertex v and move it from C_3 to OUT_3. If v has neighbors in I_3, move these neighbors from I_3 to OUT_3.
4. Find a maximum independent set in $G[OUT_3]$ (this will be shown to be doable in polynomial time, see Corollary 3.4). Output the union of this independent set and I_3.

Figure 2 depicts the situation after step 3 of the algorithm is done. At that point, I_3 is an independent set, there are no edges between I_3 and OUT_3, and every vertex $v \in C_3$ has at least four neighbors in I_3.

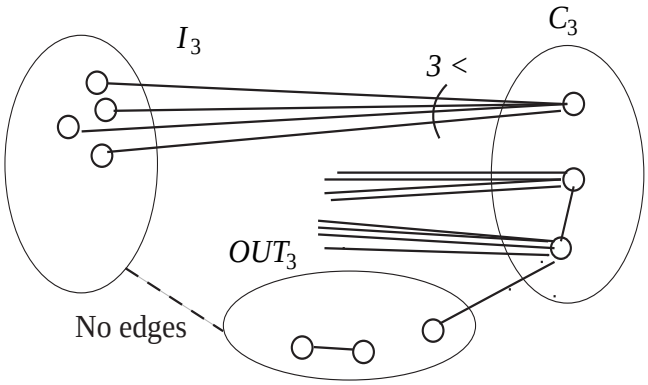

Figure 2: $FindIS$ Step 3 outcome

3 Correctness

Let I_{max} be a maximum independent set of G. We establish two theorems. Theorem 3.1 guarantees the correctness of the algorithm and Theorem 3.3 guarantees its efficient running time. Here we present these two theorems, and their proofs are deferred to later sections.

Theorem 3.1. *With high probability there exists some maximum independent set I_{max} such that $I_3 \subseteq I_{max}, C_3 \cap I_{max} = \emptyset$.*

Definition 3.2. The 2-*core* of a graph G is the maximal subgraph in which the minimal degree is 2.

It is easy to see that the 2-core is unique and can be found by iteratively removing vertices whose degree < 2.

Theorem 3.3. *With high probability the largest connected component in the 2-core of $G[OUT_3]$ has cardinality of at most $2\log n$.*

Let G be any graph. Those vertices of G that do not belong to the 2-core form trees. Each such tree is either disconnected from the 2-core or it is connected by exactly one edge to the 2-core. To find a maximum independent set of $G[OUT_3]$ we need to find a maximum independent set in each connected component of $G[OUT_3]$ separately. For each connected component D_i of $G[OUT_3]$ we find the maximum independent set as follows: let C_i be the intersection of D_i with the 2-core of $G[OUT_3]$. We enumerate all possible independent sets in C_i (there are at most $2^{|C_i|} \leq 2^{2\log n}$ possibilities), each one of them can be optimally extended to an independent set of D_i by solving (separately) a maximum independent set problem on each of the trees connected to C_i. For some trees we may have to exclude the tree vertex which is connected to D_i if it is connected to a vertex of the independent set that we try to extend. On each tree the problem can be solved by dynamic programming.

Corollary 3.4. *A maximum independent set in $G[OUT_3]$ can be found in polynomial time.*

3.1 Dense sets, degree deviations and other properties

In proving the correctness of the algorithm, we will use structural properties of the random graph G. In particular, such a random graph most likely has no small dense sets (small sets of vertices that induce many edges). This fact will be used on several occasions to derive a proof by contradiction. Namely, certain undesirable outcomes of the algorithm cannot occur, as otherwise they will lead to a discovery of a small dense set in G. Most of the following lemmas are standard, and those proofs not given in this section will be given in Section 4.

Lemma 3.5. *Let G be a random $G_{n,p,\alpha}$ ($p = \frac{d}{n}$, $d < n^{1/40}$) random graph. W.h.p. the following hold:*

1. *There is no set $U \subset V$ of cardinality smaller than $2n/d^5$ such that the number of edges induced by U is more than $\frac{4}{3}|U|$.*
2. *There is no set $U \subset V$ of size $< \alpha n/40$ containing $\alpha d|U|/14$ edges (and this happens with probability of at least $> 1 - o(n^{-\sqrt{d}})$).*
3. *There is no $C' \subseteq C$ such that $n/2d^5 \leq |C'| \leq \frac{2n\log d}{d}$ and $|\Gamma(C') \cap I| \leq |C'|$.*

Corollary 3.6. *Let G be a graph which has the property from Lemma 3.5 part 1. Let A, B be any two disjoint sets of vertices each of size smaller than n/d^5. If every vertex of B has at least 2 edges going into A, then $|A| \geq |B|/2$.*

Proof. Assume for the sake of contradiction that $|A| = \delta|B|$ for some $0 < \delta < 1/2$. The number of internal edges of $A \cup B$ is at least $2|B| > \frac{4}{3}(1+\delta)|B| = \frac{4}{3}|A \cup B|$. The last inequality contradicts part 1 of Lemma 3.5. □

The following lemma bounds the number of vertices whose degree largely deviates from its expectation.

Lemma 3.7. *Let $d < n^{1/40}$. With probability $> 1 - e^{-n^{0.1}}$:*

1. *There are at most n/d^{21} vertices in C whose degree into I is $< 0.9\alpha d$.*

2. There are at most $3e^{-d}dn$ edges that contain a vertex with degree at least $3d$.

Lemma 3.8. *W.h.p. the size of a maximum independent set in $G[C]$ is $< \frac{2n\log d}{d}$.*

Proof. The probability for an independent set of size k is at most

$$\binom{n}{k}\left(1-\frac{d}{n}\right)^{k(k-1)/2} \leq \left(\frac{ne}{k}\right)^k e^{-\frac{dk(k-1)}{2n}} = e^{k(\log(\frac{n}{k})+1-\frac{d(k-1)}{2n})}. \tag{1}$$

Setting $k = \frac{2n\log d}{d}$ the above term becomes

$$e^{k(\log d - \log\log d - \log 2 + 1)} e^{-k(\log d - o(1))} = e^{-k(\log\log d + \log 2 - 1 - o(1))} = o(1). \tag{2}$$

□

The following Lemma differs from previous lemmas in this section in the sense that it does not refer to a "static" property that the input graph G is likely to have, but rather to the likely outcome of Step 1 of algorithm $FindIS(G)$. It also differs from other lemmas in this section in the sense that its proof is considerably more complicated.

Lemma 3.9. *Let $\sqrt{\frac{c_0}{d}} \leq \alpha \leq 1/3$, $d < n^{1/40}$. With high probability $|I_1 \triangle I| < |I|/60$.*

Definition 3.10. An instance G (taken from $G_{n,d/n,\alpha}$) is *good* if it has all the properties listed in Lemmas 3.5, 3.7, 3.8, 3.9.

3.2 Proof of Theorem 3.1

In the proof of Theorem 3.1 we will assume that $d < n^{1/40}$. When d is very large (e.g., $d = n/2$), some of the lemmas that we use, such as Lemma 3.5, are not true. However, when d is very large, these lemmas are also not needed. When $d > n^{1/40}$, the planted independent set is almost surely the unique maximum independent set, and can be found using algorithms described in earlier work [2, 5].

When $\alpha \geq 1/3$ step 1 of the algorithm can be replaced by the following simpler step: put in I_1 all the vertices of degrees $< (1-\alpha/2)d$ and the rest of the vertices in C_1. A simple argument (similar to the proof of Lemma 3.7) shows that in this case w.h.p. $|I \triangle I_1| < |I|/60$ (see [11] for more details). Hence when analyzing step 1 of the algorithm (proving that w.h.p. $|I \triangle I_1| < |I|/60$), we will without loss of generality limit the range of α to $\alpha < 1/3$.

We will show that if G is good then steps $1, 2, 3$ of the algorithm give a very good approximation of I, C.

3.2.1 An aid in the analysis

For the sake of the analysis, we introduce Definition 3.11. This definition identifies certain key sets of vertices in the graph (V_2 and V_3). Definition 3.11 involves an algorithm that has complete knowledge of the planted independent set I. To avoid the possibility of confusion, let us stress here that this algorithm is not part of the algorithm $FindIS$, and moreover, the sets V_2 and V_3 are neither given to nor computed by algorithm $FindIS$.

Definition 3.11. Given a graph $G = (V, E)$ with an independent set I (and a corresponding vertex cover $C = V \setminus I$) the *P-core* of G is a subset of V which is extracted using the following steps.

CR0:	Initialization:	$V_2 = I \cup \{v \in C \mid \deg(v)_I \geq 0.9\alpha d\}$.
	Iteratively:	(i) if there is $v \in I \cap V_2$ with $\deg(v)_{V\setminus V_2} > \alpha d/4$ remove v from V_2.
		(ii) if there is $v \in C \cap V_2$ with $\deg(v)_{I\cap V_2} < 0.8\alpha d$, remove v from V_2.
CR1:	Initialization:	$V_3 = V_2$,
		remove from V_3 all the vertices of $V_3 \cap I$ that have edges to $V \setminus V_3$.
	Iteratively:	find a vertex $v \in V_3 \cap C$ with $\deg(v)_{V_3\cap I} < 4$, remove v and its neighbors in I from V_3.

The P-core of G is defined to be V_3. We use $\overline{\text{P-core}}$ to denote $V \setminus$ P-core.

The sets V_2 and V_3 have certain structural properties, as defined in Definition 3.11. We shall also need the fact that if G is a good graph (in the sense of Definition 3.10) then almost all vertices of G are in V_2 and in V_3.

Lemma 3.12. *For a good graph G, the following hold:*

1. $|V \setminus V_2| < n/d^{20}$ *(where $V_2 = CR0(G)$).*
2. $|\overline{\text{P-core}}| < n/d^{18}$.

Proof. Since G is good, after setting $V_2 = I \cup \{v \in C \mid \deg(v)_I \geq 0.9\alpha d\}$ (and before the iterations of CR0) it holds that $|V \setminus V_2| < n/d^{21}$ (see Lemma 3.7 part 1). In the iteration process, every vertex that we remove from V_2 contributes at least $0.1\alpha d$ edges to $V \setminus V_2$. If the iteration steps are repeated too many times, $V \setminus V_2$ will become dense. Assume by contradiction that at some point the set $V \setminus V_2$ doubled its size (when we compare it to the size before the first iteration). At this point it contains at least $\frac{1}{2}|V \setminus V_2|0.1\alpha d$ edges and its size is at most $2n/d^{21}$, which contradicts the definition of good G (see Lemma 3.5 part 2). This proves item 1 of the lemma.

We now prove item 2 of the lemma. Initially $V_3 = V_2$, at this point $|V \setminus V_3| \leq n/d^{20}$ (using part 1 of this Lemma). We then remove from V_3 all the vertices of I that have edges to $V \setminus V_3$. Doing so, we loose at most $3dn/d^{20} + 3de^{-d}n$ vertices (see the definition of good G in Lemma 3.7 part 2; for a random G this happens with probability $> 1 - e^{-n^{0.1}}$). At this point (just before the iteration steps) $|V \setminus V_3| \leq 4n/d^{19}$. We now begin the iteration process. In every iteration we remove a vertex u of $V_3 \cap C$ whose degree into $V_3 \cap I$ became < 4 (we also remove its neighbors from $V_3 \cap I$). Note that after the initialization of CR1 the degree of u into $V_3 \cap I$ was $\geq 0.8\alpha d$. It follows that after removing u and its neighbors from $V_3 \cap I$ there are at least $0.8\alpha d$ additional edges in $V \setminus V_3$. If the iteration step is repeated too many times the set $V \setminus V_3$ will become too dense. Assume by contradiction that at some point (for the first time during the iterations) $|V \setminus V_3|$ doubled its size when compared to the size of $V_3 \setminus V$ before the first iteration. At this point the number of edges inside $V \setminus V_3$ is at least $\frac{1}{2}|V \setminus V_3| \cdot \frac{1}{4} \cdot 0.8\alpha d$. Moreover, the size of $V \setminus V_3$ at this point is $\leq 8n/d^{19} + 3 < n/d^5$. This can not happen in a good G (see Lemma 3.5 part 2; for a random G this happens with probability $o(n^{-\sqrt{d}})$). We conclude that $|V \setminus V_3| < n/d^{18}$. □

Remark: The reader can verify by inspection that those properties of good graphs that are used in the proof of Lemma 3.12 hold with probability at least $1 - o(n^{-\sqrt{d}})$ (for a random graph taken from $G_{n,d/n,\alpha}$). Hence $\Pr[\overline{\text{P-core}} < n/d^{18}] > 1 - o(n^{-\sqrt{d}})$.

3.2.2 Analysis of Step 1.

Lemma 3.9 states that the spectral phase of the algorithm (step 1) is likely to give a good approximation of the planted independent set. The proof of Lemma 3.9 follows from known principles, but it is rather involved. It is given in Section 4.2.

3.2.3 Analysis of Step 2.

In this section, we assume that G is good (as in Definition 3.10). The approximation I_1, C_1 serves as a bootstrap for the "error reduction" done at step 2. In Lemma 3.13 we show that step 2 significantly reduces the error term i.e. $|I_2 \triangle I| < n/d^{20} < |I|/d^{19}$.

The idea of the proof is as follows. Recall that in Lemma 3.12 we showed that the set $V_2 = CR0(G)$ from Definition 3.11 is of size $> (1 - 1/d^{20})n$. Here we will use the assumption that G is good to show that every iteration of step 2 of $FindIS$ reduces the number of errors (with respect to I, C) in V_2 by a factor of 2. It then follows that after step 2 is done, all the vertices of V_2 are assigned correctly.

Lemma 3.13 (Error reduction)**.** *Let G be a good graph, $V_2 = CR0(G)$ and $I_2 = I_2^{\log n}$ from step 2 of $FindIS(G)$. Then $V_2 \cap (I \triangle I_2) = \emptyset$ and $|I_2 \triangle I| \leq n/d^{20}$.*

Proof. When step CR0 ends, every vertex of $C \cap V_2$ has at least $0.8\alpha d$ edges to $I \cap V_2$ and every vertex of $I \cap V_2$ has at most $\alpha d/4$ edges to $V \setminus V_2$.

For each iteration i of step 2 of $FindIS(G)$, define the respective "error" E_i to be the set of those vertices of V_2 that are assigned incorrectly, namely, $E_i = V_2 \cap (I_2^i \triangle I)$. We will show that each iteration of step 2 of $FindIS(G)$ reduces the error by a factor of at least 2. This is based on the fact (to be shown shortly) that every vertex of E_i has at least $\alpha d/4$ edges to E_{i-1}. Based on this fact we show that an event $|E_i| \geq |E_{i-1}|/2$ would lead to a contradiction. Fix E_i' to be any subset of E_i whose size is $|E_{i-1}|/2$ (if $|E_{i-1}|$ is odd we take a subset of size $\lceil |E_{i-1}|/2 \rceil$ and essentially the same argument goes through). The set $E_i' \cup E_{i-1}$ is of cardinality $\leq \frac{3}{2}|E_{i-1}|$ and it contains at least $\frac{\alpha d}{4}|E_{i-1}|/2$ edges. This contradicts the definition of good G as $|E_{i-1}| \leq |E_0| \leq \alpha n/60$ (see Lemma 3.5 part 2, note that $|E_i' \cup E_{i-1}| \leq \frac{3}{2}|E_{i-1}| \leq \alpha n/40$).

We now show that each vertex of E_i has at least $\alpha d/4$ edges to E_{i-1}.

case 1: $v \in E_{i-1} \cap E_i$ (either $v \in I \cap C_2^{i-1}$ or $v \in C \cap I_2^{i-1}$):
If $v \in I \cap C_2^{i-1}$ (and $v \in E_i$) then in round $i-1$ it has at least $\alpha d/2$ neighbors in I_2^{i-1}, these neighbors are in $C \cap I_2^{i-1}$ since $v \in I$. At least $\alpha d/4$ of these neighbors are in V_2 since v has at most $\alpha d/4$ edges to $V \setminus V_2$ (because $v \in I \cap V_2$). Thus v has $> \alpha d/4$ neighbors in E_{i-1}. If $v \in C \cap I_2^{i-1}$ (and $v \in E_i$) then in round $i-1$ it has at most $\alpha d/2$ neighbors in I_2^{i-1}. Since $v \in V_2 \cap C$ it has $0.8\alpha d$ neighbors in $I \cap V_2$, thus at least $0.3\alpha d$ of them are in $I \cap C_2^{i-1} \subseteq E_{i-1}$.

case 2: $v \in E_i \setminus E_{i-1}$:
If $v \in E_i \setminus E_{i-1} \cap I$ then v was moved from I_2^{i-1} to C_2^i, therefore it has at least $\alpha d/2$ neighbors in $I_2^{i-1} \cap C$. Among them at least $\alpha d/4$ belong to V_2 because v has at most $\alpha d/4$ edges to $V \setminus V_2$. If $v \in E_i \setminus E_{i-1} \cap C$ then v was moved from C_2^{i-1} to I_2^i, therefore it has at most $\alpha d/2$ neighbors in I_2^{i-1}. Since $v \in V_2 \cap C$ it has at least $0.8\alpha d$ neighbors in $I \cap V_2$, among which at least $0.3\alpha d$ are in $I \cap C_2^{i-1} \subseteq E_{i-1}$.

It follows that when step 2 is done all the vertices of V_2 are correctly assigned by the algorithm. It then follows that $|I_2 \triangle I| \leq |V \setminus V_2| \leq n/d^{20}$, where the last inequality follows from Lemma 3.12 part 1. □

3.2.4 Analysis of Step 3.

So far we have shown that at most n/d^{20} vertices of I_2, C_2 are wrongly assigned (with respect to I, C). The goal of step 3 is to "clean" I_2, C_2 yielding I_3, C_3 that can be extended into an optimal solution. Before showing that $I_3 \subseteq I_{max}$ (for some maximum independent set I_{max}) we show that the process of "cleaning" in step 3 does not move too many vertices to OUT_3. This will be used later for proving that $I_3 \subseteq I_{max}$. For the following lemma, recall the notion of $\overline{\text{P-core}}$ from Definition 3.11.

Lemma 3.14. *Let G be a good graph and let OUT_3 be the outcome of step 3 of $FindIS(G)$. Then $OUT_3 \subseteq \overline{\text{P-core}}$, and $OUT_3 < n/d^{18}$.*

Proof. As $\overline{OUT_3} = I_3 \cup C_3$ (where I_3, C_3 are from step 3 of $FindIS(G)$) it is enough to show that $V_3 \subseteq I_3 \cup C_3$. Immediately after step 3a it holds that $I_3 \cup C_3 = V$ and thus $V_3 \subseteq I_3 \cup C_3$. We will show that this is kept during steps 3b, 3c. Initially (at CR1) $V_3 = V_2$. Removing from V_3 all the vertices of $V_3 \cap I$ that have edges to $V \setminus V_3$ ensures that there are no edges between vertices of $V_3 \cap I$ and vertices which were assigned incorrectly (all the vertices of $V_3 \subseteq V_2$ are correctly assigned after step 2 of the algorithm, see Lemma 3.13 part 2). Thus, step 3b of the algorithm does not touch any vertex of V_3 because it removes only edges that contain at least one wrongly assigned endpoint. Finally, the iteration process of CR1 ensures that every vertex of $V_3 \cap C$ has at least 4 edges to vertices in $V_3 \cap I$. Since V_3 is a subset of $I_3 \cup C_3$ at the beginning of step 3c and there are no wrongly assigned vertices in V_3, during step 3c there will never be a vertex of $V_3 \cap C$ that has fewer than 4 edges to vertices of $V_3 \cap I$. We conclude that $V_3 \subseteq I_3 \cup C_3$ at the end of step 3.

Combining with Lemma 3.12 part 2, we deduce that with high probability $OUT_3 < n/d^{18}$. □

As $|I_3 \triangle I| < |I_2 \triangle I| + |OUT_3|$, using Lemmas 3.13, 3.14 we deduce:

Corollary 3.15. *For a good G, it holds that $|I_3 \triangle I| < 2n/d^{18}$.*

It turns out that I_3, C_3 is also a good approximation of I_{max}, C_{max} (I_{max} is any maximum independent set, $C_{max} = V \setminus I_{max}$).

Lemma 3.16. *For a good G it holds $|I_3 \vartriangle I_{max}| < n/d^5$.*

Proof.

$$|I_{max} \vartriangle I_3| \leq |I_{max} \vartriangle I| + |I \vartriangle I_3|$$

By Corollary 3.15 $|I_3 \vartriangle I| < n/d^{18}$. It remains to bound $|I_{max} \vartriangle I|$:

$$|I_{max} \vartriangle I| = |I_{max} \setminus I| + |I \setminus I_{max}| \leq 2|I_{max} \setminus I| = 2|I_{max} \cap C|$$

$I_{max} = (I_{max} \cap I) \cup (I_{max} \cap C)$. One can always replace $I_{max} \cap C$ with $\Gamma(I_{max} \cap C) \cap I$ to get an independent set $(I_{max} \cap I) \uplus (\Gamma(I_{max} \cap C) \cap I)$. The size of a maximum independent set of C is $< \frac{2n \log d}{d}$ (as G is good, see Lemma 3.8). This upper bounds $|I_{max} \cap C|$. From Lemma 3.5 part 3 if $|I_{max} \cap C| > n/(2d^5)$ then $|\Gamma(I_{max} \cap C) \cap I| > |I_{max} \cap C|$ which contradicts the maximality of I_{max}. □

At this point (for a good G) we know that I_3, C_3 have the following two properties:
(i) the error term $|(I_3 \cap C_{max}) \cup (I_{max} \cap C_3)| \leq |I_{max} \vartriangle I_3| < n/d^5$.
(ii) I_3 is an independent set and every vertex of C_3 has at least 4 neighbors in I_3.

The above two properties and the fact that G is good imply that $I_3 \subseteq I_{max}$. This is proven in the following Lemma.

Lemma 3.17 (Extension Lemma)**.** *Let I_m be any independent set of G (the reader may think of I_m as I_{max}) and let $C_m \triangleq V \setminus I_m$. Let I', C', OUT' be an arbitrary partition of V for which I' is an independent set. If the following hold:*

1. *$|(I' \cap C_m) \cup (I_m \cap C')| < n/d^5$.*
2. *Every vertex of C' has at least 4 neighbors in I'. None of the vertices of I' have edges to OUT'.*
3. *The graph G has no small dense subsets as described in Lemma 3.5 part 1.*

then there exists an independent set I_{new} (and $C_{new} \triangleq V \setminus I_{new}$) such that $I' \subseteq I_{new}, C' \subseteq C_{new}$ and $|I_{new}| \geq |I_m|$.

Proof. If we could show that on average a vertex of $U = (I' \cap C_m) \cup (I_m \cap C')$ contributes at least 4/3 internal edges to U, then U would form a small dense set that contradicts Lemma 3.5. This would imply that $U = (I' \cap C_m) \cup (I_m \cap C')$ is the empty set, and we could take $I_{new} = I_m$ in the proof of Lemma 3.17. The proof below extends this approach to cases where we cannot take $I_{new} = I_m$.

Every vertex $v \in C'$ has at least 4 edges into vertices of I'. Since I_m is an independent set it follows that every vertex of $I_m \cap C'$ has at least 4 edges into $I' \cap C_m$. To complete the argument we would like to show that every vertex of $I' \cap C_m$ has at least 2 edges into $I_m \cap C'$. However, some vertices $v \in I' \cap C_m$ might have less than two neighbors in $I_m \cap C'$. In this case, we will modify I_m to get an independent set I_{new} (and $C_{new} \triangleq V \setminus I_{new}$) at least as large as I_m, for which every vertex of $I' \cap C_{new}$ has 2 neighbors in $I_{new} \cap C'$. This is done iteratively; after each iteration we set $I_m = I_{new}, C_m = C_{new}$. Consider a vertex $v \in (I' \cap C_m)$ with $\deg(v)_{I_m \cap C'} < 2$:

- If v has no neighbors in $I_m \cap C'$, then define $I_{new} = I_m \cup \{v\}$. I_{new} is an independent set because v (being in I') has no neighbors in I' nor in OUT'.
- If v has only one edge into $w \in (I_m \cap C')$ then define $I_{new} = (I_m \setminus \{w\}) \cup \{v\}$. I_{new} is an independent set because v (being in I') has no neighbors in I' nor in OUT'. The only neighbor of v in $I_m \cap C'$ is w.

The three properties are maintained also with respect to I_{new}, C_{new} (replacing I_m, C_m): properties 2, 3 are independent of the sets I_m, C_m and property 1 is maintained since after each iteration it holds that $|(I' \cap C_{new}) \cup (I_{new} \cap C')| < |(I' \cap C_m) \cup (I_m \cap C')|$.

When the process ends, let U denote $(I' \cap C_m) \cup (I_m \cap C')$. Each vertex of $I' \cap C_m$ has at least 2 edges into $I_m \cap C'$, thus $|I_m \cap C'| \geq \frac{1}{2}|I' \cap C_m|$ (see Corollary 3.6). Each vertex of $I_m \cap C'$ has 4 edges into $I' \cap C_m$ so the number of edges in U is at least $4|I_m \cap C'| \geq 4|U|/3$ and also $|U| < n/d^5$, which implies that U is empty (by Lemma 3.5 part 1). □

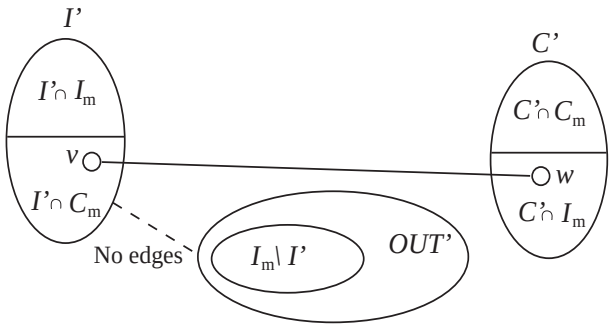

Figure 3: A vertex $v \in (I' \cap C_m)$ which has strictly less than 2 edges into $I_m \cap C'$

Proof of Theorem 3.1. Assume G is good (with high probability this is the case, see Section 3.1). By Lemma 3.16 $|I_3 \triangle I_{max}| < n/d^5$. Using Lemma 3.17 (instantiating I_m, I' from Lemma 3.17 to be I_{max}, I_3 respectively) we derive that I_3 is contained in some maximum independent set. □

It remains to show that the 2-core of OUT_3 has no large connected components.

3.3 Proof of Theorem 3.3

Having established that for a good G, OUT_3 is small (Lemma 3.14), we would now like to establish that its structure is simple enough to allow one to find a maximum independent set of $G[OUT_3]$ in polynomial time. Establishing such a structure would have been easy if the vertices of OUT_3 were chosen independently at random, because a small random subgraph of a random graph G is likely to decompose into connected components no larger than $O(\log n)$. However, OUT_3 is extracted from G using some deterministic algorithm, and hence might have more complicated structure. For this reason, we shall now consider the 2-core of $G[OUT_3]$, and bound the size of its connected components. Note that for good G it holds that $OUT_3 \subseteq \overline{\text{P-core}}$ (Lemma 3.14), thus for such G it is enough to handle $\overline{\text{P-core}}$. This simplifies the proof because the definition of $\overline{\text{P-core}}$ (namely, the procedure that generates in in Definition 3.11) is simpler than that of OUT_3.

To prove that the 2-core of $G[\overline{\text{P-core}}]$ has no large connected component, we shall introduce a notion of a *balanced* connected component. A set of vertices $U \subset V$ will be called *balanced* if it contains at least $|U|/3$ vertices from C. The following lemma shows the relevance of the notion of being balanced to our context.

Proposition 3.18. *For good G, every connected component of the 2-core of $G[\overline{\text{P-core}}]$ is balanced.*

Proof. Let A_i be such a connected component. Every vertex of A_i has degree of at least 2 in the 2-core (of $G[\overline{\text{P-core}}]$). Since A_i is a connected component of the 2-core it follows that every vertex of A_i has degree of at least 2 inside A_i. $|A_i| \leq |\overline{\text{P-core}}| < \frac{n}{d^5}$. If $\frac{|A_i \cap I|}{|A_i|}$ is more than $\frac{2}{3}$, then the number of internal edges in A_i is $> 2 \cdot \frac{2}{3}|A_i| > \frac{4}{3}|A_i|$ which contradicts Lemma 3.5 part 1. □

The following Lemma shows that a connected component is balanced only if it contains a sufficiently large balanced tree.

Lemma 3.19. *Let G be a connected graph whose vertices are partitioned into two sets: C and I. Let $\frac{1}{k}$ be a lower bound on the fraction of C vertices, where k is an integer. For any $1 \leq t \leq |V(G)|/2$ there exists a tree whose size is in $[t, 2t-1]$ and at least $\frac{1}{k}$ fraction of its vertices belong to C.*

Proof. We use the following well know fact: any tree T contains a *center* vertex v such that each subtree hanged on v contains strictly less than half of the vertices of T.

Let T be an arbitrary spanning tree of G. Observe that at least a fraction of $\frac{1}{k}$ of its vertices belongs to C. We show that any such tree T has a subtree T' of size $|T|/2 < |T'| < |T|$ with at least a fraction of $\frac{1}{k}$ of its vertices in C. Repeating this argument until the first time that $|T'| < 2t$ proves the lemma.

Let v be the center of T and let $T_1, ..., T_k$ be the subtrees hanging on v. Let T_j be the subtree with the smallest fraction of C vertices, and take T' to be the tree that remains by removing T_j from T. By properties of the center vertex, $|T|/2 < |T'| < |T|$, as desired. Moreover, if the fraction of C vertices in T_j is $\frac{1}{k}$ or less, then clearly the fraction of C vertices in T' is at least $\frac{1}{k}$. If the fraction of C vertices in T_j is strictly more than $\frac{1}{k}$, then this holds also for all other subtrees of T. By integrality of k, this implies that the fraction of C vertices in T' is at least $\frac{1}{k}$ (because then $k|C \cap T'| > |T'| - 1$ implies $k|C \cap T'| \geq |T'|$). □

To proceed with our discussion, we introduce the following definition.

Definition 3.20. We say that G is *extension friendly* if $G[\overline{\text{P-core}}]$ has no balanced trees of size in $[\log n, 2\log n]$.

In Lemma 3.21 to follow we will see that G is likely to be extension friendly. This is a key component in the proof of Theorem 3.3.

Proof of Theorem 3.3. W.h.p. it holds that G is both good and extension friendly (see Section 3.1 and Lemma 3.21). Since G is good, Proposition 3.18 implies that every connected component of the 2-core of G[P-core] is balanced. Furthermore, any balanced connected component of size at least $2\log n$ (in vertices) must contain a balanced tree of size is in $[\log n, 2\log n - 1]$ (see Lemma 3.19). Since G is extension friendly, the 2-core of G[P-core] does not contain a balanced tree with size in $[\log n, 2\log n]$. Combining these three facts we derive that the 2-core of G[P-core] has no connected components of size $> 2\log n$. Note that since G is good it holds that $OUT_3 \subseteq \overline{\text{P-core}}$ and thus also the 2-core of $G[OUT_3]$ has no connected components of size more than $2\log n$. □

We will now show that G is likely to be extension friendly.

Lemma 3.21. *Assume G is taken from $G_{n,d/n,\alpha}$. W.h.p. $G[\overline{P\text{-core}}]$ has no balanced tree of size in $[\log n, 2\log n]$.*

Proof. As we have seen in Lemma 3.12, $|\overline{\text{P-core}}|$ is likely to be very small, smaller than n/d^{18}. If we were to pick at random such a small subgraph of G, then it would have been very sparse (average degree at most d^{-17}), making it highly unlikely to contain any connected component of size $\log n$ or more, and hence also no tree (let alone a balanced tree) of size $\log n$ or more. The difficulty in proving Lemma 3.21 is that $\overline{\text{P-core}}$ is not a random set of vertices in G. To overcome this difficulty, we look more closely at the random processes by which G is generated, and decouple the random decisions that govern which set of vertices comprise $\overline{\text{P-core}}$ from the random decisions that govern which edges are present in $G[\overline{\text{P-core}}]$. To achieve this decoupling, we shall modify the notion of a P-core, slightly enlarge $\overline{\text{P-core}}$.

We now proceed with the detailed proof. We start with the empty graph (a set V of n vertices and no edges), and build the random graph G as we go.

Step 1. Choose an arbitrary partition of the vertex set V into I and C, with $|I| = \alpha n$.

Step 2. Choose a value t for the size of the balanced tree T. As $\log n \leq t \leq 2\log n$, there are $\log n + 1$ possibly ways of choosing t.

For a tree T, let $I(T)$ denote those vertices of T that are in I, and let $C(T)$ denote those vertices of T that are in C.

Step 3. Choose $t_1 = |I(T)|$, the number of vertices of T in I. Observe that $t_1 \leq 2t/3$ because the tree is balanced. Let $t_2 = |C(T)| = t - t_1 \geq t/3$. There are at most $2t/3 + 1 \leq 2\log n$ ways of choosing t_1.

Step 4. Choose $I(T)$, the vertices of I that will be in the tree T. There are $\binom{\alpha n}{t_1}$ ways of doing this.

Step 5. In this step we start constructing the edge set E of the random graph G. For every pair of vertices $u \in C$ and $v \in I \setminus I(T)$ we include the edge (u, v) in E with probability $p = d/n$. For other pairs of vertices, we do not yet decide whether to include an edge or not. We call the graph obtained after this stage G'.

Step 6. At this step we construct a P'-core which is modified version of the P-core. Its construction follows the construction of a P-core, except for two differences: it is constructed based on G' rather than on G, and the vertices of $I(T)$ are forced not to be part of the P'-core. A detailed description of the procedure for constructing the P'-core will be presented in Definition 3.23. The key properties of the P'-core that we shall need are summarized in the following lemma.

Lemma 3.22. *For the construction of P'-core as described in Definition 3.23.*

1. *Regardless of the choice of $I(T)$ and regardless of which other edges are added to G' so as to create the final graph G, P'-core is always contained in P-core (where P-core is constructed with respect to G in Definition 3.11).*

2. *The* bad event *of $|\overline{P\text{'-core}}| > n/d^{18}$ occurs with probability at most $n^{-\sqrt{d}}$ (here probability is taken over choice of edges in G'). Hence the complimentary* good event *$|\overline{P\text{-core}}| \leq n/d^{18}$ happens almost surely.*

We now assume Lemma 3.22, deferring its proof to later, and continue with the proof of Lemma 3.21.

Step 7. Choose the vertices in $C(T)$. If the good event in Lemma 3.22 occurs, then there are at most $\binom{n/d^{18}}{t_2}$ ways of doing this. If the bad event occurs, then there are $\binom{n}{t_2}$ ways of doing this.

Step 8. Having chosen all vertices of T, we need to choose those pairs of vertices that form the tree edges. We call it the interconnection pattern $E(T)$. There are at most t^{t-2} ways of doing this (which is the number of labelled trees on t vertices). The number may actually be much smaller than t^{t-2}, because there cannot be any edges between vertices in the set $I(T)$.

Step 9. Only now we complete the construction of the random graph G. For every pair of vertices $u, v \in C$ the edge (u, v) is included in E with probability p. Likewise, for every pair of vertices $u \in C$ and $v \in I(T)$, the edge (u, v) is included in E with probability p. Now we compute the probability that all edges of $E(T)$ are in. This probability depends only on step 9, and is exactly p^{t-1}, regardless of the outcome of steps 1 to 8 above.

We can now upper bound the expected number of balanced trees T in $G[\overline{\text{P-core}}]$ with t vertices and $\log n \leq t \leq 2\log n$. Let us fix a value for t and a value for $t_1 = |I(T)|$. By Steps 2 and 3 there are at most $2(\log n)^2$ ways of doing so. By Step 4, there are $\binom{\alpha n}{t_1}$ ways of choosing $I(T)$. By item 1 in Lemma 3.22 it suffices to upper bound the number of balanced trees in $G[\overline{\text{P'-core}}]$, as this will also be an upper bound on the number of balanced trees in $G[\overline{\text{P-core}}]$. By Step 7, the expected number of ways of choosing the vertices of $C(T)$ in $\overline{\text{P'-core}}$ is at most $\binom{n/d^{18}}{t_2} + n^{-\sqrt{d}}\binom{n}{t_2}$. By Step 8, the number of interconnection patterns to consider is at most t^{t-2}. Finally, by Step 9, the probability that all edges of the interconnection pattern are in G is $(d/n)^{t-1}$. Multiplying out all the terms, the expected number of balanced trees is at most:

$$2(\log n)^2 \binom{\alpha n}{t_1} \left(\binom{n/d^{18}}{t_2} + n^{-\sqrt{d}} \binom{n}{t_2} \right) t^{t-2} (\frac{d}{n})^{t-1}$$

The above expression is much smaller than 1. To see this, observe that for sufficiently large d and $t_2 \leq 2\log n$ the term $n^{-\sqrt{d}}\binom{n}{t_2}$ in negligible compared to $\binom{n/d^{18}}{t_2}$. Observe also that $\frac{t^t}{t_1^{t_1} t_2^{t_2}} \leq 2^t$ because $t_1 + t_2 = t$. Using these observations, the above expression can be upper bounded by essentially

$$\frac{(2e)^t (\log n)^2 \alpha^{t_1} d^{t-1} n}{d^{18 t_2}}$$

Observing further that $t_1 \leq t$, $t_2 \geq t/3$, $\alpha < 1$, and using the fact the d is sufficiently large and hence larger than $2e$, this last expression is upper bounded by $n(\log n)^2/d^{4t}$. Using the fact that that $t \geq \log n$ now implies that the expected number of balanced tree is at most $1/n^2$, and hence with probability at least $1 - 1/n^2$ there is no balanced tree of size between $\log n$ and $2\log n$ in $G[\overline{\text{P-core}}]$. □

We present here the detailed definition for the P'-core.

Definition 3.23. Given a graph G' with vertex set partitioned into I and C as in Step 1 of the proof of Lemma 3.21, a set $I(T) \subset I$ as in Step 4 of the proof of Lemma 3.21, and an edge set as in Step 5 of the proof of Lemma 3.21, the *P'-core* of G' is a subset of V which is extracted using the following steps.

CR0':	Initialization:	$V_2' = (I \setminus I(T)) \cup \{v \in C \mid \deg(v)_{I \setminus I(T)} \geq 0.9\alpha d\}$.
	Iteratively:	(i) if there is $v \in I \cap V_2'$ with $\deg(v)_{V \setminus V_2'} > \alpha d/4$ remove v from V_2'.
		(ii) if there is $v \in C \cap V_2'$ with $\deg(v)_{I \cap V_2'} < 0.8\alpha d$, remove v from V_2.

CR1': Initialization: $V_3' = V_2'$,
remove from V_3' all the vertices of $V_3' \cap I$ that have edges to $V \setminus V_3'$.
Iteratively: find a vertex $v \in V_3' \cap C$ with $\deg(v)_{V_3' \cap I} < 4$, remove v and its neighbors in I from V_3'.

The P'-core of G is V_3'.

We now prove Lemma 3.22.

Proof of Lemma 3.22 part 1. The notion of a P-core is defined in Definition 3.11. The notion of a P'-core is defined in Definition 3.23. Both definitions are similar in many respects, and differ only in the following details.

1. $V_2' \cap I$ is initialized to be smaller than $V_2 \cap I$, because all vertices of $I(T)$ are excluded from it. This is in agreement with our goal of showing that P'-core is contained in P-core.
2. $V_2' \cap C$ is initialized to be smaller than $V_2 \cap C$, because vertices in C have fewer edges into $I \setminus I(T)$ then they have into I, and hence less opportunity of achieving a degree of $0.9\alpha d$. Again, this is in agreement with our goal of showing that P'-core is contained in P-core.
3. The notion of P-core refers to the graph G, whereas the notion of P'-core refers to the graph G'. This difference is irrelevant. The two graphs completely agree on all edges between C and $I \setminus I(T)$. The procedure for constructing P'-core only looks at those edges and no other edges. It would produce exactly the same P'-core even if it is run on the graph G instead of G'.

Having understood the differences between the constructions of P-core and P'-core, it is easy to verify by inspection that every vertex that is removed from P-core will be removed from P'-core as well, and hence P'-core is contained in P-core, as desired. □

Proof of Lemma 3.22 part 2. The proof is similar to the proof of Lemma 3.12 (see also the remark after Lemma 3.12). The difference is that in the initialization of CR0' we remove the vertices of $I(T)$ and ignore the edges of C connected to these vertices. The fact that $|I(T)| < 2\log n$ together with the fact that the number of vertices of C affected by the removal of $I(T)$ is bounded by the sum of degrees in $I(T)$ (this number is small because the probability of G having a vertex of degree above $d^2 \log n$ is $O(n^{-d})$) imply that the total number of vertices removed the initializations of CR0' and CR0 differ by $O(d^2 \log n)^2) = o(n/d^{20})$. The rest of the proof of Lemma 3.12 applies without change. □

4 Proofs of the technical lemmas

4.1 Expansion properties and degrees deviation

Proof of lemma 3.5. Let $c > 1$ and $k \geq 3$ (k is an integer) satisfy the following three inequalities:
(i) $c < d$,
(ii) $k < n/e$,
(iii) $\left(\frac{d}{c}\right)^c \left(\frac{ke}{n}\right)^{c-1} e^2 < 1/2$.
The probability that there is a set U of cardinality at most k with $c|U|$ internal edges is at most:

$$\sum_{i=3}^{k} \binom{n}{i} \binom{\binom{i}{2}}{\lceil ic \rceil} \left(\frac{d}{n}\right)^{\lceil ic \rceil} \leq \sum_{i=3}^{k} \left(\frac{ne}{i}\right)^i \left(\frac{i^2 e}{2ic}\right)^{\lceil ic \rceil} \left(\frac{d}{n}\right)^{\lceil ic \rceil} \leq \sum_{i=3}^{k} \left(\frac{d}{c}\right)^{\lceil ic \rceil} \left(\frac{ie}{n}\right)^{\lceil ic \rceil - i} e^{2i} \tag{3}$$

$$\overset{(i),(ii)}{\leq} \sum_{i=3}^{k} \left(\frac{d}{c}\right)^{ic+1} \left(\frac{ie}{n}\right)^{ic-i} e^{2i} = \frac{d}{c} \sum_{i=3}^{k} \left[\left(\frac{d}{c}\right)^c \left(\frac{ie}{n}\right)^{c-1} e^2\right]^i \overset{(iii)}{\leq} \frac{2d}{c} \left(\left(\frac{d}{c}\right)^c \left(\frac{3e}{n}\right)^{c-1} e^2\right)^3$$

Proof of part 1:
Set $c := 4/3,\ k := 2n/d^5$. For $d > d_0$ conditions $(i) - (iii)$ hold. The last term in (3) is at most $\frac{6e^7 d^5}{n}$.

Proof of part 2:
Set $c := \alpha d/14,\ k := \alpha n/40$. For $d > d_0$ conditions $(i) - (iii)$ hold. The last term in (3) is

$$\frac{2d}{c}\left(\left(\frac{d}{c}\right)^c \left(\frac{3e}{n}\right)^{c-1} e^2\right)^3 \leq \frac{(3ed)^{3c+3}}{n^{3c-3}} = o(n^{-\sqrt{d}}),$$

where in the last inequality we used $d < n^{1/40}$ and $c = \alpha d/20 \geq c_0\sqrt{d}/20 \gg \sqrt{d}$ (for large enough c_0). □

Proof of lemma 3.5 part 3. We first show that w.h.p. there is no set $U \subset V$ of size $< \frac{10n\log d}{d}$ containing $50(\log d)|U|$ edges. Set $c := 50\log d$ and $k := \frac{10n\log d}{d}$. For $d > d_0$ conditions (i)-(iii) hold and the last term in (3) is $o(1)$. It thus follows that w.h.p. there is no set U of cardinality $\leq \frac{10n\log d}{d}$ that contains at least $50\log d|U|$ edges.

By contradiction, assume there is a bad set C' for which: $|\Gamma(C') \cap I| \leq |C'|$ and $n/2d^5 \leq |C'| \leq \frac{2n\log d}{d}$. By Lemma 3.7 part 1 at least $|C'| - n/d^{20} > \frac{9}{10}|C'|$ vertices of C' have at least $\frac{\alpha d}{2}$ edges to I. It follows that $C' \cup (\Gamma(C') \cap I)$ has at least $\frac{\alpha d}{5}|C' \cup (\Gamma(C') \cap I)| > \sqrt{d}|C' \cup (\Gamma(C') \cap I)|$ internal edges and its cardinality is at most $2|C'| < \frac{4n\log d}{d} < \frac{10n\log d}{d}$. By the first part of the proof w.h.p. such dense set does not exist. □

We now prove Lemma 3.7.

Proof of part 1. The degrees into I are independent random variables. Set $\delta = 1/d^{21}$. For a fixed set of size δn the expected sum of degrees in $\mu = \delta n\alpha d$. A bad set has only $0.9\delta n\alpha d$ edges to I. The probability for a bad set of size δn is bounded by:

$$\binom{n}{\delta n} e^{-\frac{1}{2}(0.1)^2\mu} \leq e^{-\delta n(\alpha d/200 - \log(e/\delta))} \leq e^{-\delta n(\sqrt{c_0 d}/200 - 21\log d - 1)} \leq e^{-\delta n} < e^{-n/d^{21}} < e^{-n^{0.4}}.$$

In the last inequality we used $d < n^{1/40}$. □

Proof of part 2. The proof of this lemma is very similar to the proof of (26), details are omitted. □

4.2 Spectral Approximation (proof of Lemma 3.9)

Recall that eigenvectors of the adjacency matrix of the graph G are of dimension n, equal to the number of vertices of G. Every coordinate of an eigenvector can be naturally associated with a vertex in the graph.

Let V' be the set of vertices of G with degree $< 5d$, set $n' \triangleq |V'|$. Note that n' is also the dimension of A' – the adjacency matrix of $G[V']$. Let $I' = V' \cap I,\ C' = V' \cap C$, set $\alpha' \triangleq \frac{|I'|}{|V'|}$.

Denote by $\bar{A}'$ the $n' \times n'$ matrix such that $\bar{A}'_{i,j} = 0$ for any $\{i,j\} \subset I'$ and $\bar{A}'_{i,j} = p = d/n$ for the other entries. We will use the fact that $\bar{A}'$ (which is the "expectation" of A' if we ignore the diagonal) is almost surely a good spectral approximation of A' (i.e. the spectral norm of $A' - \bar{A}'$ is small). The rank of $\bar{A}'$ is 2 and it has two non zero eigenvalues. Each of the two non-zero eigenvectors which we denote by $\bar{v}_1, \bar{v}_{n'}$ is constant on I' and constant on C' (this follows from symmetry). Given that each one of $\bar{v}_1, \bar{v}_{n'}$ has only two values, we need to find $\bar{\beta}, \bar{\lambda}$ which satisfy:

$$\overbrace{\begin{array}{|ccc|c|ccc|}\hline 0 & . & 0 & p & . & p & p \\ . & . & & & & & \\ . & & . & & & & \\ 0 & . & 0 & & & & \\ \hline p & & & p & & & \\ . & & & & . & & \\ p & & & & & p & \\ p & & & & & & p \\ \hline \end{array}}^{I'} \quad \begin{bmatrix} 1 \\ . \\ . \\ 1 \\ \bar{\beta} \\ . \\ . \\ \bar{\beta} \end{bmatrix} = \begin{bmatrix} (1-\alpha')n'p\bar{\beta} \\ . \\ . \\ . \\ \alpha'n'p + \bar{\beta}(1-\alpha')n'p \\ . \\ . \\ . \end{bmatrix} = \bar{\lambda} \begin{bmatrix} 1 \\ . \\ . \\ 1 \\ \bar{\beta} \\ . \\ . \\ \bar{\beta} \end{bmatrix},$$

or equivalently:

$$(1-\alpha')n'\bar{\beta}p = \bar{\lambda} \tag{4}$$
$$\alpha'n'p + \bar{\beta}(1-\alpha')n'p = \bar{\lambda}\bar{\beta} \tag{5}$$

a simple calculation gives a quadratic equation in $\bar{\beta}$ whose solutions are

$$\bar{\beta}_{1,2} = \frac{1}{2}\left(1 \pm \sqrt{1 + \frac{4\alpha'}{1-\alpha'}}\right) \tag{6}$$

By substituting $\bar{\beta}$ at (4) we derive:

$$\begin{array}{ll} \bar{\lambda}_1 & = (1-\alpha')n'\bar{\beta}_1 p, \\ \bar{\lambda}_{n'} & = (1-\alpha')n'\bar{\beta}_2 p, \\ \bar{v}_1 & = (\underbrace{1,1,..,1}_{\alpha'n'}, \underbrace{\bar{\beta}_1, \bar{\beta}_1, .., \bar{\beta}_1}_{(1-\alpha')n'}), \\ \bar{v}_{n'} & = (\underbrace{1,1,..,1}_{\alpha'n'}, \underbrace{\bar{\beta}_2, \bar{\beta}_2, \bar{\beta}_2, .., \bar{\beta}_2}_{(1-\alpha')n'}). \end{array} \tag{7}$$

Define $\beta_{1,2} := \frac{1}{2}(1 \pm \sqrt{1 + \frac{4\alpha}{1-\alpha}})$.

Lemma 4.1. *(i) For every $x \geq 0$ it holds that $|1 - \sqrt{1+x}| \leq \frac{x}{2}$.*
(ii) For $0 \leq x \leq 3$ it holds that $1 - \sqrt{1+x} \leq -\frac{x}{3}$.

Proof of part (i). Note that $1 - \sqrt{1+x} \leq 0$ for $x \geq 0$, thus it is enough to show that $1 - \sqrt{1+x} \geq -\frac{x}{2}$.

$$-\frac{x}{2} \leq 1 - \sqrt{1+x} \iff \sqrt{1+x} \leq 1 + \frac{x}{2} \iff 1 + x \leq 1 + x + \frac{x^2}{4}$$

□

Proof of part (ii).

$$1 - \sqrt{1+x} \leq -\frac{x}{3} \iff 1 + \frac{x}{3} \leq \sqrt{1+x} \iff$$
$$1 + \frac{2}{3}x + \frac{x^2}{9} \leq 1 + x \iff \frac{x^2}{9} \leq \frac{x}{3} \iff \frac{x}{3} \leq 1$$

□

In the reminder of this section we will use the following facts

$$\forall x \perp \bar{v}_1, \bar{v}_{n'} \text{ it holds that } \sum_{i\in C'} x_i = 0, \ \sum_{i\in I'} x_i = 0, \tag{8}$$

$$\alpha' = (1 \pm e^{-\Omega(d)})\alpha, \quad n' \geq (1 - e^{-\Omega(d)})n, \tag{9}$$

$$\sqrt{\tfrac{c_0}{d}} \leq \alpha \leq 1/3, \quad d < n^{1/40}, \tag{10}$$

$$\|\bar{v}_{n'}\|_2^2 \geq \alpha' n' \geq (1 - e^{-\Omega(d)})\alpha n, \quad \|\bar{v}_1\|_2^2 \geq n' \geq (1 - e^{-\Omega(d)})n, \tag{11}$$

$$\beta_2 = (1 \pm e^{-\Omega(d)})\bar{\beta}_2, \quad \beta_1 = (1 \pm e^{-\Omega(d)})\bar{\beta}_1 \tag{12}$$

$$|\bar{\beta}_2| \leq \frac{\alpha'}{1-\alpha'} \quad \text{(see the definition of } \bar{\beta}_2 \text{ at (6) and Lemma 4.1 part (i))}, \tag{13}$$

$$|\bar{\beta}_2\sqrt{1-\alpha'}| \leq 1, \ |\beta_2\sqrt{1-\alpha}| \leq 1 \tag{14}$$

Proof: $\beta_2 \leq 0$ thus it is enough to show that $\beta_2\sqrt{1-\alpha} \geq -1$.
$\beta_2\sqrt{1-\alpha} = \frac{1}{2}(\sqrt{1-\alpha} - \sqrt{1-\alpha+4\alpha}) \geq -1 \iff$
$2 + \sqrt{1-\alpha} \geq \sqrt{1+3\alpha} \iff 4 + 4\sqrt{1-\alpha} + 1 - \alpha \geq 1 + 3\alpha \iff$
$4 + 4\sqrt{1-\alpha} \geq 4\alpha$

$$(1-\alpha')\bar{\beta}_2 \leq -0.66\sqrt{\tfrac{c_0}{d}} \tag{15}$$

Proof: $(1-\alpha)\beta_2 = (1-\alpha)\frac{1}{2}(1 - \sqrt{1 + \frac{4\alpha}{1-\alpha}}) \overset{(10)}{\leq}$
$\frac{1}{3}(1 - \sqrt{1 + \frac{4\alpha}{1-\alpha}}) \overset{(4.1) \text{ part (ii)},(10)}{\leq} -\frac{1}{3}\frac{4\alpha}{3(1-\alpha)} \overset{(10)}{\leq} -\frac{2}{3}\sqrt{\frac{c_0}{d}}$.
(15) follows from the last inequality and (9), (12).

Additional notation: we will use $v_1, v_2, .., v_{n'}$ to denote the unit eigenvectors of A' corresponding to the eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{n'}$. The vector of all ones is denoted by $\vec{1}$.

Lemma 4.2. *Let $\bar{v}_{n'}$ be the eigenvector corresponding to the most negative eigenvalue of $\bar{A}'$ (as defined at (7)). Let $v_{n'}$ be the eigenvector corresponding to the most negative eigenvalue of A' normalized such that $\|v_{n'}\| = \|\bar{v}_{n'}\|$ and also $< v_{n'}, \bar{v}_{n'} > \ \geq 0$. With high probability it holds that $\|\bar{v}_{n'} - v_{n'}\|^2 < \frac{1}{800}\|\bar{v}_{n'}\|^2$.*

We first prove Lemma 3.9 and only then give the proof of Lemma 4.2.

Proof of Lemma 3.9. The vector $\bar{v}_{n'}$ equals: $(\underbrace{1, 1, .., 1}_{\alpha' n'}, \underbrace{\bar{\beta}_2, \bar{\beta}_2, .., \bar{\beta}_2}_{(1-\alpha')n'})$ ($\bar{\beta}_2$ is defined at (6)). In this proof we will assume that the eigenvector $v_{n'}$ used in step 1 of *FindIS* is normalized so that $\|v_{n'}\| = \|\bar{v}_{n'}\|$ and also that $< v_{n'}, \bar{v}_{n'} > \geq 0$. We may use this assumptions since the outcome of step 1 does not change if we multiply $v_{n'}$ (used by Step 1 of the algorithm) by any non-zero constant.

Denote by L the $\alpha' n'$ indices of $v_{n'}$ which correspond to the largest values in $v_{n'}$. Similarly, denote by $\bar{L}$ the $\alpha' n'$ indices of $\bar{v}_{n'}$ which correspond to the largest values in $\bar{v}_{n'}$ (hence $\bar{L} = I'$). For every vertex $i' \in L \cap C$ we match a unique vertex $i \in \bar{L} \setminus L$. Note that $v_{n'}(i') \geq v_{n'}(i)$, $\bar{v}_{n'}(i) - \bar{v}_{n'}(i') = 1 - \bar{\beta}_2$. Summing the last two inequalities gives

$$(\bar{v}_{n'}(i) - v_{n'}(i)) + (v_{n'}(i') - \bar{v}_{n'}(i')) \geq 1 - \bar{\beta}_2 \geq 1 \tag{16}$$

and thus

$$(\bar{v}_{n'}(i) - v_{n'}(i))^2 + (v_{n'}(i') - \bar{v}_{n'}(i'))^2 \geq \tfrac{1}{2}. \tag{17}$$

By Lemma 4.2 w.h.p. it holds that $\|\bar{v}_{n'} - v_{n'}\|^2 \leq \frac{1}{800}\|\bar{v}_{n'}\|^2$. It thus follows that

$$\tfrac{1}{2}|L \setminus \bar{L}| \leq \tfrac{1}{800}\|\bar{v}_{n'}\|^2. \tag{18}$$

Option 1: By setting I_1 to contain the indices of the largest αn entries in $v_{n'}$ we get:

$$|I_1 \setminus I| \leq |L \setminus \bar{L}| + |\alpha n - \alpha' n'| \leq \tfrac{1}{400}\|\bar{v}_{n'}\|^2 + |\alpha n - \alpha' n'| \overset{(9)}{\leq}$$
$$\tfrac{\alpha' n' + \bar{\beta}_2^2(1-\alpha')n'}{400} + e^{-\Omega(d)}n \overset{(13)}{\leq} \tfrac{\alpha' n(1+\frac{\alpha'}{1-\alpha'})}{400} + e^{-\Omega(d)}n \overset{(10)}{\leq} \tfrac{1.5\alpha' n'}{400} + e^{-\Omega(d)}n \overset{(9)}{\leq} \tfrac{\alpha n}{200}. \tag{19}$$

In this case the maximum matching in $G[I_1]$ contains at most $2|C' \cap I_1| \leq \frac{\alpha n}{100}$ vertices.
Remark: Note that since $|I| = |I_1|$, (19) implies that $|I \setminus I_1| \leq \frac{\alpha n}{200}$, or in other words: the intersection of I with the indices of the $(1-\alpha)n$ smallest vertices is bounded by $\frac{\alpha n}{200}$.
Option 2: By setting I_1 to contain the indices of the smallest αn entries in $v_{n'}$ we get that $|I_1 \cap I| \leq \frac{\alpha n}{200}$. (Observe that $\alpha < 1/3$ and $n' \simeq n$ imply that the largest αn entries are disjoint from the smallest αn entries, then use the last remark). The maximum independent set in $G[C]$ is bounded by $\frac{2\log dn}{d}$ (Lemma 3.8). It follows that the maximum independent set in $G[I_1]$ contains at most $\frac{\alpha n}{200} + \frac{2\log dn}{d}$ vertices. Since the complement of a maximal matching is an independent set, it follows that every maximal matching in $G[I_1]$ has at least $\frac{199\alpha n}{200} - \frac{2\log dn}{d}$ vertices.

Since step 1 of the algorithm takes the option with the smallest maximum matching (among 1,2), it chooses Option 1. It then follows that

$$|I_1 \triangle I| = 2|I_1 \setminus I| \leq \tfrac{\alpha n}{100} \tag{20}$$

□

Proof of Lemma 4.2. For simplicity, we normalize the vectors so that $\|v_{n'}\| = \|\bar{v}_{n'}\| = 1$. We need to prove that $\|\bar{v}_{n'} - v_{n'}\|^2 < \frac{1}{800}$.

Since the vectors $v_1, v_2, ..., v_{n'}$ are orthogonal, the vector $\bar{v}_{n'}$ can be written as $\sum_{i=1}^{n'} c_i v_i$, such that $\sum_{i=1}^{n'} c_i^2 = 1$. We will use the following properties:
(i) $\|(A' - \bar{\lambda}_{n'} I)\bar{v}_{n'}\| \leq 2\sqrt{d}$ (see Lemma 4.3 part (ii)),
(ii) all the eigenvalues of A' except $\lambda_1, \lambda_{n'}$ are bounded by $2c\sqrt{d}$ in absolute value (Lemma 4.4), where c is some absolute constant independent of α and d
(iii) $\bar{\lambda}_{n'} \leq -0.65\sqrt{c_0 d}$:

$$\text{by (7) second line } \bar{\lambda}_{n'} = (1-\alpha')n'\bar{\beta}_2 p \overset{(15)}{\leq}$$
$$-0.66\sqrt{\tfrac{c_0}{d}}\, n'p \overset{(9)}{\leq} -0.65\sqrt{c_0 d}.$$

(iv) $\lambda_1 \geq 0$ (because the trace of A' is 0).
(v) $c_{n'} =< v_{n'}, \bar{v}_{n'} > \ \geq 0$.

$$4d \overset{(i)}{\geq} \|(A' - \bar{\lambda}_{n'} I)\bar{v}_{n'}\|^2 = \|(A' - \bar{\lambda}_{n'} I)(\sum_{i=1}^{n'} c_i v_i)\|^2 =$$
$$\sum_{i=1}^{n'} (c_i)^2(\lambda_i - \bar{\lambda}_{n'})^2 \geq \sum_{i=1}^{n'-1} (c_i)^2(\lambda_i - \bar{\lambda}_{n'})^2. \tag{21}$$

Facts (iii),(iv) imply that $\lambda_1 - \bar{\lambda}_{n'} \geq 0.65\sqrt{c_0 d}$. Fact (ii) states that $|\lambda_i| \leq 2c\sqrt{d}$ $(1 < i < n')$ for some constant c independent of α, d. Thus by setting $c_0 > (4c)^2$ we have $\lambda_i - \bar{\lambda}_{n'} > 0.65\sqrt{c_0 d} - 2c\sqrt{d} > \sqrt{c_0 d}/5$. By combining the last inequality with (21) we derive

$$\sum_{i=1}^{n'-1} c_i^2 \leq \frac{4}{c_0/25} < \frac{100}{c_0}. \tag{22}$$

Using

$$\|\bar{v}_{n'} - v_{n'}\|^2 = \sum_{i=1}^{n'-1} c_i^2 + (1 - c_{n'})^2, \qquad \sum_{i=1}^{n'} c_i^2 = 1, \quad c_{n'} \geq 0 \tag{23}$$

we derive that for large enough c_0 it holds that $\|\bar{v}_{n'} - v_{n'}\|^2 < \frac{1}{800}$ as needed. □

Lemma 4.3. *Let $\bar{v}_1, \bar{v}_{n'}$ be the first and last eigenvectors of the matrix $\bar{A}'$ with corresponding eigenvalues $\bar{\lambda}_1, \bar{\lambda}_{n'}$. The following inequalities hold with high probability:*
(i) $\|(A' - \bar{\lambda}_1 I)\bar{v}_1\| \leq 2\sqrt{d}\|\bar{v}_1\|$,
(ii) $\|(A' - \bar{\lambda}_{n'} I)\bar{v}_{n'}\| \leq 2\sqrt{d}\|\bar{v}_{n'}\|$,
(iii) $\forall x \perp \bar{v}_1, \bar{v}_{n'}$ $\|A'x\| \leq c\sqrt{d}\|x\|$ *(where $c \geq 2$ is a universal constant independent of d, α).*

Proof of lemma 4.3 part (ii). By (11) $\|\bar{v}_{n'}\| \geq \sqrt{\alpha n}(1 - e^{-\Omega(d)})$. We will prove that $\|(A' - \bar{\lambda}_{n'} I)\bar{v}_{n'}\| < 2\sqrt{d}\|\bar{v}_{n'}\|$. We use (see (7)):

$$\bar{\lambda}_{n'} = (1 - \alpha')\bar{\beta}_2 n' p = (1 - \alpha)\beta d \pm e^{-\Omega(d)} \qquad \bar{v}_{n'} = (\underbrace{1, 1, .., 1}_{\alpha' n'}, \underbrace{\bar{\beta}_2, \bar{\beta}_2, ...}_{(1-\alpha')n'}).$$

$$\|(A' - \bar{\lambda}_{n'} I)\bar{v}_{n'}\| = \left\| \left(A' - \bar{\lambda}_{n'} I\right) \begin{bmatrix} 1 \\ . \\ . \\ \bar{\beta}_2 \\ . \\ . \end{bmatrix} \right\| = \left\| \begin{bmatrix} \bar{\beta}_2 \deg(v)_{C'} - \bar{\lambda}_{n'} \\ . \\ . \\ \deg(v)_{I'} + \bar{\beta}_2 \deg(v)_{C'} - \bar{\lambda}_{n'} \bar{\beta}_2 \\ . \\ . \end{bmatrix} \right\|$$

$$\overset{\beta_2 = \bar{\beta}_2(1 \pm e^{-\Omega(d)})}{\leq} \left\| \begin{bmatrix} \beta_2(\deg(v)_{C'} - (1 - \alpha)d) \\ . \\ . \\ \deg(v)_{I'} + \beta_2 \deg(v)_{C'} - (1 - \alpha)\beta_2^2 d \\ . \\ . \end{bmatrix} \right\| + \underbrace{\left\| \begin{bmatrix} e^{-\Omega(d)} \\ . \\ . \\ e^{-\Omega(d)} \\ . \\ . \end{bmatrix} \right\|}_{\leq e^{-\Omega(d)}\sqrt{n}}.$$

It remains to upper bound the norm of the left vector, we do that by upper bounding its squared norm with $2.3d\|\bar{v}_{n'}\|^2$. We will estimate the random variable:

$$\beta_2^2 \sum_{v \in I'} \left(\deg(v)_{C'} - (1 - \alpha)d\right)^2 + \sum_{v \in C'} (\deg(v)_{I'} + \beta_2 \deg(v)_{C'} - (1 - \alpha)\beta_2^2 d)^2. \tag{24}$$

Note that the difference between the sum in (24) and

$$\underbrace{\beta_2^2 \sum_{v \in I} \left(\deg(v)_C - (1 - \alpha)d\right)^2}_{S_1} + \underbrace{\sum_{v \in C} (\deg(v)_I + \beta_2 \deg(v)_C - (1 - \alpha)\beta_2^2 d)^2}_{S_2}, \tag{25}$$

can be classified into two types:
(i) vertices of $V \setminus V'$ that add summands to (25) which do not exist in (24).
(ii) vertices of $V \cap V'$ for which the degrees $\deg(v)_C, \deg(v)_{C'}$ or $\deg(v)_I, \deg(v)_{I'}$ are different.

It is more convenient to upper bound (25) rather than (24). The differences of type (i) can only make (25) bigger than (24). The difference induced by type (ii) is $O(e^{-\Omega(d)}nd^3)$, the explanation is as follows. For any fixed vertex v of degree $< 5d$ in G, it's summand in S_1 (or in S_2) is bounded by $O(d^3)$ (because $\beta_2 < 5\sqrt{d}$). The number of vertices in V' which have an edge to a vertex in $V \setminus V'$ is bounded by $e^{-\Omega(d)}n$ (since w.h.p. the total number of edges that touch $|V \setminus V'|$ is at most $e^{-\Omega(d)}n$).

We will now prove the following inequalities

$$S_1 \leq 1.1\alpha n(1-\alpha)d, \qquad S_2 \leq 1.1(1-\alpha)n(\alpha d + \beta_2^2(1-\alpha)). \tag{26}$$

We start with S_2. The graph G is taken from $G_{n,p,\alpha}$. We denote by $e(G)$ the number of edges in G. Let L be the set of all edges which do not have both endpoints in I. Denote $l := |L| = \binom{n}{2} - \binom{\alpha n}{2}$. The expectation of $e(G)$ is $\mu := pl$ (where $\mu \geq 0.4dn$ since $\alpha \leq \frac{1}{3}$). By Chernoff's inequality it holds that

$$\Pr_{G_{n,p,\alpha}}[e(G) \in (1 \pm n^{-0.1})\mu] \geq 1 - o(1) \tag{27}$$

(we used here $\mu \geq 0.4dn$). Denote by $G_{n,m,\alpha}$ the uniform distribution over all graphs with m edges such that none of the edges is inside I. For a graph G taken from $G_{n,p,\alpha}$, given that $e(G) = m$, the graph G has the distribution of $G_{n,m,\alpha}$. Thus, it is sufficient to show that for any $m \in (1 \pm n^{-0.1})\mu$ it holds that

$$\Pr_{G_{n,m,\alpha}}[S_2 > 1.1(1-\alpha)n(\alpha d + \beta_2^2(1-\alpha))] = o(1). \tag{28}$$

Fix some $m \in (1 \pm n^{-0.1})\mu$. Technically, it is more convenient to prove a concentration result in a product measure. Denote by $C_{n,m,\alpha}$ the distribution induced by taking a uniformly random m-tuple from L^m. Let G be taken from $C_{n,m,\alpha}$. Note that G is actually a multigraph. Given that G is simple (i.e. no parallel edges), G is distributed as $G_{n,m,\alpha}$. Note that

$$\Pr_{C_{n,m,\alpha}}[G \text{ is simple}] \geq (1 - \tfrac{m}{l})^m \overset{(i)}{\geq} e^{-\frac{2m^2}{l}} \geq e^{-(1+o(1))\frac{1}{2}p^2 l} \overset{l \leq 0.5n^2}{\geq} e^{-0.2d^2} \overset{d < n^{1/40}}{\geq} e^{-0.2n^{0.05}} \tag{29}$$

(in inequality (i) we used $1 - x \geq e^{-2x}$ which holds for $x \in [0, 0.5]$).
We will now prove that

$$\Pr_{C_{n,m,\alpha}}[S_2 > 1.1(1-\alpha)n(\alpha d + \beta_2^2(1-\alpha))] \leq 2e^{-n^{0.1}}. \tag{30}$$

We first calculate the expectation of S_2. Fix a vertex $v \in C$. Denote $X := \deg(v)_I + \beta_2 \deg(v)_C - (1-\alpha)\beta_2^2 d$.

$$\begin{aligned}
\mathbb{E}[X] &= m\,\frac{\alpha n + \beta_2((1-\alpha)n - 1)}{l} - (1-\alpha)\beta_2^2 d \overset{(9)}{=} \\
&= (1 \pm n^{-0.1})\; p\; \left[(1 \pm e^{-\Omega(d)})(\alpha' n' + \bar{\beta}_2(1-\alpha')n')\right] - (1 \pm e^{-\Omega(d)})(1-\alpha')d\bar{\beta}_2^2 \overset{(14)}{=} \\
&(\alpha' n' p + \bar{\beta}_2(1-\alpha')n'p - \underbrace{(1-\alpha')n'\bar{\beta}_2 p}_{\bar{\lambda}_{n'}}\bar{\beta}_2) \pm 4(n^{-0.1} + e^{-\Omega(d)})d \pm e^{-\Omega(d)}d \overset{(5)}{=} \qquad (31) \\
&\pm 5d(n^{-0.1} + e^{-\Omega(d)}) \overset{d < n^{1/40}}{=} \pm(o(1) + e^{-\Omega(d)}). \qquad (32)
\end{aligned}$$

Using (10) a straight forward calculation (see Lemma 4.5 for details) shows that for any $v \in C$ it holds that

$$\mathrm{VAR}[X] = (1 + o(1))(\alpha d + \beta_2^2(1-\alpha)d). \tag{33}$$

Thus for large enough d

$$\mathbb{E}[X^2] = \mathrm{VAR}[X] + (\mathbb{E}[X])^2 = (1 \pm 0.01)(\alpha d + \beta_2^2(1-\alpha)d). \tag{34}$$

Summing over all C we derive

$$\mathbb{E}[S_2] = (1 \pm 0.01)(1-\alpha)n(\alpha d + \beta_2^2(1-\alpha)d). \tag{35}$$

We will now show that w.h.p. S_2 is tightly concentrated around its expectation. Recall that G is taken from $C_{n,m,\alpha}$ for some fixed $m \in (1 \pm n^{-0.1})\mu$. By Chernoff's inequality it holds that

$$\Pr[\text{all vertices degrees in } G \text{ are bounded by } dn^{0.1}] \geq 1 - e^{-n^{0.1}} \tag{36}$$

(the expected degree of a vertex in C is $\frac{n}{l}m = \frac{n}{l}(1 \pm n^{-0.1})pl = d(1 \pm n^{-0.1})$). Thus, with probability of at least $1 - e^{-n^{0.1}}$ every summand of S_2 is bounded by $3d^2n^{0.2}$ (we use here $|\beta_2| \leq 1$, (14)). Consider the function

$$f(v) := \min\left\{(\deg(v)_I + \beta_2 \deg(v)_C - (1-\alpha)\beta_2^2 d)^2, 3d^2n^{0.2}\right\}. \tag{37}$$

Let $f(G) := \sum_{v \in C} f(v)$. Adding/removing a single edge from G changes $f(G)$ by at most $6d^2n^{0.2}$ since all the summands of $f(G)$ have values in $[0, 3d^2n^{0.2}]$. Since G is taken from $C_{n,m,\alpha}$ which is a product measure (L^m), we can use Azuma's inequality

$$\Pr[f(G) - \mathbb{E}[f(G)] > \lambda] < 2e^{-\lambda^2/2m(6d^2n^{0.2})^2}. \tag{38}$$

Since for any G it holds that $f(G) \leq S_2$ we derive

$$\Pr[f(G) - \mathbb{E}[S_2] > \lambda] < 2e^{-\lambda^2/2m(2d^4n^{0.2})^2}, \tag{39}$$

setting $\lambda = 0.01\mathbb{E}[S_2]$ and using $m \in (1 \pm n^{-0.1})\mu$, $\mu = pl \leq dn$, $\mathbb{E}[S_2] \overset{(35)}{\geq} \frac{n}{4}$ we derive

$$\Pr[f(G) \leq 1.01\mathbb{E}[S_2]] \geq 1 - 2e^{-\Omega(n^2/dnd^4n^{0.4})} \overset{d<n^{1/40}}{\geq} 1 - 2e^{-\Omega(n^{0.475})}. \tag{40}$$

Note that if all the degrees in G are bounded by $dn^{0.1}$ then $f(G) = S_2$. Inequalities (36), (40) imply that

$$\Pr_{C_{n,m,\alpha}}[S_2 > 1.01\mathbb{E}[S_2]] \leq 2e^{-n^{0.1}}. \tag{41}$$

Combining (29) with (41) we derive

$$\Pr_{C_{n,m,\alpha}}[S_2 \geq 1.01\mathbb{E}[S_2] \mid \text{G is simple}] \leq \frac{2e^{-n^{0.1}}}{\Pr[\text{G is simple}]} \leq \frac{2e^{-n^{0.1}}}{e^{-0.2n^{0.05}}} = o(1), \tag{42}$$

which together with (35) proves (28) and thus also the second part of (26). The proof of the first part of (26) is similar, we omit the details. Having (26) proved we derive

$$\beta_2^2 S_1 + S_2 \overset{(26)}{\leq} \beta_2^2 1.1\alpha n(1-\alpha)d + 1.1(1-\alpha)n(\alpha d + \beta_2^2(1-\alpha)) \overset{(14)}{\leq} \tag{43}$$

$$1.1\alpha dn + 1.1(1-\alpha)n(\alpha d + \beta_2^2(1-\alpha)d) \leq \tag{44}$$

$$2.2\alpha dn + 1.1(1-\alpha)n\beta_2^2 d \leq 2.2d(\alpha n + (1-\alpha)n\beta_2^2) \leq \tag{45}$$

$$2.2d(1 + e^{-\Omega(d)})(\alpha' n' + (1-\alpha')n'\beta_2^2) \leq 2.3d\|\bar{v}_{n'}\|_2^2. \tag{46}$$

□

Proof of lemma 4.3 part (i). By (11) $\|\bar{v}_1\| \geq \sqrt{n}(1-e^{-\Omega(d)})$. We will prove that $\|(A'-\bar{\lambda}_1 I)\bar{v}_1\| < 2\sqrt{d}\|\bar{v}_1\|$. We use (from (7)):

$$\bar{\lambda}_1 = (1-\alpha')\bar{\beta}_1 n' p = (1-\alpha)\beta_1 d \pm e^{-\Omega(d)} \qquad \bar{v}_1 = (\underbrace{1,1,..,1}_{\alpha' n'}, \underbrace{\bar{\beta}_1, \bar{\beta}_1, ...}_{(1-\alpha')n'}).$$

$$\|(A'-\bar{\lambda}_1 I)\bar{v}_1\| = \left\| \left(A'-\bar{\lambda}_1 I\right) \begin{bmatrix} 1 \\ . \\ . \\ \bar{\beta}_1 \\ . \\ . \end{bmatrix} \right\| = \left\| \begin{bmatrix} \bar{\beta}_1 \deg(v)_{C'} - \bar{\lambda}_1 \\ . \\ . \\ \deg(v)_{I'} + \bar{\beta}_1 \deg(v)_{C'} - \bar{\lambda}_1 \bar{\beta}_1 \\ . \\ . \end{bmatrix} \right\|$$

$$\overset{\beta_1 = \bar{\beta}_1(1\pm e^{-\Omega(d)})}{\leq} \left\| \begin{bmatrix} \beta_1 \left(\deg(v)_{C'} - (1-\alpha)d \right) \\ . \\ . \\ \deg(v)_{I'} + \beta_1 \deg(v)_{C'} - (1-\alpha)\beta_1^2 d \\ . \\ . \end{bmatrix} \right\| + \underbrace{\left\| \begin{bmatrix} e^{-\Omega(d)} \\ . \\ . \\ e^{-\Omega(d)} \\ . \\ . \end{bmatrix} \right\|}_{\leq e^{-\Omega(d)}\sqrt{n}}.$$

It remains to upper bound the norm of the left vector, we do that by upper bounding its squared norm with $2.3d\|\bar{v}_1\|_2^2$. The squared norm of the left vector is

$$\beta_1^2 \sum_{v \in I'} \left(\deg(v)_{C'} - (1-\alpha)d\right)^2 + \sum_{v \in C'} \left(\deg(v)_{I'} + \beta_1 \deg(v)_{C'} - (1-\alpha)\beta_1^2 d\right)^2. \tag{47}$$

An argument similar to the one used in the proof of part (ii) shows that w.h.p. the last term is bounded by $2.3\|\bar{v}_1\|^2$. □

Proof of lemma 4.3 part (iii). If $x \perp \bar{v}_1,\ x \perp \bar{v}_{n'}$, then $\sum_{i\in I'} x_i = 0,\ \sum_{i\in C'} x_i = 0$ (by (8)). Thus, for such x the following holds:

$$x^t A' x \quad = \quad x^t \left(A' + \overbrace{\begin{array}{|cccc|cccc} p & . & . & p & 0 & . & 0 & 0 \\ . & . & & & & & & \\ . & & . & & & & & \\ p & . & . & p & & & & \\ \hline 0 & & & & 0 & & & \\ . & & & & & . & & \\ 0 & & & & & & 0 & \\ 0 & & & & & & & 0 \end{array}}^{I'} \right) x \quad = x^t B' x,$$

where B' is derived from A by removing vertices of degree $> 5d$ and adding the value $p = \frac{d}{n}$ in entries of the submatrix which corresponds to $I' \subset I$. This matrix B' is very similar to the matrices analysed in [10]. The difference is that in [10] the whole matrix is random whereas in our case, a small (about $\alpha n \times \alpha n$) portion of the matrix is deterministically fixed to be p (the expectation). One would expect that having this fixed portion in the matrix would make the eigenvalue structure more similar to that of the all p matrix (a similarity that we wish to establish here) . Indeed, a simple modification of the arguments in [10] (in Sections 2.2.3, 2.2.4, 3.2, 3.3) yields that w.h.p. for any $x \perp \vec{1}$ it holds $x^t B' x \leq c\sqrt{d}$ where c is a universal constant independent of d and α (without loss of generality, we may further assume that $c \geq 2$). We omit the proof from the current manuscript, and refer the reader to [10]. □

Lemma 4.4. *Let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{n'}$ be the eigenvalues of A'. If conditions (i),(ii),(iii) from Lemma 4.3 hold then for $i = 2, 3, .., n' - 1$ it holds that $|\lambda_i| \leq 2c\sqrt{d}$ ($c \geq 2$ is the universal constant from Lemma 4.3).*

Proof. It is enough to prove that $\lambda_2 \leq 2c\sqrt{d}$ and $\lambda_{n'-1} \geq -2c\sqrt{d}$. We first show $\lambda_{n'-1} \geq -2c\sqrt{d}$. It is well known that:

$$\lambda_{n'-1} = \max_{\substack{H \text{ subspace of} \\ \text{dimension } n-1}} \min_{\substack{x\neq 0, \\ x\in H}} \frac{x^t A' x}{x^t x}.$$

Let us fix H to be the subspace perpendicular to $\bar{v}_{n'}$. Consider any vector $x \perp \bar{v}_{n'}$. The vector x can be written as $x = f + s$ where f is a multiple of $\bar{v}_1$ and $s \perp \bar{v}_1, \bar{v}_{n'}$.

$$x^t A' x = (f+s)^t A'(f+s) = f^t A' f + s^t A' s + 2 s^t A' f \geq$$

$$\bar{\lambda}_1 \|f\|^2 - c\sqrt{d}\|s\|^2 + 2 s^t (A' - \bar{\lambda}_1 I) f \geq -c\sqrt{d}\|s\|^2 - 2(1.2\sqrt{d}\|s\|\,\|f\|) \geq -2c\sqrt{d}\|x\|^2.$$

In the first equality we used the symmetry of A', in the first inequality we used Lemma 4.3 (iii) for the second term and $s \perp f$ for the third term, in the following inequality we used $\bar{\lambda}_1 \geq 0$ and Lemma 4.3 (i), and in the last inequality we used $c > 1.2$, $2\|s\|\,\|f\| \leq \|s\|^2 + \|f\|^2 = \|x\|^2$. A similar argument using

$$\lambda_2 = \min_{\substack{H \text{ subspace of} \\ \text{dimension } n-1}} \max_{\substack{x\in H \\ x\neq 0}} \frac{x^t A' x}{x^t x},$$

gives $\lambda_2 \leq 2c\sqrt{d}$. □

Lemma 4.5. *Fix $l = \binom{n}{2} - \binom{\alpha n}{2}$ and let G be taken from $C_{n,m,\alpha}$ where $m \in (1 \pm n^{-0.1})pl$. Fix any $v \in C$ and define $X := deg(v)_I + \beta_2 deg(v)_C - (1-\alpha)\beta_2^2 d$. It holds that $VAR[X] = (1 + o(1))(\alpha d + \beta_2^2(1-\alpha)d)$.*

Proof. The r.v. X can be written as the sum of m independent random variables $X_1, ..., X_m$, where $X_i := \deg(v)_I + \beta\deg(v)_C - (1-\alpha)\beta^2 d$ when G is composed of only one random edge from L. It holds that $VAR[X] = m\text{VAR}[X_1]$. We will now calculate

$$\text{VAR}[X_1] = \text{VAR}[\deg(v)_I] + \beta_2^2\text{VAR}[\deg(v)_C] + 2\beta_2\text{COV}(\deg(v)_I, \deg(v)_C). \tag{48}$$

Since $\deg(v)_I, \deg(v)_C$ are in fact indicator we will denote them respectively by z_1, z_2 where $\Pr[z_1] = p_1, \Pr[z_2] = p_2$.

$$\text{VAR}[\deg(v)_I] = p_1\ (1-p_1) = \tfrac{\alpha n}{l}(1-\tfrac{\alpha n}{l}) \overset{(10)}{=} (1 \pm o(1))\tfrac{\alpha n}{l}, \tag{49}$$

$$\text{VAR}[\deg(v)_C] = p_2(1-p_2) = \tfrac{(1-\alpha)n-1}{l}(1-\tfrac{(1-\alpha)n-1}{l}) \overset{(10)}{=} (1 \pm o(1))\tfrac{(1-\alpha)n}{l}, \tag{50}$$

$$\beta_2\text{COV}[\deg(v)_I, \deg(v)_C] = \beta_2(\Pr[z_1 \wedge z_2] - p_1p_2) = \beta_2(0 - p_1p_2) = -\beta_2(1 \pm o(1))\tfrac{\alpha(1-\alpha)n^2}{l^2} \overset{(10)}{=} o(1). \tag{51}$$

It thus follows that

$$\text{VAR}[X] = (1 \pm o(1))\tfrac{mn}{l}\left(\alpha + \beta_2^2(1-\alpha)\right) = (1 \pm o(1))d\left(\alpha + \beta_2^2(1-\alpha)\right) \tag{52}$$

□

Acknowledgements

This work was supported in part by a grant from the G.I.F., the German-Israeli Foundation for Scientific Research and Development. Part of this work was done while the authors were visiting Microsoft Research in Redmond, Washington.

References

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

[2] N. Alon, M. Krivelevich, and B. Sudakov. Finding a large hidden clique in a random graph. *Random Structures and Algorithms*, 13(3-4):457–466, 1988.

[3] N. Alon and J. Spencer. *The Probabilistic Method*. John Wiley and Sons, 2002.

[4] H. Chen and A. Frieze. Coloring bipartite hypergraphs. In *Proceedings of the 5th International Conference on Integer Programming and Combinatorial Optimization*, 345–358, 1996.

[5] A. Coja-Oghlan. Finding large independent sets in polynomial expected time. *Combinatorics, Probability and Computing*, 15(5):731–751, 2006.

[6] A. Coja-Oghlan. A spectral heuristic for bisecting random graphs. *Random Structures and Algorithms*, 29(3):351–389, 2006.

[7] U. Feige. Approximating maximum clique by removing subgraphs. *Siam J. on Discrete Math.*, 18(2):219–225, 2004.

[8] U. Feige and J. Kilian. Heuristics for semirandom graph problems. *Journal of Computing and System Sciences*, 63(4):639–671, 2001.

[9] U. Feige and R. Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures and Algorithms*, 16(2):195–208, 2000.

[10] U. Feige and E. Ofek. Spectral techniques applied to sparse random graphs. *Random Structures and Algorithms*, 27(2):251–275, 2005.

[11] U. Feige and E. Ofek. Finding a maximum independent set in a sparse random graph. In proceedings of *9th International Workshop on Randomization and Computation, RANDOM 2005, SPRINGER LNCS 3624*, 282–293, 2005.

[12] A. Flaxman. A spectral technique for random satisfiable 3cnf formulas. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 357–363, 2003.

[13] A. Goerdt and A. Lanka. On the hardness and easiness of random 4-sat formulas. In *Proceedings of the 15th International Symposium on Algorithms and Computation (ISAAC)*, pages 470–483, 2004.

[14] G. Grimmet and C. McDiarmid. On colouring random graphs. *Math. Proc. Cam. Phil. Soc.*, 77:313–324, 1975.

[15] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.

[16] M. Jerrum. Large clique elude the metropolis process. *Random Structures and Algorithms*, 3(4):347–359, 1992.

[17] R. M. Karp. The probabilistic analysis of some combinatorial search algorithms. In J. F. Traub, editor, *Algorithms and Complexity: New Directions and Recent Results*, pages 1–19. Academic Press, New York, 1976.

[18] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W.Thatcher, editors, *Complexity of Computer Computations*, pages 85–104. Plenum Press, New York, 1972.

[19] L. Kučera. Expected complexity of graph partitioning problems. *Discrete Appl. Math.*, 57(2-3):193–212, 1995.