Approximating Independent Set in Perturbed Graphs

Bodo Manthey[a], Kai Plociennik[b,1]

[a] *University of Twente, Department of Applied Mathematics, P. O. Box 217, 7500 AE Enschede, The Netherlands*
[b] *Fraunhofer Institute for Industrial Mathematics ITWM, Department "Optimization", Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany*

Abstract

For the maximum independent set problem, strong inapproximability bounds for worst-case efficient algorithms exist. We give a deterministic algorithm beating these bounds, with polynomial expected running-time for semi-random graphs: An adversary chooses a graph with n vertices, and then edges are flipped with a probability of ε. Our algorithm guarantees an approximation ratio of $O(\sqrt{n\varepsilon})$ for sufficiently large ε.

Keywords: independent set, approximation algorithms, smoothed analysis

1. Introduction and Results

Given an undirected graph $G = (V, E)$, Independent Set asks to find a largest *independent set* $I \subseteq V$, where I is independent if no edge in E connects two vertices of I. The size of a largest independent set in G is its *independence number* $\alpha(G)$. Throughout this paper, $n = |V|$.

Since Independent Set is NP-hard [6, GT20], *worst-case polynomial-time* algorithms that compute optimal solutions are unlikely to exist. Hence, approximation algorithms have been studied extensively. The *approximation ratio* of an independent set I in a graph G is $\alpha(G)/|I|$. An algorithm has *approximation ratio* f if it computes a solution I with approximation ratio at most $f(n)$ for any $n \in \mathbb{N}$ and any graph on n vertices.

Email addresses: b.manthey@utwente.nl (Bodo Manthey), kplo@hrz.tu-chemnitz.de (Kai Plociennik)

[1] Work done at Chemnitz University of Technology, Department of Computer Science, Chair of Theoretical Computer Science and Information Security.

To our knowledge, the best known worst-case efficient algorithm has approximation guarantee $O(n \cdot (\log\log n)^2/(\log n)^3)$ [3]. Unfortunately, this is not much better than the trivially achievable approximation guarantee n, which can be obtained by outputting a single vertex. Even worse, it is unlikely that this can be improved considerably by worst-case efficient algorithms: Unless P = NP, there is no polynomial-time approximation algorithm with approximation ratio $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [11].

However, one often observes that there are algorithms that compute reasonably good solutions quickly in practice. One way to explain this is *average-case analysis*, where performance is measured in terms of fully random instances. However, average-case analysis is dominated by random instances, and random instances usually have very special properties that distinguish them from real-world instances. Thus, an average-case analysis might be inconclusive.

To overcome this, Spielman and Teng [8] have introduced *smoothed analysis*: A malicious adversary, trying to make the algorithm perform poorly, chooses an arbitrary input. Then, this input is subject to a small random perturbation. If, regardless of the adversary's choice, the expected performance is good, then this explains the good observed performance: Although bad instances exist, one must be very unlucky to accidentally get one.

1.1. Our Results

We perform a probabilistic analysis of the approximability of Independent Set. The probabilistic model that we use is the *smoothed extension of* $G(n,p)$ proposed by Spielman and Teng [9]: Given a graph $G = (V,E)$, we obtain a random graph $\mathcal{G} = (V,\mathcal{E})$ with the same vertex set by negating the existence of any edge independently with a probability of $\varepsilon > 0$. Formally, each potential edge e is contained in the random edge set $\mathcal{E}$ with a probability of

$$p_e = \begin{cases} 1-\varepsilon & \text{if } e \in E \text{ and} \\ \varepsilon & \text{if } e \notin E. \end{cases}$$

We denote the resulting probability distribution by $\mathcal{G}(G,\varepsilon)$. The special case of $E = \emptyset$ is the classical $G(n,\varepsilon)$ model. In the extreme case $\varepsilon = 0$, we have $\mathcal{G} = G$ and the adversary has full power. For increasing ε, the adversary loses power. For $\varepsilon = 1/2$, the adversary has no influence, and we have a $G(n,1/2)$ graph. (For larger ε, the adversary gains influence again, but, because of symmetry, we exclude the case $\varepsilon > 1/2$.) Thus, the value of ε determines

the "amount of randomness" in $\mathcal{G}$. Note that our algorithm needs not only the perturbed graph, but also the original, unperturbed graph as input. A different view on this is that the algorithm has an estimate whether an edge is likely or unlikely to be present in the perturbed graph.

In the analysis of our algorithm, we distinguish between large and small flip probabilities ε: We say that ε is *G-high* if $\frac{\ln(1/\varepsilon)}{\varepsilon} \leq \frac{n^2}{|E|}$. Otherwise, ε is called *G-low*. Asymptotically, ε is G-high if $\varepsilon = \Omega((|E|/n^2)\log(n^2/|E|))$. For sparse graphs with $|E| = \Theta(n)$, this is equivalent to $\varepsilon = \Omega((\log n)/n)$. The algorithm `Approx-IS`, which we are going to analyze, is described in Algorithm 1.

Theorem 1. *Let $G = (V, E)$ be a graph and $\varepsilon = \varepsilon(n)$ with $\sqrt{1/n} \leq \varepsilon \leq 1/2$. Let $\mathcal{G}$ be drawn from $\mathcal{G}(G, \varepsilon)$. Then `Approx-IS`$(\mathcal{G}, G, \varepsilon)$ has polynomial expected running-time. If ε is G-high, it has approximation guarantee $O(\sqrt{n\varepsilon})$. If ε is G-low, the approximation guarantee is $O\big(\frac{|E|\log(1/\varepsilon)}{n^{3/2}\sqrt{\varepsilon}}\big)$.*

Our algorithm `Approx-IS` and parts of its analysis are based on techniques by Krivelevich and Vu [7]. For the $G(n, p)$ model, with $n^{-1/2+\delta} \leq p \leq 1/2$ ($\delta > 0$ is arbitrary but fixed), they have presented an algorithm with polynomial expected running-time and approximation guarantee $O(\sqrt{n\varepsilon}/\log n)$. Theorem 1 extends this from $G(n, \varepsilon)$ to $\mathcal{G}(G, \varepsilon)$. It slightly enlarges the range of ε from $\varepsilon \geq n^{-1/2+\delta}$ to $\varepsilon \geq n^{-1/2}$, while slightly worsening the approximation guarantee by a factor of $\log n$ if ε is G-high. If ε is G-high, then we have an approximation ratio of $O(\sqrt{n\varepsilon})$. If ε is G-low, then the approximation guarantee gets worse since the adversary gains more influence.

In our algorithm, we use a well-known greedy coloring algorithm as a subroutine. Given a graph $G = (V, E)$, a *coloring* is a partition $C = \{C_1, \ldots, C_k\}$ of V into disjoint classes C_i such that all C_i are independent sets. From now on, we assume that $V = \{1, 2, \ldots, n\}$. `GreedyColoring` computes a coloring of G as follows: We set $C_1 = \{1\}$, $\chi = 1$, and $C = \{C_1\}$. Then, we consider the vertices $v = 2, \ldots, n$ one by one. If there is an index $1 \leq i \leq \chi$ such that $C_i \cup \{v\}$ is independent, we set $C_i := C_i \cup \{v\}$ for the smallest such i. Otherwise, we set $\chi := \chi + 1$, let $C_\chi = \{v\}$, and let $C := C \cup \{C_\chi\}$. Given a graph G, the *greedy independent set* $\mathrm{gis}(G)$ is a largest color class in the greedy coloring C.

Theorem 2. *Fix $\delta > 0$. Let $G = (V, E)$ be any graph, and let $\varepsilon = \varepsilon(n)$ with $n^{-1+\delta} \leq \varepsilon \leq 1/2$. Let $\mathcal{G}$ be drawn from $\mathcal{G}(G, \varepsilon)$. Then the expected*

approximation ratio of the greedy algorithm for $\mathcal{G}$ drawn from $\mathcal{G}(G,\varepsilon)$ is $O(1)$ if ε is G-high and $O\big(\frac{|E|\log(1/\varepsilon)}{n^2\varepsilon}\big)$ if ε is G-low.

The goal of this paper is to prove these theorems. We implicitly assume n to be sufficiently large whenever necessary. In the following, we log denotes the logarithm to base 2.

2. Proofs of the Theorems

Let $\mathcal{G}$ be a graph drawn from $\mathcal{G}(G,\varepsilon)$. Our algorithm `Approx-IS` (see page 11) checks whether the greedy independent set $\mathrm{gis}(\mathcal{G})$ has the desired approximation ratio. To do this, it checks whether $\mathrm{gis}(\mathcal{G})$ is large enough and whether the independence number $\alpha(\mathcal{G})$ is small enough. In the analysis, we use two corresponding tail bounds, which we state and prove next. `Approx-IS` is analyzed in Section 2.3. After that, we prove Theorem 2.

2.1. A Tail Bound on the Greedy Independent Set Size

Lemma 3 states that the greedy independent set $\mathrm{gis}(\mathcal{G})$ (a largest color class in the greedy coloring of $\mathcal{G}$) is sufficiently large with high probability. We define the threshold t_{gis}. For a graph $G=(V,E)$ and $\varepsilon,\delta>0$, let

$$t_{\mathrm{gis}}(G,\varepsilon)=\frac{\delta}{16}\cdot\min\left\{\frac{\ln n}{\varepsilon},\frac{n^2\ln n}{|E|\ln(1/\varepsilon)}\right\}.$$

We assume $\delta>0$ to be small and fixed and thus omit it as a parameter. By the definition of G-low and G-high, $t_{\mathrm{gis}}(G,\varepsilon)=\Omega\big(\frac{\log n}{\varepsilon}\big)$ if ε is G-high and $t_{\mathrm{gis}}(G,\varepsilon)=\Omega\big(\frac{n^2\log n}{|E|\log(1/\varepsilon)}\big)$ if ε is G-low. Krivelevich and Vu [7] proved a lemma similar to the below Lemma 3 for $G(n,p)$. Our proof is based on the same technique.

Lemma 3. *Fix $\delta\in(0,1)$. For any graph $G=(V,E)$ and any flip probability $\varepsilon=\varepsilon(n)$ with $n^{-1+\delta}\le\varepsilon\le 1/2$, we have*

$$\Pr\left[|\mathrm{gis}(\mathcal{G})|<t_{\mathrm{gis}}(G,\varepsilon)\right]\le e^{-n\ln n}.$$

Proof. For brevity, let $s=t_{\mathrm{gis}}(G,\varepsilon)$, and let $r=n/(2s)$. We call a set $D=\{D_1,\dots,D_r\}$ of r disjoint independent sets $D_i\subseteq V$ with $|D_i|\le s$ for all D_i a *partial r-coloring*. Let $\overline{D}=V\setminus(D_1\cup\ldots\cup D_r)$. We call D *bad* if every vertex $v\in\overline{D}$ is connected to all classes $D_1,\dots,D_r$.

Let C be the greedy coloring of $\mathcal{G}$. Assume that our bad event "$|\mathrm{gis}(\mathcal{G})| < s$" happens. Then all color classes in C are smaller than s. Thus, there are at least $n/s > r$ color classes in C. Let $C^* = \{C_1, \ldots, C_r\}$ contain the first r color classes of C. C^* is a partial r-coloring. Furthermore, C^* is bad since otherwise some vertex $v \in \overline{C^*}$ is inserted into a class $C_i \in C^*$ by `GreedyColoring`. Thus, $\Pr[|\mathrm{gis}(\mathcal{G})| < s] \leq \Pr[\text{there is a bad partial } r\text{-coloring}]$.

We fix an arbitrary partial r-coloring $D = \{D_1, \ldots, D_r\}$ and estimate $\Pr[D \text{ is bad}]$. We have $|D_1 \cup \ldots \cup D_r| \leq rs = n/2$. Thus, $|\overline{D}| \geq n/2$. For a vertex $v \in \overline{D}$ and a class D_i, let $n_{v,i}$ be the number of vertices $w \in D_i$ such that the edge $\{v, w\}$ is contained in the original (unperturbed) edge set E of G. The number of vertices in D_i to which v is not adjacent in G is $|D_i| - n_{v,i}$. Fix a vertex v and a class D_i. Then the probability that the random $\mathcal{G}$ contains an edge that connects v to some vertex in D_i is $1 - (1-\varepsilon)^{|D_i| - n_{v,i}} \varepsilon^{n_{v,i}}$. Let $f(x) = (1-\varepsilon)^{s-x}\varepsilon^x$ for short. Together with $1 - x \leq e^{-x}$ for $x \in \mathbb{R}$ and $|D_i| \leq s$ for all D_i, we get

$$\Pr\big[D \text{ is bad}\big] \leq \prod_{v \in \overline{D}} \prod_{i=1}^{r} \big(1 - (1-\varepsilon)^{|D_i| - n_{v,i}} \varepsilon^{n_{v,i}}\big) \leq \exp\left(-\sum_{v \in \overline{D}} \sum_{i=1}^{r} f(n_{v,i})\right).$$

Without loss of generality, we assume $|\overline{D}| = n/2$. Let $\hat{N} = \sum_{v,i} n_{v,i} \leq |E|$. Since $f(x)$ is convex, Jensen's inequality, the fact that f is monotonically decreasing, and the fact that the number of terms in the sum equals $rn/2$ yield

$$\frac{2}{rn} \cdot \sum_{v,i} f(n_{v,i}) \geq f\left(\frac{2}{rn} \cdot \sum_{v,i} n_{v,i}\right) = f\left(\frac{2\hat{N}}{rn}\right) \geq f\left(\frac{2|E|}{rn}\right).$$

Thus, we get

$$\Pr\big[D \text{ is bad}\big] \leq \exp\left(-\frac{rn}{2} \cdot (1-\varepsilon)^{s - 2|E|/(rn)} \varepsilon^{2|E|/(rn)}\right). \tag{1}$$

Now we show that the absolute value of the exponent in (1) is at least $2n \ln n$. For brevity, let $a = (1-\varepsilon)^{s-2|E|/(rn)}$ and $b = \varepsilon^{2|E|/(rn)}$. Then this is equivalent to $rab \geq 4 \ln n$ or $\ln r + \ln a + \ln b \geq \ln(4 \ln n)$. Since $s = t_{\mathrm{gis}}(G, \varepsilon) \leq \frac{\delta \ln n}{16\varepsilon}$ and $\varepsilon \geq n^{-(1-\delta)}$, we get $s \leq \frac{\delta n^{1-\delta} \ln n}{16}$. This yields

$$\ln r = \ln\left(\frac{n}{2s}\right) \geq \ln\left(\frac{8n^\delta}{\delta \ln n}\right) = \delta \ln n - o(\ln n) \geq \frac{\delta}{2} \cdot \ln n. \tag{2}$$

With $\ln a \geq s \ln(1-\varepsilon)$ and $s \leq \frac{\delta \ln n}{16\varepsilon}$ and $1 - x \geq e^{-2x}$ for $x \in [0, 1/2]$, we get

$$\ln a \geq -2\varepsilon s \geq -(\delta/8) \ln n. \tag{3}$$

From $\ln b = \frac{2|E|}{rn} \cdot \ln \varepsilon$ and $s = t_{\text{gis}}(G, \varepsilon) \leq \frac{\delta n^2 \ln n}{16|E| \ln(1/\varepsilon)}$ and $r = n/(2s)$, we get

$$\ln b = \frac{4|E|s}{n^2} \cdot \ln \varepsilon \geq \frac{4|E| \ln \varepsilon \cdot \delta n^2 \ln n}{16|E| \ln(1/\varepsilon) \cdot n^2} = -\frac{\delta}{4} \cdot \ln n. \tag{4}$$

Finally, (2), (3), and (4) lead to $(\ln r)+(\ln a)+(\ln b) \geq (\delta/2-\delta/8-\delta/4) \ln n \geq \ln(4 \ln n)$, which proves $\Pr[D \text{ is bad}] \leq e^{-2n \ln n}$ for a fixed partial r-coloring. The number of choices for one color class of a partial r-coloring is bounded by n^{s+1}. Thus, the number of partial r-colorings is at most $n^{(s+1)r} \leq n^n = \exp(n \ln n)$. A union bound over all partial r-colorings D combined with $\Pr[D \text{ is bad}] \leq e^{-2n \ln n}$ for any fixed D completes the proof. □

2.2. A Tail Bound on the Independence Number

Now we analyze how to certify that the independence number $\alpha(\mathcal{G})$ is small. It is an adaption of Krivelevich and Vu's method in their algorithm for $G(n, p)$ [7]. The idea is as follows: Denote by $\lambda_1(A)$ the largest eigenvalue of a suitable real, symmetric matrix $A = A(\mathcal{G}, G, \varepsilon)$. Then we compute $\lambda_1(A(\mathcal{G}))$. Lemma 4 states that always $\alpha(\mathcal{G}) \leq \lambda_1(A)$, and that $\lambda_1(A)$ is sufficiently small with high probability .

Let $G = (V, E)$ be a graph, $\varepsilon > 0$ be a flip probability, and $\mathcal{G} = (V, \mathcal{E})$ be drawn from $\mathcal{G}(G, \varepsilon)$. Remember that p_e is the probability that a potential edge e is contained in $\mathcal{E}$ ($p_e = \varepsilon$ if $e \notin E$ and $p_e = 1 - \varepsilon$ if $e \in E$). Let $A(\mathcal{G}, G, \varepsilon) = (a_{ij})_{1 \leq i,j \leq n}$ be the $n \times n$ matrix given by

$$a_{ij} = \begin{cases} 1 & \text{if } e = \{i, j\} \notin \mathcal{E} \text{ and} \\ -(1 - p_e)/p_e & \text{if } e = \{i, j\} \in \mathcal{E}. \end{cases}$$

In particular, we have $a_{ii} = 1$ for all i, because our graphs do not contain loops.

Note that a_{ij} depends on whether $e = \{i, j\} \in \mathcal{E}$ and whether $e \in E$. The matrix A is a canonical extension of the matrix used by Krivelevich and Vu [7] to handle two different edge probabilities.

Lemma 4. *Fix a graph G and $\varepsilon = \varepsilon(n) \leq 1/2$ with $\varepsilon = \Omega((\log n)^2/n)$. Let $A = A(\mathcal{G}, G, \varepsilon)$. Then always $\alpha(\mathcal{G}) \leq \lambda_1(A)$. Furthermore,*

$$\mathrm{E}[\lambda_1(A)] \leq 2^7 \cdot (\log n) \cdot \sqrt{n/\varepsilon} \tag{5}$$

and

$$\Pr\left[\lambda_1(A) \geq 2^8 \cdot (\log n) \cdot \sqrt{n/\varepsilon}\right] \leq 4 \cdot \exp(-2^9 \cdot n\varepsilon \cdot (\log n)^2). \qquad (6)$$

Throughout the rest of Section 2.2, we prove Lemma 4.

The claim that we always have $\alpha(\mathcal{G}) \leq \lambda_1(A(\mathcal{G}))$ follows immediately from Krivelevich and Vu [7, Lemma 2.4]. They have proved a similar result for $G(n,p)$, for which they used a matrix with A with entry $a_{ij} = 1$ for non-edges and $a_{ij} = -(1-p)/p$ if i and j are connected. This corresponds to our setting if the adversary chooses the empty graph and $p = \varepsilon$.

In $A = A(\mathcal{G}, G, \varepsilon)$, an entry corresponding to a non-edge has a value of 1. Since the corresponding proof of Krivelevich and Vu [7] for their matrix does not depend on the values of the other entries, we have $\alpha(\mathcal{G}) \leq \lambda_1(A)$.

It remains to prove (5) and (6). Krivelevich and Vu [7, Lemma 2.3] have proved their counterpart for $G(n,p)$ using the matrix described above as follows: Füredi and Komlós [5] have bounded the expected value of the largest eigenvalue $\lambda_1(M)$ of the matrix M used by Krivelevich and Vu. Then a tail bound similar to (6) is proved by estimating the probability that $\lambda_1(M)$ deviates significantly from $\mathrm{E}[\lambda_1(M)]$. We first have to bound $\mathrm{E}[\lambda_1(A)]$ from above, which will give us (5) (Section 2.2.1). Then we prove (6) by the large deviation technique [7] (Section 2.2.2).

2.2.1. The Expectation of the Largest Eigenvalue

The *trace* of a matrix $A \in \mathbb{R}^{n\times n}$ is $\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$. To bound $\mathrm{E}[\lambda_1(A)]$ from above, we use Wigner's trace method [10] for estimating $\lambda_1(A)$, which was also used by Füredi and Komlós [5]: For any (random) real, symmetric matrix A and even $k \in \mathbb{N}$, we have $\mathrm{E}[\lambda_1(A)] \leq \mathrm{E}[\mathrm{tr}(A^k)]^{1/k}$. To prove (5) in Lemma 4, we thus have to estimate $\mathrm{E}[\mathrm{tr}(A(\mathcal{G}, G, \varepsilon)^k)]$. We have

$$\begin{aligned} &\mathrm{E}[\mathrm{tr}(A^k)] \\ &= \mathrm{E}\left[\sum_{l_0=1}^{n}\sum_{l_1=1}^{n}\cdots\sum_{l_{k-1}=1}^{n} a_{l_0l_1}a_{l_1l_2}\ldots a_{l_{k-1}l_0}\right] = \sum_{\vec{l}\in L}\mathrm{E}[a_{l_0l_1}a_{l_1l_2}\ldots a_{l_{k-1}l_0}], \quad (7) \end{aligned}$$

where we abbreviate the set of sequences $\vec{l} = (l_0, \ldots, l_{k-1})$ by $L = \{1, \ldots, n\}^k$. We fix $\vec{l} \in L$ and estimate the corresponding summand $\mathrm{E}[a_{l_0l_1}a_{l_1l_2}\ldots a_{l_{k-1}l_0}]$ in (7). Since A is symmetric, we identify the two equal entries a_{ij} and a_{ji} and consider a_{ij} $(i \leq j)$ as representative. (This means that we replace

all occurrences of a_{ji} by a_{ij}. Let $a_{i_1j_1}, \ldots, a_{i_mj_m}$ be the representatives in $\mathrm{E}[a_{l_0l_1}a_{l_1l_2}\ldots a_{l_{k-1}l_0}]$ with multiplicities $r_1, \ldots, r_m \geq 1$, respectively. Since the presence of different edges in $\mathcal{G}$ is independent, we have

$$\mathrm{E}[\mathrm{tr}(A^k)] = \sum_{\vec{l}\in L} \prod_{s=1}^{m} \mathrm{E}\left[a_{i_sj_s}^{r_s}\right]. \tag{8}$$

To estimate $\mathrm{E}[\mathrm{tr}(A^k)]$, we bound (8) from above. First, consider the sequences $\vec{l} \in L$ for which all representatives $a_{i_sj_s}$ lie on the main diagonal. Then $l_0 = \ldots = l_{k-1} = i$ for $i \in \{1, \ldots, n\}$. For such $\vec{l}$, the corresponding summand in (8) is 1 by the definition of A. Therefore, the n summands for the sequences $l_0 = \ldots = l_{k-1} = i$, $i = 1, \ldots, n$, contribute n to (8). Now, consider the sequences $\vec{l} \in L$ choosing at least one off-diagonal representative entry $a_{i_sj_s}$. If such an $a_{i_sj_s}$ with multiplicity $r_s = 1$ appears, then $\prod_{s=1}^{m} \mathrm{E}\left[a_{i_sj_s}^{r_s}\right] = 0$ by the definition of A: We have $\mathrm{E}[a_{i_sj_s}] = 1 \cdot (1 - p_e) - \frac{1-p_e}{p_e} \cdot p_e = 0$. Hence, it suffices to consider the set L' of sequences $\vec{l}$ with at least one off-diagonal entry and every such entry appearing at least twice.

To bound $|L'|$ from above, let us view a sequence $\vec{l} \in L'$ as a closed walk $l_0, l_1, \ldots, l_{k-1}, l_k = l_0$ of length k in an undirected complete graph. A step (l_j, l_{j+1}) is *identical* if $l_j = l_{j+1}$ and *real* otherwise. Entry $a_{l_jl_{j+1}}$ is off-diagonal if and only if the corresponding step is real. Let k' be the number of real steps, and let m' be the number of different edges that the walk visits (no edge is traversed in identical steps). We call such a walk a (k, k', m')-*walk*. We have $2 \leq k' \leq k$ and $1 \leq m' \leq k'/2$ since each of the m' edges is traversed at least twice.

First, we count the possible (k, k', m')-walks for given k' and m'. For the positions of the $k - k'$ identical steps, we have $\binom{k}{k-k'} \leq 2^k$ choices. It remains to choose a closed walk of length k' with real steps only and each of the m' traversed edges appearing at least twice. Call such a walk a (k', m')-*real-walk*. Friedman et al. [4, p. 425ff] showed an upper bound of $2^k k^k n^{m'+1}$ for the number of such walks. (They have called them *duplicated walks*. In fact, they showed a bound of $k'^{2k'} n^{m'+1}$, which can be improved by using an upper bound of $2^{k'}$ instead of $k'^{k'}$ for $\binom{k'}{m'}$. Moreover, we have used $m' \leq k' \leq k$.) Together with at most 2^k choices for the positions of the identical steps, the total number of (k, k', m')-walks is at most

$$2^k \cdot 2^k \cdot k^k \cdot n^{m'+1} = 2^{2k} \cdot k^k \cdot n^{m'+1}. \tag{9}$$

For a (k,k',m')-walk $\vec{l} \in L'$, we estimate its summand $\prod_{s=1}^{m} \mathrm{E}[a_{i_s j_s}^{r_s}]$ in (8). Since $a_{i_s j_s} = 1$ for $i_s = j_s$, we can omit their factors $\mathrm{E}[a_{i_s j_s}^{r_s}] = 1$. For an off-diagonal representative $a_{i_s j_s}$, $i_s < j_s$, we have

$$\begin{aligned}\mathrm{E}\left[a_{i_s j_s}^{r_s}\right] &= 1^{r_s} \cdot (1-p_e) + \left(-\frac{1-p_e}{p_e}\right)^{r_s} \cdot p_e \\ &\leq 1 + \frac{1}{p_e^{r_s-1}} \leq \frac{2}{p_e^{r_s-1}} \leq \frac{2}{\varepsilon^{r_s-1}}. \end{aligned} \tag{10}$$

Observe that our estimate $p_e \geq \varepsilon$ in the inequality in (10) neglects the potential edges e which are actually present in the adversarial graph G. For such an e, we have $p_e = 1 - \varepsilon \geq \varepsilon$, and one might think that this could improve (10) and our final result. However, asymptotically we lose nothing: Assume that G's edges form a clique of size $n/2$. Then $|E| = \Theta(n^2)$ but G still contains an independent set of size $n/2$. This part of our random graph $\mathcal{G}$ behaves as $G(n/2, \varepsilon)$. Thus, we cannot expect to get a better bound than for $G(n/2, \varepsilon)$.

We continue our proof. Without loss of generality, we assume that the off-diagonal representatives $a_{i_s j_s}$ have indices $s = 1, \ldots, m'$. Then

$$\sum_{s=1}^{m'} (r_s - 1) = k' - m'.$$

This together with (10) yields, for a fixed (k,k',m')-walk $\vec{l} \in L'$,

$$\prod_{s=1}^{m} \mathrm{E}\left[a_{i_s j_s}^{r_s}\right] = \prod_{s=1}^{m'} \mathrm{E}\left[a_{i_s j_s}^{r_s}\right] \leq \prod_{s=1}^{m'} \frac{2}{\varepsilon^{r_s-1}} = \frac{2^{m'}}{\varepsilon^{\sum_{s=1}^{m'}(r_s-1)}} = \frac{2^{m'}}{\varepsilon^{k'-m'}}.$$

We can now estimate the contribution of the collection of all sequences $\vec{l} \in L'$ to (8). The number of (k,k',m')-walks $\vec{l}$ is at most $2^{2k} \cdot k^k \cdot n^{m'+1}$ by (9). We sum up all possibilities for k' and m' and get

$$\begin{aligned}\sum_{\vec{l} \in L'} \prod_{s=1}^{m} \mathrm{E}\left[a_{i_s j_s}^{r_s}\right] &\leq \sum_{k'=2}^{k} \sum_{m'=1}^{k'/2} 2^{2k} \cdot k^k \cdot n^{m'+1} \cdot \frac{2^{m'}}{\varepsilon^{k'-m'}} \\ &\leq \sum_{k'=2}^{k} \sum_{m'=1}^{k'/2} 2^{3k} \cdot k^k \cdot n \cdot \left(\frac{n}{\varepsilon}\right)^{k/2} \leq 2^{4k} \cdot k^k \cdot n \cdot \left(\frac{n}{\varepsilon}\right)^{k/2}, \end{aligned} \tag{11}$$

using that $2 \leq k' \leq k$ and $1 \leq m' \leq k'/2$ and $(1/(n\varepsilon))^{k/2-m'} \leq 1$.

Now we can bound $\mathrm{E}[\mathrm{tr}(A^k)]$ from above: We have shown that the contribution of the sequences $\vec{l} \in L \setminus L'$ is n. The contribution of the sequences $\vec{l} \in L'$ is given by (11). Using (8), we get

$$\begin{aligned} &\mathrm{E}[\mathrm{tr}(A^k)] \\ &= \sum_{\vec{l} \in L} \prod_{s=1}^{m} \mathrm{E}\left[a_{i_s j_s}^{r_s}\right] \leq n + 2^{4k} k^k n \cdot \left(\frac{n}{\varepsilon}\right)^{k/2} \leq 2^{5k} k^k n \cdot \left(\frac{n}{\varepsilon}\right)^{k/2}. \end{aligned} \tag{12}$$

Now we set $k = 2\lceil \log n \rceil$ and apply the trace method to (12), which yields

$$\begin{aligned} \mathrm{E}[\lambda_1(A)] \leq \mathrm{E}\left[\mathrm{tr}(A^k)\right]^{1/k} &\leq \left(2^{5k} \cdot k^k \cdot n \cdot \left(\frac{n}{\varepsilon}\right)^{k/2}\right)^{1/k} \\ &= 2^5 \cdot k \cdot n^{1/k} \cdot \sqrt{n/\varepsilon} \leq 2^7 \cdot (\log n) \cdot \sqrt{n/\varepsilon}. \end{aligned}$$

For the last inequality, we have used that $n^{1/k} = n^{1/(2\lceil \log n \rceil)} \leq \sqrt{2}$. This completes the proof of (5) in Lemma 4.

2.2.2. A Tail Bound on the Largest Eigenvalue

To prove (6) of Lemma 4, we adapt a result by Krivelevich and Vu [7, Lemma 2.3] to our model. Since p_e can be either ε or $1-\varepsilon$, there are two types of corresponding entries a_{ij}. In order to adapt their proof, we have to bound the difference of two different outcomes of an entry of $A = A(\mathcal{G}, G, \varepsilon)$: This difference is at most $1 + (1-p_e)/p_e = 1/p_e \leq 1/\varepsilon$. Let m' be the median of the largest eigenvalue $\lambda_1(A)$ of the matrix A. Then we can apply Krivelevich and Vu's proof [7, Proof of Lemma 2.3], for which only an upper bound on the difference of the two different outcomes of each entry of A is needed. This yields

$$\begin{aligned} \Pr\big[|\lambda_1(A) - m'| \geq t\big] &\leq 4\exp\big(-(t\varepsilon)^2/8\big) \quad \text{and} \\ \big|\mathrm{E}[\lambda_1(A)] - m'\big| &= O(1/\varepsilon). \end{aligned} \tag{13}$$

From this, we can conclude that the median and the mean do not differ by too much: $|\mathrm{E}[\lambda_1(A)] - m'| = O(1/\varepsilon) = o(\log n \cdot \sqrt{n/\varepsilon})$ by the assumption that $\varepsilon = \Omega((\log n)^2/n)$. Together with (5), we obtain

$$m' \leq \mathrm{E}[\lambda_1(A)] + o(\log n \cdot \sqrt{n/\varepsilon}) \leq (2^7 + o(1)) \cdot (\log n)\sqrt{n/\varepsilon}.$$

Now assume that $\lambda_1(A) \geq 2^8 (\log n)\sqrt{n/\varepsilon}$ happens. Then the bound for m' above implies $|\lambda_1(A) - m'| \geq 2^6 (\log n)\sqrt{n/\varepsilon}$ for sufficiently large n. Plugging $t = 2^6 (\log n)\sqrt{n/\varepsilon}$ into (13) completes the proof.

Algorithm 1 `Approx-IS`$(\mathcal{G}, G, \varepsilon)$

1: Compute the greedy independent set $I = \mathrm{gis}(\mathcal{G})$. If $|I| < t_{\mathrm{gis}}(G, \varepsilon)$ then go to Step 5.
2: Compute $\lambda_1(A(\mathcal{G}, G, \varepsilon))$. If $\lambda_1 < 2^8 \cdot (\log n) \cdot \sqrt{n/\varepsilon}$ then output I.
3: For all $S' \subseteq V$, $|S'| = (8 \log n)/\varepsilon$, compute $|\overline{N}(S')|$. If $|\overline{N}(S')| \leq (2 \log n) \cdot \sqrt{n/\varepsilon}$ for all tested subsets S' then output I.
4: Check all subsets $S'' \subseteq V$ with $|S''| = (8 \log n)\sqrt{n/\varepsilon}$. If none of them is independent then output I.
5: Find a largest independent set by exhaustive search and output it.

2.3. Approximating the Independence Number

Now we prove Theorem 1 and state our algorithm `Approx-IS` (Algorithm 1). To do this, let, for a graph $G = (V, E)$ and a set $S \subseteq V$, the *non-neighborhood* $\overline{N}(S)$ of S be the set of all vertices $v \in V \setminus S$ for which there is no edge $\{v, w\} \in E$ with $w \in S$. `Approx-IS` gets an adversarial graph G, a flip probability ε, and a random graph $\mathcal{G}$ drawn from $\mathcal{G}(G, \varepsilon)$ as input. Recall the definition of the threshold for the greedy independent set size: $t_{\mathrm{gis}}(G, \varepsilon) = \frac{\delta}{16} \cdot \min\{\frac{\ln n}{\varepsilon}, \frac{n^2 \ln n}{|E| \ln(1/\varepsilon)}\}$. From now on, we fix $\delta = 1/2$.

Approximation Guarantee. We start with the approximation guarantee. We show that we always get a solution with approximation ratio $O(\frac{\log n \cdot \sqrt{n/\varepsilon}}{t_{\mathrm{gis}}(G,\varepsilon)})$. Plugging in the definition of t_{gis} completes the proof.

Step 5 outputs an optimal solution with approximation ratio 1. If any other step outputs the greedy independent set $I = \mathrm{gis}(\mathcal{G})$, we have $|I| \geq t_{\mathrm{gis}}(G, \varepsilon)$, since otherwise we jump to exhaustive search (Step 5) in Step 1. Furthermore, the independence number $\alpha(\mathcal{G})$ is small: If Step 2 outputs I, then Lemma 4 yields

$$\alpha(\mathcal{G}) \leq \lambda_1(A(\mathcal{G})) = O(\log n \cdot \sqrt{n/\varepsilon}).$$

The same holds if Step 3 outputs I: Then, for all sets $S' \subseteq V$ of size $(8 \log n)/\varepsilon$, the non-neighborhood has size $|\overline{N}(S')| \leq 2 \log n \sqrt{n/\varepsilon}$. Hence,

$$\alpha(\mathcal{G}) \leq (8 \log n)/\varepsilon + 2 \log n \cdot \sqrt{n/\varepsilon} = O(\log n \cdot \sqrt{n/\varepsilon}),$$

since $\varepsilon \geq \sqrt{1/n}$. For Step 4, this upper bound on $\alpha(\mathcal{G})$ is obvious if I is output. With our bounds on $\alpha(\mathcal{G})$ and $|I|$, we get the desired approximation ratio of $\frac{\alpha(\mathcal{G})}{|I|} = O(\frac{\log n \sqrt{n/\varepsilon}}{t_{\mathrm{gis}}(G,\varepsilon)})$.

The Expected Running-Time. Now we analyze the expected running-time of `Approx-IS`. The expected running-time of a step is the product of the time it takes to execute it (its *effort*) and the probability of executing it. We show that the expected running-time of every step is polynomial.

Let T_i be the random variable for the time spent in Step i. Steps 1 and 2 have polynomial worst-case running-time. In particular, eigenvalues can be computed in polynomial time [1].

We turn to Steps 3, 4, and 5. Let $s' = (8 \log n)/\varepsilon$. Step 3's effort is

$$O\left(\text{poly}(n) \cdot \binom{n}{s'}\right) = O(\text{poly}(n) \cdot n^{s'}) = O\left(\text{poly}(n) \cdot \exp\left(\frac{8(\ln n)^2}{\varepsilon \ln 2}\right)\right),$$

since it tests $\binom{n}{s'}$ sets, each of which in polynomial time. The step is only executed if Step 2 does not output I. Then $\lambda_1 \geq 2^8 \cdot \log n \sqrt{n/\varepsilon}$, which happens with a probability of at most $4\exp(-2^9 n\varepsilon(\log n)^2)$ by Lemma 4. We conclude that the expected running-time of Step 3 is

$$\begin{aligned}\mathrm{E}[T_3] &= O\left(\text{poly}(n) \cdot \exp\left(\frac{8(\ln n)^2}{\varepsilon \ln 2}\right) \cdot \exp(-2^9 \cdot n\varepsilon \cdot (\log n)^2)\right) \\ &= O\left(\text{poly}(n) \cdot \exp\left(\frac{8(\ln n)^2}{\varepsilon \ln 2} - \frac{2^9 \cdot n\varepsilon \cdot (\ln n)^2}{(\ln 2)^2}\right)\right). \qquad (14)\end{aligned}$$

The exponent in (14) is non-positive if $\varepsilon \geq \sqrt{(8 \ln 2)/2^9} \cdot \sqrt{1/n}$, which holds since $\varepsilon \geq \sqrt{1/n}$. Thus, $\mathrm{E}[T_3]$ is bounded by a polynomial.

Now let $n' = (2 \log n)\sqrt{n/\varepsilon}$. Then

$$\Pr[\text{Step 3 does not output } I] = \Pr[\exists S' \subseteq V,\, |S'| = s' \colon |\overline{N}(S')| > n'].$$

If Step 3 does not output I, then there are sets $S', N' \subseteq V$ with $|S'| = s'$ and $|N'| = n'$ such that none of the $s'n'$ potential edges between S' and N' exists in $\mathcal{E}$. Each edge is absent with probability at most $1 - \varepsilon$. A union bound over all sets S' and N' combined with $1 - x \leq e^{-x}$ yields

$$\begin{aligned}&\Pr[\text{Step 3 does not output } I] \\ &\leq \binom{n}{s'} \cdot \binom{n}{n'} \cdot (1-\varepsilon)^{s'n'} \leq n^{s'} \cdot n^{n'} \cdot \exp(-\varepsilon s' n') \\ &= \exp\left(\frac{8 \cdot (\ln n)^2}{\varepsilon \ln 2} + \frac{2 \cdot (\ln n)^2 \sqrt{n/\varepsilon}}{\ln 2} - \frac{16 \cdot (\ln n)^2 \sqrt{n/\varepsilon}}{(\ln 2)^2}\right) \\ &\leq \exp\left(\left(\frac{8}{\ln 2} + \frac{2}{\ln 2} - \frac{16}{(\ln 2)^2}\right) \cdot (\ln n)^2 \cdot \sqrt{\frac{n}{\varepsilon}}\right) \leq \exp\left(-\frac{8(\ln n)^2 \sqrt{n\varepsilon}}{\ln 2}\right),\end{aligned}$$

using $\frac{8\cdot(\ln n)^2}{\varepsilon \ln 2} \leq \frac{8}{\ln 2} \cdot (\ln n)^2 \sqrt{n/\varepsilon}$ due to $\varepsilon \geq \sqrt{1/n} \geq 1/n$ for the second-to-last inequality. Since the number of tested sets S'' in Step 4 is

$$\binom{n}{8 \log n \sqrt{n/\varepsilon}} \leq \exp\left(\frac{8}{\ln 2} \cdot (\ln n)^2 \sqrt{n/\varepsilon}\right),$$

we can infer that also $\mathrm{E}[T_4]$ is bounded by a polynomial.

In a fixed tested set S'', there are

$$\binom{8 \log n \sqrt{n/\varepsilon}}{2} \geq \frac{16n(\ln n)^2}{(\ln 2)^2 \varepsilon}$$

potential edges. Thus, S'' is independent with a probability of at most

$$(1-\varepsilon)^{\frac{16n(\ln n)^2}{(\ln 2)^2 \varepsilon}} \leq \exp\left(-\varepsilon \cdot \frac{16n(\ln n)^2}{(\ln 2)^2 \varepsilon}\right) = \exp\left(-\frac{16(\ln n)^2 n}{(\ln 2)^2}\right).$$

The number of tested sets in Step 4 is at most

$$\exp\left(\frac{8(\ln n)^2 \sqrt{n/\varepsilon}}{\ln 2}\right) = \exp\left(o((\ln n)^2 n)\right)$$

since $\varepsilon \geq \sqrt{1/n}$. A union bound over all tested sets yields that the probability that Step 4 does not output I is $\exp(-\Omega((\log n)^2 n))$. Step 5 is only executed if Step 4 does not output I or if Step 1 fails, i.e., $|I| < t_{\mathrm{gis}}(G, \varepsilon)$. Lemma 3 shows that this happens with a probability of at most $e^{-n \ln n}$. Thus, Step 5 is executed with a probability of at most $\exp\big(-\Omega((\log n)^2 n)\big) + \exp(-n \ln n) = O(e^{-n \ln n})$. Since Step 5 tests 2^n sets, its effort is $O(\mathrm{poly}(n) \cdot 2^n)$. Hence, also $\mathrm{E}[T_5]$ is bounded by a polynomial.

2.4. The Expected Behavior of Greedy Independent Set

Now we prove Theorem 2. Since $\varepsilon \leq 1/2$, $\alpha(\mathcal{G})$ is stochastically dominated by the independence number of a $G(n, \varepsilon)$ graph. The probability that a $G(n, \varepsilon)$ graph contains a clique of size at least $c(\log n)/\varepsilon$ for some sufficiently large constant c is at most $1/n$, as follows for instance from Bollobás and Erdős [2]. Lemma 3 states that the probability that `GreedyColoring` does not find an independent set of cardinality at least $\Omega((\log n)/\varepsilon)$ is exponentially small. Combining this yields that the probability that `GreedyColoring` does not achieve a constant approximation ratio is at most $O(1/n)$. If this nevertheless happens, we can lower-bound the size of the greedy independent set by the trivial bound of 1 and upper-bound the independent set by the trivial bound of n. This contributes only $O(1)$ to the expected value of the approximation ratio.

3. Conclusions and Open Problems

We have performed a probabilistic analysis of the approximability of Independent Set. The probabilistic model that we have used is a smoothed extension of $G(n, \varepsilon)$ [9]. Our algorithm *guarantees* an approximation ratio of $O(\sqrt{n\varepsilon})$ in *expected* polynomial time. Furthermore, we proved that the greedy algorithm, which has *worst-case* polynomial time, has constant *expected* approximation ratio. This shows a trade-off between guaranteed or expected running-time and approximation ratio.

Our algorithm `Approx-IS` needs to know the adversarial graph G in addition to $\mathcal{G}$. A different view on this is that `Approx-IS` has an estimate about the probability of the existence of an edge, which can be high or low. We leave it as an open problem to eliminate the need of knowing G.

[1] Noga Alon. Spectral techniques in graph algorithms. In Claudio L. Lucchesi and Arnaldo V. Moura, editors, *Proc. of the 3rd Latin American Symposium on Theoretical Informatics*, volume 1380 of *Lecture Notes in Computer Science*, pages 206–215. Springer, 1998.

[2] Béla Bollobás and Paul Erdős. Cliques in random graphs. *Mathematical Proc. Cambridge Philosophical Society*, 80(3):419–427, 1976.

[3] Uriel Feige. Approximating maximum clique by removing subgraphs. *SIAM J. Discrete Math.*, 18(2):219–225, 2004.

[4] Joel Friedman, Andreas Goerdt, and Michael Krivelevich. Recognizing more unsatisfiable random k-sat instances efficiently. *SIAM J. Comput.*, 35(2):408–430, 2005.

[5] Zoltán Füredi and János Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.

[6] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[7] Michael Krivelevich and Van H. Vu. Approximating the Independence Number and the Chromatic Number in Expected Polynomial Time. *J. Comb. Optim.*, 6(2):143–155, 2002.

[8] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004.

[9] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52(10):76–84, 2009.

[10] Van H. Vu. Spectral norm of random matrices. *Combinatorica*, 27(6):721–736, 2007.

[11] David Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(1):103–128, 2007.