Parameterized Clique on Scale-Free Networks

Tobias Friedrich and Anton Krohmer

Friedrich-Schiller-Universität Jena, Germany

Abstract. Finding cliques in graphs is a classical problem which is in general NP-hard and parameterized intractable. However, in typical applications like social networks or protein-protein interaction networks, the considered graphs are scale-free, i.e., their degree sequence follows a power law. Their specific structure can be algorithmically exploited and makes it possible to solve clique much more efficiently. We prove that on inhomogeneous random graphs with n nodes and power law exponent γ, cliques of size k can be found in time $\mathcal{O}(n^2)$ for $\gamma \geq 3$ and in time $\mathcal{O}(n \exp(k^4))$ for $2 < \gamma < 3$.

1 Introduction

The clique problem numbers among the most studied problems in theoretical computer science. Its decision version calls for determining whether a given graph with n vertices contains a clique of size k, i.e., a complete subgraph on k vertices. It is one of Karp's original NP-complete problems [16] and is complete for the class W[1], the parameterized analog of NP [8]. Its optimization variant is a classical example of a problem that is NP-hard to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [11, 31]. Also, on Erdős-Rényi random graphs, the problem is believed to be intractable in general, which is even used for cryptographic schemes [15]. For all functions p, Rossman [27] presented an average-case lower bound of $\omega(n^{k/4})$ on the size of monotone circuits for solving k-Clique.

The term "clique" was first used 1949 by Luce and Perry [19], to describe a group of mutual friends in a social network. Since then, social networks, and likewise the study thereof, increased tremendously. There exist numerous models, most of them having in common a so-called *scale-free* behavior. This means that there is a constant γ such that the fraction of nodes that have degree d is proportional to $d^{-\gamma}$. Besides social networks, many other real-world networks are scale-free, too. Examples are the internet, citation graphs, co-author graphs, protein-protein interaction networks and power supply networks [7, 20, 22].

It is therefore natural to study the clique problem on scale-free graphs. This is not only of theoretical interest, as this question occurs in different application domains. One example for this is bioinformatics. Here, cliques in protein-protein interaction networks are sought in order to identify clusters of proteins that interact tightly with each other [29]. Another bioinformatics example is the clustering of large scale gene expression data using cliques [4]. A different direction is internet marketing, where it is e.g. valuable to find large cliques on Facebook.

K.-M. Chao, T.-s. Hsu, and D.-T. Lee (Eds.): ISAAC 2012, LNCS 7676, pp. 659–668, 2012.

We are interested in the complexity of the clique problem on *inhomogeneous random graphs*, where each node i has a weight w_i and an edge $\{i, j\}$ is present independently with probability proportional to $w_i w_j$. This is a generalization of several scale-free random graph models like Chung-Lu [1, 2, 6], Norros-Reittu [23], and generalized random graphs [30].

Our Results

The behavior of scale-free networks depends significantly on the exponent γ of the power law degree distribution. If $\gamma > 3$, the expected maximal size of a clique is constant [5, 13]. This implies that large cliques are very unlikely, but does not imply a fast algorithm that always answers correctly. The difficulty is certifying a negative answer. We prove the following theorem.

Theorem 1. *The k-Clique problem can be solved in expected time $\mathcal{O}(n^2)$ on inhomogeneous random graphs with power law exponent $\gamma \geq 3$.*

All our algorithms are deterministic and always return the correct answer. Note that the above theorem implies that k-Clique, which is NP-complete in general, in this setting becomes avgP, which is the average-case analog of P [18].

On the other hand, many scale-free networks (e.g. the internet) have a power law exponent γ with $2 < \gamma < 3$ [21]. In this case, the expected maximal size of a clique diverges [5, 13]. and there exists a giant component of polynomial size, the *core*. The core is a subgraph that has a diameter of $\mathcal{O}(\log \log n)$ and contains a dense Erdős-Rényi graph [6, 30]. As this is a known hard problem, we cannot expect similarly good results as for $\gamma \geq 3$. We prove the following theorem.

Theorem 2. *The k-Clique problem can be solved in time $\mathcal{O}(n \exp(k^4))$ with overwhelming*[1] *probability on inhomogeneous random graphs with power law exponent $2 < \gamma < 3$.*

While in general k-Clique is not believed to be parameterized tractable, i.e. in FPT, the above theorem shows that in this setting k-Clique is typically parameterized tractable, i.e. in typFPT, which is an average-case analog of FPT as defined in [9].

Related Work

Much previous research on cliques in random graphs focuses on Erdős-Rényi random graphs [12]. Rossman [26, 27, 28] provides lower bounds for solving the problem where he uses bounded-depth Boolean circuits and unbounded-depth monotone circuits as the computation model. Using a greedy approach, a clique of size $\log n$ can be found in a $G(n, \frac{1}{2})$ [10], whereas Jerrum [14] showed that one cannot use the Metropolis algorithm to find cliques of size $(1+\varepsilon) \log n$ in $G(n, \frac{1}{2})$,

[1] We use the terms *high probability* for probability $1 - o(1)$, *negligible probability* for probability $1/f(n)$, and *overwhelming probability* for probability $1 - 1/f(n)$, where $f(n)$ is any superpolynomially increasing function.

and Peinado [24, 25] proved that several randomized algorithms are also bound to fail on this problem. Kučera [17] shows that a planted clique (i.e., a clique which is explicitly added after drawing a random graph) of size $\Omega(\sqrt{n \log n})$ is easy to find in the $G(n, \frac{1}{2})$, and Alon et al. [3] further improve this bound to $\Omega(\sqrt{n})$.

The inspiration for our work was Fountoulakis et al. [9]. They introduce an average-case analog of FPT and show that the k-Clique problem on $G(n, p)$ can be solved for all edge probabilities $p(n)$ in expected FPT-time and in FPT-time with high probability. Janson et al. [13] also showed that on Norros-Reittu random graphs [23] a simple algorithm with access to the weights of the model can find a $(1 - o(1))$-approximation of maximum clique in polynomial time with high probability.

2 Preliminaries

In order to achieve high general validity, we use the inhomogeneous random graph model of van der Hofstad [30], which generalizes the models of Chung-Lu [1, 2, 6] and Norros-Reittu [23] as well as the generalized random graphs. The model has two adjustable parameters: the exponent of the scale-free network γ and the average degree a. Depending on these two parameters, each node i has a weight w_i. This determines the edge probability $p_{ij} := \Pr[\{i, j\} \in E]$, which should be set proportional to $w_i w_j$.

Weights w_i. A simple way to fix the weights would be for example $w_i = a(n/i)^{\frac{1}{\gamma-1}}$. However, we aim for a more general setting and proceed differently. Given the weights w_i, we can use the empirical distribution function $F_n(w) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[w_i \geq w]$. This gives us $F_n(w) = \Pr[W \geq w]$, where W is a random variable chosen uniformly from the weights $w_1, \ldots, w_n$. Instead of fixing w_i, it is now easier to start from $F_n(w)$ and assume the following.

Definition 1 (Power-Law Weights). *We say that an empirical distribution function $F_n(w)$ follows the* power law with exponent γ*, if there exist two positive constants α_1, α_2 such that*

$$\alpha_1 w^{-\gamma+1} \leq F_n(w) \leq \alpha_2 w^{-\gamma+1}.$$

Then, we require that weights $w_1, \ldots, w_n$ have the empirical distribution function $F_n(w)$. Following van der Hofstad [30], we moreover require that the empirical distribution function F_n satisfies the following properties.

Definition 2 (Regularity Conditions for Vertex Weights).

(1) **Weak convergence of vertex weights.** *There exists a function F such that $\lim_{n\to\infty} F_n(x) = F(x)$.*

(2) **Convergence of average vertex weight.** *Let W_n and W have distribution functions F_n and F, respectively. Then, $\lim_{n\to\infty} \mathrm{E}[W_n] = \mathrm{E}[W]$ holds. Furthermore, $\mathrm{E}[W] > 0$.*

The regularity of F_n guarantees that the intuition $F_n(w) = \Pr[W \geq w]$ indeed holds. Furthermore, it guarantees that the average degree in the inhomogeneous random graphs converges, and that the largest weight is asymptotically bounded by $o(n)$, i.e. $\max_{i\in\{1,\dots,n\}} w_i = o(n)$. Both assumptions are sufficient to generate a scale-free network. [30]

Edge Probability p_{ij}. Other inhomogeneous random graph models use the following definitions:

$$p_{ij} = \min\left\{\frac{w_i w_j}{\sum_{k=1}^n w_k}, 1\right\} \qquad \text{(Chung-Lu)}$$

$$p_{ij} = \frac{w_i w_j}{\sum_{k=1}^n w_k + w_i w_j} \qquad \text{(Generalized Random Graph)}$$

$$p_{ij} = 1 - \exp\left\{-\frac{w_i w_j}{\sum_{k=1}^n w_k}\right\} \qquad \text{(Norros-Reittu)}$$

We use a more general approach and only assume the following.

Definition 3. *We call p_{ij} the* edge probability *between nodes i and j of the inhomogeneous random graph, if it is 0 for $i = j$, and otherwise fulfills*

$$p_{ij} = \mathcal{O}\left(\frac{w_i w_j}{n}\right) \qquad \text{and} \qquad p_{ij} = \Omega\left(\frac{w_i w_j}{n + w_i w_j}\right).$$

In order to see that this is a generalization of all aforementioned scale-free random graph models, we observe that when $\gamma \geq 2$, $w_{\min} = \Theta(1)$ and $w_{\min}^{-\gamma+2} > w_{\max}^{-\gamma+2}$ and compute

$$\sum_{k=1}^{n} w_k = n \cdot \mathrm{E}[W] = n \cdot \int_{w_{\min}}^{w_{\max}} F_n(w)\,\mathrm{d}w = \Theta(n \cdot w_{\min}^{-\gamma+2}) = \Theta(n). \tag{1}$$

It is useful to observe that the expected degree of each vertex can be asymptotically upper bounded by its weight:

$$\mathrm{E}[\deg(i, G)] = \sum_{j=1}^{n} p_{ij} = \mathcal{O}\Big(\frac{w_i}{n}\sum_{j=1}^{n} w_j\Big) = \mathcal{O}(w_i) \tag{2}$$

This bound is in fact tight, as shown by the following lemma. The proof is omitted due to space limitations.

Lemma 1. $\mathrm{E}[\deg(i, G)] = \Theta(w_i)$ *for* $\gamma \geq 2$.

Notation. We use $\mathcal{G}_{\mathcal{SF}}(\gamma)$ to refer to the probability space of inhomogeneous random graphs that were created as described above, and G to represent a graph drawn from $\mathcal{G}_{\mathcal{SF}}(\gamma)$. By $\deg(v, G)$ we refer to the degree of a node v in a graph G. We expect the nodes to be ordered from smallest to greatest weight. Finally, we use the induced subgraph $G_i := G[i, \dots, n]$ that describes an inhomogeneous random graph $G \leftarrow \mathcal{G}_{\mathcal{SF}}$ where nodes $1, \dots, i-1$ have been removed from the vertex set.

3 Analysis for Power Law Exponent $\gamma \geq 3$

In this section, we describe an algorithm to solve the clique problem in $\mathcal{O}(n^2)$ on average whenever the scale-free network exhibits an exponent of $\gamma \geq 3$.

Greedy Algorithm. We exploit the scale-free structure by processing low-degree nodes first. The algorithm repeats the following steps: Choose a node v with minimum degree. If there is a $(k-1)$-subset of neighbors of v that is a clique, return the resulting k-clique. Otherwise, remove v from the graph. This implies that when the algorithm reaches the high-degree nodes, the graph is almost empty, which means that those nodes are of small degree, too. We use adjacency lists to store the edges. A careful implementation then allows finding a node with minimum degree in expected amortized constant time: An array of length n stores in each position i all nodes of degree i. By memorizing an index in the array, one can extract the currently smallest degree node in constant time. As removing a node from the graph means removing the node from the array and all adjacency lists of its neighbors, the overall runtime for updating the graph is proportional to the sum of all degrees. By equation (2), this is $\mathcal{O}(n)$. We show that the greedy algorithm has an expected runtime of $\mathcal{O}(n^2)$.

Weight Algorithm. The difficulty in the analysis of the greedy algorithm is that the node with the smallest degree may not have the smallest weight, and vice versa. We therefore take a slight detour and analyze another approach, the weight algorithm. We prove in the following Lemma 2 that it is at most $\mathcal{O}(n)$ faster than the greedy algorithm. The weight algorithm works like the greedy algorithm, the only difference being that instead of taking the node with smallest degree, it chooses a node v with minimum weight w_v. This makes it more practical for bounding the runtime. As we only use the weight algorithm for our analysis, it does not matter that the weights are not available for real networks.

Lemma 2. *On all inputs, the greedy algorithm is at most a factor of $\mathcal{O}(n)$ slower than the weight algorithm.*

Proof. Consider a graph $G = (V, E)$ and let nodes $v_1, \ldots, v_n$ be ordered as they are processed by the greedy algorithm. Let $d_{\max}$ be the largest degree that occurs during the greedy algorithm, and let t be an iteration in which this happens. Then, using some constant c, we can upper bound the runtime of this algorithm by $T_{\text{greedy}} = c \cdot n \cdot \binom{d_{\max}}{k-1} \cdot (k-1)^2$. Therefore, by definition, the subgraph $G_t = G[v_t, \ldots, v_n]$ of graph G has minimum degree $d_{\max}$, and the weight algorithm needs at least $\binom{d_{\max}}{k-1} \cdot (k-1)^2$ time for processing a node from this subgraph. □

We now examine the expected degree of a node i in the weight algorithm, when the processed nodes $1, \ldots, i-1$ were already removed from the graph.

Lemma 3. *Let $G = (V, E)$ be a random graph drawn from $\mathcal{G}_{\mathcal{SF}}(\gamma)$, where $\gamma \geq 3$, and let $G_i = G[i, \ldots, n]$. Then, we have*

$$\mathrm{E}\left[\deg(i, G_i)\right] = \mathcal{O}(1).$$

Proof. We use the indicator variable $\mathbb{1}[\{i,j\} \in E]$, which attains value 1 if $\{i,j\}$ is an edge and 0 otherwise. Whenever possible, we hide constants in $\mathcal{O}(1)$.

$$\mathrm{E}\left[\deg(i,G_i)\right] = \sum_{j=i+1}^{n} \mathrm{E}\left[\mathbb{1}[\{i,j\} \in E]\right] = \sum_{j=i+1}^{n} p_{ij} = \mathcal{O}\Big(\frac{w_i}{n} \sum_{j=1}^{n} w_j \cdot \mathbb{1}[w_j > w_i]\Big)$$

We now use the random variable W as described in equation (1), yielding

$$\begin{aligned}\mathrm{E}\left[\deg(i,G_i)\right] &= \mathcal{O}\big(w_i \cdot \mathrm{E}\left[W \cdot \mathbb{1}[W \geq w_i]\right]\big) \\ &= \mathcal{O}\big(w_i \cdot \mathrm{E}\left[W \mid W \geq w_i\right] \cdot \Pr[W \geq w_i]\big) \\ &= \mathcal{O}\Big(w_i \cdot F_n(w_i) \cdot \int_{w_1}^{w_n} \Pr[W \geq w \mid W \geq w_i]\,\mathrm{d}w\Big).\end{aligned}$$

To determine the probability that a weight W drawn uniformly at random admits $W \geq w$ given that $W \geq w_i$ holds, we distinguish two cases:

(1) $w \leq w_i$: Since we know that $W \geq w_i$, it is also larger than w.
(2) $w > w_i$: The conditional probability simplifies to $\frac{\Pr[W \geq w]}{\Pr[W \geq w_i]} = \frac{F_n(w)}{F_n(w_i)}$.

This simplifies the integral and yields

$$\begin{aligned}\mathrm{E}\left[\deg(i,G_i)\right] &= \mathcal{O}\left(w_i \cdot F_n(w_i) \cdot \left(\int_{w_1}^{w_i} 1\,\mathrm{d}w + \int_{w_i}^{w_n} \frac{F_n(w)}{F_n(w_i)}\,\mathrm{d}w\right)\right) \\ &= \mathcal{O}\left(w_i(w_i - w_1) \cdot F_n(w_i) + \frac{w_i}{-\gamma+2}\left[w^{-\gamma+2}\right]_{w_i}^{w_n}\right) \\ &= \mathcal{O}(w_i^{-\gamma+3}).\end{aligned}$$

For $\gamma \geq 3$, this term is constant, as the weights w_i are in $\Omega(1)$. □

For proving Theorem 1, it remains to show that the processed nodes are very unlikely to have high degrees during the course of the weight algorithm.

Proof of Theorem 1. Let T_{greedy} and T_{weight} be the running time of the greedy and the weight algorithms, respectively. The runtime for processing node i in the weight algorithm is denoted by T_i.

The number of $(k-1)$-subsets of vertices a node with x neighbors allows is $\binom{x}{k-1} \leq 2^x$, and the time needed to check whether a subset is a clique is $(k-1)^2 \leq x^2$. For the expected degree of node i we write μ_i. Thus, we can write

$$\begin{aligned}\mathrm{E}[T_{\text{greedy}}] &= \mathcal{O}(n \cdot \mathrm{E}\left[T_{\text{weight}}\right]) = \mathcal{O}\left(n \sum_{i=1}^{n} \mathrm{E}\left[T_i\right]\right) \\ &= \mathcal{O}\left(n \sum_{i=1}^{n} \sum_{x=1}^{n} \binom{x}{k-1} \cdot (k-1)^2 \cdot \Pr\left[\deg(i,G_i) = x\right]\right) \\ &= \mathcal{O}\left(n \sum_{i=1}^{n} \sum_{x=1}^{n} 2^x x^2 \cdot \Pr\left[\deg(i,G_i) \geq \left(1 + \left(\frac{x}{\mu_i} - 1\right)\right)\mu_i\right]\right).\end{aligned}$$

By Lemma 3, μ_i is constant. Applying a Chernoff bound gives

$$\begin{aligned}\mathrm{E}[T_{\text{greedy}}] &= \mathcal{O}\left(n\sum_{i=1}^{n}\sum_{x=1}^{n} 2^x x^2 \cdot \left(\exp\left(\frac{x}{\mu_i}-1\right)\Big/\left(\frac{x}{\mu_i}\right)^{x/\mu_i}\right)^{\mu_i}\right)\\ &= \mathcal{O}\left(n\sum_{i=1}^{n}\sum_{x=1}^{\infty}(2e\mu_i)^x x^{2-x}\right).\end{aligned}$$

Since the inner sum converges to a constant, $\mathrm{E}[T_{\text{greedy}}] = \mathcal{O}(n^2)$. □

4 Analysis for Power Law Exponent $\gamma \in (2, 3)$

Using the greedy algorithm of the previous section for this case would imply a superpolynomial runtime of $\text{poly}(n)^k$ since the neighborhood sizes in the algorithm increase with n. This only yields the result that k-Clique is in expectation in the parameterized class XP [8]. We therefore have to apply a third algorithm to prove a better result. Instead of hoping that there will be few edges and therefore cliques, we hope that the probability that the core contains a k-clique is high. However, this approach is only feasible for small values of k. This shows that k-Clique is parameterized tractable for the parameter k with high probability.

Partitioning Algorithm. To find a k-clique, the partitioning algorithm first removes all nodes with degree below $\sqrt{n/\log\log n}$. The obtained subgraph $G' = (V', E')$ is arbitrarily partitioned into components of size k. Each component is then individually checked to determine if it is a clique. If no clique is found, the algorithm searches exhaustively all k-subsets of V. It is easy to see that this algorithm is correct.

We now want to prove that when k is small, the exhaustive search is triggered only with negligible probability. For this, we first show that there are polynomially many nodes in V' and that their mutual edge probabilities are $\geq 1/\log n$. We then use this to prove that one of the partitions is likely to be a clique. Note that a slightly larger threshold like $\sqrt{n}$ would yield an edge probability of $1 - o(1)$, but the core V' is then empty if γ is close to 3. The chosen threshold $\sqrt{n/\log\log n}$ is therefore more suitable for our analysis. As in the previous section, the analysis would be much easier if, when choosing V', the algorithm was allowed to choose the nodes according to their weight, but unfortunately it only has access to their degree in the given graph. We do know, however, that the weight and the expected degree of a node are equal up to a constant. Using that, we can prove the two following lemmas:

Lemma 4 (Partitioning algorithm keeps polynomially many nodes). *Let $\gamma \in (2, 3)$, and $G = (V, E)$ be a scale-free graph drawn from $\mathcal{G}_{\mathcal{SF}}(\gamma)$. Then,*

$$\Pr\left[\exists i > n - n^{\frac{3-\gamma}{2}} : \deg(i, G) < \sqrt{n/\log\log n}\right] \leq \exp\big(-\Theta(\sqrt{n})\big).$$

Lemma 5 (Partitioning algorithm keeps only high-weight nodes). *Let $\gamma \in (2,3)$, and $G = (V,E)$ be a scale-free graph drawn from $\mathcal{G}_{\mathcal{SF}}(\gamma)$. Then,*

$$\Pr\left[\exists i \in V' \colon w_i < \sqrt{\alpha_1 n / \log n}\right] \leq \exp\big(-\Theta(n^{\frac{1}{3}})\big).$$

Both proofs will be given in the full version of the paper. It remains to show that the partitioning algorithm needs more than $k^2 n$ time only with negligible probability, if k is small enough. The idea is that the core V' of the scale-free network has larger edge probabilities than a dense Erdős-Rényi random graph, which is known to allow finding cliques fast [9].

Proof of Theorem 2. If $k > \log^{\frac{1}{3}} n$, then $n < e^{k^3}$, which implies that the exhaustive search of the partitioning algorithm runs in time $n^k < e^{k^4}$ and proves the claim. We can therefore assume $k \leq \log^{\frac{1}{3}} n$.

As excluding unlikely events does not affect small failure probabilities, we can condition on the statements of Lemmas 4 and 5 and assume that there are more than $n^{\frac{3-\gamma}{2}}$ nodes in V' and all nodes in V' have weight $\geq \sqrt{\alpha_1 n / \log n}$. This implies that $w_i w_j \geq \frac{\alpha_1 n}{\log n}$ and the edge probability between nodes $i, j \in V'$ is

$$p_{ij} = \Omega\left(\frac{w_i w_j}{n + w_i w_j}\right) = \Omega\left(\frac{1}{\log n}\right).$$

By choosing $g(n) = \frac{\log\log n - \Theta(1)}{\log n}$ suitably, we can write the edge probability as $p := p_{ij} \geq n^{-g(n)}$.

A k-partition is a k-clique with probability $\geq p^{\binom{k}{2}}$. The probability of not finding a clique before the exhaustive search is thus

$$\leq \left(1 - p^{\binom{k}{2}}\right)^{\left\lfloor n^{\frac{3-\gamma}{2}}/k \right\rfloor} \leq \exp\left(-\left\lfloor \frac{n^{\frac{3-\gamma}{2}}}{k} \right\rfloor p^{\binom{k}{2}}\right)$$

$$\leq \exp\left(-\frac{n^{\frac{3-\gamma}{2}}}{2k} p^{\binom{k}{2}}\right) = \exp\left(-\frac{n^{\frac{3-\gamma}{2} - g(n)\binom{k}{2}}}{2k}\right),$$

since we have $k \leq \log^{\frac{1}{3}} n \leq n^{\frac{3-\gamma}{2}}/2$ for large n and therefore $\lfloor p_1(n)/k \rfloor \geq p_1(n)/k - 1 \geq p_1(n)/(2k)$.

Since $k \leq \log^{\frac{1}{3}} n$, we can also assume that $k \leq \sqrt{(3-\gamma)/(10\, g(n))}$ holds for large n. Similarly, we have that $k \leq n^{\frac{3-\gamma}{20}}/2$. Then, we obtain $\binom{k}{2} g(n) \leq (3-\gamma)/10$ and thus $\frac{3-\gamma}{2} - g(n)\binom{k}{2} \geq \frac{3-\gamma}{10}$. Therefore $n^{\frac{3-\gamma}{2} - g(n)\binom{k}{2}} \geq n^{\frac{3-\gamma}{10}}$. Hence, as $2k < n^{\frac{3-\gamma}{20}}$, it follows that the probability of doing the exhaustive search is $\leq \exp\left(-n^{\frac{3-\gamma}{20}}\right)$. □

5 Conclusion

Social networks are becoming ubiquitous. There is a significant body of research on the structural properties of such networks, but very little on how this can

be exploited algorithmically. We have shown that for scale-free networks with n nodes and power-law exponent γ, the notoriously hard k-Clique problem becomes parameterized tractable for $2 < \gamma < 3$ (runtime $\mathcal{O}(n \exp(k^4))$) and even polynomial time solvable for $\gamma \geq 3$ (runtime $\mathcal{O}(n^2)$). In the future, we plan to improve the latter runtime bound for the greedy algorithm, as well as examine other NP-hard combinatorial problems (e.g. from bioinformatics) on scale-free networks.

Acknowledgements. We thank Jiong Guo and Danny Hermelin for helpful discussions. This work is part of the project "Average-Case Analysis of Parameterized Problems and Algorithms (ACAPA)" of the German Research Foundation (DFG).

References

[1] Aiello, W., Chung, F., Lu, L.: A random graph model for massive graphs. In: 32nd Symp. Theory of Computing (STOC), pp. 171–180 (2000)
[2] Aiello, W., Chung, F., Lu, L.: A random graph model for power law graphs. Experimental Mathematics 10(1), 53–66 (2001)
[3] Alon, N., Krivelevich, M., Sudakov, B.: Finding a large hidden clique in a random graph. In: 9th Symp. Discrete Algorithms (SODA), pp. 594–598 (1998)
[4] Ben-Dor, A., Shamir, R., Yakhini, Z.: Clustering gene expression patterns. Journal of Computational Biology 6(3-4), 281–297 (1999)
[5] Bianconi, G., Marsili, M.: Number of cliques in random scale-free network ensembles. Physica D: Nonlinear Phenomena 224(1-2), 1–6 (2006)
[6] Chung, F., Lu, L.: Connected components in random graphs with given expected degree sequences. Annals of Combinatorics 6(2), 125–145 (2002)
[7] Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. SIAM Review 51(4), 661–703 (2009)
[8] Fellows, M.R., Downey, R.G.: Parameterized Complexity. Springer (1998)
[9] Fountoulakis, N., Friedrich, T., Hermelin, D.: On the average-case complexity of parameterized clique (unpublished manuscript)
[10] Grimmett, G.R., McDiarmid, C.J.H.: On colouring random graphs. Mathematical Proceedings of the Cambridge Philosophical Society 77(02), 313–324 (1975)
[11] Håstad, J.: Clique is hard to approximate within $n^{1-\varepsilon}$. Acta Mathematica 182(1), 105–142 (1999)
[12] Janson, S., Łuczak, T., Rucinsky, A.: Random Graphs. John Wiley & Sons (2000)
[13] Janson, S., Łuczak, T., Norros, I.: Large cliques in a power-law random graph. J. Appl. Probab. 47(4), 1124–1135 (2010)
[14] Jerrum, M.: Large cliques elude the metropolis process. Random Structures & Algorithms 3(4), 347–360 (1992)
[15] Juels, A., Peinado, M.: Hiding cliques for cryptographic security. Designs, Codes And Cryptography 20(3), 269–280 (2000)
[16] Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of Computer Computations, pp. 85–103. Plenum Press (1972)
[17] Kučera, L.: Expected complexity of graph partitioning problems. Discrete Applied Mathematics 57(2-3), 193–212 (1995)

[18] Levin, L.A.: Average case complete problems. SIAM Journal on Computing 15(1), 285–286 (1986)
[19] Luce, R., Perry, A.: A method of matrix analysis of group structure. Psychometrika 14(2), 95–116 (1949)
[20] Mitzenmacher, M.: A brief history of generative models for power law and lognormal distributions. Internet Mathematics 1(2), 226–251 (2004)
[21] Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45(2), 167–256 (2003)
[22] Newman, M.E.J.: Power laws, Pareto distributions and Zipf's law. Contemporary Physics 46(5), 323–351 (2005)
[23] Norros, I., Reittu, H.: On a conditionally Poissonian graph process. Advances in Applied Probability 38(1), 59–75 (2006)
[24] Peinado, M.: Hard graphs for the randomized Boppana-Halldórsson algorithm for maxclique. Nordic Journal of Computing 1(4), 493–515 (1994)
[25] Peinado, M.: Go with the winners algorithms for cliques in random graphs. In: 12th Intl. Symp. Algorithms and Computation (ISAAC), pp. 525–536 (2001)
[26] Rossman, B.: On the constant-depth complexity of k-clique. In: 40th Symp. Theory of Computing (STOC), pp. 721–730 (2008)
[27] Rossman, B.: The monotone complexity of k-clique on random graphs. In: 51st Symp. Foundations of Computer Science (FOCS), pp. 193–201 (2010a)
[28] Rossman, B.: Average-case complexity of detecting cliques. PhD thesis, Massachusetts Institute of Technology (2010b)
[29] Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. Proceedings of the National Academy of Sciences 100(21), 12123–12128 (2003)
[30] van der Hofstad, R.: Random graphs and complex networks (2009), `http://www.win.tue.nl/~rhofstad/NotesRGCN.pdf`
[31] Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. Theory of Computing 3(1), 103–128 (2007)