UNIVERSITÄT DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

SONDERFORSCHUNGSBEREICH 531

Design und Management komplexer technischer Prozesse und Systeme mit Methoden der Computational Intelligence

Finding Large Cliques in Sparse Semi-Random Graphs by Simple Randomized Search Heuristics

Tobias Storch

Nr. CI-211/06

Interner Bericht ISSN 1433-3325 June 2006

Sekretariat des SFB 531 · Universität Dortmund · Fachbereich Informatik/XI
44221 Dortmund · Germany

Diese Arbeit ist im Sonderforschungsbereich 531, „Computational Intelligence“, der Universität Dortmund entstanden und wurde auf seine Veranlassung unter Verwendung der ihm von der Deutschen Forschungsgemeinschaft zur Verfügung gestellten Mittel gedruckt.

Finding Large Cliques in Sparse Semi-Random Graphs by Simple Randomized Search Heuristics

Tobias Storch *

Department of Computer Science 2
University of Dortmund
44221 Dortmund, Germany
`tobias.storch@uni-dortmund.de`

Abstract

Surprisingly, general heuristics often solve hard combinatorial problems quite sufficiently, although they do not outperform specialized algorithms. Here, the behavior of simple randomized optimizers on the maximum clique problem is investigated. We focus on semi-random models for sparse graphs, in which an adversary is even allowed to insert a limited number of edges and not only to remove them. In the course of these investigations also the approximation behavior on general graphs and the optimization behavior on sparse graphs and further semi-random graph models are considered. With regard to the optimizers particular interest is given to the influences of the population size and the search operator.

Key words: maximum clique problem, semi-random graph, randomized local search, evolutionary algorithm, population size, run time analysis

1 Introduction

One of the best-known combinatorial optimization problems is to find a maximum clique in a simple undirected graph. A *clique* is a subset of vertices, where each two vertices are connected by an edge, and a *maximal clique* is a clique not contained in any larger clique. At last, a *maximum clique* is a (maximal) clique of maximum cardinality. The task to find a large clique in a graph is of practical and theoretical importance.

* This work was supported by the Deutsche Forschungsgemeinschaft (DFG) as a part of the Collaborative Research Center "Computational Intelligence" (SFB 531).

The maximum clique problem was among the first problems proven to be **NP**-hard [12]. However, the best-known exact algorithms find a maximum clique in $\mathcal{O}(1.189^n)$ time and exponential space or in $\mathcal{O}(1.203^n)$ time and polynomial space [14]. There exist several common ways to cope with this worst-case complexity. One possibility is to investigate approximative solutions. Unfortunately, even an approximation by a factor of $n^{1-\mathcal{O}((\log\log n)^{-1/2})}$ in polynomial time cannot be achieved under some reasonable complexity theoretical assumptions [5]. The best-known approximation algorithm finds a clique approximating a maximum clique by a factor of $\mathcal{O}(n(\log\log n)^2/(\log n)^3)$ in polynomial time [6]. Another way to handle such a worst-case complexity is the consideration of subclasses of all graphs (e.g., sparse graphs, where the number of edges is limited) or to investigate the average-case behavior with respect to probability distributions on all graphs [9]. E.g., random graphs with a probability of 1/2 for inserting an edge, where it is known that even a simple greedy algorithm typically finds a clique approximating a maximum clique by a factor close to 2. Unfortunately, no polynomial time algorithm is known which does significantly better with a probability of at least 1/2. Just if a clique of size $k(n) = \Omega(n^{1/2})$ is planted in the random graph, i.e., $k(n)$ randomly chosen vertices are forced to be a clique, polynomial time algorithms exist which usually find a maximum clique [1]. However, even for $k(n)$ slightly larger than $2\log n$ the planted clique is normally the maximum clique.

Sparse Semi-Random Graphs

Random models are due to explain the success of algorithms on real-world instances, but such inputs are generally not as well-formed as random ones. In order to enrich and to robustify random models, semi-random models are considered, where two main variants exist. The first variant allows an adversary to present an arbitrary input, but this is modified moderately at random. The second variant presents a random input, but an adversary is allowed to vary it within limits [7]. These models generate combinations of average- and worst-case instances. E.g., planted random graphs with $k(n) = \Omega(n^{1/2})$, the planted maximum clique is usually even found in polynomial time, if an adversary is allowed to remove arbitrary edges outside the planted clique. Moreover, it is also certified that the clique found is maximum [8]. Such modifications should at least ease to find a maximum clique, however, many algorithms fail. Thus, semi-random models are typically more adequate to distinguish between naïve and more sophisticated algorithms. In this spirit and as an extension, an adversary could be allowed to behave moderately harmful and not only helpful. So, an adversary may naturally be permitted to insert a limited number of edges in a random graph. These little changes should typically be not very misleading and further strengthen the semi-random graph models. A major motivation for investigating especially sparse semi-random graphs is based on the observation that many real-world inputs are quite sparse, but far away from being sparse random. This motivates to consider a quite powerful adversary.

Simple Randomized Search Heuristics

The area of applications of randomized search heuristics is as huge as their variety. On a wide range of real-world clique inputs a remarkable experimental success of such optimizers and hybrid algorithms is reported [3],[11]. Theoretically only little is known about these heuristics and in particular about their behavior in combinatorial optimization. Concerning the maximum clique problem an outstanding exception is an article of Jerrum [10]. He investigated the Metropolis process for finding large cliques in random graphs. In contrast to hill-climbers, the Metropolis process also accepts randomly with respect to its so-called temperature an individual with a worse function value. Jerrum proves the process' super-polynomial runtime even to find a clique approximating a maximum one by a factor of slightly better than 2. This holds for planted random graphs and denser graphs, too. The occasional acceptance of slightly worse elements is one strategy to overcome local optima. Two further well-known methods are the usage of a larger population and the application of a global search operator. We focus on (the effects of) these two popular strategies and thereby deepen the insight of the effectiveness of also more complex heuristics on real-world instances for the maximum clique problem. Beside the Metropolis algorithm the probably best-known types of the broad class of general search heuristics are the randomized local search algorithms (RLSs) and evolutionary algorithms (EAs). Understanding their successes, working principles, and the considered problems' structure is a major motivation for the analyses of simple optimizers' behavior. General search heuristics are not problem-specific – and therefore, they can be applied to a wide range of even not well-understood problems without modifications. So, we doubt that they outperform problem-specific algorithms. But since they are typically easy to implement, these algorithms have applications.

The Objective Functions

Since randomized search heuristics are intended to optimize objective functions $f : S \to \mathbb{R}$, such a fitness function has to be designed for the clique problem. The search space $S = \{0,1\}^{|V|}$ seems to be a canonical choice, where an element is interpreted as characteristic vector of the graph's vertices V in an arbitrary order. Using the size of the clique as function value, if the subset represents a clique, the aim is maximization. Choosing $-\infty$ (or -1) as function value, if the subset does not represent a clique, leads for typical randomized optimizers using random initialization even on the empty graph to inefficiency. This can be overcome by an initialization with empty cliques

[10] or by applying for a graph $G = (\{v_1, \ldots, v_n\}, E)$ the canonical function:

$$\textsc{MaximumClique}_G(x) := \begin{cases} +\|x\| & \text{if } \{v_i \,|\, x_i = 1, 1 \le i \le n\} \text{ is } \quad\;\; \text{a clique} \\ -\|x\| & \text{if } \{v_i \,|\, x_i = 1, 1 \le i \le n\} \text{ is not a clique} \end{cases}$$

Here, $\|x\| := \sum_{i=1}^{n} x_i$ for a bit-string $x = x_1 \cdots x_n$ and $|x| := n$ denotes the length of x. Outside cliques the function values direct to the empty clique and the objective function $\textsc{MaximumClique}_G$ can be evaluated efficiently. In the following, let $\mathcal{C}^{\ge k(n)}(G) \subseteq \mathcal{P}(V)$ denote the set of all cliques of size at least $k(n)$ in a graph G and let $\omega(G)$ denote the size of the maximum cliques.

The Simple Randomized Search Heuristics studied

Whereas RLSs come up with local search operators, EAs search more globally. Another vigorousness of EAs is the application of a population. One of our aims is to analyze the (interactive) effects of the choice of the population size and the search operator. In order to concentrate thereon we consider simple heuristics that support these analyses, but avoid unnecessary complications due to the effects of other optimizer's components. Therefore, we investigate the following RLS and EA presented by Storch [15], where a hierarchy result for the population size on artificial example functions was proven. A mutation probability of $1/n$ is a standard choice [4]. Since the algorithms avoid duplicates of elements in the population the population structure is a set and not only a multiset.

(μ+1) RLS and **(μ+1) EA**

(1) Choose μ different individuals $x_{[i]} \in \{0,1\}^n$, $1 \le i \le \mu$, uniformly at random. These individuals constitute the population $\boldsymbol{P} = \{x_{[1]}, \ldots, x_{[\mu]}\}$.
(2) Choose an individual $x \in \boldsymbol{P}$ uniformly at random and create y by flipping
(μ+1) RLS : a bit in x chosen uniformly at random.
(μ+1) EA : each bit in x with probability $1/n$.
(3) If $y \notin \boldsymbol{P}$, then let $z \in \boldsymbol{P} \cup \{y\}$ be randomly chosen among those individuals with the worst $\textsc{MaximumClique}_G$-value and let the population be $\boldsymbol{P} \cup \{y\} \setminus \{z\}$, goto 2., and else let the population be $\boldsymbol{P}$, goto 2.

Such algorithms are called *efficient* on $\textsc{MaximumClique}_G : \{0,1\}^n \to \mathbb{R}$ if their expected number of steps to evaluate for the first time an optimum is bounded above by a polynomial in n.

The reader can easily verify that all upper bounds hold for arbitrary initialization strategies. Moreover, the (similar) lower (and upper) bounds follow directly (even more simply) for initialization with empty cliques.

Contributions and Further Article Structure

In Section 3 we investigate the just now specified simple randomized search heuristics on the afore developed new type of semi-random graph model. However, it is a natural challenge to consider such optimizers due to their simplicity and their relation to more complex real-world heuristics. Even the semi-random instances considered in detail model real-world inputs more appropriately than more traditional ones. With a focus on sparse instances the tight runtime bounds exhibit how much random and adversarial power the optimizers can bear. Such considerations are also performed for more traditional semi-random graph models which allows a comparison with further algorithms. These first runtime analyses of general optimizers on semi-random inputs give additionally – even in a very general setting – a first proof of the possible major advantage of the application of a large population in combinatorial optimization. This was an outstanding open problem for a long time. However, we begin with worst-case analyses in Section 2. These are helpful for the semi-average case analyses and interesting for themselves. We end with a summary and some conclusions in Section 4.

2 Analyses for Worst Inputs

We begin with worst-case analyses of the (μ+1) RLS and the (μ+1) EA on general graphs in Section 2.1 and in particular on sparse graphs in Section 2.2.

2.1 General Graphs

Let us investigate how long it takes to generate an individual representing a clique for the first time. We take all graphs into account but the expectation over the algorithms' random choices. The following considerations for the (μ+1) RLS resp. (μ+1) EA are comparable to those by [4] for the (1+1) EA on $\|x\|$. While the population does not contain the empty clique or it consists of cliques only, with a probability of at least $1/\mu \cdot \|x\| \cdot 1/n(1-1/n)^{n-1} \geq \|x\|/(e\mu n)$ resp. $1/\mu \cdot \|x\| \cdot 1/n \geq \|x\|/(\mu n)$ the following happens. An element of the population x with minimal number of ones is selected for mutation resp. bit flip. This individual x creates a search point with a larger function value than each element not representing a clique. At least $\|x\|$ specific 1-bit mutations resp. flips of x which change a one to a zero do so. Such an offspring is included in the population for sure. After at most n such function value increases the empty clique is generated or the whole population consists of cliques. Hence, the expected number of steps is bounded above by $\sum_{i=1}^{n} e\mu n/i = \mathcal{O}(\mu n \log n)$ resp. $\sum_{i=1}^{n} \mu n/i = \mathcal{O}(\mu n \log n)$. Afterwards, the population spreads among elements representing cliques. While not the whole population consists of cliques only or contains a maximum clique, a maximum clique of G is generated within at most $\omega(G)$ mutations of a currently largest clique in the population which is a subset of a maximum clique. The expected number of steps is bounded by $\mathcal{O}(\mu n \log \omega(G)) = \mathcal{O}(\mu n \log n)$.

If the population is large enough, namely $\mu \geq |\mathcal{C}^{\geq 0}(G)| - 3$, then for sure an element representing a maximum clique of G is created in an expected number of $\mathcal{O}(\mu n \log n)$ steps – even for the (μ+1) RLS. Such a population size ensures that always a subset of a maximum clique is contained in the population. This holds since a maximum clique of size at least 2 contains at least 4 subsets. And for the empty graph even a population of size $\mu = 1 \leq (n+1)-3 = |\mathcal{C}^{\geq 0}(G)|-3$ is sufficient.

Let us summarize these investigations by the following lemma.

Lemma 1 *Let G be an arbitrary graph.*

(a) *The* (μ+1) RLS *and the* (μ+1) EA *create in an expected number of $\mathcal{O}(\mu n \log n)$ steps a maximum clique or the whole population consists of cliques.*
(b) *The* (μ+1) RLS *and the* (μ+1) EA *create in an expected number of $\mathcal{O}(\mu n \log n)$ steps a maximum clique, if $\mu \geq |\mathcal{C}^{\geq 0}(G)| - 3$.*

We remark that the (μ+1) RLS needs an infinite expected number of steps to find the maximum clique of $G = (\{v_1, \ldots, v_n\}, \{\{v_1, v_2\}\})$ which is $\{v_1, v_2\}$, if $\mu < |\mathcal{C}^{\geq 0}(G)| - 3 = n - 1$. This is the case since with positive probability the population is initialized with the cliques $\{v_3\}, \ldots, \{v_{\mu+2}\}$. Afterwards, no 1-bit flip is accepted. Let us come to worst-case upper bounds for the investigated algorithms. These are helpful while considering semi-random inputs in the following section, too.

Theorem 2 *For every graph G the expected number of steps until a clique of size at least $s(n) \leq \omega(G)$ is created by the* (μ+1) EA *is bounded above by*

$$\mathcal{O}\Big(n^{2s(n)-2}\binom{\omega(G)}{s(n)-1}^{-1} + \mu n \log n\Big) \quad .$$

PROOF. By Lemma 1.(a) in an expected number of $\mathcal{O}(\mu n \log n)$ steps a maximum clique is found or the whole population consists of cliques. In the first case or if the population contains an $s(n)$-clique the aim of the theorem is reached. Otherwise, by counting just the subsets of size $s(n) - 1$ of one maximum clique, the number of $(s(n) - 1)$-cliques which are subsets of a larger clique is bounded below by $\binom{\omega(G)}{s(n)-1}$. Since for every element x in the population it is $\|x\| \leq s(n) - 1$ and for each such clique y it is $\|y\| = s(n) - 1$, at most $2s(n) - 2$ specific bits of x have to flip to generate y. Hence, the probability is bounded below by $\binom{\omega(G)}{s(n)-1}/(en^{2s(n)-2})$ to create an arbitrary such clique. To obtain the desired result and in this situation, it is sufficient to bound the probability by $\Omega(1)$ to create a larger clique of size at least $s(n)$ prior to remove a non-maximal $(s(n)-1)$-clique by a maximal $(s(n)-1)$-clique. In the case of a failure we can repeat the argumentation. We observe that if the population contains cliques smaller than $s(n) - 1$, these will be removed prior to $(s(n) - 1)$-cliques. With a probability of at least $1/(e\mu n)$ a clique of size at least $s(n)$ is created. Since a specific $(s(n) - 1)$-clique is removed with a probability of at most $1/(\mu + 1)$, it is sufficient to bound the probability by $\mathcal{O}(1/n)$ to create a maximal $(s(n) - 1)$-clique.

Therefore, let V_1 denote the set of vertices of an arbitrary clique – here, of size $s(n)-1$. For each $S_0 \subseteq V \setminus V_1$ there exists at most one set $S_1 \subseteq V_1$, where $|S_0| = |S_1|$, such that $(V_1 \setminus S_1) \cup S_0$ forms a maximal clique of size $s(n) - 1$. Assume that there exist two different sets S_1 and S_2, where $|S_1| = |S_2|$ for which this holds. Afterwards, let $a \in S_2$, but $a \notin S_1$, then also $(V_1 \setminus S_1) \cup S_0 \cup \{a\}$ forms a clique of size $s(n)$ what is a contradiction since $(V_1 \setminus S_1) \cup S_0$ is assumed to be maximal. Furthermore, for each individual of the population the probability to create an element which represents a maximal clique of size $s(n)-1$ is bounded by $\sum_{k=1}^{s(n)-1} \binom{n}{k} \cdot 1/n^{2k}(1 - 1/n)^{n-2k} \leq \sum_{k\geq 1} n^k \cdot 1/n^{2k} = \sum_{k\geq 1} 1/n^k \leq 2/n$. $\square$

Let us combine the ideas of Lemma 1.(b) and Theorem 2, namely the strengths of a population and a global search operator. Therefore, we observe that whenever the population contains an individual representing a small clique, a larger clique which is also a subset of a maximum clique is created fast – assuming such a clique exists. So, let $|\mathcal{C}^{\geq s(n)}(G)| \geq 1$ for an arbitrarily chosen $s(n)$. In an expected number of $\mathcal{O}(\mu n \log n)$ steps a maximum clique is created or the population consists of cliques only – for the (μ+1) EA. Afterwards and if $\mu \geq 2|\mathcal{C}^{\geq s(n)}(G)|$, then we can bound the number of elements which represent cliques of size at most $s(n) - 1$ by $\mu/2$ from below. Thus, the probability is bounded below by $(\mu/2)/\mu$ to select such an element. A mutation of such an element creates one representing a clique of size at least $s(n)$ which is a subset of a maximum clique with a probability of at least $1/n^{2s(n)-1}(1-1/n)^{n-2s(n)+1} \geq 1/(en^{2s(n)-1})$. Individuals that represent cliques of size at least $s(n)$ are inserted in the population and never removed from it. Finally, in an expected number of $\mathcal{O}(\mu n \log n)$ steps a maximum clique is generated.

Let us summarize these investigations by the following corollary.

Corollary 3 *Let G be an arbitrary graph. If $\mu \geq \max\{2|\mathcal{C}^{\geq s(n)}(G)|, 1\}$, the* ($\mu$+1) EA *creates in an expected number of $\mathcal{O}(n^{2s(n)-1} + \mu n \log n)$ steps a maximum clique.*

In order to review these general upper bounds, we first make the following observation. By Lemma 1.(b) the expected number of steps of the (μ+1) RLS is bounded above by $\mathcal{O}(\mu n \log n)$ to create a clique of size at least $s(n)$, if $\mu \geq 1$ and $s(n) \leq 1$, or, if $\mu \geq \sum_{i=0}^{s(n)-1} \binom{n}{i} - 1$ and $s(n) \geq 2$. Furthermore, $\sum_{i=1}^{s(n)-1} \binom{n}{i} \leq n^{s(n)-1}/(s(n)-2)! \leq \left(\frac{3n}{s(n)-1}\right)^{s(n)-1}$. However, we will present graphs G, where the (μ+1) RLS with $\mu \leq \left(\frac{n/2}{s(n)-1}\right)^{s(n)-1}$ needs an infinite expected number steps. Moreover, the (1+1) EA needs an expected number of $\Omega(n^{2s(n)-2}\binom{\omega(G)}{s(n)-1}^{-1} + n \log n)$ steps to create a clique of size at least $s(n) \leq \omega(G)$. To demonstrate this, we need the following technical lemma.

Lemma 4 *For a graph G let V_0 and V_1 be the elements of a partition of V, where $|V_0| = |V|/2$. If there exist edges in V_0 and in V_1 only, then the* (1+1) EA *creates each clique of size* 1 *with probability* $\Omega(1/n)$ *as population.*

PROOF. (*sketch of*) (*See Appendix A for a full proof.*) Let the first $|V|/2$ bits of an element represent the vertices of V_0. We observe that each search point does not represent a clique, if its first and second half consist of at least a single one. Some crude investigations show, that with probability $\Omega(1)$ the (1+1) EA first creates a bit-string consisting of zeros in one of its halves only, when the other half contains exactly a single one. Because of symmetry, the result follows. □

Theorem 5 *There exist graphs G with $\omega(G) \leq n/2$, where the expected number of steps until a clique of size at least $s(n)$ is created by the algorithm A is bounded below by $t(n)$, where $s(n)$, A, and $t(n)$ are given in the following table. Let $\ell := \lfloor n/(2s(n)-2) \rfloor \geq 1$.*

size $s(n)$	*algorithm* A	*time* $t(n)$
$s(n) \leq \omega(G)$	(1+1) EA	$\Omega(n^{2s(n)-2}\binom{\omega(G)}{s(n)-1}^{-1} + n \log n)$
$2 \leq s(n) \leq \omega(G)$	(μ+1) RLS, $\mu \leq (\ell+1)^{s(n)-1} - 1$	∞

PROOF. For a maximum clique of size $k(n)$, let $V_i := \{v_{(i-1)\cdot\ell+1}, \ldots, v_{i\cdot\ell}\}$, $1 \leq i < s(n)$, $E_0 := \{\{v_i, v_j\} \mid v_i \in V_i, v_j \in V_j, 1 \leq i < j < s(n)\}$, and $E_1 := \{\{v_i, v_j\} \mid n - k(n) < i < j \leq n\}$ for $k(n) \geq s(n)$. We investigate the graph $W_{n,k(n),s(n)-1} := (V, E_0 \cup E_1)$ (see Figure 1 for an illustration). There are $\sum_{i=0}^{s(n)-1} \binom{s(n)-1}{i} \cdot \ell^i = (\ell+1)^{s(n)-1}$ cliques in the range of 0 to $s(n)-1$ formed by edges of E_0 whereas the edges of E_1 form the maximum clique. Therefore, $\omega(W_{n,k(n),s(n)-1}) = k(n)$.

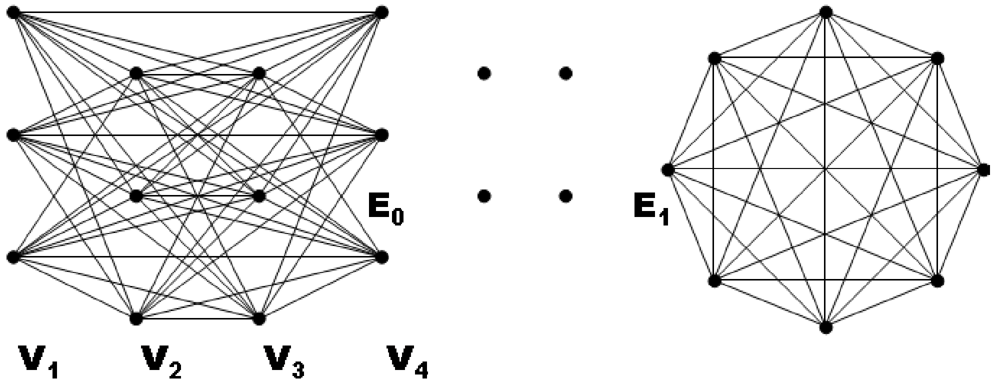

Fig. 1. An illustration of $W_{24,8,4}$.

Let us investigate the (μ+1) RLS, where $\mu \le (\ell+1)^{s(n)-1} - 1$. With positive probability its population is initialized with cliques on $\bigcup_{i=1}^{s(n)-1} V_i$ except the empty clique. Afterwards, each such element has a Hamming distance of at least 2 to a clique which is a subset of the maximum clique. This leads to an infinite expected number of steps.

Let us investigate the (1+1) EA. By Lemma 4 each 1-clique is created as population with probability $\Omega(1/n)$. Moreover, by [4] an expected number of $\Omega(n \log n)$ steps is needed to create an element which consists of a single one. Such an individual is generated with probability $\Omega(1/n) \cdot n = \Omega(1)$. Hence, for $s(n) \le 1$ the aim of the theorem is reached. Let $s(n) \ge 2$. With probability $\ell \cdot (s(n)-1) \cdot \Omega(1/n) = \Omega(1)$ a 1-clique $S \subseteq \{v_1, \ldots, v_{\ell\cdot(s(n)-1)}\}$ is generated. Moreover, the probability is bounded above by $1/n^{|S|}$ to create any clique S', where $S' \subseteq \{v_{\ell\cdot(s(n)-1)+1}, \ldots, v_n\}$, whereas the probability is bounded below by $1/(en)$ to create a larger clique on $\{v_1, \ldots, v_{\ell\cdot(s(n)-1)}\}$, if $|S| < s(n)-1$. Since $\sum_{i=1}^{s(n)-2} \frac{1/n^i}{1/n^i + 1/(en)} = \sum_{i=1}^{s(n)-2} \frac{e}{e+n^{i-1}} \le 3/4$, for n large enough, the probability is bounded below by $\Omega(1)$ to create an $(s(n)-1)$-clique $S \subseteq \{v_1, \ldots, v_{n-\ell\cdot(s(n)-1)}\}$ prior to a clique $S' \subseteq \{v_{n-\ell\cdot(s(n)-1)+1}, \ldots, v_n\}$. Afterwards, the probability to generate a non-maximal clique of size $s(n)-1$ or a clique of size at least $s(n)$ is bounded above by

$$\sum_{i=s(n)-1}^{k(n)} \binom{k(n)}{i} 1/n^{2i} \le \sum_{i=0}^{\infty} \binom{k(n)}{s(n)-1} n^i / n^{2(s(n)-1+i)} \le 2\binom{k(n)}{s(n)-1} / n^{2s(n)-2}$$

since $\binom{k(n)}{s(n)-1+i} \le \binom{k(n)}{s(n)-1} n^i$ and $\sum_{i\ge 0} 1/n^i \le 2$. $\square$

2.2 Sparse Graphs

Recently, Storch [16] has investigated how various popular simple randomized search heuristics find maximum cliques in planar graphs in the worst- and average-case. Planar graphs are necessarily quite sparse. Let us investigate

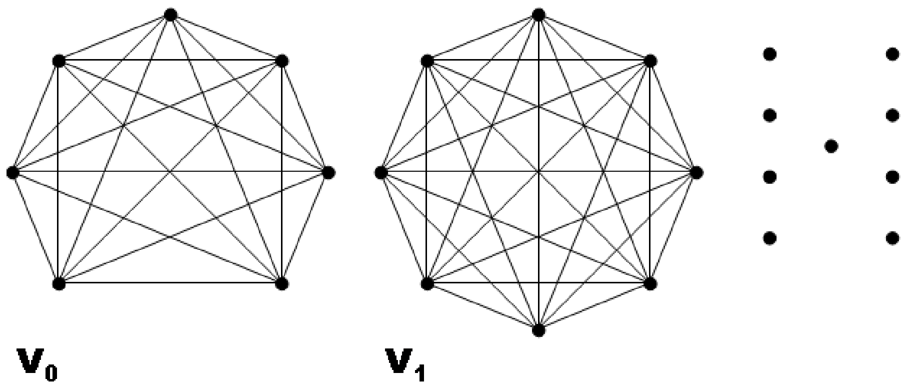

Fig. 2. An illustration of $F_{24,12}$.

the class of all graphs which are sparse, namely where the number of edges is limited. This represents a subclass of all graphs. A proof by induction demonstrates, that for an arbitrary graph G it is $|\mathcal{C}^{\geq 2}(G)| \leq 2^{k(n)} - k(n) - 1$, where the number of edges is bounded above by $\binom{k(n)}{2}$. (See Appendix B for a full proof.) By Lemma 1.(b) the following theorem follows directly.

Theorem 6 *Let G be an arbitrary graph, where $|E| \leq \binom{k(n)}{2}$. If $\mu \geq 2^{k(n)} - k(n) + n - 3$, the* ($\mu$+1) RLS *and the* ($\mu$+1) EA *create in an expected number of $\mathcal{O}(\mu n \log n)$ steps a maximum clique.*

However, if we consider the (1+1) EA we observe the following. Since the maximum clique in each graph with at most $\binom{k(n)}{2}$ edges is bounded above by $k(n)$, by Theorem 2 the expected number of steps until the (1+1) EA has created a maximum clique is bounded by $\mathcal{O}(n^{2k(n)-2}/k(n) + \mu n \log n)$. The following theorem demonstrates that this bound is somehow tight.

Theorem 7 *There exist graphs G with $|E| \leq \binom{k(n)}{2}$, $3 \leq k(n) \leq n/\sqrt{2}$, where the expected number of steps until a maximum clique is created by the* (1+1) EA *is bounded by $\Omega(n^{\sqrt{2}k(n)-2}/k(n))$.*

PROOF. Let us investigate the graph $F_{n,k(n)} := (V, \{\{v_i, v_j\} \mid 1 \leq i < j \leq k(n)/\sqrt{2} - 1$ or $k(n)/\sqrt{2} \leq i < j \leq \sqrt{2}k(n) - 1\})$ (see Figure 2 for an illustration). This graph consists of $\binom{k(n)/\sqrt{2}-1}{2} + \binom{k(n)/\sqrt{2}}{2} \leq \binom{k(n)}{2}$ edges. Let $V_0 := \{v_1, \ldots, v_{k(n)/\sqrt{2}-1}\}$ and $V_1 := \{v_{k(n)/\sqrt{2}}, \ldots, v_{\sqrt{2}k(n)-1}\}$. By Lemma 4 the (1+1) EA creates each 1-clique S as population with probability $\Omega(1/n)$. If $S = \{v\} \subseteq V \setminus (V_0 \cup V_1)$, then

(a) with a probability of at least $(k(n)/\sqrt{2}-1)/n^2(1-1/n)^{n-2} \geq (k(n)/\sqrt{2}-1)/(en^2)$ one of the 1-cliques $\{v\} \subseteq V_0$ is created and
(b) with a probability of at most $(k(n)/\sqrt{2})/n \cdot 1/n = (k(n)/\sqrt{2})/n^2$ a clique S', $\emptyset \neq S' \subseteq V_1$, is created.

In this situation the probability is bounded by $\Omega(1)$ to perform (a) prior to (b). Therefore, we can bound the probability by $\Omega(1)$ to create and accept as population a 1-clique $\{v\} \subseteq V_0$ prior to generate a clique S', in total. This holds since $(k(n)/\sqrt{2}-1) \cdot \Omega(1/n) + (n - \sqrt{2}k(n)+1) \cdot \Omega(1/n) \cdot \Omega(1) = \Omega(1)$. Afterwards and equivalently to the proof of Theorem 5, the probability is bounded by $\Omega(1)$ to create the clique V_0 prior to a clique S', $S' \subseteq V_1$. And in this situation, the probability is bounded by $\mathcal{O}(k(n)/n^{\sqrt{2}k(n)-2})$ to create a non-maximal $(k(n)/\sqrt{2}-1)$-clique or the $(k(n)/\sqrt{2})$-clique. This leads to the proposed expected number of steps, in total. $\square$

We remark that the (μ+1) RLS and (μ+1) EA are efficient when applying a large enough (but still polynomial bounded) population, if $k(n) = \mathcal{O}(\log n)$, whereas the (1+1) EA can be inefficient, if $k(n) = \omega(1)$. This proves even a super-polynomial decrease of the expected number of steps applying a large population.

3 Analyses for Semi-Random Inputs

We begin with analyses on traditional semi-random graph models – and extensions thereof, too – for the maximum clique problem in Section 3.1 and come to the new, more powerful semi-random graph model mentioned in the introduction in Section 3.2. Here, the focus is on sparse graphs again.

3.1 Removing Edges

As mentioned in the introduction let us at first make precise the planted random graph model $\mathcal{G}_{n,p(n),k(n)}$ with a vertex set V of size n. This is a good starting point for semi-random graph considerations. Let $P \subseteq V$ be a randomly chosen subset of size $k(n) \geq 0$. Afterwards, the edges $\{v, w\}$, $v \neq w$, are inserted independently to the graph; with probability

- 1, if $v, w \in P$, (subset of edges E_P) and
- $p(n)$, otherwise (subset of edges $E_{\neg P}$).

For $k(n) = 0$ we obtain that the well-known random graph model $\mathcal{G}_{n,p(n)}$, where with high probability, namely with probability $1 - o(1)$, the maximum clique size equals asymptotically $2\log((1-p(n))n)/\log(1/p(n)) =: c_{n,p(n)}$, if $p(n) = 1 - \omega(1/n)$ (this is not a restriction when interested in sparse graphs) [9]. Thus, the planted clique is probable also the maximum clique, if $k(n)$ is at least slightly larger than $c_{n,p(n)}$. In the rest of the article we assume n to be large enough. Let us consider two types of adversaries having different degrees of freedom to modify the input and thereupon, the resulting optimization behavior of the investigated heuristics. We will investigate the (μ+1) EA on these semi-random graphs and discuss the possible advantage of a global search operator and a large population.

Removing Edges from a Small Set

In the first semi-random graph model $\mathcal{G}^*_{n,p(n),k(n)}$ an adversary is allowed to remove arbitrarily chosen edges of $E_{\neg P}$ (a small set) out of $\mathcal{G}_{n,p(n),k(n)}$ [8]. Theorem 8 shows an upper bound for the (μ+1) EA on $\mathcal{G}^*_{n,p(n),k(n)}$ whereas Theorem 9 demonstrates that this bound is somehow tight. In particular, the effects are considered when either the global search operator or the population is omitted.

Theorem 8 *With respect to* $\mathcal{G}^*_{n,n^{-\varepsilon(n)},k(n)}$ *the* (μ+1) EA *creates a maximum clique in an expected* (*with respect to the algorithm's random bits*) *number of*

(a) $\mathcal{O}(n^{9/\varepsilon(n)} + \mu n \log n)$ *steps with probability* $1 - n^{-\Omega(1/\varepsilon(n))}$ *and*
(b) $\mathcal{O}(n^{12/\varepsilon(n)+5} + \mu n \log n)$ *steps in expectation*

(*with respect to the input's random part*).

PROOF. We remark that $c_{n,n^{-\varepsilon(n)}}$ equals roughly $2/\varepsilon(n)$. At first, let us bound the probability from above for the following event $\mathcal{E}_i$, $i \geq 1$. There exist two disjoint sets of vertices S, T, where $|S| = |T| = i$, $S \subseteq V \setminus P$, $T \subseteq V$, and $\{\{s,t\} \mid s \in S, t \in T\} \subseteq E$. Let X_i be the random variable which describes the number of such pairs of disjoint sets. In order to estimate the probability that $\mathcal{E}_i$ occurs, we make use of the first moment method. Namely, applying Markov inequality [13] we obtain $\mathbf{Pr}[\mathcal{E}_i] = \mathbf{Pr}[X_i \geq 1] \leq \mathbf{E}[X_i]/1 \leq \binom{n-k(n)}{i}\binom{n}{i} n^{-\varepsilon(n)\cdot i^2} \leq n^{2i-\varepsilon(n)\cdot i^2}$ since $\binom{n-k(n)}{i} \leq \binom{n}{i} \leq n^i$.

By Theorem 2 in an expected number of $\mathcal{O}(n^{4i-2} + \mu n \log n)$ steps a maximum clique is created; or a clique of size at least $2i$ is created and the population consists of cliques only. Let us assume that $\mathcal{E}_i$ does not occur and that no maximum clique C is created. We observe that each clique C', where $|C'| < 2i$, creates a clique of size at least $|C'| + 1$ which is a subset of C in an expected number of $\mathcal{O}(n^{|C'|+(|C'|+1)}) = \mathcal{O}(n^{4i-1})$ steps. Moreover, for each clique C', where $|C'| \geq 2i$, it is $|C' \setminus P| \leq i - 1$. Since $|C| \geq |P|$ it holds $|P \setminus C| \leq i - 1$, too. Thus, while no maximum clique is created, in an expected number of $\mathcal{O}(n^{(i-1)+(i-1)+2(i-1)+1}) = \mathcal{O}(n^{4i-3})$ steps a clique is generated which is a subset of C and of size at least $|C'| + 1$.

We consider the following two cases.

- The population does not contain a subset of C. In an expected number of $\max\{\mathcal{O}(n^{4i-1}), \mathcal{O}(n^{4i-3})\} = \mathcal{O}(n^{4i-1})$ steps a clique of size at least $\ell+1$ is generated and inserted in the population, if the smallest clique in the population has size ℓ.
- The population contains a subset of C of size $\ell < |C|$. In an expected number of $\mathcal{O}(\mu n/(|C|-\ell))$ steps a subset of C of size at least $\ell+1$ is generated and inserted in the population.

Afterwards, the population always consists of a subset of C of size at least $\ell+1$ or each clique in the population is of size at least $\ell+1$. Hence, each of the cases occurs at most once for each $0 \le \ell < |C| \le n$. In total, this leads to an expected number of $\mathcal{O}(n^{4i-2}+\mu n\log n)+\sum_{\ell=0}^{|C|-1}(\mathcal{O}(n^{4i-1})+\mathcal{O}(\mu n/(|C|-\ell))) = \mathcal{O}(n^{4i}+\mu n\log n)$ steps to create a maximum clique, if $\mathcal{E}_i$ does not hold.

For part (a) we observe that $\mathcal{E}_{9/(4\varepsilon(n))}$ holds with a probability of at most $n^{9/(2\varepsilon(n))-\varepsilon(n)\cdot 81/(16\varepsilon(n)^2)} = n^{-9/(16\varepsilon(n))}$ and the result follows by the investigations made above.

For part (b) we obtain an expected number of at most

$$\begin{aligned}
&\sum_{i=1}^{n}\mathbf{Pr}[\mathcal{E}_{i-1} \text{ and } \neg\mathcal{E}_i]\cdot\Big(\mathcal{O}(n^{4i})+\mathcal{O}(\mu n\log n)\Big)\\
=&\,\mathcal{O}(\mu n\log n)+1\cdot\mathcal{O}(n^{4(3/\varepsilon(n))})\\
&\qquad+\sum_{i=3/\varepsilon(n)+1}^{n}[n^{2(i-1)-\varepsilon(n)\cdot(i-1)^2}\cdot 1]\cdot\mathcal{O}(n^{4i})
\end{aligned}$$

steps. An index transformation shows that this is equivalently to

$$\begin{aligned}
&\mathcal{O}(\mu n\log n)+\mathcal{O}(n^{12/\varepsilon(n)})+\sum_{i=0}^{n-3/\varepsilon(n)-1}\mathcal{O}(n^{9/\varepsilon(n)-\varepsilon(n)\cdot i^2+4})\\
=&\,\mathcal{O}(\mu n\log n)+\mathcal{O}(n^{12/\varepsilon(n)})+\sum_{i=0}^{n}\mathcal{O}(n^{9/\varepsilon(n)+4})\\
=&\,\mathcal{O}(n^{12/\varepsilon(n)+5}+\mu n\log n)
\end{aligned}$$

which proves the result. □

Theorem 9 *With respect to* $\mathcal{G}^*_{n,n^{-\varepsilon(n)},k(n)}$ *the algorithm* A *needs with high probability (with respect to the input's random part) an expected (with respect to the algorithm's random bits) number of* $t(n)$ *steps to create a maximum clique, where* A *and* $t(n)$ *are given in the following table. Let* $1/\log n \le \varepsilon(n) \le 1$ *and* $3/\varepsilon(n) \le k(n) \le n/2$.

algorithm A	time $t(n)$
(1+1) EA	$n^{\Theta(1/\varepsilon(n))}$
(μ+1) RLS, $\mu \leq n^{1/(5\varepsilon(n))}$	∞

PROOF. We remark that $\varepsilon(n)$ covers edge probabilities from 1/2 to tiny, where the maximum clique size is normally bounded above by 2. Moreover, $k(n)$ is just ensured to be typically the maximum clique. Let $V_0 \supseteq V \setminus P$, where $|V_0| = |V|/2$, be chosen uniformly at random and let $V_1 := V \setminus V_0$. We recall that P, where $|P| = k(n)$, is chosen uniformly at random among V. Thus, the graph on V_0 is a random one according to $\mathcal{G}_{n/2,n^{-\varepsilon(n)}}$ and with high probability all of the following events occur. If not all events occur, nothing will or has to be shown.

$\mathcal{E}_1$: The size of a maximum clique on V_0 is bounded above by $2.1/\varepsilon(n)$.
$\mathcal{E}_2$: The size of a smallest maximal clique on V_0 is bounded below by $0.9/\varepsilon(n)$.
$\mathcal{E}_3$: The number of cliques of size $1/\varepsilon(n)$ on V_0 is bounded below by $n^{1/(5\varepsilon(n))}$.

(See Appendix C for a proof of the probability bounds on the events.)

Let us investigate the (1+1) EA. For the upper bound, we apply Theorem 8. For the lower bound, we consider the graph, where all edges not in P or in V_0 only are removed. By $\mathcal{E}_1$ the planted clique is also the maximum clique since $2.1/\varepsilon(n) < 3/\varepsilon(n)$. Moreover, in this situation, by Lemma 4 the (1+1) EA creates each 1-clique with probability $\Omega(1/n)$. So, the probability is bounded below by $n/2 \cdot \Omega(1/n) = \Omega(1)$ to begin with one of the 1-cliques $\{v\} \subseteq V_0$. Afterwards, for a clique C of size $\ell < 0.9/\varepsilon(n)$, where $C \subseteq V_0$, the probability is bounded below by $1/(en)$ to create a clique C' of size larger than ℓ, where $C' \subseteq V_0$. This holds since C is not a maximal clique by $\mathcal{E}_2$. On the other hand, the probability is bounded above by $1/n^{\ell}$ to create a clique which is a subset of V_1. Similar to the proof of Theorem 7 a lower bound of $\Omega(1)$ follows for reaching a clique of size at least $0.9/\varepsilon(n)$ on V_0 prior to create a clique on V_1. Afterwards, at least $0.9/\varepsilon(n)$ specific bits on V_0 have to flip to create a subset of the maximum clique on V_1. In total, this results in an expected number of $\Omega(1) \cdot \Omega(n^{0.9/\varepsilon(n)})$ steps.

Let us investigate the (μ+1) RLS, where $\mu \leq n^{1/(5\varepsilon(n))}$. We consider the same modifications of the graph as above. By $\mathcal{E}_3$ there exist at least $n^{1/(5\varepsilon(n))}$ cliques on V_0 of size $1/\varepsilon(n) \geq 1$ since $\varepsilon(n) \leq 1$ and with positive probability the population of the (μ+1) RLS is initialized with these elements. Afterwards, each element in the population has a Hamming distance of at least 2 to a non-empty subset of the maximum clique. Thus, there is no possibility to create the maximum clique by any sequence of queries. $\square$

In the second semi-random graph model $\mathcal{G}^{**}_{n,p(n),k(n)}$ an adversary is allowed to remove arbitrarily chosen edges of $E_{\neg P} \cup E_P = E$ (a large set) out of $\mathcal{G}_{n,p(n),k(n)}$. Similar as in the previous section, Theorem 10 shows an upper bound for the (μ+1) EA on $\mathcal{G}^{**}_{n,p(n),k(n)}$, while Theorem 11 demonstrates that this bound is somehow tight.

Theorem 10 *With respect to* $\mathcal{G}^{**}_{n,n^{-\varepsilon(n)},k(n)}$*, if* $\varepsilon(n) \geq 2\log\log n/\log n$*, the* ($\mu$+1) EA*, where* $\mu \geq 2^{3k(n)}$*, creates a maximum clique in an expected (with respect to the algorithm's random bits) number of* $\mathcal{O}(n^{9/\varepsilon(n)-1} + \mu n \log n)$ *steps with probability* $1 - n^{-\Omega(1/\varepsilon(n))}$ *(with respect to the input's random part).*

PROOF. We remark that the values of $\varepsilon(n)$ cover even quite dense graphs. In the proof of Theorem 8 we have shown that the probability is bounded above by $n^{-\Omega(1/\varepsilon(n))}$ that the following event $\mathcal{E}_1$ occurs. There exist two disjoint sets of vertices S, T, where $|S| = |T| = 9/(4\varepsilon(n))$, $S \subseteq V \setminus P$, $T \subseteq V$, and $\{\{s,t\} \mid s \in S, t \in T\} \subseteq E$. If $9/(4\varepsilon(n)) \leq k(n) \leq 9/(4\varepsilon(n)) \log n$, then let us also consider the event $\mathcal{E}_2$ that there exist $2k(n)$ vertices of $V \setminus P$ that are adjacent to at least $9/(4\varepsilon(n))$ vertices of P. Let $\ell(n) := k(n) \cdot 4\varepsilon(n)/9$. Applying the first moment method as in the proof of Theorem 8 we obtain

$$\mathbf{Pr}[\mathcal{E}_2] \leq \binom{n-k(n)}{2k(n)} \binom{k(n)}{\frac{9}{4\varepsilon(n)}}^{2k(n)} n^{-\varepsilon(n)\cdot 2k(n)\cdot\frac{9}{4\varepsilon(n)}} \leq n^{2k(n)} (e\ell(n))^{\frac{9}{4\varepsilon(n)}\cdot 2k(n)} n^{-\frac{9k(n)}{2}}$$

since $\binom{k(n)}{9/(4\varepsilon(n))} \leq \left(\frac{e\cdot 9/(4\varepsilon(n))\ell(n)}{9/(4\varepsilon(n))}\right)^{9/(4\varepsilon(n))}$. Moreover, this is bounded above by $(e\log n)^{9/(4\cdot 2\log\log n/\log n)} \leq n^{19/16}$ as $\ell(n) \leq \log n$ and $\varepsilon(n) \leq 2\log\log n/\log n$. So, we obtain $\mathbf{Pr}[\mathcal{E}_2] \leq n^{2k(n)+19/16\cdot 2k(n)-9k(n)/2} = n^{-k(n)/8} = n^{-\Omega(1/\varepsilon(n))}$ since $k(n) \geq 9/(4\varepsilon(n))$.

We consider the following three cases for $k(n)$.

- $k(n) < 9/(4\varepsilon(n))$. If $\mathcal{E}_1$ does not occur, nothing will or has to be shown. Thus, there exist cliques of size at most $9/(2\varepsilon(n))$ only. Moreover, by Theorem 2 an expected number of $\mathcal{O}(n^{9/\varepsilon(n)-2} + \mu n \log n)$ steps is sufficient to generate a maximum clique.
- $9/(4\varepsilon(n)) \leq k(n) \leq 9/(4\varepsilon(n)) \log n$. If $\mathcal{E}_1$ and $\mathcal{E}_2$ do not occur, nothing will or has to be shown. Let P' denote the vertices of $V \setminus P$ that are adjacent to at least $9/(4\varepsilon(n))$ vertices of P. Each clique which contains at least one vertex of $V \setminus (P \cup P')$, is of size at most $9/(2\varepsilon(n)) - 1$. Thus, each clique of size at least $9/(2\varepsilon(n))$ consists of vertices of $P \cup P'$ only. Since $\mathcal{E}_2$ does not occur, it is $|P \cup P'| \leq k(n) + (2k(n) - 1) = 3k(n) - 1$ and $|\mathcal{C}^{\geq 9/(2\varepsilon(n))}(G)| \leq \sum_{i=9/(2\varepsilon(n))}^{3k(n)-1} \binom{3k(n)-1}{i} \leq 2^{3k(n)-1}$. Moreover, by Corollary 3 an expected number of $\mathcal{O}(n^{9/\varepsilon(n)-1} + \mu n \log n)$ steps is sufficient to generate

a maximum clique, if $\mu \geq 2^{3k(n)}$. We remark that the size of the maximum clique is bounded above by $3k(n)$.

- $k(n) > 9/(4\varepsilon(n)) \log n$. If $\mathcal{E}_1$ does not occur, nothing will or has to be shown. There exist at most $\sum_{i=0}^{9/(2\varepsilon(n))} \binom{n-k(n)}{i} \leq n^{5/\varepsilon(n)}$ cliques C on $V \setminus P$. Since $|C| \leq 9/(2\varepsilon(n))$ the number of cliques on $C \cup P$ is bounded above by $2^{|C \cup P|} \leq 2^{9/(2\varepsilon(n))+k(n)} \leq 2^{2k(n)}$ and $|\mathcal{C}^{\geq 0}(G)| \leq n^{5/\varepsilon(n)} \cdot 2^{2k(n)} \leq 2^{3k(n)}$. Moreover, by Lemma 1.(b) an expected number of $\mathcal{O}(\mu n \log n)$ steps is sufficient to generate a maximum clique, if $\mu \geq 2^{3k(n)}$. We remark that the size of the maximum clique is bounded above by $2k(n)$.

This proves the result. □

Theorem 11 *With respect to $\mathcal{G}^{**}_{n,n^{-\varepsilon(n)},k(n)}$ the algorithm A needs with high probability (with respect to the input's random part) an expected (with respect to the algorithm's random bits) number of $t(n)$ steps to create a maximum clique, where A and $t(n)$ are given in the following table. Let $2 \log \log n / \log n \leq \varepsilon(n) \leq 1$ and $3/\varepsilon(n) \leq k(n) \leq n/2$.*

algorithm A	time $t(n)$
(1+1) EA	$n^{\Theta(k(n))}$
(μ+1) RLS, $\mu \leq n^{1/(5\varepsilon(n))} + 2^{k(n)/4} - 1$	∞

PROOF. Let us investigate the (1+1) EA. For the upper bound, we observe that in the proof of Theorem 10 it was shown that with high probability the size of a maximum clique is bounded above by $3k(n)$. The result follows directly by Theorem 2. For the lower bound, we observe that the graph $F_{n,k(n)/\sqrt{2}}$ of the proof of Theorem 7 can be constructed by an adversary and the result follows directly, too.

Let us investigate the (μ+1) RLS, where $\mu \leq n^{1/(5\varepsilon(n))} + 2^{k(n)/4} - 1$. We consider the some modifications described in the proof of Theorem 9. Additionally, let $P_1 \subseteq P$, where $|P_1| = 3k(n)/4$, and $P_0 := P \setminus P_1$; and we remove all edges in P which are not in P_0 or P_1 only. If $\mathcal{E}_1$ and $\mathcal{E}_3$ of the proof of Theorem 9 do not occur, nothing will or has to be shown. Therefore, the clique P_1 is the maximum one since $3k(n)/4 > 2.1/\varepsilon(n)$. Similar as we have seen in the proof of Theorem 9, with positive probability the population is initialized with the $1/\varepsilon(n)$-cliques on V_0 and the $2^{k(n)/4} - 1$ non-empty cliques on P_0. Afterwards, there is no possibility to create the maximum clique by any sequence of queries. □

At last, we investigate the more powerful semi-random graph model $\mathcal{G}^{***}_{n,p(n),m(n)}$, where an adversary is allowed to insert up to $m(n)$ edges in an otherwise random graph $\mathcal{G}_{n,p(n)}$ (even without a planted clique). For $p(n) = 0$ we are in the same situation than in Theorem 6 and even the considerations made there indicate the necessity of a large population. Moreover, in particular the investigations of Theorem 10 and Theorem 11 indicate the advantage of a large population and a global search operator. Let us make this more precise by the following theorems. We remark that the following results hold also, if an adversary is additionally allowed to remove arbitrary edges.

Theorem 12 *Let* $\log^{1/4} n \leq k(n) \leq 2^{\log^{15/16} n/32}$ *and* $1/\log^{1/16} n \leq \varepsilon(n) \leq 1$. *With respect to* $\mathcal{G}^{***}_{n,n^{-\varepsilon(n)},\binom{k(n)}{2}}$ *the* $(\mu{+}1)$ EA, *where* $\mu \geq 2^{11k(n)/10}$, *creates in an expected (with respect to the algorithm's random bits) number of* $\mathcal{O}(n^{9/\varepsilon(n)-1} + \mu n \log n)$ *steps a maximum clique with probability* $1 - n^{-\Omega(1/\varepsilon(n))}$ *(with respect to the input's random part).*

PROOF. We remark that the values of $k(n)$ cover sub-linear to super-polynomial population sizes and $\varepsilon(n)$ covers quite dense graphs, too. To simplify the notation, let $\text{bin}(k(n)) := \binom{k(n)}{2}$. At first, some calculations show that with probability $1 - n^{-\Omega(1/\varepsilon(n))}$ none of the following events occur. If at least one of the events occur, nothing will or has to be shown.

- $\mathcal{E}_1$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains two disjoint subsets $S, T \subseteq V$, where $|S| = |T| = 9/(4\varepsilon(n))$, and $\{\{s,t\} \mid s \in S, t \in T\} \subseteq E$. Probability: $n^{-\Omega(1/\varepsilon(n))}$.
- $\mathcal{E}_2$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains two disjoint sets $S, T \subseteq V$, where $|S| = |T| = 2\text{bin}(k(n))$ and for each $t \in T$ holds $|\{s,t\} \mid s \in S\}| \geq 9/(4\varepsilon(n))$. Probability: $n^{-\Omega(\text{bin}(k(n)))}$
- $\mathcal{E}_3$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains a subset $S \subseteq V$, where $|S| = 4\text{bin}(k(n))$, and $|E_S| \geq \text{bin}(k(n))^{19/16}$. Probability: $n^{-\Omega(\text{bin}(k(n)))}$.
- $\mathcal{E}_4$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains a subset $S \subseteq V$, where $|S| = \text{bin}(k(n))^{13/16}$, and $|E_S| \geq \text{bin}(k(n))/5$. Probability: $n^{-\Omega(\text{bin}(k(n))^{13/16})}$.

(See Appendix D for a proof of the probability bounds on the events.)

Let V_0^+ consist of all vertices that are incident to edges inserted by an adversary. And let $V_0^- \subseteq V \setminus V_0^+$ consist of all vertices that are adjacent to at least $9/(4\varepsilon(n))$ vertices of V_0^+. It is $|V_0^+| \leq 2\text{bin}(k(n))$ and $|V_0^-| \leq 2\text{bin}(k(n))$ since $\mathcal{E}_2$ does not occur. However, let $V_0 := V_0^+ \cup V_0^-$ and $V_1 := V \setminus V_0$. Roughly speaking, on V_0 a lot can happen and on V_1 (also in combination with V_0) only little. Since $\mathcal{E}_1$ and $\mathcal{E}_2$ do not occur, vertices of V_1 are contained in cliques of size less than $2 \cdot 9/(4\varepsilon(n))$. Let us consider the cliques on V_0. Thus, V_0 is small and on V_0 typically only a few edges exist. In particular, since $\mathcal{E}_3$ does not occur, their number is bounded above by $\text{bin}(k(n))^{19/16} + \text{bin}(k(n))$ – including the adversary's edges. Furthermore, only a very few vertices $V_0' \subseteq V_0$ are adjacent to many – namely, at least $5\text{bin}(k(n))^{3/8}$ – vertices of V_0. Hence, by contradiction it follows that $|V_0'| \leq \text{bin}(k(n))^{13/16}$. The number of cliques of size at least $9/(2\varepsilon(n))$ with at least one vertex of $V_0 \setminus V_0'$ is bounded above by $2^{k(n)^{7/8}}$. This holds since the degree of a vertex of $V_0 \setminus V_0'$ is bounded above by $5\text{bin}(k(n))^{3/8} - 1$. This is why it is contained in cliques of size at most $5\text{bin}(k(n))^{3/8}$. Finally, the number of cliques of size at least $9/(2\varepsilon(n))$ and at most $5\text{bin}(k(n))^{3/8}$ on V_0, where $|V_0| \leq 4\text{bin}(k(n))$, is bounded above by

$$\sum_{i=9/(2\varepsilon(n))}^{5\text{bin}(k(n))^{3/8}} \binom{4\text{bin}(k(n))}{i} \leq 5\text{bin}(k(n))^{3/8} \cdot \binom{4\text{bin}(k(n))}{5\text{bin}(k(n))^{3/8}}$$
$$\leq 5\text{bin}(k(n))^{3/8} \cdot (4\text{bin}(k(n)))^{5\text{bin}(k(n))^{3/8}} \leq 2^{k(n)^{7/8}} \quad .$$

Since $\mathcal{E}_4$ does not occur, we can upper bound the total number of edges on V_0' by $\text{bin}(k(n))/5 + \text{bin}(k(n)) = 6\text{bin}(k(n))/5$. By Theorem 6 the number of cliques on V_0' is bounded above by $2^{\sqrt{6/5 \cdot k(n)}}$. Therefore, $|\mathcal{C}^{\geq 9/(2\varepsilon(n))}(G)| \leq 2^{\sqrt{6/5 \cdot k(n)}} + 2^{k(n)^{7/8}} \leq 2^{11k(n)/10-1}$ and the proposed result follows directly by Corollary 3. $\square$

Theorem 13 *Let $\log^{1/4} n \leq k(n) \leq n^{1/4}$ and $\varepsilon(n) \geq 17/\log^{1/4} n$. With respect to $\mathcal{G}^{***}_{n,n^{-\varepsilon(n)},\binom{k(n)}{2}}$ the algorithm A creates in an expected (with respect to the algorithm's random bits) number of $t(n)$ steps a maximum clique with probability $1-n^{-\Omega(k(n))}$ (with respect to the input's random part), where A and $t(n)$ are given in the following table.*

algorithm A	time $t(n)$
(1+1) EA	$n^{\Theta(k(n))}$
(μ+1) RLS, $\mu \leq 2^{k(n)/4}$	∞

PROOF. Let $\text{bin}(k(n)) := \binom{k(n)}{2}$. Again, some calculations show that with probability $1 - n^{-\Omega(k(n))}$ none of the following events occur. If at least one of the events occur, nothing will or has to be shown.

- $\mathcal{E}_1$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains a subset $S \subseteq V$, where $|S| = 2k(n)$, and $|\{\{s_1, s_2\} \mid s_1 \neq s_2 \text{ and } s_1, s_2 \in S\} \setminus E| \leq \text{bin}(k(n))$. Probability: $n^{-\Omega(k(n))}$.
- $\mathcal{E}_2$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains two disjoint subsets $S, T \subseteq V$, where $|S| = |T| = k(n)/8$, and $\{\{s,t\} \mid s \in S, t \in T\} \subseteq E$. Probability: $n^{-\Omega(k(n))}$.

(See Appendix E for a proof of the probability bounds on the events.)

Let us investigate the (1+1) EA. Since $\mathcal{E}_1$ does not occur, the adversary can construct cliques of size at most $2k(n) - 1$ only. By Theorem 2 the expected number of steps to generate a maximum clique is bounded above by $n^{\mathcal{O}(k(n))}$. Let $V_0 := \{v_1, \ldots, v_{5k(n)/8}\}$ and $V_1 := \{v_{n-6k(n)/8+1}, \ldots, v_n\}$. Let us consider the lower bound. Therefore, we investigate the graph, where all the at most $\text{bin}(5k(n)/8) + \text{bin}(6k(n)/8) \leq \text{bin}(k(n))$ non-existent edges of $\{\{s_1, s_2\} \mid s_1 \neq s_2 \text{ and } s_1, s_2 \in V_0\} \cup \{\{s_1, s_2\} \mid s_1 \neq s_2 \text{ and } s_1, s_2 \in V_1\}$ are inserted. We consider the first step, where either (a) $|x_1 \cdots x_{n-6k(n)/8}| < k(n)/8$ or (b) $|x_{6k(n)/8+1} \cdots x_n| < k(n)/8$. Since $\mathcal{E}_2$ does not occur, all elements seen before have not been cliques. Because of symmetry, we are in both situations with equal probabilities and in situation (b) with a probability of at least 1/2. Afterwards, a specific at most $7k(n)/8$-bit mutation creates the clique V_0. This mutation corrects all bits in $x_1 \cdots x_{6k(n)/8}$ and at most $k(n)/8$ outside. Its probability is bounded below by $1/(en^{7k(n)/8})$. In this situation, only cliques with at least $5k(n)/8 - k(n)/8 = k(n)/2$ vertices of either (a) V_0 or (b) V_1 are the population's element. This is the case since $\mathcal{E}_2$ does not occur. In order to generate a maximum clique at least once the situation (b) has to occur necessarily. This is why at least $2 \cdot k(n)/2 = k(n)$ out of $5k(n)/8 + 6k(n)/8 = 11k(n)/8$ bits have to flip. Its probability is bounded above by $\binom{11k(n)/8}{k(n)} n^{-k(n)} \leq 4^{k(n)} n^{-k(n)} \leq n^{-15k(n)/16}$. This leads to an expected number of at least $n^{15k(n)/16}/(2en^{7k(n)/8}) = n^{\Omega(k(n))}$ steps.

Let us investigate the (μ+1) RLS, where $\mu \leq 2^{k(n)/4}$. We consider the same modifications of the graph as above. With positive probability the population is initialized with subsets of V_0 of size $3k(n)/8$ only. Their number is bounded above by $\binom{5k(n)/8}{3k(n)/8} \geq 2^{k(n)/4}$. Since $\mathcal{E}_2$ does not occur, afterwards, only cliques with at least $2k(n)/8$ vertices of V_0 are in the population. Thus, there is no possibility to create the maximum clique by any sequence of queries – for the (μ+1) RLS. $\square$

4 Summary and Conclusions

The optimization behavior of simple general randomized search heuristics on – in particular – sparse semi-random graphs for the maximum clique problem with a powerful adversary is investigated in detail. In contrast to traditional semi-random models that are considered, too, an adversary is allowed to be *moderately harmful* and not only *helpful*. This exhibits a new approach and models real-world inputs more appropriate than former ones. Actually in such a general setting the major advantage of applying a large population *and* a global search operator is demonstrated. So, also the long time outstanding open question if a large population can outperform a small one in combinatorial optimization is solved. Future research will consider – beside the expectation – the success probability of the investigated and further randomized optimizers in order to obtain results for their (independent) (parallel) multistart variants.

Acknowledgements

This research was supported by grants from the German-Israeli Foundation for Scientific Research and Development and the German Research Foundation. The author thanks Stefan Droste, Thomas Jansen, and Ingo Wegener for their helpful discussions while preparing this article.

References

[1] N. Alon, M. Krivelevich, and B. Sudakov. Finding a large hidden clique in a random graph. In *Proceedings of the 9th Symposium on Discrete Algorithms – SODA*, pages 594–598, 1998.

[2] B. Bollobás. *Random Graphs.* Academic Press, London, 1985.

[3] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. *Handbook of Combinatorial Optimization*, chapter The Maximum Clique Problem, pages 1–74. Kluwer, Dordrecht, 1999.

[4] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.

[5] L. Engebretsen and J. Holmerin. Clique is hard to approximate within $n^{1-o(1)}$. In *Proceedings of the 27th International Colloquium on Automata, Languages and Programming – ICALP*, pages 2–12, 2000.

[6] U. Feige. Approximating maximum clique by removing subgraphs. *SIAM Journal on Discrete Mathematics*, 18:219–225, 2004.

[7] U. Feige and J. Kilian. Heuristics for semi-random graph problems. *Journal of Computer and System Science*, 63:639–671, 2001.

[8] U. Feige and R. Krauthgamer. Finding and certifying a large hidden clique in a semi-random graph. *Random Structures and Algorithms*, 16:195–208, 2000.

[9] A. M. Frieze and C. McDiarmid. Algorithmic theory of random graphs. *Random Structures and Algorithms*, 10:5–42, 1997.

[10] M. Jerrum. Large cliques elude the Metropolis process. *Random Structures and Algorithms*, 3:347–360, 1992.

[11] D. S. Johnson and M. A. Trick. *Cliques, Coloring, and Satisfiability.* American Mathematical Society, Providence, 1996.

[12] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.

[13] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, Cambridge, 1995.

[14] J. M. Robson. Finding a maximum independent set in time $\mathcal{O}(2^{n/4})$. In *Technical Report 1251-01, Université Bordeaux, LaBRI*, 2001.

[15] T. Storch. On the choice of the population size. In *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO*, pages 748–760, 2004.

[16] T. Storch. How randomized search heuristics find maximum cliques in planar graphs. In *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO*, page to appear, 2006.

A Lemma 4

Claim. For a graph G let V_0 and V_1 be the elements of a partition of V, where $|V_0| = |V|/2$. If there exist edges in V_0 and in V_1 only, then the (1+1) EA creates each clique of size 1 with probability $\Omega(1/n)$ as population.

PROOF. Let $x_{(i)}$, $i = 0, 1$, denote the bits of x which represent the vertices of V_i and $\ell_i := \|x_{(i)}\|$. We observe that in the case of at least a single one in the first and second half of a bit-string, i.e., $\ell_0, \ell_1 \geq 1$, the element does not represent a clique. The function value equals $-(\ell_0 + \ell_1)$. Thus, as long as no element with $\ell_0 = 0$ or $\ell_1 = 0$ is created, the (1+1) EA behaves equivalently than on $-\|x\|$. The probability is bounded below by $(\ell_0 + \ell_1)/n(1 - 1/n)^{n-1} \geq (\ell_0 + \ell_1)/(en)$ to create an element y, where $\|y\| < \|x\|$. Whereas the probability is bounded above by $(\ell_0 + \ell_1)/n$ to create an element z, where $\|z\| = \|x\|$, but $z \neq x$. Hence, the probability is bounded above by $\left(\frac{(\ell_0+\ell_1)/n}{(\ell_0+\ell_1)/n+(\ell_0+\ell_1)/(en)}\right)^{\ell_0+\ell_1} = \left(\frac{e}{e+1}\right)^{\ell_0+\ell_1} \leq 2^{-2(\ell_0+\ell_1)/5}$ to create at least $\ell_0 + \ell_1$ such elements z before creating an element y. By [4] during the optimization the (1+1) EA on $-\|x\|$ generates each element as population with the same number of ones with equal probability.

Therefore, the probability that among these elements is at least one element, where $\ell_0 = 0$ or $\ell_1 = 0$, equals

$$\begin{aligned}\sum_{i=0}^{\ell_0+\ell_1-1} \frac{\binom{n/2}{\ell_0+\ell_1}\binom{n/2}{0}+\binom{n/2}{0}\binom{n/2}{\ell_0+\ell_1}}{\binom{n}{\ell_0+\ell_1}-i} &\le \frac{2(\ell_0+\ell_1)\binom{n/2}{\ell_0+\ell_1}}{\binom{n}{\ell_0+\ell_1}-(\ell_0+\ell_1)} \\ &\le \frac{3(\ell_0+\ell_1)\binom{n/2}{\ell_0+\ell_1}}{\binom{n}{\ell_0+\ell_1}} \\ &\le 3(\ell_0+\ell_1)2^{-(\ell_0+\ell_1)} \quad .\end{aligned}$$

Let us bound the probability from above to create an element z, where $\|z\| > \|x\|$ and at least one of the halves consists of zeros only. We distinguish the two cases that at least one of the halves consist of exactly a sole one (case 1) and that both halves consist of at least two ones (case 2).

Case 1. Similar as above, we can bound the fraction of elements, where either $\ell_0 = 1$ or $\ell_1 = 1$, with respect to all elements with equal number of ones by $2\binom{n/2}{\ell_0+\ell_1-1}\binom{n/2}{1}/\binom{n}{\ell_0+\ell_1} \le 2(\ell_0+\ell_1)2^{-(\ell_0+\ell_1)}$. For these elements, the probability is bounded above by $n^{-\ell_0} + n^{-\ell_1} \le 2/n$ to create such an individual z. Furthermore, the probability is bounded above by

$$\frac{4(\ell_0+\ell_1)/(n2^{\ell_0+\ell_1})}{4(\ell_0+\ell_1)/(n2^{\ell_0+\ell_1})+(\ell_0+\ell_1)/(en)} = 4e/(4e+2^{\ell_0+\ell_1}) \le 2^{-2(\ell_0+\ell_1)/5}$$

to create an element z prior to an element y, where $\|y\| < \|x\|$. The last inequality holds for $\ell_0 + \ell_1 \ge 6$.

Case 2. The probability is bounded above by $n^{-\ell_0} + n^{-\ell_1} \le 2/n^2$ to create such an individual z. Furthermore, the probability is bounded above by

$$\frac{2/n^2}{2/n^2+(\ell_0+\ell_1)/(en)} = \frac{2e}{2e+(\ell_0+\ell_1)n} \le 1/(2n+1)$$

to create an element z prior to an element y, where $\|y\| < \|x\|$. The last inequality holds for $\ell_0 + \ell_1 \ge 11$.

Thus, for all $\ell \ge 11$, the probability that

- the (1+1) EA has more than ℓ different individuals with ℓ ones as population during optimization or
- among these at most ℓ elements is at least one element, where one of the halves consists of zeros only or
- an element with more than ℓ ones is generated, where one of the halves consist of zeros only

is upper bounded by

$$\sum_{\ell=11}^{n} 2^{-2\ell/5} + \sum_{\ell=11}^{n} 3\ell \cdot 2^{-\ell} + \sum_{\ell=11}^{n} (2^{-2\ell/5} + (2n+1)^{-1})$$
$$\leq 2 \sum_{\ell=11}^{\infty} 2^{-2\ell/5} + 3 \sum_{\ell=11}^{\infty} \ell \cdot 2^{-\ell} + \frac{n-10}{2n+1}$$
$$\leq 2 \cdot 1/5 + 3 \cdot 1/50 + 1/2 = 24/25$$

With a probability of at least $1/25$ an element x, where $\|x\| \leq 10$, is generated and has each element with equal number of ones with equal probability by [4].

Moreover, for an element x the probability is bounded below by $\|x\|/(en)$ to create an element y with the following properties. It is $y_i = x_i$ for all but one indices i and for the remaining index i holds $0 = y_i < x_i = 1$. Each such individual y is created with equal probability and also accepted as population. The probability is bounded above by $\|x\|/n$ to generate an element z, where either $\|z\| \leq \|x\|$, but $z \neq x$ and $z \neq y$ for all such y or $\|z\| > \|x\|$, but at least one of the halves consist of zeros only. This is the case since at least one of the ones in x has to flip. The probability is bounded below by

$$\frac{\|x\|/(en)}{\|x\|/(en) + \|x\|/n} = \frac{1}{1+e}$$

to create an element y prior to an element z. If the heuristic starts with an individual which consists of at most $\ell \leq 10$ ones and performs $\ell - 1 \leq 9$ times the described mutation first, then an element x, where $\|x\| = 1$, is generated. The probability therefore is bounded below by $1/(1+e)^9$. We recall that each element x is created with equal probability. Let x', where $\|x'\| = 2$, be the parent of x. Hence, a failure occurred to the worst, if one of the halves of x' consists of zeros only. The fraction of such elements equals $2\binom{n/2}{2}/\binom{n}{2} \leq 1/2$. So, we can bound the failure probability by $1/25 \cdot 1/(1+e)^9 \cdot 1/2 = \Omega(1)$ from above, in total. $\square$

B Theorem 6

Claim. $|\mathcal{C}^{\geq 2}(G)| \leq 2^{k(n)} - k(n) - 1$

PROOF. (*by induction*) The graph contains exactly $2^{k(n)} - k(n) - 1$ cliques of size at least 2, if and only if $k(n)$ vertices form a complete subgraph. For $k(n) = 2$ the graph contains exactly one edge and one edge describes exactly $1 = 2^2 - 2 - 1$ clique of size 2. For $k(n) \geq 3$ we distinguish the two cases that either G consists of a clique of size $k(n)$ (case 1) or not (case 2).

Case 1. The graph G consists of a clique of size $k(n)$. In this situation, G contains $\sum_{i=2}^{k(n)} \binom{k(n)}{i} = 2^{k(n)} - k(n) - 1$ cliques of size at least 2.

Case 2. The graph G does not consist of a clique of size $k(n)$. In this situation, as long as the graph contains more than $\mathrm{bin}(k(n)-1) = \binom{k(n)}{2} - (k(n)-1)$ edges, successively delete all edges of a vertex with minimal degree $d_j \geq 1$ in step j. Since G does not consist of a complete subgraph on $k(n)$ vertices, it always holds $d_j \leq k(n) - 2$. Hence, this removal of edges in step j destroys at most $\binom{d_j}{i}$, $0 \leq i \leq d_j$, cliques of size exactly $i+1$. Thus, at most $\sum_{i=1}^{d_j} \binom{d_j}{i} = 2^{d_j} - 1$ cliques of size at least 2 are removed, in total. Since $(2^a - 1) + (2^b - 1) \leq 2^{a+b} - 1$ for all $a, b \geq 0$, it is $d_j \leq k(n) - 2$, and also $\sum_j d_j \leq 2(k(n) - 2)$ holds, we obtain $\sum_j (2^{d_j} - 1) \leq 2(2^{k(n)-2} - 1)$. The remaining at most $\mathrm{bin}(k(n)-1)$ edges form at most $2^{k(n)-1} - (k(n)-1) - 1$ cliques of size at least 2. In total, at most $(2^{k(n)-1} - 2) + (2^{k(n)-1} - (k(n)-1) - 1) = 2^{k(n)} - k(n) - 2 < 2^{k(n)} - k(n) - 1$ cliques of size at least 2 exist in G, for $k(n) \geq 3$. $\square$

C Theorem 9

Claim.

$\mathcal{E}_1$: The size of a maximum clique on V_0 is bounded above by $2.1/\varepsilon(n)$.
$\mathcal{E}_2$: The size of a smallest maximal clique on V_0 is bounded below by $0.9/\varepsilon(n)$.
$\mathcal{E}_3$: The number of cliques of size $1/\varepsilon(n)$ on V_0 is bounded below by $n^{1/(5\varepsilon(n))}$.

PROOF. That the events $\mathcal{E}_1$ and $\mathcal{E}_2$ do not occur with high probability was proven by [9]. Let us investigate the event $\mathcal{E}_3$. Therefore, let X denote the number of clique of size $1/\varepsilon(n)$ on V_0. By [2] it was proven that

$$\begin{aligned}\mathbf{E}[X] &= \binom{n/2}{1/\varepsilon(n)} n^{-\varepsilon(n)\cdot\binom{1/\varepsilon(n)}{2}} \geq n^{1/\varepsilon(n)} \cdot \left(\frac{\varepsilon(n)}{2}\right)^{1/\varepsilon(n)} \cdot n^{-1/(2\varepsilon(n))} \cdot n^{-1/2} \\ &\geq n^{1/(4\varepsilon(n))}\end{aligned}$$

since $\binom{n/2}{1/\varepsilon(n)} \geq (n \cdot \varepsilon(n)/2)^{1/\varepsilon(n)}$, $(\varepsilon(n)/2)^{1/\varepsilon(n)} \geq (2\log n)^{-1/\varepsilon(n)} \geq n^{-1/(8\varepsilon(n))}$, and $n^{-1/2} \geq n^{-1/(8\varepsilon(n))}$. Moreover, by [2] it was also proven that

$$\begin{aligned}\mathbf{Var}[X] &\leq \mathbf{E}[X]^2 \Big(\frac{(n^{\varepsilon(n)} - 1)}{2\varepsilon(n)^4 n^2} + \frac{1}{\mathbf{E}[X]} + \frac{(n^{3\varepsilon(n)} - 1)}{6\varepsilon(n)^7 n^3} + \frac{n \cdot n^{-\varepsilon(n)(1/\varepsilon(n)-1)}}{\varepsilon(n)\mathbf{E}[X]} \Big) \\ &\leq \mathbf{E}[X]^2/n\end{aligned}$$

since it is $(n^{\varepsilon(n)}-1)/(2\varepsilon(n)^4n^2) \le n^{0.1}(\log n)^4/(2n^2) \le 1/(4n)$, $1/\mathbf{E}[X] \le n^{-1/(4\varepsilon(n))} \le 1/(4n)$, $(n^{3\varepsilon(n)}-1)/(6\varepsilon(n)^7n^3) \le n^{0.3}(\log n)^7/(6n^3) \le 1/(4n)$, and $n \cdot n^{-\varepsilon(n)(1/\varepsilon(n)-1)}/(\varepsilon(n)\mathbf{E}[X]) \le n^{0.1}(\log n)/n^{1/(4\varepsilon(n))} \le 1/(4n)$. We make use of the second moment method. Namely, applying Chebyshev inequality [13] and since $\mathbf{E}[X]/2 \ge n^{1/(4\varepsilon(n))}/2 \ge n^{1/(5\varepsilon(n))}$ we obtain

$$\mathbf{Pr}[X \le \mathbf{E}[X]/2] \le \mathbf{Pr}[|X-\mathbf{E}[X]| \ge \mathbf{E}[X]/2] \le \mathbf{Var}[X]/(\mathbf{E}[X]/2)^2 \le 4/n^2 \quad .$$

So, the event $\mathcal{E}_3$ does not occur with high probability. $\square$

D Theorem 12

Claim.

- $\mathcal{E}_1$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains two disjoint subsets $S,T \subseteq V$, where $|S| = |T| = 9/(4\varepsilon(n))$, and $\{\{s,t\} \mid s \in S, t \in T\} \subseteq E$. Probability: $n^{-\Omega(1/\varepsilon(n))}$.
- $\mathcal{E}_2$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains two disjoint sets $S,T \subseteq V$, where $|S| = |T| = 2\mathrm{bin}(k(n))$ and for each $t \in T$ holds $|\{s,t\} \mid s \in S\}| \ge 9/(4\varepsilon(n))$. Probability: $n^{-\Omega(\mathrm{bin}(k(n)))}$
- $\mathcal{E}_3$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains a subset $S \subseteq V$, where $|S| = 4\mathrm{bin}(k(n))$, and $|E_S| \ge \mathrm{bin}(k(n))^{19/16}$. Probability: $n^{-\Omega(\mathrm{bin}(k(n)))}$.
- $\mathcal{E}_4$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains a subset $S \subseteq V$, where $|S| = \mathrm{bin}(k(n))^{13/16}$, and $|E_S| \ge \mathrm{bin}(k(n))/5$. Probability: $n^{-\Omega(\mathrm{bin}(k(n))^{13/16})}$.

PROOF. *Event* $\mathcal{E}_1$. Similar to the proof of Theorem 8 the probability for the event is bounded above by

$$\binom{n}{9/(4\varepsilon(n))}\binom{n}{9/(4\varepsilon(n))} n^{-\varepsilon(n)\cdot(\frac{9}{4\varepsilon(n)})^2} \le n^{2\cdot 9/(4\varepsilon(n))-\varepsilon(n)\cdot\frac{81}{16\varepsilon(n)^2}} = n^{-9/(16\varepsilon(n))}$$

since $\binom{n}{9/(4\varepsilon(n))} \le n^{9/(4\varepsilon(n))}$.

Event $\mathcal{E}_2$. We observe that necessarily for each $t \in T$ a subset $S_t \subseteq S$ of size $9/(4\varepsilon(n))$ exists, where t is connected to each vertex of S_t. Thus, similar to the proof of Theorem 8 the probability for the event is bounded above by

$$\binom{n}{2\mathrm{bin}(k(n))}\binom{n-2\mathrm{bin}(k(n))}{2\mathrm{bin}(k(n))}\binom{2\mathrm{bin}(k(n))}{9/(4\varepsilon(n))}^{2\mathrm{bin}(k(n))} n^{-\varepsilon(n)\cdot 2\mathrm{bin}(k(n))\cdot 9/(4\varepsilon(n))} \quad .$$

We observe $\binom{2\mathrm{bin}(k(n))}{9/(4\varepsilon(n))} \le \left(\frac{8e\cdot\varepsilon(n)\cdot\mathrm{bin}(k(n))}{9}\right)^{9/(4\varepsilon(n))} \le 2^{9/(4\varepsilon(n))\cdot\log(3\mathrm{bin}(k(n)))}$ since $\varepsilon(n) \le 1$ and moreover, $2^{9/(4\varepsilon(n))\cdot\log(3\mathrm{bin}(k(n)))} \le 2^{9(\log^{1/16} n)/4\cdot 32/27\cdot 2(\log^{15/16} n)/32} =$

$n^{1/6}$. Therefore, the former expression is bounded above by

$$n^{2\text{bin}(k(n))}n^{2\text{bin}(k(n))}n^{1/6\cdot 2\text{bin}(k(n))}n^{-9\text{bin}(k(n))/2} = n^{-\text{bin}(k(n))/6} \quad .$$

Event $\mathcal{E}_3$. For a subset of $4\text{bin}(k(n))$ vertices the probability that the number of edges between these vertices is at least $\text{bin}(k(n))^{19/16}$ is bounded above by

$$\binom{\binom{4\text{bin}(k(n))}{2}}{\text{bin}(k(n))^{19/16}} n^{-\varepsilon(n)\text{bin}(k(n))^{19/16}} \leq n^{-\varepsilon(n)\text{bin}(k(n))^{19/16}/2}$$

since $\binom{\binom{4\text{bin}(k(n))}{2}}{\text{bin}(k(n))^{19/16}} \leq \left(\frac{e\cdot 16\text{bin}(k(n))^2}{2\text{bin}(k(n))^{19/16}}\right)^{\text{bin}(k(n))^{16/19}} \leq 2^{\text{bin}(k(n))^{19/16}\log^{15/16} n/2}$ and moreover, $n^{\varepsilon(n)\text{bin}(k(n))^{19/16}/2} \geq 2^{\text{bin}(k(n))^{19/16}\log^{15/16} n/2}$. Similar to the proof of Theorem 8 this leads to a probability of at most

$$\binom{n}{4\text{bin}(k(n))} n^{-\varepsilon(n)\text{bin}(k(n))^{19/16}/2} \leq n^{\text{bin}(k(n))(4-\varepsilon(n)\text{bin}(k(n))^{3/16}/2)} \leq n^{-\text{bin}(k(n))}$$

since $\binom{n}{4\text{bin}(k(n))} \leq n^{4\text{bin}(k(n))}$ and

$$\begin{aligned} 4-\varepsilon(n)\text{bin}(k(n))^{3/16}/2 &\leq 4-(1/\log^{1/16} n)\cdot((\log^{1/2} n)/3)^{3/16}/2 \\ &= 4-(\log^{1/32} n)/(2\cdot 3^{3/16}) \leq -1 \quad . \end{aligned}$$

Event $\mathcal{E}_4$. For a subset of $\text{bin}(k(n))^{13/16}$ vertices the probability that the number of edges between these vertices is at least $\text{bin}(k(n))/5$ is bounded above by

$$\binom{\binom{\text{bin}(k(n))^{13/16}}{2}}{\text{bin}(k(n))/5} n^{-\varepsilon(n)\text{bin}(k(n))/5} \leq n^{-\varepsilon(n)\text{bin}(k(n))/10}$$

since $\binom{\binom{\text{bin}(k(n))^{13/16}}{2}}{\text{bin}(k(n))/5} \leq \left(\frac{e\cdot 5\text{bin}(k(n))^{13/8}}{\text{bin}(k(n))}\right)^{\text{bin}(k(n))/5} \leq 2^{\text{bin}(k(n))\log^{15/16} n/10}$ and moreover, $n^{\varepsilon(n)\text{bin}(k(n))/10} \geq 2^{\text{bin}(k(n))\log^{15/16} n/10}$. Similar to the proof of Theorem 8 this leads to a probability of at most

$$\begin{aligned} \binom{n}{\text{bin}(k(n))^{13/16}} n^{-\varepsilon(n)\text{bin}(k(n))/10} &\leq n^{\text{bin}(k(n))^{13/16}(1-\varepsilon(n)\text{bin}(k(n))^{3/16}/10)} \\ &\leq n^{-\text{bin}(k(n))^{13/16}} \end{aligned}$$

since $\binom{n}{\text{bin}(k(n))^{13/16}} \leq n^{\text{bin}(k(n))^{13/16}}$ and

$$\begin{aligned} 1-\varepsilon(n)\text{bin}(k(n))^{3/16}/10 &\leq 1-(1/\log^{1/16} n)\cdot((\log^{1/2} n)/3)^{3/16}/10 \\ &\leq 1-(\log^{1/32} n)/(10\cdot 3^{3/16}) \leq -1 \quad . \qquad \square \end{aligned}$$

E Theorem 13

Claim.

$\mathcal{E}_1$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains a subset $S \subseteq V$, where $|S| = 2k(n)$, and $|\{\{s_1, s_2\} \mid s_1 \neq s_2 \text{ and } s_1, s_2 \in S\} \setminus E| \leq \text{bin}(k(n))$. Probability: $n^{-\Omega(k(n))}$.

$\mathcal{E}_2$: $\mathcal{G}_{n,n^{-\varepsilon(n)}}$ contains two disjoint subsets $S, T \subseteq V$, where $|S| = |T| = k(n)/8$, and $\{\{s,t\} \mid s \in S, t \in T\} \subseteq E$. Probability: $n^{-\Omega(k(n))}$.

PROOF. *Event* $\mathcal{E}_1$. Let X_i, $1 \leq i \leq \binom{2k(n)}{2}$, indicate whether the ith edge on $S \subseteq V$, where $|S| = 2k(n)$, exists ($X_i = 1$) or not ($X_i = 0$). Let $X := X_1 + \cdots + X_{\binom{2k(n)}{2}}$. It is $\mathbf{E}[X] = \binom{2k(n)}{2} n^{-\varepsilon(n)}$. By Chernoff bounds [13] with

$$\delta = \frac{\binom{2k(n)}{2} - \binom{k(n)}{2}}{\mathbf{E}[X]}$$

it is

$$\begin{aligned}\mathbf{Pr}[X \geq \binom{2k(n)}{2} - \binom{k(n)}{2}] &= \mathbf{Pr}[X \geq (1 + (\delta - 1))\mathbf{E}[X]] \\ &\leq e^{-\mathbf{E}[X]}(e/\delta)^{\delta \mathbf{E}[X]} \leq (2en^{-\varepsilon(n)})^{\binom{2k(n)}{2} - \binom{k(n)}{2}} \quad ,\end{aligned}$$

where the last inequality holds since $e^{-\mathbf{E}[X]} \leq 1$ and $\frac{e\binom{2k(n)}{2}}{\binom{2k(n)}{2} - \binom{k(n)}{2}} \leq 2$. Moreover,

$$(2en^{-\varepsilon(n)})^{\binom{2k(n)}{2} - \binom{k(n)}{2}} \leq n^{-\varepsilon(n)k(n)^2} \leq n^{-17k(n)} \quad .$$

Similar to the proof of Theorem 8 this leads to a probability of

$$\binom{n}{2k(n)} n^{-17k(n)} \leq n^{2k(n)} n^{-17k(n)} = n^{-\Omega(k(n))}$$

for the event $\mathcal{E}_1$.

Event $\mathcal{E}_2$. Similar to the proof of Theorem 8 the probability for the event is bounded above by

$$\begin{aligned}&\binom{n}{k(n)/8}\binom{n}{k(n)/8} n^{-\varepsilon(n)\cdot(k(n)/8)^2} \\ &\leq n^{2\cdot k(n)/8 - \varepsilon(n)\cdot k(n)^2/64} \leq n^{k(n)/4 - 17/(\log^{1/4} n)\cdot \log^{1/4} n \cdot k(n)/64} = n^{-k(n)/64}\end{aligned}$$

since $\binom{n}{k(n)/8} \leq n^{k(n)/8}$. $\square$