

# Fixed-Parameter Tractability of the (1+1) Evolutionary Algorithm on Random Planted Vertex Covers

Jack Kearney

Algorithmic Evolution Lab

Department of Computer Science  
University of Minnesota Duluth  
Duluth, MN, USA

Frank Neumann

Optimisation and Logistics Group  
School of Computer Science  
University of Adelaide  
Adelaide, SA, Australia

Andrew M. Sutton

Algorithmic Evolution Lab  
Department of Computer Science  
University of Minnesota Duluth  
Duluth, MN, USA

## ABSTRACT

We present the first parameterized analysis of a standard (1+1) Evolutionary Algorithm on a distribution of vertex cover problems. We show that if the planted cover is at most logarithmic, restarting the (1+1) EA every  $O(n \log n)$  steps will find a cover at least as small as the planted cover in polynomial time for sufficiently dense random graphs  $p > 0.71$ . For superlogarithmic planted covers, we prove that the (1+1) EA finds a solution in fixed-parameter tractable time in expectation.

We complement these theoretical investigations with a number of computational experiments that highlight the interplay between planted cover size, graph density and runtime.

## CCS CONCEPTS

• Theory of computation → Theory of randomized search heuristics.

## KEYWORDS

runtime analysis, parameterized complexity, vertex cover

### ACM Reference Format:

Jack Kearney, Frank Neumann, and Andrew M. Sutton. 2023. Fixed-Parameter Tractability of the (1+1) Evolutionary Algorithm on Random Planted Vertex Covers. In *Proceedings of the 17th ACM/SIGEVO Conference on Foundations of Genetic Algorithms (FOGA '23), August 30-September 1, 2023, Potsdam, Germany*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3594805.3607134>

## 1 INTRODUCTION

Combinatorial problems with planted solutions have been an important subject of study on a wide range of settings. In this scenario, a fixed solution is hidden within a large random structure such as a graph. The canonical example of this is the *planted clique* problem where a fixed complete subgraph of size  $k$  is placed within a large Erdős-Rényi random graph on  $n \gg k$  vertices. The task is to either recover the hidden solution [2] or one of size at least  $k$  [12]. These problems have important applications in cryptography [13] for example. In the context of randomized search heuristics, Storch [21]

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

FOGA '23, August 30-September 1, 2023, Potsdam, Germany

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0202-0/23/08...\$15.00

<https://doi.org/10.1145/3594805.3607134>

investigated the planted clique problem for random local search (RLS) and the (1+1) EA. More recently, Doerr et al. [7] considered randomly generated propositional satisfiability problems with planted assignments and proved that the (1+1) EA requires at most  $O(n \log n)$  time to solve this problem provided that the constraint density is high enough.

Planted vertex covers have recently been studied in the context of *systematically incomplete data* [4] in networks. In this view, true node interactions can only be observed among some core set  $C$ , whereas a potentially much larger set of fringe nodes lies outside this sphere of observability. This may occur, for example, in social networks and communication data sets [20] where a company only knows about links within the company and between an employee and the outside world, but not about links between external entities. This translates to a planted vertex cover problem on a graph  $G = (V, E)$ . An adversary knows of a subset  $C \subseteq V$  which is a vertex cover, and the task is to identify a set as close to  $C$  as possible.

In the  $\mathcal{G}(n, k, p)$  model, a graph  $G = (V, E)$  is constructed on a set  $V$  of  $n$  vertices by taking a size- $k$  subset  $C \subseteq V$  to be the *core*. An edge appears in  $G$  with probability  $p$  unless it connects two vertices in  $V \setminus C$ , in which it occurs with probability zero. Therefore,  $G$  is guaranteed to have a  $k$ -vertex cover. Note that a graph can be constructed from this model by drawing a standard Erdős-Rényi graph and subsequently deleting all edges that connect fringe vertices.

This model is a special case of the *stochastic block model* of random graphs from network theory [10] in which the vertex set is partitioned into  $r$  disjoint communities and edge probabilities are specified by a symmetric  $r \times r$  matrix  $P$  where a vertex in community  $i$  is connected to a vertex in community  $j$  with probability  $P_{ij}$ . The stochastic block model allows for the generation of graphs from which the community subgraphs might be recovered partially or in full from the graph data [1]. This models the detection of *community structure* in networks, which is a fundamental problem in computer science. The  $\mathcal{G}(n, k, p)$  model we study in this work is a stochastic block model with  $r = 2$  and probability matrix

$$P = \begin{bmatrix} p & p \\ p & 0 \end{bmatrix}.$$

In this paper, we are interested in the performance of simple randomized search heuristics on planted vertex cover problems in the context of parameterized complexity. We prove that, for sufficiently “dense” graphs (i.e., large enough  $p$ ), the (1+1) EA is with high probability a fixed-parameter tractable heuristic for the  $k$ -vertex cover problem where  $k$  is the size of the planted solution. More precisely, if  $k$  is at most logarithmic, we prove there is a threshold on  $p$  such that above this threshold the (1+1) EA is very

likely to find a  $k$ -cover in almost linear time. For larger values of  $k$ , we show that the (1+1) EA runs in  $O(f(k, p)n \log n)$  time where  $f$  is a function of  $k$  and  $p$  (but not  $n$ ).

The first parameterized result on vertex cover is due to Kratsch and Neumann [16] who demonstrated that Global SEMO using instance-specific mutation operators has expected optimization time  $O(OPT \cdot n^4 + n \cdot 2^{OPT^2+OPT})$  on any graph  $G$  where  $OPT$  is the size of the optimal vertex cover of  $G$ . This result can be tightened to  $O(n^2 \log n + OPT \cdot n^2 + 4^{OPT}n)$  by incorporating the cost of an optimal fractional vertex cover provided by an LP solver into the fitness function. A recent study by Baguley et al. [3] extended these multi-objective approaches to the W-separator problem. Using a special focused jump-and-repair mechanism, Branson and Sutton [5] showed that evolutionary algorithms can solve the vertex cover problem in expected time  $O(2^{OPT} n^2 \log n)$  by probabilistically simulating an iterative compression routine.

The above results hold for all graphs  $G$  with vertex cover size  $OPT$ . In this paper, we sacrifice the generality of the problem slightly in order to investigate a more general algorithm, i.e., the (1+1) EA. To our knowledge, we present here the first parameterized complexity result on vertex cover problems for a standard evolutionary algorithm that does not rely on any special mutation operators.

*Our results.* For random planted graph models with  $n$  vertices, edge density  $p$  and planted cover size  $k$ , we show that if  $k \leq \ln n$ , then if  $p > \sqrt{\frac{1-\ln\delta}{2}}$  for any constant  $\delta \in (1/e, 1)$ , a restart framework for the (1+1) EA finds a  $k$ -cover in  $n^{c+1} \log n$ , where  $c$  is a constant. If  $k > \ln n$ , then we show for any  $0 < p < 1$ , the expected time of the (1+1) EA is  $O\left(k^{4k\left(1+\frac{1}{p}\right)} n \log n\right)$ , i.e., the (1+1) EA runs in FPT time parameterized by  $k$  and  $p$ .

We also provide the results of computational experiments that investigate regimes that our theorem does not cover, for example when both  $p$  and  $k$  are small. These results elucidate the relationship between  $k$  and  $p$  and the runtime of the (1+1) EA, and hint at new interesting directions for future theoretical study.

## 2 PRELIMINARIES

Given a graph  $G = (V, E)$  on  $n$  vertices, we encode subsets of  $V$  as elements of  $\{0, 1\}^n$  in the usual way. For  $x \in \{0, 1\}^n$ , denote as  $|x|$  as the number of bits set to 1 in  $x$  (i.e., the cardinality of the set to which it corresponds). The fitness function typically employed by evolutionary algorithms on the minimum vertex cover problem first penalizes infeasible sets (sets that do not cover all edges in  $E$ ), then penalizes larger feasible covers:

$$f(x) = |x| + n \cdot \left| \left\{ (u, v) \in E : x[u] = x[v] = 0 \right\} \right|. \quad (1)$$

This fitness function is quite natural for searching for a minimal cover, and was originally designed by Khuri and Bäck [15]. It has been studied extensively both empirically and theoretically [8, 15, 19].

We point out that this is a so-called *vertex-based* representation for which there are currently no bounds on the approximation ratio for the (1+1) EA. It is possible to obtain a guaranteed

2-approximation with the (1+1) EA by using edge-based representations instead [11]. This is rather notable, as minimum vertex cover is likely hard to approximate below a  $(2 - \epsilon)$  factor [14].

---

### Algorithm 1: (1+1) EA

---

```

Input: A fitness function  $f: \{0, 1\}^n \rightarrow \mathbb{R}$ 
1 Choose  $x$  uniformly at random from  $\{0, 1\}^n$ ;
2 while termination criteria not met do
3   Create  $y$  by flipping each bit of  $x$  with probability  $1/n$ ;
4   if  $f(y) \leq f(x)$  then  $x \leftarrow y$ ;
5 return  $x$ ;

```

---

Many of our theoretical results make use of multiplicative drift with tail bounds, which we state in the following theorem for reference.

**THEOREM 1 (MULTIPLICATIVE DRIFT [6, 17]).** *Let  $(X_t)_{t \in \mathbb{N}}$  be a stochastic process over  $\mathbb{R}$ ,  $x_{\min} > 0$  and let  $T := \min\{t : X_t < x_{\min}\}$ . Suppose that  $X_0 \geq x_{\min}$  and, for all  $t \leq T$ , it holds that  $X_t \geq 0$ , and there exists some  $\delta > 0$  such that, for all  $t < T$ ,  $E[X_t - X_{t+1} | X_0, \dots, X_t] \geq \delta X_t$ , then,*

- (1)  $E[T | X_0] \leq \frac{\ln(X_0/x_{\min})+1}{\delta}$ , and
- (2)  $\Pr\left(T \geq \frac{\ln(X_0/x_{\min})+r}{\delta}\right) \leq e^{-r}$

The fitness function in Equation (1) ensures that Algorithm 1 quickly finds a feasible cover, which is captured in Theorem 2, which was proved asymptotically in [8, Theorem 1]. We restate this result here with a simple upper bound with leading constants using drift.

**THEOREM 2.** *The expected time until the (1+1) EA finds a feasible cover for any graph on  $n$  vertices is at most  $\frac{1}{2}(en \ln n + en)$ .*

**PROOF.** Let  $(X_t)_{t \in \mathbb{N}}$  be the stochastic process that counts the number of edges uncovered by the candidate solution in iteration  $t$  of the (1+1) EA. For any vertex  $u$ , denote as  $d_t(u)$  the count of uncovered edges incident to  $u$  in iteration  $t$ . Since any vertex  $u$  is flipped with probability  $(1 - 1/n)^{n-1}(1/n) \geq (en)^{-1}$ , and an increase in uncovered edges is never accepted, we may bound the drift of  $(X_t)$  as

$$E[X_t - X_{t+1} | X_t] \geq \sum_u \frac{d_t(u)}{en} = \frac{2X_t}{en}$$

since each of the  $X_t$  uncovered edges is counted twice in the sum over  $d_t$ . The claim follows by Theorem 1.  $\square$

**DEFINITION 1.** *Let  $n, k \in \mathbb{N}$  and  $p \in (0, 1)$ . The  $\mathcal{G}(n, k, p)$  model of random planted graphs is a distribution of random graphs on  $n$  vertices defined by construction as follows.*

*Let  $V$  be a set of  $n$  (labeled) vertices. Choose a  $k$ -subset  $C \subset V$  uniformly at random, and for each  $u, v \in V$ , if  $\{u, v\} \cap C \neq \emptyset$ , add edge  $uv$  to  $E$  with probability  $p$ .*

*In the resulting graph  $G = (V, E)$ , we refer to  $C$  as the core, and each  $v \in C$  as a core vertex. We refer to vertices in  $V \setminus C$  as fringe vertices.*

### 3 SMALL $k$

In this section we consider  $\mathcal{G}(n, k, p)$  where  $k \leq \ln n$ . Our results rely heavily on the following property of planted vertex cover graphs, which we call  $\delta$ -heaviness.

**DEFINITION 2.** Let  $G = (V, E)$  be a graph drawn from the  $\mathcal{G}(n, k, p)$  model. For a constant  $0 < \delta < 1$ , we say  $G$  is  $\delta$ -heavy if for every subset  $S \subset V \setminus C$  where  $|S| = \delta|V \setminus C|$ , every core vertex in  $C$  is adjacent to at least  $\ln n$  vertices in  $S$ .

**LEMMA 1.** Let  $G = (V, E)$  be a graph drawn from the  $\mathcal{G}(n, k, p)$  model. Let  $\delta, p \in (0, 1)$  be constants. If  $p > \sqrt{\frac{1-\ln\delta}{2}}$ , then  $G$  is  $\delta$ -heavy with probability  $1 - e^{-\Omega(n)}$ .

**PROOF.** Fix an arbitrary  $v \in C$  and an arbitrary  $\delta(n - k)$ -sized subset  $S \subset V \setminus C$ . We first bound the probability that  $v$  is adjacent to no more than  $\ln n$  vertices in  $S$ . Let  $X$  be the random variable that counts the edges between  $v$  and vertices in  $S$ . Each edge from  $v$  to a vertex in  $S$  appears independently with probability  $p$ , so  $X$  is the sum of  $|S|$  independent Bernoulli random variables, each with success probability  $p$  so  $E[X] = p|S|$ . By Hoeffding's inequality [9], for any  $t > 0$ ,  $\Pr(X \leq E[X] - t) < e^{-2t^2/|S|}$ , thus the probability that  $v$  is adjacent to at most  $\ln n$  vertices in  $S$  can be estimated by

$$\begin{aligned} \Pr(X \leq \ln n) &= \Pr(X \leq E[X] - (E[X] - \ln n)) \\ &< e^{-(p|S| - \ln n)^2/|S|} \\ &= \exp\left(-2\left(p^2|S| + \frac{\ln^2 n}{|S|} - 2p\ln n\right)\right) \\ &\leq \exp\left(-2\delta p^2(n - k) + 4p\ln n\right). \end{aligned}$$

We have assumed  $k \leq \ln n$ , so this probability is at most

$$\exp\left(-2\delta p^2(n - \ln n) + 4p\ln n\right) < \exp\left(-2\delta p^2n + 6p\ln n\right).$$

Note that we have used here the fact that  $\delta < 1$  and  $p^2 < p$ . Taking a union bound over all  $k$  vertices  $v \in C$ , the probability that any core vertex is adjacent to fewer than  $\ln n$  vertices in  $S$  is at most

$$\exp\left(-2\delta p^2n + 6p\ln n + \ln k\right).$$

A final union bound over all subsets  $S$  of size  $\delta|V \setminus C| = \delta(n - k)$  shows the probability that  $G$  is not  $\delta$ -heavy is at most

$$\begin{aligned} &\binom{n}{\delta n} \exp\left(-2\delta p^2n + 6p\ln n + \ln k\right) \\ &\leq \frac{e^{\delta n} n^{\delta n}}{(\delta n)^{\delta n}} \exp\left(-2\delta p^2n + 6p\ln n + \ln k\right) \\ &= \exp\left(-2\delta p^2n + 6p\ln n + \ln k + \delta n \ln(e/\delta)\right) \\ &\leq \exp\left(-\delta n(2p^2 - \ln(e/\delta)) + (6p + 1)\ln n\right). \end{aligned}$$

Since  $p > \sqrt{\frac{1-\ln\delta}{2}}$ , and  $p$  and  $\delta$  are taken to be positive constants, we have  $2p^2 - \ln(e/\delta) = \Omega(1)$ , and the probability that  $G$  is not  $\delta$ -heavy is  $e^{-\Omega(n)}$ , which completes the proof.  $\square$

**THEOREM 3.** Consider the  $\mathcal{G}(n, k, p)$  model with  $k \leq \ln n$  and  $p > \sqrt{\frac{1-\ln\delta}{2}}$  for some constant  $\delta \in (1/e, 1)$ . Then for all but an exponentially-fast vanishing fraction of all graphs  $G$  sampled from

$\mathcal{G}(n, k, p)$ , if  $T$  is the runtime for the (1+1) EA to find a  $k$ -cover on  $G$ , we have

$$\Pr(T \leq 2en\ln n + \lfloor en(1 - \delta) \rfloor) = \Omega(n^{-(e(1-\delta)\ln(2e)+\ln 2)}).$$

**PROOF.** Since  $p$  is sufficiently large, by Lemma 1, all but an  $e^{-\Omega(n)}$ -fraction of graphs drawn from  $\mathcal{G}(n, k, p)$  are  $\delta$ -heavy. Thus, we assume for the remainder of the proof that  $G$  is  $\delta$ -heavy.

Let  $\mathcal{E}$  be the event that after exactly  $\lfloor en(1 - \delta) \rfloor$  iterations of the (1+1) EA, the following conditions hold:

- (1) The core vertices  $C$  belong to the current solution of the (1+1) EA,
- (2) There are at least  $\delta n$  fringe vertices that are not part of the current solution of the (1+1) EA.

This is a rather fortunate event for the (1+1) EA, because such a candidate solution is already a feasible vertex cover (as all vertices in  $C$  are present), so after this point no infeasible covers would be accepted. Moreover, since  $G$  is  $\delta$ -heavy, every core vertex is adjacent to at least  $\ln n$  uncovered edges (by condition (2) above). Thus in order to remove a core vertex  $v$  from the cover, a single mutation operation would need to change at least  $\ln n$  neighbors of  $v$  to remain feasible. In contrast, it is always possible to remove any fringe vertex from the current cover. Thus if there are  $i$  fringe vertices in the current solution, the probability to improve the fitness is at least  $i/(en)$ . Furthermore, the probability of flipping at least  $\ln n$  vertices in a single mutation is  $n^{-\omega(1)}$ .

Let  $\{X_t\}_{t \in \mathbb{N}}$  denote the stochastic process that tracks the number of fringe vertices in the cover at time  $t$ . The drift of  $\{X_t\}$  conditioned on  $\mathcal{E}$  and starting at iteration  $\lfloor en(1 - \delta) \rfloor$  is at least  $X_t/en - n^{-\omega(1)} = \Omega(X_t/n)$ . By Theorem 1,

$$\Pr(T < 2en\ln n + \lfloor en(1 - \delta) \rfloor \mid \mathcal{E}) = 1 - o(1)$$

It remains to bound the probability of  $\mathcal{E}$ . Let  $\mathcal{E}_1$  be the event that the initial solution to the (1+1) EA contains every vertex in  $C$  and let  $\mathcal{E}_2$  be the event that the core vertices in  $C$  are not mutated during the first  $\lfloor en(1 - \delta) \rfloor$  iterations of the (1+1) EA. Conditioning on  $\mathcal{E}_1 \cap \mathcal{E}_2$ , the (1+1) EA already starts with a feasible solution and does not remove any core vertices during the first  $\lfloor en(1 - \delta) \rfloor$  steps.

Let  $T_1$  be the random variable that measures the number of iterations until the first time the number of fringe vertices in the cover drops below a  $\delta$ -fraction. Again applying tail bounds on multiplicative drift, and noting that  $1 + \ln\left(\frac{1}{1-\delta}\right) \geq 1 - \delta$  for constant  $0 < \delta < 1$ , under the condition  $\mathcal{E}_1 \cap \mathcal{E}_2$ , the (1+1) EA has reduced the number of fringe vertices in the cover from at most  $n - k$  to at most  $\delta(n - k)$  with probability at least  $1 - 1/e$ . Applying the law of total probability we have

$$\begin{aligned} \Pr(\mathcal{E}) &\geq \Pr(\mathcal{E} \mid \mathcal{E}_1 \cap \mathcal{E}_2) \Pr(\mathcal{E}_1 \cap \mathcal{E}_2) \\ &= \Pr(\mathcal{E} \mid \mathcal{E}_1 \cap \mathcal{E}_2) \Pr(\mathcal{E}_2 \mid \mathcal{E}_1) \Pr(\mathcal{E}_1) \\ &\geq \left(1 - \frac{1}{e}\right) \cdot \left[\left(1 - \frac{1}{n}\right)^k\right]^{\lfloor en(1-\delta) \rfloor} (1/2)^k \quad (1/2)^k \\ &\geq (1 - 1/e) \cdot (2e)^{-ek(1-\delta)} \cdot 2^{-k} \\ &\geq (1 - 1/e) \cdot n^{-(e(1-\delta)\ln(2e)+\ln 2)}, \end{aligned}$$

where we have used  $k \leq \ln n$  in the final inequality.  $\square$

**Algorithm 2:** (1+1) EA with cold restarts

---

```

Input: A fitness function  $f: \{0, 1\}^n \rightarrow \mathbb{R}$  and a run length  $\ell$ 
1  $t \leftarrow 0$ ;
2 while termination criteria not met do
3   if  $t = 0$  then
4      $\quad$  Choose  $x$  uniformly at random from  $\{0, 1\}^n$ ;
5     Create  $y$  by flipping each bit of  $x$  with probability  $1/n$ ;
6     if  $f(y) \leq f(x)$  then  $x \leftarrow y$ ;
7      $t \leftarrow (t + 1) \bmod \ell$ ;
8 return  $x$ ;

```

---

Theorem 3 provides a lower bound on the probability that a run of length at least  $2en \ln n + \lfloor en(1 - \delta) \rfloor$  finds a  $k$ -cover of a random graph with sufficient density. This bound vanishes with  $n$ , but slowly enough that a simple cold-restart strategy (periodically starting over from a randomly generated cover) is guaranteed to be efficient. This is captured by the following corollary.

**COROLLARY 1 (TO THEOREM 3).** *Consider the  $\mathcal{G}(n, k, p)$  model with  $k \leq \ln n$  and  $0.71 \leq p \leq 1$ . Running the (1+1) EA with cold restarts (Algorithm 2) with  $\ell = 3en \ln n$  finds a  $k$ -cover on all but an exponentially-fast vanishing fraction of graphs in  $O(n^{c+1} \log n)$  fitness evaluations where  $0.73 < c \leq e(1 + \ln 2) - 1 < 3.61$  is a constant depending on  $p$ .*

**PROOF.** Let  $\delta = e^{1-2p^2}$ . Since  $p > 0.71$ , we have  $\delta \in (1/e, 1)$ . Thus the conditions for Theorem 3 are satisfied, and the success probability for an independent run of length  $3en \ln n$  of the (1+1) EA is  $\Omega(n^{-(e(1-\delta) \ln(2e)+\ln 2)})$ . Under this condition, the number of independent runs until a success is geometrically distributed with expectation  $n^{e(1-\delta) \ln(2e)+\ln 2} = n^{e(1-e^{1-2p^2})(1+\ln 2)+\ln 2}$ , and  $c$  can be chosen appropriately.  $\square$

## 4 LARGE $k$

We now consider  $\mathcal{G}(n, k, p)$  in which  $k > \ln n$ . We will make use of the following probabilistic bound on the size of independent sets in the core.

**LEMMA 2.** *Suppose  $G$  is drawn from the  $\mathcal{G}(n, k, p)$  model with  $k = \omega(1)$ . Then with probability  $1 - o(1)$ , the largest independent set in  $C$  has size at most  $(1 + 2/p) \ln k + 1$ .*

**PROOF.** Set  $\ell := \lceil (1 + 2/p) \ln k + 1 \rceil$ . There are  $\binom{k}{\ell}$  size- $\ell$  vertex sets in  $C$ . We label these sets from 1 to  $\binom{k}{\ell}$  and consider a sequence  $X_1, \dots, X_{\binom{k}{\ell}}$  of indicator random variables over  $\mathcal{G}(n, k, p)$  where

$$X_i = \begin{cases} 1 & \text{if the } i\text{-th size-}\ell \text{ subset of } C \text{ is an independent set in } G, \\ 0 & \text{otherwise.} \end{cases}$$

Consider the sum  $X = X_1 + \dots + X_{\binom{k}{\ell}}$  and note that  $X = 0$  if and only if there are no independent sets of size  $\ell$  or larger in  $G$ . By

Markov's inequality,

$$\begin{aligned} \Pr(X \geq 1) &\leq \mathbb{E}[X] = \binom{k}{\ell} (1-p)^{\binom{\ell}{2}} \leq k^\ell \left( (1-p)^{(\ell-1)/2} \right)^\ell \\ &\leq (\exp(\ln k - p(\ell-1)/2))^\ell, \text{ since } 1-p \leq e^{-p}, \\ &= \exp\left(-\left[\left(1 + \frac{p}{2}\right) \ln k + \frac{p}{2}\right] \ln k\right) \\ &\leq e^{-\ln^2 k}, \end{aligned}$$

since  $p \geq 0$ .  $\square$

**THEOREM 4.** *Consider a graph  $G$  drawn from the  $\mathcal{G}(n, k, p)$  model with  $k > \ln n$ . Then with probability  $1 - o(1)$  (taken over the model), the expected runtime of the (1+1) EA to find a cover of size at most  $k$  on  $G$  is  $O\left(k^{4k\left(1 + \frac{1}{p}\right)} n \log n\right)$ .*

**PROOF.** By Theorem 2, the (1+1) EA takes at most  $\frac{1}{2}(en \ln n + en)$  steps in expectation to find a feasible solution, after which the (1+1) EA never accepts an infeasible solution.

Consider the potential function  $\phi(x) = \max\{0, f(x) - k\}$  and note that when  $\phi(x) = 0$ ,  $x$  is a feasible cover of size at most  $k$ . Moreover,  $\phi$  cannot increase during the run of the (1+1) EA.

By Lemma 2, the largest independent set in the core of  $G$  contains at most  $(1 + \frac{2}{p}) \ln k + 1$  vertices with probability  $1 - o(1)$ , and we condition on this event for the remainder of the proof. Consider the stochastic process  $(X_t)_{t \in \mathbb{N}}$ , which corresponds to the potential in the  $t$ -th iteration.

We seek to bound the drift of  $(X_t)$  after finding a feasible solution. Assume that the (1+1) EA has already found a feasible solution, and let  $C$  be the core vertices of  $G$ . Let  $x$  be the current solution. We make the following case distinction on  $x$ .

**Case 1:**  $C \cap \{i : x[i] = 0\} = \emptyset$ . In this case, all of the vertices in  $C$  are in the cover described by  $x$ . Thus, any fringe vertex can be removed from the current cover and the resulting set is still a cover. A particular vertex is removed from the cover with probability  $(1/n)(1 - 1/n)^{n-1}$  and there are  $f(x) - k$  fringe vertices, so the drift in this case is

$$\mathbb{E}[X_t - X_{t+1} \mid X_t] \geq \frac{f(x) - k}{n} \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{X_t}{en}.$$

**Case 2:**  $C \cap \{i : x[i] = 0\} \neq \emptyset$ . In this case, some of the core vertices are not in the cover described by  $x$ . Let  $Z := C \cap \{x[i] = 0\}$  be the set of core vertices that are not in the current cover. Note that since  $x$  is feasible  $Z$  must be an independent set in  $C$  (otherwise there would be an uncovered edge in  $C$ ).

Let  $Z'$  be an arbitrary set of exactly  $|Z|$  fringe vertices that belong to the current solution  $x$ , i.e.,  $Z' \subseteq \{i : x[i] = 1\} \cap (V \setminus C)$  with  $|Z'| = |Z|$ . Such a  $Z'$  must exist, otherwise we would have  $f(x) < k$ . Let  $\mathcal{E}$  denote the event that mutation changes all of the zero-bits corresponding to  $Z$  into one-bits, and all of the one-bits corresponding to  $Z'$  to zero. Since each bit is mutated independently, we may invoke the principle of deferred decisions [18] and assume that the choices are first made for the bits in  $Z$  and  $Z'$  to produce a partially mutated offspring. Hence, we assume that  $\mathcal{E}$  has occurred, and consider the random choices on the remaining

bits corresponding to  $V \setminus (Z \cap Z')$ . There are  $f(x) - (k - |Z|) = f(x) - k + |Z|$  fringe vertices in  $x$ , and after removing  $|Z'| = |Z|$  fringe vertices, there are still  $f(x) - k = X_t$  fringe vertices that have not yet been considered for mutation, so we may assume that we are in Case 1, now with exactly  $f(x) - k = X_t$  fringe vertices remaining in the cover. Since  $X_t - X_{t+1} \geq 0$ , by the law of total expectation, we can bound the drift from below as follows.

$$\begin{aligned} \mathbb{E}[X_t - X_{t+1} \mid X_t] &\geq \mathbb{E}[X_t - X_{t+1} \mid X_t \cap \mathcal{E}] \Pr(\mathcal{E}) \\ &\geq n^{-2|Z|} \frac{X_t}{en}, \end{aligned}$$

since  $\Pr(\mathcal{E}) = n^{-(|Z|+|Z'|)} = n^{-2|Z|}$ .

In either case, the drift is at least  $n^{-2|Z|} \frac{X_t}{en}$ , but we have assumed via Lemma 2 that  $|Z| \leq (1 + \frac{2}{p}) \ln k + 1 < 2(1 + 1/p) \ln k$  for sufficiently large  $n$  (and hence  $k$ , as  $k \geq \ln n$ ). Therefore, by the multiplicative drift theorem, the expected time until a  $k$ -cover is found is at most

$$\begin{aligned} O(n^{4(1+1/p) \ln k} n \log n) &= O(k^{4(1+1/p) \ln n} n \log n) \\ &= O\left(k^{4k\left(1+\frac{1}{p}\right)} n \log n\right), \end{aligned}$$

since  $\ln n < k$ .  $\square$

## 5 COMPUTATIONAL EXPERIMENTS

To fill in the gaps left open by the previous sections, we report here on a number of experiments that investigate the relationship between the parameters of the planted vertex cover problem. For each experiment, we sample from the  $\mathcal{G}(n, k, p)$  model by constructing a random graph on  $n$  vertices choosing each edge with probability  $p$  as long as at least one incident vertex is in the set  $\{1, \dots, k\}$ . After this, we run the standard (1+1) EA (Algorithm 1) until  $f(x) \leq k$ . For each setting of  $n, k, p$ , we run the algorithm for 100 trials (but sample a new graph from  $\mathcal{G}(n, k, p)$  each time).

To better understand how the runtime depends on  $n$  on dense graphs in which  $k$  is a small function of  $n$ , we plot the average runtime, varying  $n = 100, \dots, 1000$  and fixing  $p = 0.5$ . This is plotted in Figure 1, where we observe a stable runtime varying almost linearly with  $n$ . In Figure 2, we show the same data for runs where  $p$  is also varied with  $n$ , i.e.,  $p = 1/n$ . This corresponds to much sparser graphs, and we see that the runtime has much higher variability, especially for slower growing  $k$ .

This scaling behavior is not so surprising, as we expect that random planted graphs are particularly easy for the (1+1) EA. Similar to the case of random planted satisfiability [7], the relatively uniform structure of the problem is likely to provide a good fitness signal for hill-climbing type algorithms.

Random distributions of problems often undergo a so-called phase transition as various system parameters are varied. Very often, problems sampled near a critical density tend to be (empirically) harder to solve by different algorithms. For example, empirical evidence suggests critically-constrained planted propositional satisfiability formulas are difficult for the (1+1) EA when they are sampled near a critical density [7]. To study the performance of the (1+1) EA on  $\mathcal{G}(n, k, p)$  as a function of graph density, we plot the dependence of the average runtime on  $p$  in Figures 3 and 4, holding

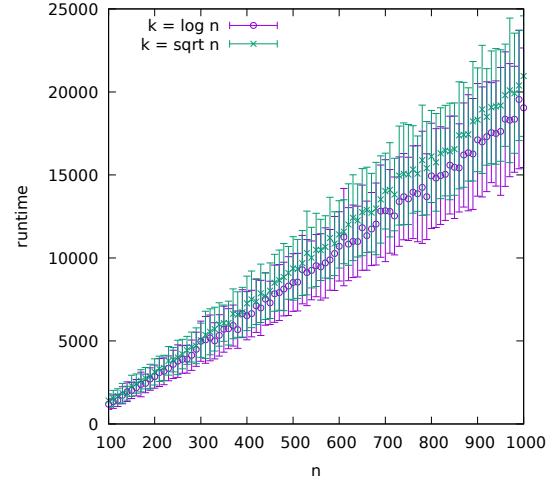


Figure 1: Runtime dependence on  $n$  dense regime ( $p = 0.5$ ) for  $k = \ln n$  and  $k = \sqrt{n}$ . Error bars denote standard deviation.

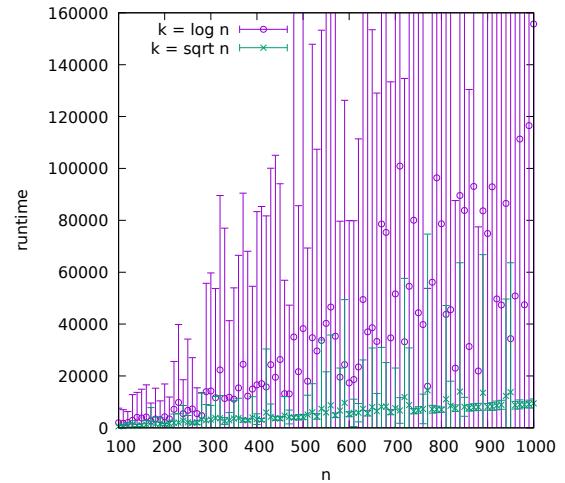
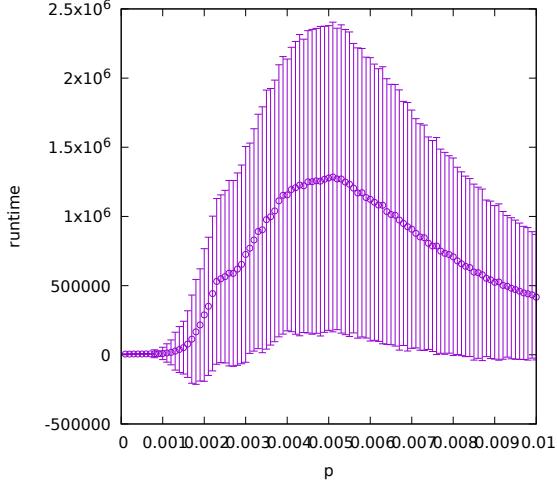
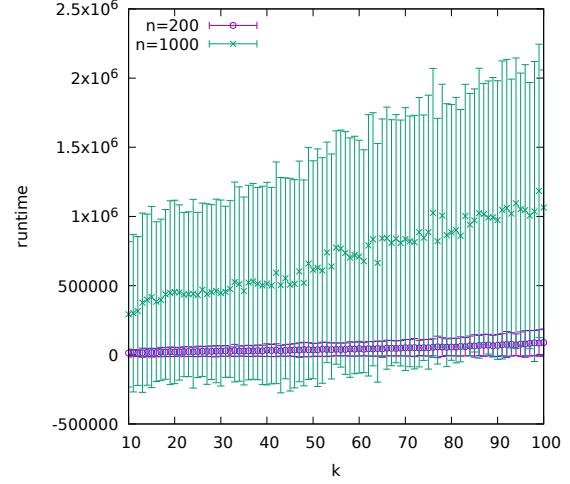


Figure 2: Runtime dependence on  $n$  sparse regime ( $p = 1/n$ ) for  $k = \ln n$  and  $k = \sqrt{n}$ . Error bars denote standard deviation.

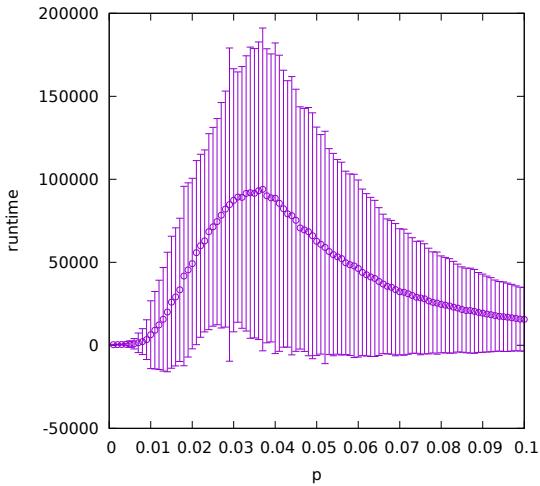
$n$  fixed and averaging over all values of  $k$ . We also see in this case a dependence on graph density in which the (1+1) EA performs worse in a band of not-too-sparse but not-too-dense graphs.



**Figure 3: Runtime dependence on  $p$  for fixed  $n = 1000$  varying  $k = 10, \dots, 100$ . Error bars denote standard deviation.**



**Figure 5: Runtime dependence on  $k$  ( $p$  aggregated). Error bars denote standard deviation.**



**Figure 4: Runtime dependence on  $p$  for fixed  $n = 200$  varying  $k = 10, \dots, 100$ . Error bars denote standard deviation.**

The dependence of runtime on  $k$ , however, is more uniform as we can see in Figure 5. Here we have aggregated over all  $p$  values, which likely explains the large variance, especially in the larger  $n = 1000$  problems.

A more detailed picture is provided by Figures 6 and 7, where we display two-dimensional color plots showing the runtime dependence on both  $k$  and  $p$  simultaneously. On these plots one can see how the density and the cover size influences the efficiency of the (1+1) EA. We conjecture that there is a critical value (or range) of  $p$  at which the (1+1) EA struggles to find a  $k$ -cover.

The (1+1) EA completes execution as soon as it finds a  $k$ -cover. However, this is not necessarily guaranteed to be the  $k$ -cover that was planted in the graph. Indeed, for smaller densities, we would expect many other  $k$ -covers in the graph. To investigate this, in Figure 8 we plot the proportion of runs in which the planted  $k$ -core was recovered (as opposed to some different  $k$ -cover) as a function of  $p$ . The dependence of this characteristic as a function of  $k$  is plotted in Figure 9, and Figures 10 and 11 display this in a color plot for both  $k$  and  $p$  simultaneously.

When the graph is relatively sparse, we would also expect the (1+1) EA to “overshoot”  $k$  by finding an even smaller cover before finding a  $k$ -cover. To understand better how this depends on  $k$  and  $p$ , we plot the average difference between  $k$  and the best fitness found as a function of  $p$  on sparse ( $p = 1/n$ ) instances where  $n$  is varied in Figure 12, and on fixed- $n$  instances in Figures 13 and 14.

## 6 CONCLUSION

In this paper we have presented a parameterized analysis the (1+1) EA on problems drawn from the  $\mathcal{G}(n, p, k)$  random planted vertex cover model. We showed that for dense graphs ( $p > 0.71$ ) and small  $k$ , there is sufficient signal in enough of the space so that the (1+1) EA has a relatively good chance of finding a  $k$ -cover in a polynomial-length run. When  $k$  is large, we showed that a feasible cover cannot leave too much of the planted core uncovered, and therefore the

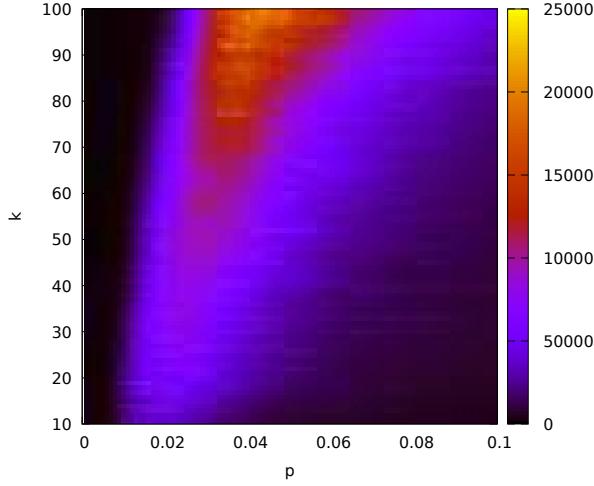


Figure 6: Runtime dependence on both  $k$  and  $p$  for  $n = 200$ .

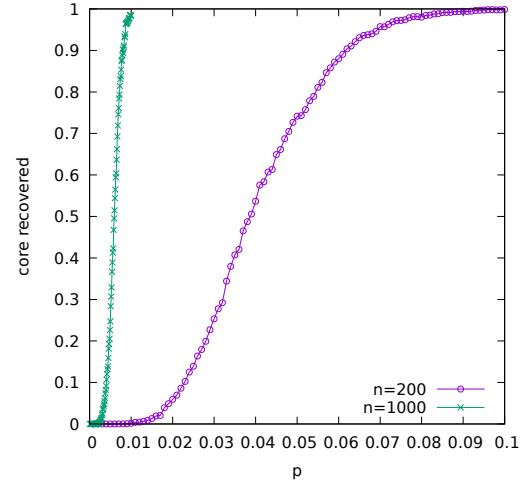


Figure 8: Proportion of runs in which the  $k$ -core was recovered as a function of  $p$ .

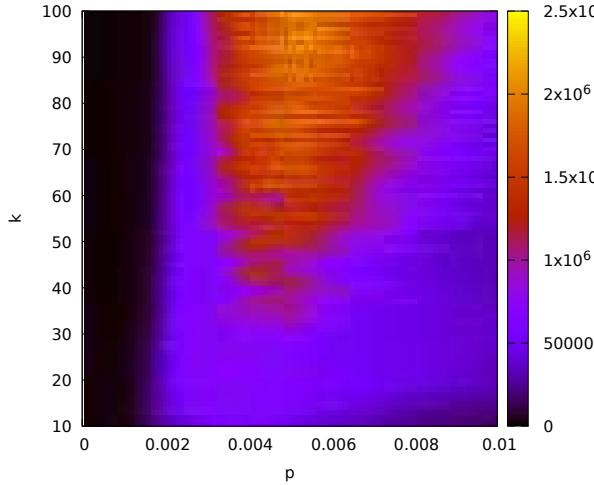


Figure 7: Runtime dependence on both  $k$  and  $p$  for  $n = 1000$ .

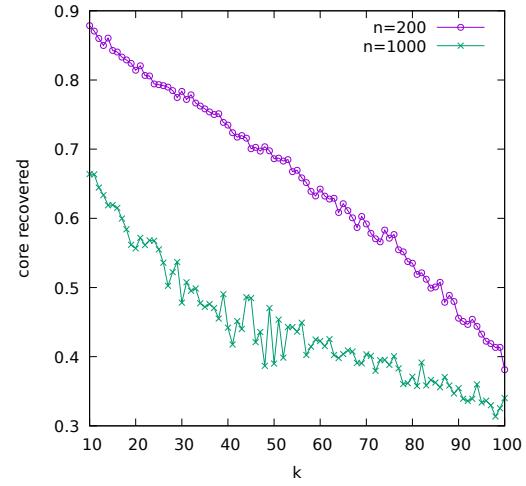
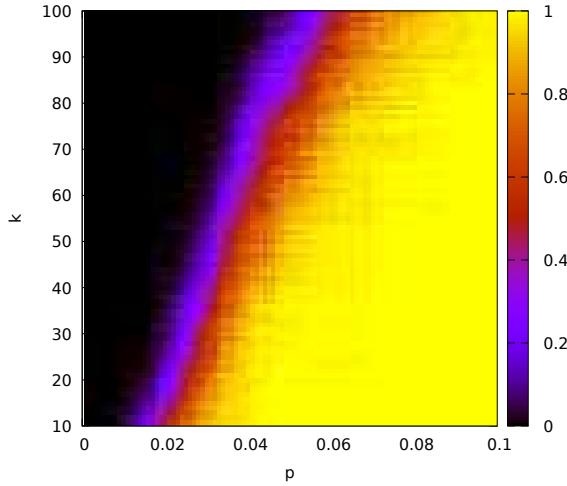


Figure 9: Proportion of runs in which the  $k$ -core was recovered as a function of  $k$ .

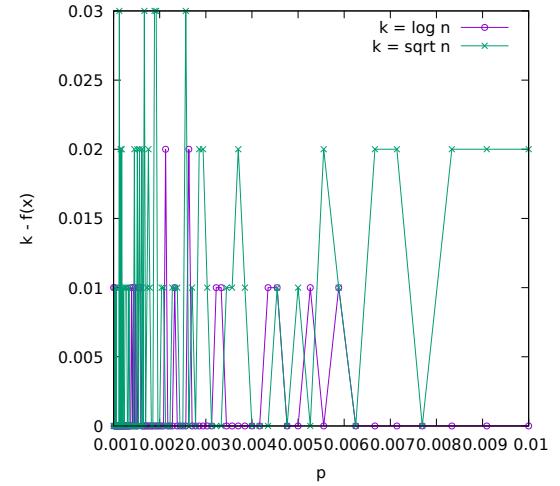
(1+1) EA does not require a large effort to make progress. In the end, this translates to a fixed-parameter tractable runtime for the (1+1) EA with high probability over  $\mathcal{G}(n, p, k)$ .

To fill in the picture, we also reported a number of computational experiments that measure the runtime on graphs drawn from  $\mathcal{G}(n, p, k)$ . These experiments point to a critical value for  $p$  at which

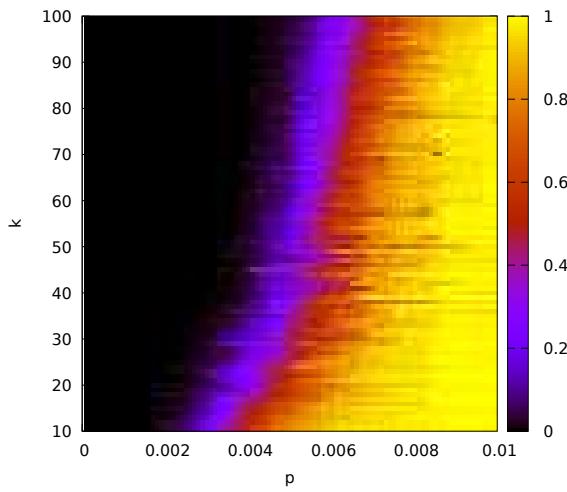
the (1+1) EA requires more time to find any  $k$ -cover, which suggest an interesting direction for future theoretical work to understand this phenomenon better.



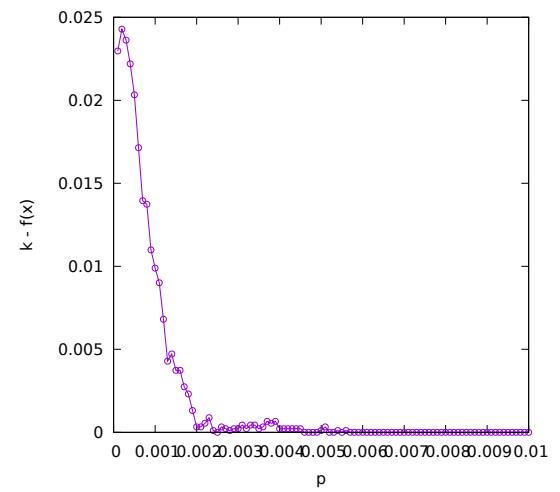
**Figure 10:** Proportion of runs in which the  $k$ -core was recovered as a function of both  $k$  and  $p$  for  $n = 200$ .



**Figure 12:** Average difference between  $k$  and best fitness found as a function of  $p$ .



**Figure 11:** Proportion of runs in which the  $k$ -core was recovered as a function of both  $k$  and  $p$  for  $n = 200$ .



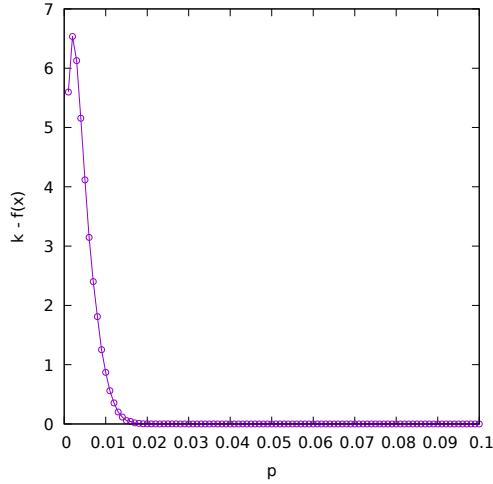
**Figure 13:** Average difference between  $k$  and best fitness found as a function of  $p$  on  $n = 1000$ .

## ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation under grant 2144080 and by the Australian Research Council under grant FT200100536.

## REFERENCES

[1] Emmanuel Abbe and Colin Sandon. 2015. Community Detection in General Stochastic Block models: Fundamental Limits and Efficient Algorithms for Recovery. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*. 670–688. <https://doi.org/10.1109/FOCS.2015.47>



**Figure 14: Average difference between  $k$  and best fitness found as a function of  $p$  on  $n = 200$ .**

[2] Noga Alon, Michael Krivelevich, and Benny Sudakov. 1998. Finding a Large Hidden Clique in a Random Graph. *Random Structures & Algorithms* 13, 3–4 (1998), 457–466. [https://doi.org/10.1002/\(SICI\)1098-2418\(199810/12\)13:3/4%3C457::AID-RSA14%3E3.0.CO;2-W](https://doi.org/10.1002/(SICI)1098-2418(199810/12)13:3/4%3C457::AID-RSA14%3E3.0.CO;2-W)

[3] Samuel Baguley, Tobias Friedrich, Aneta Neumann, Frank Neumann, Marcus Pappik, and Ziena Zeif. 2023. Fixed Parameter Multi-Objective Evolutionary Algorithms for the W-Separator Problem. In *Genetic and Evolutionary Computation Conference (GECCO '23)*. <https://doi.org/10.1145/3583131.3590501>

[4] Austin R. Benson and Jon M. Kleinberg. 2018. Found Graph Data and Planted Vertex Covers. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.), 1363–1374. <https://proceedings.neurips.cc/paper/2018/hash/af4836712c5e77550897e25711e1d96-Abstract.html>

[5] Luke Branson and Andrew M. Sutton. 2021. Focused Jump-and-Repair Constraint Handling for Fixed-Parameter Tractable Graph Problems. In *Proceedings of the 16th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3450218.3477304>

[6] Benjamin Doerr and Leslie Ann Goldberg. 2010. Drift Analysis with Tail Bounds. In *Proceedings of the Eleventh International Conference on Parallel Problem Solving from Nature (PPSN XI) (Lecture Notes in Computer Science, Vol. 6238)*. Springer, 174–183.

[7] Benjamin Doerr, Frank Neumann, and Andrew M. Sutton. 2017. Time Complexity Analysis of Evolutionary Algorithms on Random Satisfiable  $k$ -CNF Formulas. *Algorithmica* 78, 2 (June 2017), 561–586. <https://doi.org/10.1007/s00453-016-0190-3>

[8] Tobias Friedrich, Jun He, Nils Hebbinghaus, Frank Neumann, and Carsten Witt. 2010. Approximating Covering Problems by Randomized Search Heuristics Using Multi-Objective Models. *Evolutionary Computation* 18, 4 (June 2010), 617–633. [https://doi.org/10.1162/EVCO\\_a\\_00003](https://doi.org/10.1162/EVCO_a_00003)

[9] Wassily Hoeffding. 1963. Probability Inequalities for Sums of Bounded Random Variables. *J. Amer. Statist. Assoc.* 58, 301 (1963), 13–30. <https://doi.org/10.1080/01621459.1963.10500830>

[10] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social Networks* 5, 2 (1983), 109–137. [https://doi.org/10.1016/0378-8733\(83\)90021-7](https://doi.org/10.1016/0378-8733(83)90021-7)

[11] Thomas Jansen, Pietro S. Oliveto, and Christine Zarges. 2013. Approximating vertex cover using edge-based representations. In *Proceedings of the Twelfth Workshop on Foundations of Genetic Algorithms (FOGA XII)*, Adelaide, SA, Australia, January 16–20, 2013, Frank Neumann and Kenneth A. De Jong (Eds.). ACM, 87–96. <https://doi.org/10.1145/2460239.2460248>

[12] Mark Jerrum. 1992. Large Cliques Elude the Metropolis Process. *Random Structures & Algorithms* 3, 4 (1992), 347–359. <https://doi.org/10.1002/rsa.3240030402>

[13] Ari Juels and Marcus Peinado. 2000. Hiding Cliques for Cryptographic Security. *Designs, Codes and Cryptography* 20, 3 (2000), 269–280.

[14] Subhash Khot and Oded Regev. 2008. Vertex cover might be hard to approximate to within  $(2 - \epsilon)$ . *J. Comput. System Sci.* 74, 3 (2008), 335–349. <https://doi.org/10.1016/j.jcss.2007.06.019> Computational Complexity 2003.

[15] Sami Khuri and Thomas Bäck. 1994. An Evolutionary Heuristic for the Minimum Vertex Cover Problem. In *Workshops of the Eighteenth Annual German Conference on Artificial Intelligence (KI-94)*, Saarbrücken, Germany, J. Kunze and H. Stoyan (Eds.), 86–90.

[16] Stefan Kratsch and Frank Neumann. 2012. Fixed-Parameter Evolutionary Algorithms and the Vertex Cover Problem. *Algorithmica* 65, 4 (May 2012), 754–771. <https://doi.org/10.1007/s00453-012-9660-4>

[17] Timo Kötzing and Martin S. Krejca. 2019. First-hitting times under drift. *Theoretical Computer Science* 796 (2019), 51–69.

[18] Michael Mitzenmacher and Eli Upfal. 2005. *Probability and Computing : Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press.

[19] Pietro S. Oliveto, Jun He, and Xin Yao. 2009. Analysis of the (1+1) EA for Finding Approximate Solutions to Vertex Cover Problems. *IEEE Transactions on Evolutionary Computation* 13, 5 (2009), 1006–1029. <https://doi.org/10.1109/TEVC.2009.2014362>

[20] Daniel M. Romero, Brian Uzzi, and Jon M. Kleinberg. 2019. Social Networks under Stress: Specialized Team Roles and Their Communication Structure. *ACM Trans. Web* 13, 1 (2019), 6:1–6:24. <https://doi.org/10.1145/3295460>

[21] Tobias Storch. 2007. Finding large cliques in sparse semi-random graphs by simple randomized search heuristics. *Theoretical Computer Science* 386 (2007), 114–131. <https://doi.org/10.1016/j.tcs.2007.06.008>