Algorithmica manuscript No.
(will be inserted by the editor)

Time Complexity Analysis of Evolutionary Algorithms on Random Satisfiable k-CNF Formulas

Benjamin Doerr · Frank Neumann ·
Andrew M. Sutton

the date of receipt and acceptance should be inserted later

Abstract We contribute to the theoretical understanding of randomized search heuristics by investigating their optimization behavior on satisfiable random k-satisfiability instances both in the planted solution model and the uniform model conditional on satisfiability. Denoting the number of variables by n, our main technical result is that the simple (1+1) evolutionary algorithm with high probability finds a satisfying assignment in time $O(n \log n)$ when the clause-variable density is at least logarithmic. For low density instances, evolutionary algorithms seem to be less effective, and all we can show is a subexponential upper bound on the runtime for densities below $\frac{1}{k(k-1)}$. We complement these mathematical results with numerical experiments on a broader density spectrum. They indicate that, indeed, the (1+1) EA is less efficient on lower densities. Our experiments also suggest that the implicit constants hidden in our main runtime guarantee are low. Our main result extends and considerably improves the result obtained by Sutton and Neumann [*Lecture notes in computer science*, Vol. 8672 (pp. 942–951)] in terms of runtime, minimum density, and clause length. These improvements are made possible by establishing a close fitness-distance correlation in certain parts of the search space. This approach might be of independent interest and could be useful for other average-case analyses of randomized search heuristics. While the notion of a fitness-distance correlation has been around for a long time, to the best of our knowledge, this is the first time that fitness-distance correlation is explicitly used to rigorously prove a performance statement for an evolutionary algorithm.

Benjamin Doerr
École Polytechnique, Université Paris-Saclay, Palaiseau, France

Frank Neumann
School of Computer Science, University of Adelaide, Adelaide, Australia

Andrew M. Sutton
Hasso-Plattner-Institut, Universität Potsdam, Potsdam, Germany

1 Introduction

Randomized search heuristics such as randomized local search, evolutionary algorithms, and ant colony optimization have been widely used to solve complex combinatorial optimization and engineering problems. Their popularity with practitioners lies in the broad and easy applicability to many complex problems in a number of application domains. In contrast to this, the theoretical understanding still lags behind their practical success due to complex random processes underlying the run of such algorithms. Nevertheless, the analysis of randomized search heuristics has made significant progress over the past fifteen years. A wide range of randomized search heuristics has been analyzed for specific fitness functions as well as problems from combinatorial optimization. We refer the reader to the timely books [5, 21, 33] for a comprehensive presentation.

In this paper, we study the behavior of randomized search heuristics on one of the most classical combinatorial optimization problems, namely satisfiability. Our aim is to get new insights into the behavior of randomized search heuristics when dealing with random k-CNF formulas. Random instances give insight in the average-case ("typical") behavior of an algorithm as opposed to worst-case analyses, which give upper bounds valid for all instances (and thus absolute guarantees), sometimes at the price of being very pessimistic. So far, there are only a few runtime results of randomized search heuristics on random instances of combinatorial optimization problems. Witt [40] investigated random instances for makespan scheduling. Based on an analysis of the well-known 2-OPT local search algorithm on random instances of the Traveling Salesperson problem (TSP) carried out in [17], results have been obtained for ant colony optimization [25]. Furthermore, runtime results using the fixed budget perspective [**?**] have been obtained for evolutionary algorithms [32] and random instances of the TSP.

In the first runtime analysis for the (1+1) EA on random satisfiability instances [39], a runtime bound of $O(n^2 \log n)$ was shown to hold with probability $1 - o(1)$ for the 3-satisfiability problem when the number of random clauses is at least $\Omega(n^2)$, that is, the density is linear. The key argument used in the proof is that a large proportion of the search space (with sufficiently high probability) has no pair of Hamming neighbors with fitness gradient pointing away from the planted assignment. This condition implies that any search point has a Hamming neighbor closer to the planted solution with strictly higher fitness. Unfortunately, it turns out that this condition is not sufficient to handle search trajectories that do not only move towards the optimum via moves to Hamming neighbors. Such trajectories arise, e.g., from mutation steps that change more than one bit at once. This weakness leads to an additional linear factor in the runtime bound, which we overcome in this paper by establishing a fitness-distance correlation (FDC) that implies that trajectories involving multiple bit flips also make sufficient progress towards the target.

Our proof techniques utilize rigorous bounds on the FDC of an instance to show there is a sufficiently strong fitness signal that yields a stochastic drift toward the optimal solution. This technique may be of independent interest, and should be extensible to other algorithms and analyses. Historically, the notion of fitness-distance correlation has been used to qualitatively explain the hardness of a problem by considering how the distance of search points to an optimum relates to their fitness values [23]. The intuition is that problems are easy to solve by evolutionary algorithms if the fitness improves with decreasing distance to the optimum and hard to solve if the fitness is pointing in the opposite direction. While the intuition sounds sensible, it does not always translate directly into an accurate prediction of algorithm performance. Different counterexamples have been presented in the literature that show FDC is not always a good predictor of algorithm performance (see, e.g., [4,20,35]).

Furthermore, a strong FDC is only a reliable predictor if a randomized search heuristic does not encounter any deviations from the assumed usual behavior. In the case that a deviation from the predicted behavior becomes very unlikely, a strong FDC can potentially be used to accurately predict the runtime of randomized search heuristics. This property is explored in this paper, and we show that there is a strong FDC for sufficiently dense k-CNF formulas. Usually, the FDC is established by sampling search points and calculating the empirical correlation between fitness and the distance to a known optimum. In order to make it useful for upper bounds on the runtime of randomized search heuristics, we must be able to make rigorous statements about the properties of the relationship between fitness and distance and show that those properties hold with high probability. We also require such statements to explicitly depend on the input size, and hence are valid for all problem sizes larger than a reasonable minimum bound. Experimental investigations into FDC, on the other hand, can only make statements about fixed problem sizes.

We prove rigorous bounds on a suitable notion of FDC for k-CNF formulas having at least logarithmic density. This admits a proof of an improved runtime bound of $O(n \log n)$, attained with probability $1 - o(1)$, for these instances. We also present a straightforward matching lower bound for asymptotically almost all satisfiable k-CNF formulas of sufficiently high constraint density.

We begin by studying the *planted* model of random k-CNF distributions and extend our results to the *filtered* model using a straightforward generalization of a correspondence on 3-CNF formulas due to Ben-Sasson et al. [6]. Planted distributions for the maximum clique problem in graphs have also been studied by Storch [38] in the context of randomized search heuristics. In propositional satisfiability, the planted distribution of k-CNF formulas is known to be easy to solve for classical algorithms [27], and our objective is to advance the theoretical analysis of evolutionary algorithms on random satisfiability models. This article is based on its conference version [14] which carried out the investigations for 3-CNF formulas. The present article generalizes the results obtained in the conference version to k-CNF formulas where $k \geq 3$ is a constant.

The outline of the paper is a follows. We introduce the model and algorithm under investigation in Section 2. We start our analysis by investigating formulas of high (linear) density in Section 3 and prove the $O(n \log n)$ bound.

We then extend this analysis to formulas of logarithmic density and present a matching lower bound on the expectation. Finally, we give a short proof in Section 4 that the runtime of the (1+1) EA is faster than exponential for very low constant densities. Our theoretical results are complemented by experimental investigations in Section 5. We conclude the paper in Section 6.

2 Preliminaries

A k-CNF formula F is obtained from a set of n Boolean variables $\{v_1, v_2, \dots, v_n\}$ by forming a logical conjunction of exactly m clauses $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$. Each clause is the logical disjunction of exactly k literals, $C_i = \ell_{i_1} \vee \cdots \vee \ell_{i_k}$ and each literal ℓ_{i_j} is either an occurrence of a variable v or its negation $\neg v$. A k-CNF formula F is *satisfiable* if and only if there is an assignment of variables to truth values so that every clause contains at least one true literal.

The set of all assignments to a set of n Boolean variables is isomorphic to $\{0,1\}^n$ by interpreting the bit at each position i of the string as the state of exactly one Boolean variable v_i. Given a string $x \in \{0,1\}^n$, we say a disjunctive clause C is *satisfied* by x if it evaluates to true under the variable assignment corresponding to x. Otherwise, we say it is *unsatisfied* or *false*. For a length-m formula F on n variables, we define the fitness function $f := f_F \colon \{0,1\}^n \to \{0, \dots, m\}$ defined by $f_F(x) := |\{C \in F : \ C \text{ is satisfied by } x\}|$. If F is satisfiable, the task of finding a satisfying assignment reduces to the task of maximizing f.

The standard (1+1) EA, illustrated in Algorithm 1, is a basic evolutionary algorithm that maintains a size-one population and produces a single offspring in each step. It can be characterized as a stochastic hill-climbing search that uses the standard bit-wise uniform mutation operator. Given a length-m formula F on n variables we seek an asymptotic bound on the runtime of the (1+1) EA searching for a satisfying assignment to F by optimizing the corresponding pseudo-Boolean function $f = f_F$. We study the infinite stochastic process $\{x^{(t)} : t \in \mathbb{N}_0\}$ on $\{0,1\}^n$ where $x^{(t)}$ is the assignment generated in iteration t of Algorithm 1. The runtime of the (1+1) EA is the random variable $T = \inf\{t \in \mathbb{N}_0 : f(x^{(t)}) = m\}$.

Algorithm 1: The (1+1) EA.

```
choose $x \in \{0,1\}^n$ uniformly at random;
repeat forever
    $y \leftarrow x$;
    flip each bit of $y$ independently with probability $1/n$;
    if $f(y) \geq f(x)$ then $x \leftarrow y$;
```

In order to bound the runtime of the (1+1) EA, we will consider the sequence $(x^{(0)}, x^{(1)}, \dots)$ of assignments generated by the (1+1) EA and study the drift of corresponding stochastic processes that measure fitness values and distance values along this sequence. To make precise statements about the runtime, we rely heavily on the following drift theorem.

Theorem 1 (Multiplicative Drift [12,13]) *Let $\{X_t : t \in \mathbb{N}_0\}$ be a sequence of random variables over $\mathbb{R}_{\geq 0}$. Let T be the random variable that denotes the earliest point in time $t \geq 0$ such that $X_t < 1$. Assume that there exists $\delta > 0$ such that, for all $a \geq 1$, $\mathrm{E}(X_t - X_{t+1} \mid X_t = a) \geq \delta\, a$. Then for all $a \geq 1$, we have*

$$\mathrm{E}(T \mid X_0 = a) \leq \frac{1 + \ln(a)}{\delta}$$

and

$$\Pr\left(T > \frac{\lambda + \ln(a)}{\delta} \;\middle|\; X_0 = a\right) \leq e^{-\lambda} \text{ for all } \lambda > 0.$$

2.1 Random k-CNF distributions

Throughout the paper we assume that $k \geq 3$ is an integer constant. We consider distributions of k-CNF formulas consisting of m clauses of length exactly k over n distinct variables. We also impose the assumption that each clause consists of distinct variables. We assume clauses are sampled with replacement (i.e., repeated clauses are allowed). This assumption is common, and simplifies the proofs.

Definition 1 Let $\Omega_{n,m,k}$ be the finite set of all k-CNF formulas over n variables and m clauses.

We say a property holds for *asymptotically almost all* formulas with n variables and $m := m(n)$ clauses if that property holds for all formulas in $\Omega_{n,m,k}$ except for a set of measure tending to zero with $n \to \infty$.

We associate random k-CNF distributions with categorical distributions over the sample space $\Omega_{n,m,k}$. In particular, the well-known *uniform distribution* $\mathcal{U}_{n,m,k}$ is defined by

$$\Pr(F) = |\Omega_{n,m,k}|^{-1}$$

for all $F \in \Omega_{n,m,k}$. The *filtered distribution* $\mathcal{U}^{\mathrm{SAT}}_{n,m,k}$ is the uniform distribution conditioned on satisfiability, that is, we have

$$\Pr(F) = |\{F \in \Omega_{n,m,k} : F \text{ is satisfiable}\}|^{-1}$$

for all satisfiable formulas F. The *planted distribution* $\mathcal{P}_{n,m,k}$ is the uniform distribution conditioned on satisfiability by a “planted assignment” $x^\star$. For all formulas F satisfied by the assignment $x^\star$, we have

$$\Pr(F) = |\{F \in \Omega_{n,m,k} : F \text{ is satisfied by } x^\star\}|^{-1}.$$

When considering a formula F constructed from $\mathcal{P}_{n,m,k}$, without loss of generality we will hereafter assume that the planted solution is $x^\star = (1, 1, \ldots, 1)$

since the behavior of the (1+1) EA is invariant under negating literals of F. We define the function $d\colon \{0,1\}^n \to \{0,\dots,n\}; x \mapsto |\{i : x_i = 0\}|$ that measures the Hamming distance to the planted solution.

Definition 2 Fix a small constant $\epsilon > 0$. We define a subset of directed hypercube edges $\mathcal{H} = \mathcal{H}_\epsilon \subseteq \{0,1\}^n \times \{0,1\}^n$ such that $(x,y) \in \mathcal{H}$ if and only if

1. $|\{i : x_i \neq y_i\}| = 1$,
2. $d(y) = d(x) - 1$, and
3. $d(x) \leq (1/2 + \epsilon)n$

The following lemma introduces a two-sided bound on the expected difference in fitness between pairs in $\mathcal{H}$, provided that F is drawn from the $\mathcal{P}_{n,m,k}$ distribution.

Lemma 1 *Let $(x,y) \in \mathcal{H}$. Let $F \sim \mathcal{P}_{n,m,k}$ and $f := f_F$. Then there exists a $\gamma_k \colon \mathbb{R} \to \mathbb{R}$ such that*

$$\frac{km}{(2^k-1)n}(1-\gamma_k(n)) \leq \mathrm{E}(f(y) - f(x)) \leq \frac{km}{(2^k-1)n},$$

where $\lim_{n\to\infty} \gamma_k(n) = 1 - \left(\frac{1-2\epsilon}{2}\right)^{k-1}$.

Proof Let A be the set of all k-CNF clauses on n variables with at least one positive literal that are not satisfied by x but are satisfied by y. Similarly, let B be the set of all k-CNF clauses on n variables with at least one positive literal that are satisfied by x but not satisfied by y. Let $C \supset A \cup B$ be the set of all k-CNF clauses on n variables with at least one positive literal.

We begin by computing the sizes of the sets A, B and C. Let $i \in [n]$ be the unique index in which x and y differ. Then every clause in A must contain the variable v_i as a positive literal. Furthermore, since the clause is unsatisfied under x, the polarity of the remaining $k-1$ literals in the clause is uniquely determined by their corresponding bit values in x. There are $k-1$ remaining variables to pick from the set of all variables (excluding v_i), so $|A| = \binom{n-1}{k-1}$.

Similarly, every clause in B must contain the negative literal $\neg v_i$, and the polarity of the remaining literals in the clause is again uniquely determined by the state of the corresponding bit values in y. However, this also counts clauses that contain no positive literal, and so these must be subtracted out. Any k-clause not satisfied by y has no positive literal if and only if it is comprised entirely of literals that correspond to variables that are true under y. There are $n - d(y)$ such variables, so

$$|B| = \binom{n-1}{k-1} - \binom{n-d(y)-1}{k-1} \leq \binom{n-1}{k-1} - \binom{n(1/2-\epsilon)}{k-1},$$

where the inequality follows from $d(y) = d(x) - 1 \leq (1/2+\epsilon)n - 1$. Setting

$$\gamma_k(n) = 1 - \binom{n(1/2-\epsilon)}{k-1} \Big/ \binom{n-1}{k-1},$$

we have

$$0 \leq |B| < \gamma_k(n)\binom{n-1}{k-1}. \tag{1}$$

Finally, we note that the set C is constructed by all k-clauses that contain at least one positive literal, so $|C| = (2^k - 1)\binom{n}{k}$.

To finish the proof, suppose $F \sim \mathcal{P}_{n,m,k}$. Let Z_A be the random variable that counts the occurrences of clauses from A in F and Z_B be the random variable that counts the occurrences of clauses from B in F. Since F contains exactly m clauses chosen from C uniformly at random, we have $E(Z_A) = m|A|/|C|$ and $E(Z_B) = m|B|/|C|$. Hence,

$$\mathrm{E}(f(y) - f(x)) = \mathrm{E}(Z_A - Z_B) = m\left(\frac{|A| - |B|}{|C|}\right),$$

and the claimed two-sided bound follows from (1). □

Lemma 2 *Let n be sufficiently large and m a function of n. Let $(x, y) \in \mathcal{H}$. Let $F \sim \mathcal{P}_{n,m,k}$ and $f = f_F$. Then*

$$\Pr\left(\frac{c_1 m}{n} < f(y) - f(x) < \frac{c_2 m}{n}\right) = 1 - e^{-\Omega(m/n)}$$

for particular positive constants c_1 and c_2 that depend on k.

Proof We consider the random variables $Z = Z_A - Z_B$ from the proof of Lemma 1. Note that Z_A and Z_B can each be written as the sum of m independent indicator random variables (indicating whether or not the i-th clause belongs to A or B).

By multiplicative Chernoff bounds, for any constant $0 < \delta < 1$,

$$\Pr\left(Z_A \notin \left[(1-\delta)\frac{km}{(2^k-1)n}, (1+\delta)\frac{km}{(2^k-1)n}\right]\right) = e^{-\Omega(m/n)},$$

For n sufficiently large, $\gamma_k(n) < (1 + \delta/2)\left(1 - \left(\frac{1-2\epsilon}{2}\right)^{k-1}\right)$, and so we also have

$$\Pr\left(Z_B \notin \left[0, (1+\delta)\left(1 - \left(\frac{1-2\epsilon}{2}\right)^{k-1}\right)\frac{km}{(2^k-1)n}\right]\right) = e^{-\Omega(m/n)},$$

Thus both random variables take on values in these intervals with probability $1 - 2e^{-\Omega(m/n)}$. Under this event,

$$\left(\left(\frac{1-2\epsilon}{2}\right)^{k-1} - 2\delta\right)\frac{km}{(2^k-1)n} < Z < (1+\delta)\frac{km}{(2^k-1)n}.$$

The proof is completed by choosing δ small enough. □

2.2 Constraint density

The *constraint density* of a formula is the ratio of clauses to variables m/n. The constraint density (apart from the constant factor of k) quantifies the average number of constraints that are imposed on a variable. Boolean formulas with low constraint density are expected to be easy to satisfy, since each variable has, on average, few constraints. On the other hand, formulas with high constraint density are, on average, easy to refute because backtracking search algorithms can quickly derive a contradiction. The study of a threshold phenomenon in the uniform random k-CNF distribution $\mathcal{U}_{n,m,k}$ has been the focus of intense study in the last two decades. The *satisfiability threshold conjecture* [2] asserts that for all $k \geq 3$ there exists a real number r_k such that if is a formula drawn uniformly at random from the set of all k-CNF formulas with n variables and m clauses, then

$$\lim_{n\to\infty} \Pr\{F \text{ is satisfiable}\} = \begin{cases} 1 & m/n < r_k; \\ 0 & m/n > r_k. \end{cases}$$

Recently, Coja-Oghlan and Panagiotou [**?**] proved that $r_k = 2^k \ln 2 - \frac{1}{2}(1 + \ln 2) + o(1)$ where the error term vanishes as $k \to \infty$. Ding, Sly, and Sun [**?**] obtained an exact representation of the threshold for all $k \geq k_0$, where k_0 is a large enough constant. There are still no exact results for the location of this threshold at low values of k, but experimental studies on 3-CNF formulas suggest one exists around $r_3 \approx 4.26$. For a more detailed treatment of random satisfiability, see the chapter by Achlioptas [1].

Backtracking SAT solvers like the Davis-Putnam-Logemann-Loveland (DPLL) procedure exhibit an empirical hardness peak around the critical value [29, **?**]. This corresponds to the so-called *phase transition* phenomenon in the uniform random $\mathcal{U}_{n,m,k}$ model where formulas near the critical threshold have high decision complexity [24].

The planted $\mathcal{P}_{n,m,k}$ and filtered $\mathcal{U}^{\text{SAT}}_{n,m,k}$ models obviously have no satisfiability threshold. However, it is interesting to observe how the complexity of formulas depends on the constraint density parameter on these distributions since they are formed from the uniform distribution $\mathcal{U}_{n,m,k}$ that has been conditioned on satisfiability.

Planted 3-CNF formulas with density $m/n = \Omega(n)$ are known to be easy for simple greedy algorithms [26] (always flipping the variable assignment that gives the largest improvement). More sophisticated algorithms can even handle planted formulas with densities bounded below by a sufficiently high constant [27, 18].

For low-density planted 3-CNF formulas, a basic hillclimber that accepts only strictly improving moves fails with high probability for $\varepsilon < m/n < (7/6) \ln n$ (where ε is an arbitrary positive constant) because it is likely to become trapped in a local optimum [7]. This strict hillclimber is claimed to be successful again at extremely low densities, i.e., $m/n \approx n^{-1/4}$ [7]. A slightly more sophisticated hillclimber called GSAT is successful with high probability on planted 3-CNF formulas of density $\Omega(\log n)$ [37]. On the other hand, the random WalkSAT local search algorithm [34] (which iteratively selects an unsatisfied clause uniformly at random and flips one of its variables uniformly at random) needs an exponential number of steps to find the planted assignment [42].

Similar to its the uniform counterpart, an empirical easy-hard-easy pattern has also been also observed on the filtered $\mathcal{U}^{\mathrm{SAT}}_{n,m,3}$ model near the same critical parameter [8]. In the remainder of the paper we characterize the time complexity of the (1+1) EA on the $\mathcal{P}_{n,m,k}$ and $\mathcal{U}^{\mathrm{SAT}}_{n,m,k}$ models in different constraint density regimes.

3 Upper Bounds Based on FDC Arguments

We now study the runtime of the (1+1) EA on high-density planted formulas. We begin with linear densities in Section 3.1, namely, length-m formulas on n variables where $m/n \geq cn$ for a specific constant c. In this regime we prove a strong FDC condition and use this to show that for asymptotically almost all formulas, the (1+1) EA finds a satisfying assignment in $O(n \log n)$ time with high probability. For $k = 3$ this improves by a linear factor the previous runtime bound [39] for the (1+1) EA at these densities.

In Section 3.2, we consider formulas where $m/n \geq c \log n$ for a particular c. In this sparser regime, we can only prove a weaker type of FDC (Lemma 4). We overcome this difficulty by arguing that the typically relatively quick run of the (1+1) EA only visits a very small part of the search space, hence this weaker FDC condition suffices. Of course, this argument requires that we work in the joint probability space of the random decisions of the algorithm and the formula sampling process.

3.1 Linear density

The following strong FDC condition is the key to our analysis in the case of linear densities. It contains two crucial parts. The first is that in a sufficient part of the search space (given by the set $\mathcal{H}$) we gain a significant fitness increase when going to a Hamming neighbor closer to the planted solution. This part is very similar to the condition used in [39]. The second part shows a weaker fitness distance correlation, however, for a much larger set of pairs of search points (in particular, not only for Hamming neighbors). This part will enable us to argue that also multi-bit flip mutations cannot be harmful.

Definition 3 We say a formula F has *strong FDC* if $f = f_F$ satisfies the following two properties.

Property A. For all $(x, y) \in \mathcal{H}$, we have $c_1 m/n < f(y) - f(x) < c_2 m/n$.
Property B. For all $x, y \in \{0,1\}^n$ with $n/2 + \epsilon n \geq d(x) \geq n/2 + 3\epsilon n/4$ and $d(y) \leq n/2 + \epsilon n/2$, we have $f(x) < f(y)$.

Here $0 < c_1 < c_2$ are the constants used in Lemma 2.

Lemma 3 *Let $F \sim \mathcal{P}_{n,m,k}$, where $m/n \geq cn$ for a sufficiently large positive constant c. The probability that F has strong FDC is at least $1 - e^{-\Omega(n)}$.*

Proof By Lemma 2 together with a union bound over the elements of $\mathcal{H}$, Property A of Definition 3 holds with probability $1 - e^{-\Omega(n)}$.

To show Property B, we first observe that for any $z \in \{0,1\}^n$, the expected fitness $\mathrm{E}(f(z))$ depends only on $d(z)$ and not the particular z. More precisely, let $z, z' \in \{0,1\}^n$ such that $d(z) = d(z') =: i$. Then $\mathrm{E}(f(z)) = \mathrm{E}(f(z'))$. We may thus define $E_i := \mathrm{E}(f(z))$.

Let $x, y \in \{0,1\}^n$ with $n/2 + \epsilon n \geq d(x) \geq n/2 + 3\epsilon n/4$ and $d(y) \leq n/2 + \epsilon n/2$. Let $u, v \in \{0,1\}^n$ such that $d(u) = d(x) =: a$ and $d(v) = d(y) =: b$ and such that $u \leq v$ bit-wise, that is, such that u can be transformed into v by changing $a - b \geq \epsilon n/4$ zeros to ones.

We now argue that $E_b \geq E_a + \Theta(m)$. By a repeated application of Lemma 1, we have

$$E_b = \mathrm{E}(f(v)) \geq \mathrm{E}(f(u)) + \frac{km}{(2^k - 1)n}(1 - \gamma_k(n))(\epsilon n/4) = E_a + \Theta(m).$$

Let $q := (E_a + E_b)/2$. Note that $f(y)$ is a random variable that can be written as sum of m independent 0/1 random variables. Consequently, the additive Chernoff bound [**?**, Theorem 1.11] shows that

$$\Pr(f(y) \leq q) = \Pr(f(y) \leq E_b - (E_b - E_a)/2) \leq e^{-\Theta(m)}.$$

The same argument shows that x has a fitness of at least q with probability $e^{-\Theta(m)}$ only. Consequently, we have $f(x) < f(y)$ with probability $1 - e^{-\Theta(m)}$. Applying a union bound over the applicable pairs $x, y \in \{0,1\}^n$, we conclude that Property B of Definition 3 holds with probability at least $1 - e^{-\Omega(n)}$. A final union bound over both properties concludes the proof. □

Note that the proof above actually shows that with probability $1 - e^{-\Omega(n)}$, any $x, y \in \{0,1\}^n$ with $d(y) \leq d(x) - \epsilon/4$ and $d(x) \leq n/2 + \epsilon n$ satisfy $f(x) < f(y)$. We shall not need this stronger statement, though.

Theorem 2 *Let $m/n \geq cn$ for a sufficiently large positive constant c. For all but an $e^{-\Omega(n)}$-fraction of the k-CNF formulas with n variables and m clauses satisfied by 1^n, the runtime of the (1+1) EA is $O(n \log n)$ with probability $1 - o(1)$.*

Proof By Lemma 3, every planted formula at density at least cn has strong FDC except for an $e^{-\Omega(n)}$-fraction, so we will assume for the remainder of the proof that we are working with a formula that has the strong FDC property.

If F has strong FDC, then for states that are not too far away from the planted assignment, the fitness and distance are tightly correlated in the following sense. For all $x \in \{0,1\}^n$ with $d(x) \leq n/2 + \epsilon n$, we have

$$f(x) + c_2 d(x) m/n \geq m \text{ and } f(x) + c_1 d(x) m/n \leq m. \tag{2}$$

This follows again from regarding a shortest path from x to 1^n and applying Property A to each edge.

We consider the drift of the stochastic process $\{X_t : t \in \mathbb{N}_0\}$, where $X_t = m - f(x^{(t)})$. Assume at iteration t that $0 < d(x^{(t)}) \leq (1/2 + \epsilon)n$ (we will later show that the second inequality holds with high probability throughout the run of the algorithm). By the structure of the hypercube, the set S of $y \in \{0,1\}^n$ such that $(x^{(t)}, y) \in \mathcal{H}$, has cardinality exactly $d(x^{(t)})$.

For each $y \in S$, since $f(y) > f(x^{(t)}) + c_1 m/n > f(x^{(t)})$, a mutation from $x^{(t)}$ to y is clearly accepted by selection. Furthermore, selection does not accept mutations to lower fitness values, so $X_t - X_{t+1} \geq 0$ with probability one. Let $\mathcal{E}$ denote the event that mutation produces some $y \in S$ from $x^{(t)}$. Let $a \geq 1$. By the law of total expectation,

$$\begin{aligned} \mathrm{E}(X_t - X_{t+1} \mid X_t = a) &\geq \mathrm{E}(X_t - X_{t+1} \mid X_t = a \wedge \mathcal{E}) \Pr(\mathcal{E}) \\ &\geq \mathrm{E}(X_t - X_{t+1} \mid X_t = a \wedge \mathcal{E}) \frac{d(x^{(t)})}{en}. \end{aligned}$$

By the first inequality in (2),

$$\frac{X_t}{c_2 m/n} = \frac{m - f(x^{(t)})}{c_2 m/n} \leq d(x^{(t)}).$$

Also, if $\mathcal{E}$ holds, then we have $X_t - X_{t+1} \geq c_1 m/n$ by Property A. Consequently, we can bound the drift by

$$\mathrm{E}(X_t - X_{t+1} \mid X_t = a) \geq c_1 (m/n) \frac{a}{en(c_2 m/n)} = a \frac{c_1/c_2}{en}. \tag{3}$$

We only need to show that with high probability, the process never leaves $\mathcal{H}$. Using the multiplicative Chernoff bound, the initial search point generated uniformly at random has $d(x^{(0)}) \leq n/2 + \epsilon n/2$ with high probability. In this case, by Property B of Definition 3, the EA can never reach a search point with distance $n/2 + 3\epsilon n/4$ or worse in $\mathcal{H}$. Since $\mathcal{H}$ by definition contains points at distance at most $(1/2 + \epsilon)n$, in order for the process to leave $\mathcal{H}$, it must jump over the gap between $n/2 + 3\epsilon n/4$ and $n/2 + \epsilon n$. This can only occur after mutating at least $\epsilon n/4$ bits: an event that occurs with probability at most $e^{-\Omega(n \log n)}$ under uniform mutation.

We thus assume that the process does not leave $\mathcal{H}$, and so the inequality of (3) is valid for all times t. Finally, we apply Theorem 1 using inequality (3) by setting $\delta = c_1/(c_2 en)$ and $\lambda = \log n$ to obtain the tail bound. □

3.2 Logarithmic density

For smaller densities we can obtain a similar bound on the runtime, but we have to take a slightly different proof approach. At high densities, in the proof of Theorem 2 we could argue that (1) all but a vanishing fraction of the formulas have a certain property and (2) then argue that all formulas having this property have an $O(n \log n)$ runtime with probability $1-o(1)$. By this, we could treat the probability spaces of the random formulas and the actions of the algorithm separately. At the lower densities we regard in this subsection, we cannot make this separation, but can only prove that in the joint probability space we have a runtime of $O(n \log n)$ with probability $1 - O(n^{-1})$, which is technically more demanding. Of course, once we have proven these results, we can deduce that, e.g., only a fraction of at most $O(n^{-1/2})$ of the formulas are such that the probability of leading to a runtime higher than $O(n \log n)$ is $O(n^{-1/2})$. In all these statements we could replace the exponents of n by arbitrary constants. To increase the readability, we decided to choose specific constants leading to an overall $O(n^{-1})$ failure probability.

For any $x \in \{0,1\}^n$ with $d(x) = k$ we define $P(x) := (x = x_0, x_1, \ldots, x_k = 1^n)$ to be the unique path, where x_{i+1} is constructed from x_i by flipping the leftmost zero-bit.

Lemma 4 *Let c be a sufficiently large positive constant, $m/n > c \ln n$, $F \sim \mathcal{P}_{n,m,k}$ and $f = f_F$. If $d(x) \leq (1/2+\epsilon)n$, then with probability at least $1-n^{-3}$ we have that for every $(x_i, x_{i+1}) \in P(x)$, $c_1 m/n \leq f(x_{i+1}) - f(x_i) \leq c_2 m/n$. Here c_1, c_2 are as in Lemma 2.*

Proof For all $x, y \in \{0,1\}^n$, we define the indicator random variable $\chi_{x,y} \colon \Omega_{n,m,k} \to \{0,1\}$ by

$$\chi_{x,y}(F) = \begin{cases} 1 & \text{if } \frac{c_1 m}{n} < f(y) - f(x) < \frac{c_2 m}{n}, \\ 0 & \text{otherwise.} \end{cases}$$

A simple union bound shows

$$\begin{aligned} \Pr\left(\bigcup_{(x_i,x_{i+1}) \in P(x)} \chi_{x_i,x_{i+1}}(F) = 0 \right) &\leq \sum_{(x_i,x_{i+1}) \in P(x)} \Pr\left(\chi_{x_i,x_{i+1}}(F) = 0\right) \\ &\leq n e^{-\Omega(m/n)} \leq n^{-3}, \end{aligned}$$

where we have applied Lemma 2 and used the fact that $m > cn \ln n$, c sufficiently large. □

Lemma 4 bounds the probability that the fitness and distance are sufficiently correlated along a path to the planted solution on a random formula of density at least $c \ln n$ for a specific constant c. We now apply this result to derive a bound on the runtime of the (1+1) EA over such a formula. Note that the lemma does not immediately imply our results as it is very unlikely that the (1+1) EA follows exactly the path $P(x)$ when x is the initial search point. In fact, because of mutations flipping more than one bit, it does happen regularly that the (1+1) EA does not follow a path in the Hamming cube at all.

Theorem 3 *Let $m/n > c \ln n$ for a sufficiently large positive constant c. Let $F \sim \mathcal{P}_{n,m,k}$ and $f = f_F$. Then with probability $1 - O(n^{-1})$, the runtime of the (1+1) EA optimizing the random function f is $O(n \log n)$. Note that here the probability is taken over the joint probability space of the random sampling of the formula and the random actions of the (1+1) EA.*

Proof Let r be a sufficiently large constant. We consider the first $rn \ln n$ steps of the (1+1) EA on a formula F drawn from $\mathcal{P}_{n,m,k}$ uniformly at random and show that, with probability polynomially close to one, the (1+1) EA has found a satisfying assignment.

We say a bitstring $x \in \{0,1\}^n$ has an *FDC path* if the following two properties hold at x.

Property A′. For all $(x_i, x_{i+1}) \in P(x)$, we have $c_1 m/n \leq f(x_{i+1}) - f(x_i) \leq c_2 m/n$.
Property B′. $d(x) \leq n/2 + 3\epsilon n/4$.

We say the process has *failed* at time t if there is an $0 \leq s \leq t$ such that $x^{(s)}$ does not have an FDC path. We argue by induction that the probability that the process fails at time t conditioned on the event that it has not failed at time $t-1$ is sufficiently low. Specifically, in each step we show that if the process has not yet failed, Property B′ holds with probability $1 - e^{-\Omega(n)}$, and the probability of Property A′ conditioned on B′ is polynomially close to one.

For the initial point, we assume a slightly stronger condition than Property B′, namely that $d(x^{(0)}) \leq n/2 + \epsilon n/2$, which by Chernoff bounds holds with probability exponentially close to one. Let $h = f(x^{(0)})$. If $t > 0$ and the process has not yet failed at time $t-1$, then Property B′ can be violated only if the (1+1) EA accepts a point sufficiently far away from the planted solution. Let y be the offspring generated in iteration t of the (1+1) EA. We make a case distinction on three disjoint events occurring in the mutation step.

Case $d(y) > n/2 + \epsilon n$. Since Property B′ holds at $x^{(t-1)}$, in this case at least $\epsilon n/4$ bits must change during mutation to produce y. The number of bits that change is distributed binomially; at least j bits change with probability at most $\binom{n}{j}(1/n)^j \leq 1/j!$. Therefore, this case occurs only with probability $e^{-\Omega(n \log n)}$.
Case $n/2 + \epsilon n \geq d(y) > n/2 + 3\epsilon n/4$. By the same argument as in the proof of Lemma 3, we have $f(y) \geq h$ only with $e^{-\Omega(m)}$ probability. Since the fitness of points can only monotonically increase during a run of the (1+1) EA, with probability $1 - e^{-\Omega(m)}$ we have $f(y) < h \leq f(x^{(t-1)})$ and so $x^{(t)} = x^{(t-1)}$ since y would not be accepted.
Case $d(y) \leq n/2 + 3\epsilon n/4$. In this case, Property B′ will also not be violated by $x^{(t)}$ because, $x^{(t)} = y$ or $x^{(t)} = x^{(t-1)}$, both of which satisfy Property B′.

Since the first two events occur with exponentially small probability (and the events partition the probability space), we can conclude that, as long as the process has not failed by iteration $t-1$, then with probability $1 - e^{-\Omega(n)}$, Property B′ holds at $x^{(t)}$.

Assuming Property B$'$ holds at $x^{(t)}$, we can apply Lemma 4 and conclude the conditional probability that Property A$'$ also holds at $x^{(t)}$ is at least $1 - n^{-3}$. The joint probability that both properties hold is therefore at least $1 - O(n^{-3})$.

Now by induction on t, the process does not fail for $rn \ln n$ steps with probability at least $1 - O(n^{-1})$. Conditioning on the event that the process has not failed for a phase of $rn \ln n$ steps, the bound on the drift during this phase is the same as with inequality (3). We can again apply Theorem 1 with $\lambda = rn \ln n$ to obtain the tail bound. □

3.3 Extension to the uniform filtered distribution

In an unpublished manuscript, Ben-Sasson et al. [6] proved that for at least logarithmic densities and $k = 3$, the uniform planted distribution $\mathcal{P}_{n,m,3}$ becomes statistically close to the uniform filtered distribution $\mathcal{U}^{\mathrm{SAT}}_{n,m,3}$. In this section, we show this remains true for any constant k. This allows us to apply our runtime bounds that hold with high probability on dense planted instances to the uniform filtered k-CNF distribution. To ease the notation when working with different distributions, let us write $\Pr(\mathcal{E} \mid F \sim \mathcal{D})$ to denote the probability that the event $\mathcal{E}$ holds for a formula F chosen randomly with distribution $\mathcal{D}$.

We begin with the following technical lemma, which yields a straightforward extension of the results of [6] to higher arity clauses. Denote by $X(F)$ the random variable over $\Omega_{n,m,k}$ that counts the number of satisfying assignments to F.

Lemma 5 *If $m/n > 2(2^k - 1)\ln n$, the following three properties hold.*

1. $\Pr(X(F) = 1 \mid F \sim \mathcal{P}_{n,m,k}) = 1 - o(1)$.
2. $\Pr(X(F) = 1 \mid F \sim \mathcal{U}^{\mathrm{SAT}}_{n,m,k}) = 1 - o(1)$.
3. *For all $f \in \Omega_{n,m,k}$ such that $X(f) = 1$,*

$$\Pr(F = f \mid F \sim \mathcal{P}_{n,m,k}) = \Pr(F = f \mid F \sim \mathcal{U}^{\mathrm{SAT}}_{n,m,k}).$$

Proof To prove property 1, consider a planted formula F with planted solution $x^\star = (1, 1, \ldots, 1)$ (w.l.o.g.). Let $\mathcal{A}_\delta$ denote the event that an arbitrary assignment x' with $d(x') = \delta$ satisfies a k-clause that is selected uniformly at random from the set of all clauses satisfied by $x^\star$.

For simplicity, we assume without loss of generality that x' and $x^\star$ agree on the first $n - \delta$ positions, that is,

$$x' = (\underbrace{1, 1, \ldots, 1}_{n-\delta}, \underbrace{0, 0, \ldots, 0}_{\delta}).$$

Let C be drawn uniformly at random from all clauses satisfying $x^\star$. There are two events that can occur for C to be satisfied by x'. The first event is that all k variables of C are chosen from the first $n - \delta$ variables. In this case, C

must be satsified by x' since it is also satisfied by $x^\star$. The second event is that not all k variables of C are chosen from the first $n-\delta$, but the signs of the literals are set correctly so that C is still satisfied by x'. The first event happens with probability $\binom{n-\delta}{k}/\binom{n}{k}$. In the second event, the probability that not all variables of C are chosen from the first $n-\delta$ is $1-\binom{n-\delta}{k}/\binom{n}{k}$. Given the chosen variables, the probability that C is *not* satisfied by x' is $1/(2^k-1)$ since there are exactly 2^k-1 possible ways to set the literals to ensure C is still satisfied by $x^\star$, and each clause is made false by exactly one setting of its k variables. Thus,

$$\begin{aligned}\Pr(\mathcal{A}_\delta) &= \frac{\binom{n-\delta}{k}}{\binom{n}{k}} + \frac{2^k-2}{2^k-1}\left(1-\frac{\binom{n-\delta}{k}}{\binom{n}{k}}\right)\\ &\le \frac{2^k-2}{2^k-1} + \frac{1}{2^k-1}\left(\frac{n-\delta}{n}\right) = 1-\frac{\delta}{(2^k-1)n}.\end{aligned}$$

By the union bound, the probability that there are more satisfying assignments than $x^\star$ is at most

$$\begin{aligned}\Pr\left(X(F)>1 \mid F\sim\mathcal{P}_{n,m,k}\right) &\le \sum_{\delta=1}^{n}\binom{n}{\delta}\Pr(\mathcal{A}_\delta)^m\\ &\le \sum_{\delta=1}^{n}\binom{n}{\delta}\left(1-\frac{\delta}{(2^k-1)n}\right)^{2(2^k-1)n\ln n}\\ &\le \sum_{\delta=1}^{n} n^\delta e^{-2\delta\ln n} = \sum_{\delta=1}^{n} e^{-\delta\ln n} = o(1).\end{aligned}$$

The proof of property 2 is identical to the proof of Lemma 3.3 in [6], but we derive it again here in our own notation for completeness. We first define the following sets. For $i\in[2^n]$, let $S^{(i)}_{n,m,k}=\{F : F\in\Omega_{n,m,k} \text{ and } X(F)=i\}$ and $S_{n,m,k}=\bigcup_{i=1}^{2^n} S^{(i)}_{n,m,k}$. When sampling a formula from $\mathcal{P}_{n,m,k}$, the planted assignment $x^\star$ is sampled first uniformly at random from all 2^n assignments. Let $s:=|\{F\in S_{n,m,k} \colon F \text{ is satisfied by } x^\star\}|$. By a simple symmetry argument, for an arbitrary $y\in\{0,1\}^n$, $|\{F\in S_{n,m,k}\colon F \text{ is satisfied by } y\}|=s$.

Suppose there exists a formula $F\in\Omega_{n,m,k}$ with $X(F)=i$. This F is generated by the planted model if (1) one of the i assignments that satisfies F is selected as the planted assignment $x^\star$, and (2) F is then selected from the set of formulas satisfied by $x^\star$. By definition, the planted model selects a formula uniformly from the formulas that satisfy the planted assignment so (1) occurs with probability $i/2^n$ and (2) occurs with probability $1/s$. We thus have

$$\Pr\left(X(F)=i \mid F\sim\mathcal{P}_{n,m,k}\right) = \frac{i|S^{(i)}_{n,m,k}|}{2^n s} \le \frac{i|S^{(i)}_{n,m,k}|}{|S_{n,m,k}|}, \qquad (4)$$

because $|S_{n,m,k}| \leq 2^n s$. Since the uniform filtered distribution chooses uniformly from every satisfiable formula,

$$\begin{aligned}\Pr(X(F) = 1 \mid F \sim \mathcal{U}^{\mathrm{SAT}}_{n,m,k}) &= \frac{|S^{(1)}_{n,m,k}|}{|S_{n,m,k}|} \\ &\geq \Pr(X(F) = 1 \mid F \sim \mathcal{P}_{n,m,k}) = 1 - o(1)\end{aligned}$$

by Equation (4) and Property 1.

Finally, observing that both distributions give equal weight among all satisfying assignments with unique solutions yields property 3. □

Theorem 4 *Let $\mathcal{E}$ be an event defined for a formula F. If $m/n > c \log n$ for a particular constant $c > 0$, then*

$$\Pr\left(\mathcal{E} \mid F \sim \mathcal{U}^{\mathrm{SAT}}_{n,m,k}\right) = \Pr\left(\mathcal{E} \mid F \sim \mathcal{P}_{n,m,k}\right) \pm o(1).$$

Proof For notational simplicity define the event $\mathcal{A} := \{X(F) = 1\}$ and the distributions $P(\cdot) := \Pr(\cdot \mid F \sim \mathcal{P}_{n,m,k})$ and $Q(\cdot) := \Pr(\cdot \mid F \sim \mathcal{U}^{\mathrm{SAT}}_{n,m,k})$. By the law of total probability,

$$\begin{aligned}&Q(\mathcal{E}) = Q(\mathcal{E} \mid \mathcal{A})Q(\mathcal{A}) + Q(\mathcal{E} \mid \overline{\mathcal{A}})Q(\overline{\mathcal{A}}), \quad \text{and} \\ &P(\mathcal{E}) = P(\mathcal{E} \mid \mathcal{A})P(\mathcal{A}) + P(\mathcal{E} \mid \overline{\mathcal{A}})P(\overline{\mathcal{A}}).\end{aligned}$$

By property 3 of Lemma 5, $P(F|A) = Q(F|A)$ so

$$\begin{aligned}Q(\mathcal{E}) &= \left(P(\mathcal{E}) - P(\mathcal{E} \mid \overline{\mathcal{A}})P(\overline{\mathcal{A}})\right)\frac{Q(\mathcal{A})}{P(\mathcal{A})} + Q(\mathcal{E} \mid \overline{\mathcal{A}})Q(\overline{\mathcal{A}}) \\ &= \left(P(\mathcal{E}) - o(1)\right)\frac{1 - o(1)}{1 - o(1)} + o(1),\end{aligned}$$

by properties 1 and 2, which completes the proof. □

We conclude the typical runtime of the (1+1) EA very rarely deviates above $O(n \log n)$ for all satisfiable formulas of sufficiently high density, except for a set of measure vanishing with n.

Corollary 1 *Let $m/n > c \ln n$ for a sufficiently large positive constant c. The runtime of the (1+1) EA is bounded by $O(n \log n)$ with probability $1 - o(1)$ on asymptotically almost all satisfiable k-CNF formulas on n variables and m clauses.*

We may also use these results to derive a simple lower bound on the runtime of the (1+1) EA.

Theorem 5 *On asymptotically almost all satisfiable k-CNF formulas of density $m/n > c \ln n$, c a sufficiently large constant, the runtime of the (1+1) EA is $\Omega(n \log n)$ with high probability.*

Proof By Lemma 5, setting $c := 2(2^k - 1)$, for densities above $c \ln n$, asymptotically almost all formulas have a unique satisfying assignment. For the remainder of the proof, assume that F is satisfied by a unique $x^\star \in \{0,1\}^n$, that is, the corresponding fitness function f_F has a unique optimum $x^\star$. Let $t = \gamma(n-1)\ln(n)$ with $\gamma < 1$ a constant. The probability that the initial individual and the t individuals generated in the first t iterations all have the ith bit different from $x_i^\star$, is $(1/2)(1-1/n)^t \geq (1/2)e^{-\gamma \ln(n)} = n^{-\gamma}/2$. Note that this event implies that the (1+1) EA has not found the optimum within the first t iterations. Since the (1+1) EA, apart from the selection step, treats the bits independently, the probability that none of these n bad events happens, is at most $(1 - n^{-\gamma}/2)^n \leq \exp(-n^{1-\gamma}/2) = o(1)$.

The proof above mostly builds on the well-known fact that the $(1+1)$ EA needs $\Omega(n \log n)$ iterations to optimize a function with unique global optimum. However, we have not found such a simple proof for a high-probability statement in the literature. For the expectation, this was shown already in Droste et al. [16] (for linear functions with positive weights, but it is clear that the proof extends to arbitrary functions with unique optimum).

4 Low-density regime

On the uniform k-CNF distribution, formulas that appear to be difficult for complete search algorithms lie near the critical threshold r_k. For example 3-CNF formulas are empirically harder for DPLL near $r_3 \approx 4.26$ [29]. However, at very low densities, random formulas become easy to solve again on average, even by very simple linear-time backtracking-free heuristics such as the *unit clause* heuristic, which succeeds with asymptotically positive probability at all densities $m/n < (1-\varepsilon_k)(e/2)2^k/k$ where ε_k tends to zero for large k [28]. A slight generalization of the unit clause heuristic (called the *small clause* heuristic) succeeds with high probability for densities up to $m/n < (1-\varepsilon_k)(e^2/8)2^k/k$ and can even be improved to $(1.817 - \varepsilon_k)2^k/k$ if limited backtracking is allowed [19]. The *pure literal* heuristic again succeeds with high probability if m/n is below some constant c_k, however, with $c_k = O(\log k)$ only [30,31]. For $k = 3$, for example, we have $c_k \approx 1.637$.

In the context of randomized search heuristics, Alekhnovich and Ben-Sasson [3] discovered a deep connection between a randomized local search algorithm (known as RWalkSAT that iteratively flips a single randomly chosen variable in a random unsatisfied clause) and the pure literal heuristic. They proved that such a constraint-directed random walk finds a satisfying assignment on uniform random 3-CNF formulas with high probability in linear time for constraint densities at most 1.637. More recently, Coja-Oghlan and Frieze proved that this approach works for all $k > k_0$ (where $k_0 > 3$ is a constant) when the density is below $(1/25)2^k/k$ [10]. We should point out, though, that this randomized search heuristic is quite different from typical evolutionary algorithms. In particular, RWalkSAT is completely ignorant of the fitness (except that is stops when a satisfying assignment is found).

In the interest of a more complete picture, we would also like to understand the behavior of the (1+1) EA at very sparse densities. While it may be easy to believe that such instances are easy for the (1+1) EA, we can support this potential belief only weakly via our experimental results in Section 5. The fact that all known efficient heuristics of such instances ignore the fitness may even shed some doubt. In this section, we show that if the density is $O(k^{-2})$, then the structure of the constraints is so sparse that the formula breaks up into logarithmic-size components that the (1+1) EA can solve separately. However, even from this we can only derive a subexponential, but not a polynomial time bound.

Lemma 6 *Let $H = H_k(n,m)$ denote a random k-uniform hypergraph with n vertices and exactly m hyperedges selected uniformly at random with replacement from the family of all $\binom{n}{k}$ possible k-sets.*

Let $\alpha = km/n$ denote the average degree *of H. If $\alpha < (k-1)^{-1}$, then with high probability, the number of vertices in the largest connected component of H is $O(\log n)$.*

Proof Let H be the random hypergraph obtained from m times selecting a k-set chosen independently and uniformly at random (that is, with replacement). Let H_0 be the hypergraph obtained from deleting multiple hyperedges. Note that H and H_0 have the same connected components. Let H_1 be the hypergraph obtained from H_0 by repeatedly adding random hyperedges not yet present until the number of hyperedges is exactly m. By construction, every connected component of H_0 (and thus of H) is contained in a connected component of H_1. In particular, the largest component of H is not larger than the largest component of H_1. Finally, due to a result of Schmidt-Pruzan and Shamir [36], the largest component of H_1 has size $O(\log n)$ with high probability. □

The *constraint hypergraph* of a formula is a hypergraph $H = (X, E)$ where X corresponds to the set of variables in F and E is a sequence of m nonempty subsets of X constructed as follows. Each clause C of F corresponds to a unique $S \in E$ that contains exactly the variables that appear as literals in C. Thus, every k-CNF formula on n variables with m clauses has a unique k-uniform constraint hypergraph with m hyperedges (parallel hyperedges are allowed). It is easy to see that at very low constant densities, the constraint structure of Boolean formulas breaks up into small components that the (1+1) EA can solve separately. This is captured by the following theorem.

Theorem 6 *Let F be a k-CNF formula drawn from $\mathcal{U}_{n,m,k}$ with density $m/n < \frac{1}{k(k-1)}$. Then with high probability, the (1+1) EA finds a satisfying assignment for F in subexponential time.*

Proof We consider the average degree α of the constraint hypergraph H of F. Since F is sampled uniformly at random from $\Omega_{n,m,k}$, its constraint hypergraph is a random k-uniform hypergraph with n vertices and m edges sampled uniformly at random with replacement since each of the 2^k distinct clauses associated with each unique k-set is also selected uniformly at random. Since $\alpha = km/n < 1/(k-1)$, by Lemma 6, with high probability the largest connected component in H contains $O(\log n)$ vertices.

In this case, let q be the number of connected components in H. We partition the clause set into $S_1, S_2, \ldots, S_q$ such that S_i is the set of clauses that contain only variables from the i-th connected component of H (in some arbitrary order). The fitness function f can be expressed as $f(x) = \sum_{i=1}^{q} f_i(x)$ where $f_i(x)$ counts the number of clauses in S_i that are satisfied by x. Since each f_i depends on at most $O(\log n)$ bits of x, f is decomposable into linearly separable subfunctions of bounded size.

The proof is then completed by a simple fitness level argument [41]. In particular, let $(A_0, \ldots, A_m)$ be a partition of $\{0,1\}^n$ such that for all $x \in A_j$, $f(x) = j$. Let t be an arbitrary iteration in the execution of the (1+1) EA and set $\ell := f(x^{(t)})$. As long as there is an unsolved subfunction f_i with respect to the assignment corresponding to $x^{(t)}$, the (1+1) EA can generate a strictly improving offspring by solving f_i and flipping no other bit outside of S_i. The resulting offspring must lie in some $A_{\ell'}$ with $\ell' > \ell$. The probability of this event is at least $(1-1/n)^{n-|S_i|}(1/n)^{|S_i|} \geq n^{-|S_i|}/e$, and the waiting time to increase the fitness level by at least one is bounded by $en^{|S_i|}$. Since there are at most $m = O(n)$ suboptimal fitness levels, the expected time until F is solved is bounded by $n^{O(\log n)} = 2^{o(n)}$. □

Note that if we only aim at a statement on the expectation, then most of the last paragraph of the proof could have been simply replaced by applying a general result on optimization times of separable functions (Theorem 12 in [15]).

5 Experiments

In this section we report numerical experiments that investigate the constants in the asymptotic bounds proved in this paper, and explore the runtime character of the (1+1) EA at lower densities. In Figure 1 we investigate the runtime divided by $n \ln n$ as a function of $n = 300, 310, \ldots, 1000$ for the $\mathcal{P}_{n,m,k}$ model with $k \in \{3,4,5\}$ and $m/n = n$. For each value of n and k we generate 100 random k-CNF formulas, and for each such random formula we conduct 100 runs of the (1+1) EA, measuring the first iteration in which it finds a satisfying assignment. Thus, for each value of n and k we have a total of 10000 measurements. We then calculate the quartiles of the number of iterations to solve each formula as a robust statistic for the runtime as a function of n. For each value of $k \in \{3,4,5\}$, the plotted value appears to converge to some constant $c \leq e$. Note that c might depend on k, but we cannot draw such a conclusion from these experiments, as there exists a significant overlap in the interquartile ranges. Most importantly, the plot provides empirical evidence that the runtime bound proved in this paper is tight, and suggests that

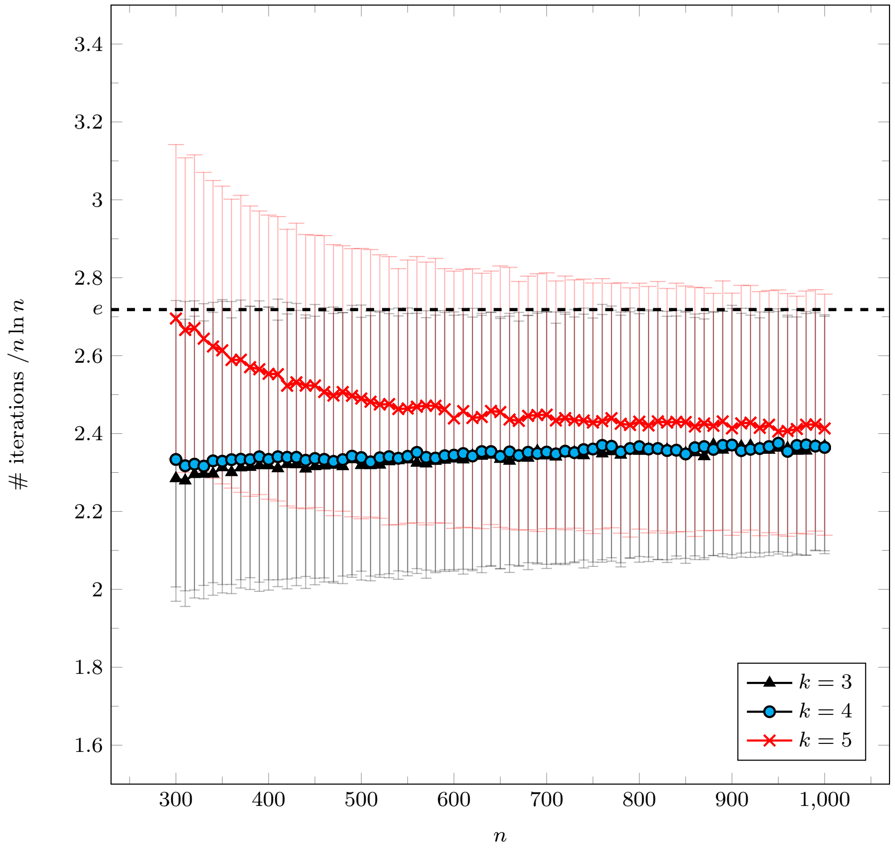

Fig. 1: Median runtime of the (1+1) EA divided by $n \ln n$ as a function of n on formulas sampled from the $\mathcal{P}_{n,m,k}$ model where $k \in \{3, 4, 5\}$, $m = n^2$ (constraint density is $\Theta(n)$). The error bars denote the interquartile range. The statistics are taken from 100 runs each on 100 random formulas generated for each value of n.

the true runtime on the linear-density $\mathcal{P}_{n,m,k}$ model is concentrated around $cn \ln n \pm O(n)$.

We repeat this experiment for asymptotically lower densities and plot the results in Figure 2. In this case, we set $m/n = 2(2^k - 1) \ln n$ for each random formula corresponding to the statement of Theorem 3. The value of the leading constant is taken from the proof of Theorem 5. The behavior that can be observed in Figure 2 is very similar to the linear density case. Specifically, for some a constant $c \leq e$ (again, that may depend on k), the true runtime on log-density instances from $\mathcal{P}_{n,m,k}$ appears to be concentrated around $cn \ln n \pm O(n)$.

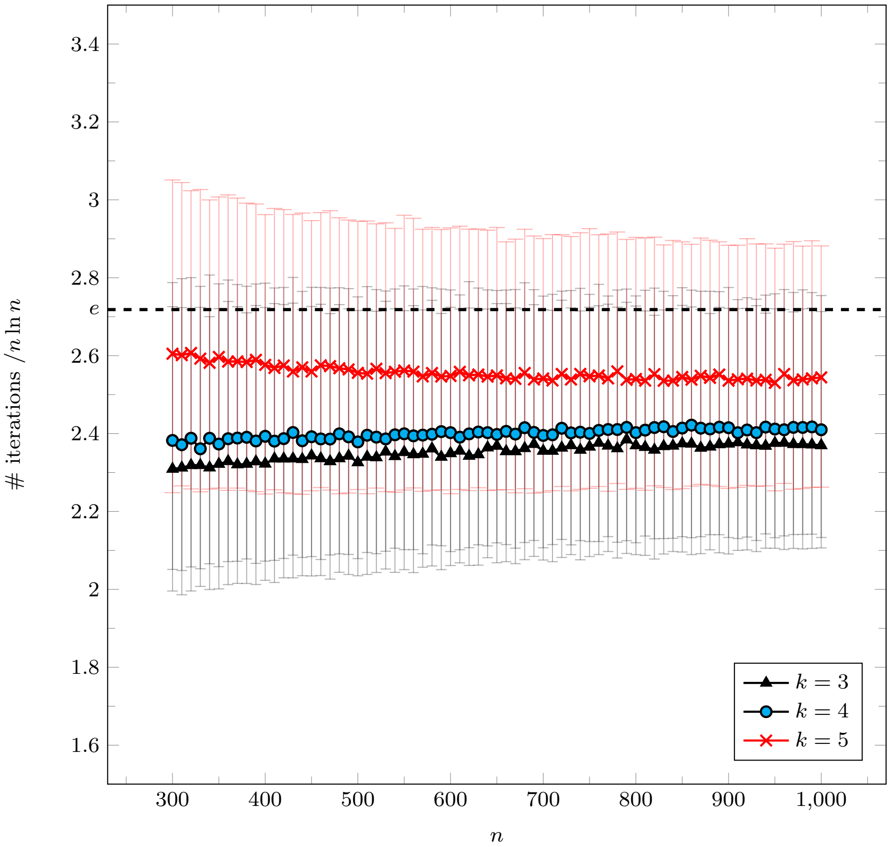

Fig. 2: Median runtime of the (1+1) EA divided by $n \ln n$ as a function of n on formulas sampled from the $\mathcal{P}_{n,m,k}$ model where $k \in \{3, 4, 5\}$, $m = 2(2^k - 1)n \ln n$ (constraint density is $\Theta(\log n)$). The error bars denote the interquartile range. The statistics are taken from 100 runs each on 100 random formulas generated for each value of n.

5.1 Phase transition behavior

In order to gain a more precise understanding of the behavior of the (1+1) EA on random planted k-CNF formulas across the density spectrum, we report numerical experiments that measure the time until a satisfying assignment is found at different densities for some distinct values of n.

We focus on the case $k = 3$. On the $\mathcal{P}_{n,m,3}$ model, for three distinct values of n, i.e., $n \in \{100, 300, 1000\}$, we generate formulas using 100 equidistant values of m such that the constraint density ranges from 1 to 10. For each distinct density value, we generate 100 formulas from the random $\mathcal{P}_{n,m,3}$ model and run the (1+1) EA 100 times on each formula. Runs that do not complete in at most 10^7 iterations are halted and removed from consideration. Of the runs

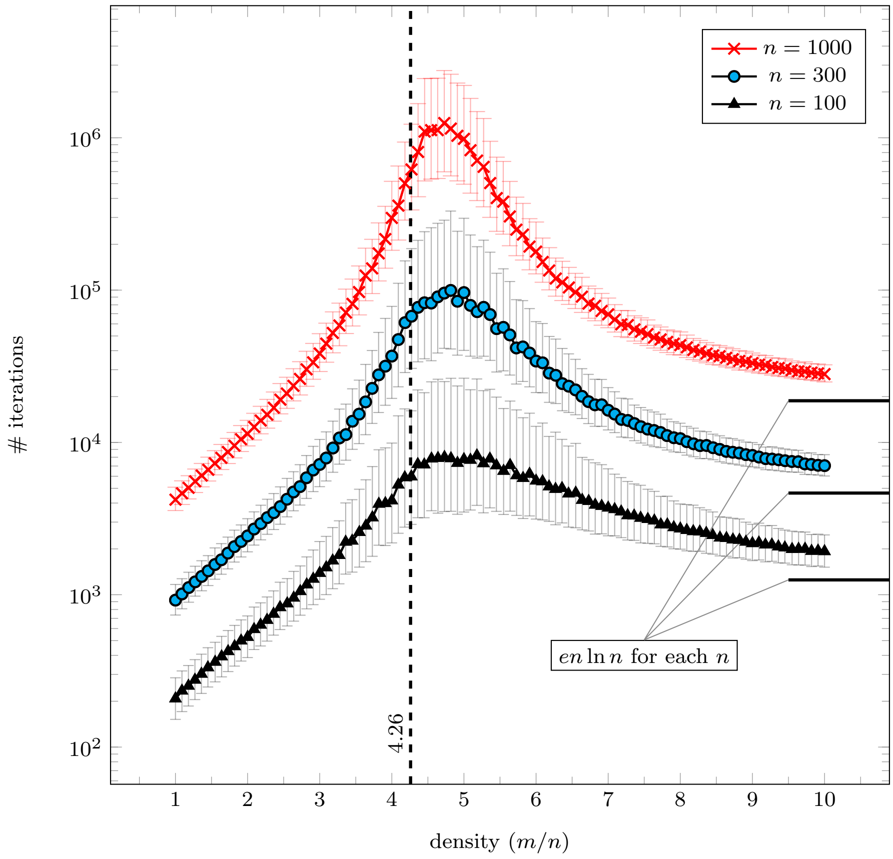

Fig. 3: Runtime statistics for the (1+1) EA on the $\mathcal{P}_{n,m,3}$ model controlling m for constraint density. The statistics are taken from 100 runs each on 100 random formulas generated for each value of m/n. The marked lines denote the median runtime. The error bars denote the interquartile range.

that do not fail, the median runtime as a function of constraint density for these trials is plotted in Figure 3. We also plot the percentage of runs that failed as a function of constraint density in Figure 4.

In these results, we also observe the classical easy-hard-easy pattern similar to the one that occurs for complete DPLL solvers on the uniform random model [29, **?**]. Remarkably, our experiments suggest that there is also a critical density in the *planted* model $\mathcal{P}_{n,m,3}$ for the (1+1) EA at which formulas are on average more difficult to optimize. We also observe that the hardness peak for the (1+1) EA occurs close to density values of $m/n \approx 4.26$, which is the critical density for DPLL solvers on the uniform model $\mathcal{U}_{n,m,3}$. This corresponds to the conjectured satisfiability threshold r_3 for random unfiltered, unplanted formulas.

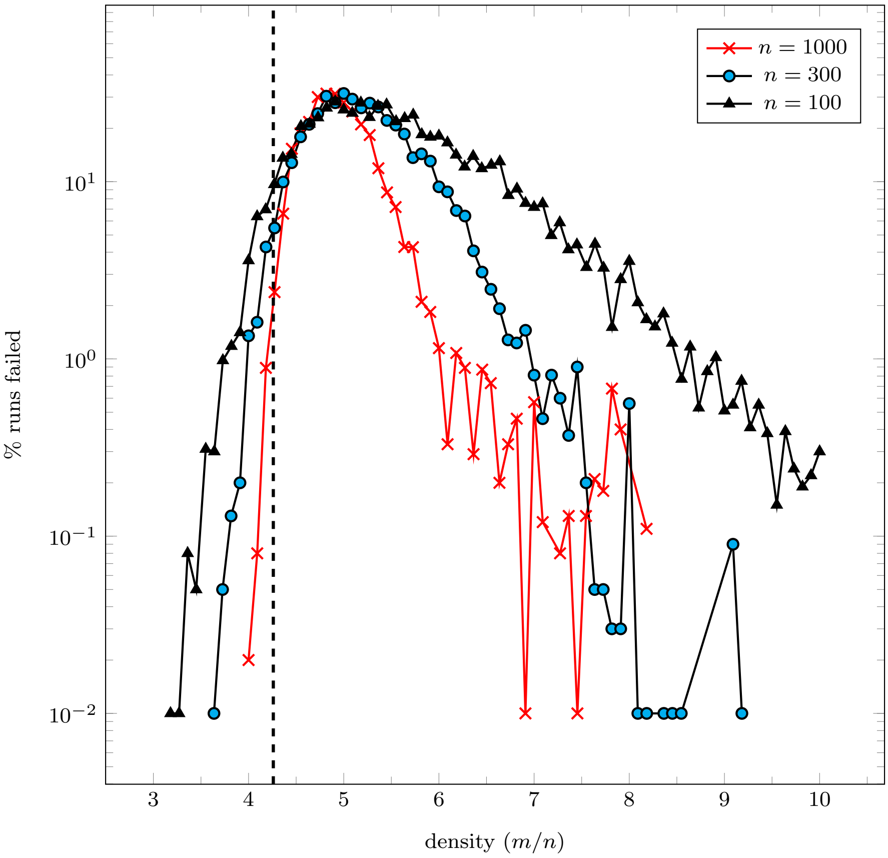

Fig. 4: Percentage of runs (out of 10000) for the (1+1) EA on the $\mathcal{P}_{n,m,3}$ model requiring $\geq 10^7$ iterations at each density value.

Below the hardness peak, the (1+1) EA appears to find a satisfying assignment quickly. Theorem 6 guarantees only subexponential time (and only for densities below 1/6 on $\mathcal{U}_{n,m,3}$) and it remains an open theoretical question whether or not polynomial time is possible again for low densities. As density increases beyond the critical point, the empirical running time in Figure 3 appears to converge again toward $en \ln n$ for each n value. Theorem 3 establishes an asymptotic bound on the density at which most formulas become easy again. An interesting open problem is the location of the critical density below which formulas become difficult on average for the (1+1) EA.

6 Conclusions

We have presented a time complexity analysis of the (1+1) EA for randomly constructed k-CNF formulas. Investigating the fitness distance correlation for high density formulas, we have shown an improved bound of $O(n \log n)$ on the (1+1) EA. In extension to the investigations in [39], the $O(n \log n)$ bound holds for formulas of logarithmic density with probability $1 - o(1)$, and for k-CNF formulas where the only restriction on k is that it is constant. Our complementary experimental investigations imply the leading constants in our asymptotic bounds are low, and extend the investigations to other density ratios.

Acknowledgements

The research leading to these results has received funding from the Australian Research Council (ARC) under grant agreement DP140103400 and from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 618091 (SAGE).

References

1. Dimitris Achlioptas. Random satisfiability. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 245–270. IOS Press, 2009.
2. Dimitris Achlioptas, Amin Coja-Oghlan, and Federico Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. *Random Structures and Algorithms*, 38(3):251–268, 2011.
3. Mikhail Alekhnovich and Eli Ben-Sasson. Linear upper bounds for random walk on small density random 3-CNFs. *SIAM Journal on Computing*, 36(5):1248–1263, 2007.
4. Lee Altenberg. Fitness distance correlation analysis: An instructive counterexample. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 57–64. Morgan Kaufmann, 1997.
5. Anne Auger and Benjamin Doerr, editors. *Theory of Randomized Search Heuristics: Foundations and Recent Developments.* World Scientific Publishing Co., Inc., 2011.
6. Eli Ben-Sasson, Yonatan Bilu, and Danny Gutfreund. Finding a randomly planted assignment in a random 3-CNF. Manuscript, 2002.
7. Andrei A. Bulatov and Evgeny S. Skvortsov. Phase transition for local search on planted SAT. In Giuseppe F. Italiano, Giovanni Pighizzini, and Donald T. Sannella, editors, *Mathematical Foundations of Computer Science 2015*, volume 9235 of *Lecture Notes in Computer Science*, pages 175–186. Springer Berlin Heidelberg, 2015.
8. David A. Clark, Jeremy Frank, Ian P. Gent, Ewan MacIntyre, Neven Tomov, and Toby Walsh. Local search and the number of solutions. In *Principles and Practice of Constraint Programming (CP 1996)*, pages 119–133. Springer, 1996.
9. Amin Coja-Oghlan and Alan Frieze. Analyzing walksat on random formulas. *SIAM Journal on Computing*, 43(4):1456–1485, 2014.
10. Amin Coja-Oghlan and Konstantinos Panagiotou. The asymptotic k-SAT threshold. *Advances in Mathematics*, 288:985–1068, 2016.
11. James M. Crawford and Larry D. Auton. Experimental results on the crossover point in random 3-SAT. *Artificial Intelligence*, 81(12):31 – 57, 1996.
12. Jian Ding, Allan Sly, and Nike Sun. Proof of the satisfiability conjecture for large k. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC 2015, pages 59–68, New York, NY, USA, 2015. ACM.
13. Benjamin Doerr. Analyzing randomized search heuristics: Tools from probability theory. In Auger and Doerr [5], pages 1–20.
14. Benjamin Doerr and Leslie Ann Goldberg. Adaptive drift analysis. *Algorithmica*, 65(1):224–250, 2013.

15. Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. *Algorithmica*, 64(4):673–697, 2012.
16. Benjamin Doerr, Frank Neumann, and Andrew M. Sutton. Improved runtime bounds for the (1+1) EA on random 3-CNF formulas based on fitness-distance correlation. In Juan Luis Jiménez Laredo, Sara Silva, and Anna Isabel Esparcia-Alcázar, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 1415–1422. ACM, 2015.
17. Benjamin Doerr, Dirk Sudholt, and Carsten Witt. When do evolutionary algorithms optimize separable functions in parallel? In *Proceedings of the Twelfth ACM SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA 2013)*, pages 48–59. ACM, 2013.
18. Stefan Droste, Thomas Jansen, and Ingo Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276(1-2):51–81, 2002.
19. Matthias Englert, Heiko Röglin, and Berthold Vöcking. Worst case and probabilistic analysis of the 2-opt algorithm for the TSP. *Algorithmica*, 68(1):190–264, 2014.
20. Abraham D. Flaxman. A spectral technique for random satisfiable 3CNF formulas. *Random Structures and Algorithms*, 32(4):519–534, 2008.
21. Alan Frieze and Stephen Suen. Analysis of two simple heuristics on a random instance of k-SAT. *Journal of Algorithms*, 20(2):312–355, 1996.
22. Thomas Jansen. On classifications of fitness functions. In Leila Kallel, Bart Naudts, and Alex Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, Natural Computing Series, pages 371–385. Springer Berlin Heidelberg, 2001.
23. Thomas Jansen. *Analyzing Evolutionary Algorithms - The Computer Science Perspective.* Natural Computing Series. Springer, 2013.
24. Thomas Jansen and Christine Zarges. Performance analysis of randomised search heuristics operating with a fixed budget. *Theoretical Computer Science*, 545:39 – 58, 2014.
25. Terry Jones and Stephanie Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann, 1995.
26. Scott Kirkpatrick and Bart Selman. Critical behavior in the satisfiability of random Boolean expressions. *Science*, 264(5163):1297–1301, 1994.
27. Timo Kötzing, Frank Neumann, Heiko Röglin, and Carsten Witt. Theoretical analysis of two ACO approaches for the traveling salesman problem. *Swarm Intelligence*, 6(1):1–21, 2012.
28. Elias Koutsoupias and Christos H. Papadimitriou. On the greedy algorithm for satisfiability. *Information Processing Letters*, 43(1):53–55, 1992.
29. Michael Krivelevich and Dan Vilenchik. Solving random satisfiable 3CNF formulas in expected polynomial time. In *Proceedings of the Seventeenth Symposium on Discrete Algorithms (SODA 2006)*, pages 454–463, 2006.
30. Chao Ming-Te and John Franco. Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k satisfiability problem. *Information Sciences*, 51(3):289–314, 1990.
31. David Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of SAT problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI 1992)*, pages 459–465, 1992.
32. Michael Mitzenmacher. Tight thresholds for the pure literal rule. Technical Report 1997-011, Digital SRC, 1997.
33. Michael Molloy. Cores in random hypergraphs and Boolean formulas. *Random Structures and Algorithms*, 27(1):124–135, 2005.
34. Samadhi Nallaperuma, Frank Neumann, and Dirk Sudholt. A fixed budget analysis of randomized search heuristics for the traveling salesperson problem. In Dirk V. Arnold, editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*, pages 807–814. ACM, 2014.
35. Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity.* Springer, 2010.
36. Christos H. Papadimitriou. On selecting a satisfying truth assignment. In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pages 163–169, Oct 1991.

37. R. J. Quick, Victor J. Rayward-Smith, and George D. Smith. Fitness distance correlation and ridge functions. In A. E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel, editors, *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature (PPSN V)*, volume 1498 of *Lecture Notes in Computer Science*, pages 77–86. Springer, 1998.
38. Jeanette Schmidt-Pruzan and Eli Shamir. Component structure in the evolution of random hypergraphs. *Combinatorica*, 5(1):81–94, 1985.
39. Evgeny S. Skvortsov. A theoretical analysis of search in GSAT. In Oliver Kullmann, editor, *Theory and Applications of Satisfiability Testing (SAT 2009)*, volume 5584 of *Lecture Notes in Computer Science*, pages 265–275. Springer Berlin Heidelberg, 2009.
40. Tobias Storch. Finding large cliques in sparse semi-random graphs by simple randomized search heuristics. *Theoretical Computer Science*, 386:114–131, 2007.
41. Andrew M. Sutton and Frank Neumann. Runtime analysis of evolutionary algorithms on randomly constructed high-density satisfiable 3-CNF formulas. In Thomas Bartz-Beielstein, Jürgen Branke, Bogdan Filipic, and Jim Smith, editors, *Proceedings of the Thirteenth International Conference on Parallel Problem Solving from Nature (PPSN XIII)*, volume 8672 of *Lecture Notes in Computer Science*, pages 942–951. Springer, 2014.
42. Carsten Witt. Worst-case and average-case approximations by simple randomized search heuristics. In Volker Diekert and Bruno Durand, editors, *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005, Proceedings*, volume 3404 of *Lecture Notes in Computer Science*, pages 44–56. Springer, 2005.
43. Carsten Witt. Fitness levels with tail bounds for the analysis of randomized search heuristics. *Information Processing Letters*, 114(1–2):38 – 41, 2014.
44. YuRen Zhou. Exponential bounds for the random walk algorithm on random planted 3-sat. *Science China Information Sciences*, 56(9):1–13, 2013.