Runtime Analysis of Evolutionary Algorithms on Randomly Constructed High-Density Satisfiable 3-CNF Formulas

Andrew M. Sutton[1] and Frank Neumann[2]

[1] Friedrich-Schiller-Universität Jena, 07743 Jena, Germany
[2] Optimisation and Logistics, School of Computer Science, The University of Adelaide, Adelaide, SA, 5005, Australia

Abstract. We show that simple mutation-only evolutionary algorithms find a satisfying assignment on two similar models of random planted 3-CNF Boolean formulas in polynomial time with high probability in the high constraint density regime. We extend the analysis to random formulas conditioned on satisfiability (i.e., the so-called filtered distribution) and conclude that most high-density satisfiable formulas are easy for simple evolutionary algorithms. With this paper, we contribute the first rigorous study of randomized search heuristics from the evolutionary computation community on well-studied distributions of random satisfiability problems.

1 Introduction

Boolean satisfiability is an archetypical NP-complete problem with extensive theoretical and practical relevance. Randomized search heuristics such as evolutionary algorithms [6] and randomized local search techniques [10] are often successfully applied to quickly identify satisfiable Boolean formula. Modern high-performance heuristics can handle problems with millions of variables [13], but the relationship between problem structure and computational cost is still poorly understood from a rigorous perspective.

In the field of Boolean satisfiability, a significant amount of research has been carried out on the runtime of algorithms over randomly generated formulas. Theoretical and empirical work on uniform random satisfiability suggests that, despite the hardness of the problem of determining whether or not a Boolean formula has a satisfying assignment, a vast fraction of formulas are easy to solve on average. Understanding the behavior of evolutionary algorithms with respect to their runtime for this central problem pushes forward the theoretical understanding of these algorithms on an NP-hard problem in the context of randomly generated instances.

Extensive progress has already been made in runtime analysis of evolutionary algorithms from a worst-case perspective [1,12]. However, still very little is known about typical behavior on randomly generated instances of NP-hard problems. The only study that we are aware of is the one of Witt [16] for makespan scheduling. In this paper we study the behavior of simple evolutionary algorithms over

T. Bartz-Beielstein et al. (Eds.): PPSN XIII 2014, LNCS 8672, pp. 942–951, 2014.

uniform distributions of satisfiable 3-CNF formulas. We prove that the runtime of the (1+1) EA is $O(n^2 \log n)$ with high probability on almost all satisfiable 3-CNF formulas (except for a fraction that vanishes exponentially fast), as long as their constraint density is $\Omega(n)$. Though this distribution is previously known to be easy for classical algorithms [9], to our knowledge this constitutes the first rigorous analysis of evolutionary algorithms on random satisfiability models.

1.1 3-CNF Distributions

A k-CNF formula F over a set of n Boolean variables $\{x_1, x_2, \ldots, x_n\}$ is a conjunction of exactly m clauses $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$, where each clause is the 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 x or its negation $\bar{x}$. A k-CNF formula 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 *constraint density* of a formula is the ratio of clauses to variables m/n. The constraint density quantifies the average number of constraints (disjunctive clauses) in which a variable occurs.

The set of all assignments to a set of n Boolean variables is isomorphic to $\{0,1\}^n$ by interpreting each position of the string as the state of exactly one Boolean variable x_i (i.e., a 1 corresponds to $x_i = true$; a 0 corresponds to $x_i = false$). Given a 3-CNF formula F with n variables, we represent candidate solutions as length-n bitstrings and define the function $f\colon \{0,1\}^n \to \mathbb{N}$ where $f(x)$ counts the clauses of F that are satisfied under the assignment corresponding to $x \in \{0,1\}^n$. If F is satisfiable, the task of finding a satisfying assignment is reduced to the task of optimizing a pseudo-Boolean function.

Uniform distributions of random Boolean formulas are similar to the Erdős-Rényi model of random graphs. In the $\mathcal{U}_{n,m}$ model, exactly m random clauses are selected independently and uniformly with replacement[1] from all possible 3-CNF clauses over n variables. In the $\mathcal{U}_{n,p}$ model, each 3-CNF clause over n variables is chosen for inclusion independently with probability p.

Stochastic search algorithms such as evolutionary algorithms and randomized local search are generally incapable of proving a formula unsatisfiable, but are often applied as incomplete heuristics and can be treated as Monte Carlo algorithms when their runtime is fixed. Because of this, one is often interested in their performance on *satisfiable* formulas.

One way to generate random satisfiable formulas is to condition the distribution $\mathcal{U}_{n,m}$ on satisfiability. This results in the *filtered uniform* model $\mathcal{U}^{\mathrm{SAT}}_{n,m}$. The filtered uniform model is difficult to analyze, and potentially hard to sample from (since it requires solving an NP-hard problem to check whether a formula is satisfiable). To circumvent this, *uniform planted* models attempt to "hide" a satisfiable assignment in an instance. In this model, a *planted assignment* $x^\star$ is first selected uniformly at random from $\{0,1\}^n$, and clauses are selected uniformly from the set of all clauses that are satisfied by $x^\star$. In the $\mathcal{P}_{n,m}$ model, exactly

[1] Generating the clause set with replacement is easier in practice, and facilitates our analysis later in the paper.

m random clauses are selected independently and uniformly with replacement from the set of clauses satisfied by $x^\star$. Similarly, in the $\mathcal{P}_{n,p}$ model, each 3-CNF clause over n variables that is satisfied by the planted assignment is selected independently with probability p. Other conditional distributions have also been studied, for example see Krivelevich et al. [8].

1.2 Background

The (1+1) EA has been the subject of the first analyses of worst-case expected runtime for pseudo-Boolean functions. Droste et al. [5] showed that the expected runtime of the (1+1) EA is bounded above by $O(n^n)$ steps over all pseudo-Boolean functions. Moreover, they showed that linear pseudo-Boolean polynomials are optimized in $O(n \log n)$ steps in expectation by the (1+1) EA. More recently, Witt [17] has derived an $en \ln n + O(n)$ bound for the (1+1) EA optimizing linear functions, which is tight up to lower order terms.

The case of general functions over $\{0,1\}^n$ is currently less clear. The class of pseudo-Boolean polynomial functions of degree at most $k \geq 2$ is already NP-hard since it contains maximum k-satisfiability. Some theoretical analyses have been carried out to investigate large-scale search space properties for k-satisfiability [14,15]. To our knowledge, no results connecting the k-satisfiability search space to EA runtime analysis have yet been carried out.

In this paper, we reduce the problem of finding a satisfying assignment to a 3-CNF formula to finding the maximum of a degree-3 pseudo-Boolean polynomial. Koutsoupias and Papadimitriou [7] showed that the $\mathcal{P}_{n,p}$ distribution has desirable search space properties for a greedy algorithm. Using similar techniques, we extend this analysis to the $\mathcal{P}_{n,m}$ distribution, and prove that the (1+1) EA can also exploit these properties to solve high-density formulas efficiently. For each distribution $\mathcal{P}_{n,m}$, $\mathcal{P}_{n,p}$, and $\mathcal{U}_{n,m}^{\mathrm{SAT}}$ in the high density regime, we prove that the (1+1) EA can find a satisfying assignment in polynomial time with probability $1 - o(1)$ on every formula, except for a set of measure vanishing exponentially fast in n. We also give a corresponding lower bound that suggests our upper bounds are tight up to a factor of $O(n)$ and conjecture that our upper bounds can be improved by a linear factor.

2 Preliminaries

A sequence of events $\{\mathcal{E}_n\}$ is said to hold *with high probability* if $\lim_{n\to\infty} \Pr(\mathcal{E}_n) = 1$. We will often make use of the following theorem. A proof can be found, for example, in the text by Motwani and Raghavan [11].

Theorem 1 (Chernoff Bounds). *Let $X_1, X_2, \ldots X_n$ be independent Poisson trials such that for $1 \leq i \leq n$, $\Pr(X_i = 1) = p_i$, where $0 < p_i < 1$. Let $X = \sum_{i=1}^n X_i$, $\mu = \mathrm{E}(X) = \sum_{i=1}^n p_i$. Then for $0 < \delta \leq 1$, $\Pr(X \geq (1+\delta)\mu) \leq e^{-\mu\delta^2/3}$ and $\Pr(X \leq (1-\delta)\mu) \leq e^{-\mu\delta^2/2}$.*

Chernoff bounds provide sharp bounds for tail probabilities in situations where we can estimate the expected number of successes from a series of independent trials. We will need the following two definitions.

Definition 1. *For any arbitrary $x \in \{0,1\}^n$, we define a pair of sets S_x and U_x that partition the set of all possible 3-CNF clauses on n variables as follows. S_x is the set of all 3-CNF clauses on n variables that are satisfied by x. Similarly, U_x is the set of all 3-CNF clauses on n variables that are not satisfied by x.*

Definition 2. *The* hypercube graph *of order n is the undirected graph $G = (V, E)$ where $V = \{0,1\}^n$ and $\{x, y\} \in E \iff |\{i : x_i \neq y_i\}| = 1$.*

Let F be a satisfiable 3-CNF formula on n variables. Denote as $x^\star \in \{0,1\}^n$ an assignment (possibly unique) that satisfies F. We define the potential function $\varphi(x) = |\{i : x_i \neq x_i^\star\}|$. F induces an orientation and an edge labeling on G in the following way. Let $G_{F,x^\star}$ be the directed, edge-labeled graph such that the directed edge (x, y) appears in $E(G_{F,x^\star})$ if and only if x and y are neighbors in G and $\varphi(y) < \varphi(x)$. Furthermore, (x, y) is labeled *deceptive* if x satisfies at least as many clauses in F as y.

3 Random Planted Formulas

In this section, we study the distribution of graphs $G_{F,x^\star}$ where F is a formula constructed by a random planted model and $x^\star$ is the planted assignment. We will rely on these results to apply multiplicative drift theorems that bound the runtime of the (1+1) EA for all but a vanishing fraction of high-density formula.

Definition 3. *An assignment x is* bad *if, for any constant $\epsilon > 0$, $\varphi(x) > (1/2 + \epsilon)n$. An assignment x is* good *if it is not bad.*

Definition 4. *The directed graph $G'_{F,x^\star}$ is the subgraph of $G_{F,x^\star}$ induced by the set of all good assignments.*

For a particular 3-CNF distribution, we want to derive the probability of deceptive edges appearing in the hypercube graph induced by formulas drawn from that distribution. If a region of the search space contains no deceptive edges, then the local gradient is consistent with the distance to a solution since every strictly improving Hamming neighbor of a solution in that region is also strictly closer to $x^\star$. This is obviously a nice property to have in the search space, and we call formulas *well-structured* that have this property.

Definition 5. *A planted 3-CNF formula F is said to be* well-structured *if there are no deceptive edges in $G'_{F,x^\star}$ where $x^\star$ is the planted assignment.*

Koutsoupias and Papadimitriou [7] studied the $\mathcal{P}_{n,p}$ distribution, and the next theorem follows from their work.

Theorem 2 (Koutsoupias and Papadimitriou [7]). *Suppose F is a 3-CNF formula constructed by the $\mathcal{P}_{n,p}$ model. The probability that F is well-structured is bounded below by $1 - e^{-cpn^2+\Theta(n)}$ for some constant $c > 0$.*

We extend this analysis to the $\mathcal{P}_{n,m}$ distribution. In this model, we first choose an assignment $x^\star$ uniformly at random, then choose exactly m clauses *with replacement* from the set of $(2^3-1)\binom{n}{3}$ clauses that satisfy $x^\star$.

Lemma 1. *Suppose (x, y) is a directed edge in $G'_{F,x^\star}$. Then,*

$$|S_{x^\star} \cap (U_x \cap S_y)| = \binom{n-1}{2}, \text{ and, } |S_{x^\star} \cap (S_x \cap U_y)| \leq \gamma(n)\binom{n-1}{2}$$

where $\gamma(n) = (1 + o(1))(3/4 + \epsilon - \epsilon^2)$ for any constant $\epsilon > 0$.

Proof. Without loss of generality, suppose $x^\star = (1, 1, \ldots, 1)$. In this case, $S_{x^\star}$ is the set of all clauses with at least one positive literal. Since x and y are Hamming neighbors, they differ by exactly one bit i which is set to zero in x and set to 1 in y. Thus $S_{x^\star} \cap (U_x \cap S_y)$ contains clauses where (1) x_i appears as a positive literal, and (2) the polarity of the remaining two literals in the clause are uniquely determined by their state in x. There are $\binom{n-1}{2}$ ways to choose these remaining two literals. Similarly, $S_{x^\star} \cap (S_x \cap U_y)$ contains clauses in which the literal $\bar{x}_i$ appears and the polarity of the remaining two literals again are uniquely determined. However, we cannot choose all $\binom{n-1}{2}$ such literals, because some of these correspond to clauses where all three literals are negative (and hence do not belong to $S_{x^\star}$). These literals correspond to the elements in x that are set to 1 since all such literals must be negative if they appear in any clause not satisfied by y. There are $n - \varphi(x)$ such elements. By subtracting out the $\binom{n-\varphi(x)}{2}$ ways to choose two negative literals, we obtain

$$|S_{x^\star} \cap (S_x \cap U_y)| = \binom{n-1}{2} - \binom{n-\varphi(x)}{2} \leq \binom{n-1}{2} - \binom{n(1/2-\epsilon)}{2}.$$

The final inequality holds since x is good, so $\varphi(x) \leq n(1/2 + \epsilon)$. Setting

$$\gamma(n) = 1 - \binom{n(1/2-\epsilon)}{2} \Big/ \binom{n-1}{2}$$

completes the proof since $\lim_{n\to\infty} \gamma(n) = 3/4 + \epsilon - \epsilon^2$. □

Theorem 3. *Suppose F is a 3-CNF formula constructed by the $\mathcal{P}_{n,m}$ model. The probability that F is well-structured is bounded below by $1 - e^{-cm/n+\Theta(n)}$ for some constant $c > 0$.*

Proof. Let (x, y) be an arbitrary edge in $G'_{F,x^\star}$. We define the following random variables that count clauses in F.

$$Z_1 = |\{\text{clauses } C \text{ in } F \ : C \in (S_{x^\star} \cap (U_x \cap S_y))\}|,$$
$$Z_2 = |\{\text{clauses } C \text{ in } F \ : C \in (S_{x^\star} \cap (S_x \cap U_y))\}|.$$

Since the m clauses are chosen independently with replacement, the probability of choosing a clause from $(S_{x^\star} \cap (U_x \cap S_y))$ is, by Lemma 1, $\binom{n-1}{2} / \left(7\binom{n}{3}\right) = 3/(7n)$. Similarly, the probability of choosing a clause from $(S_{x^\star} \cap (S_x \cap U_y))$ is at most $3\gamma(n)/(7n)$. Hence Z_1 and Z_2 are binomially distributed independent random variables, both with m trials, and their expected values are $\mathrm{E}(Z_1) = 3m/(7n)$ and $\mathrm{E}(Z_2) \le \gamma(n)3m/(7n)$.

The event that (x, y) is labeled deceptive under F is equivalent to the event $Z_1 \le Z_2$, and thus the probability that (x, y) is labeled deceptive is $\Pr(Z_1 \le Z_2) \le \Pr(Z_1 \le t) + \Pr(Z_2 \ge t)$ for any $t > 0$. Appealing to Theorem 1, this is at most

$$\begin{aligned} &\exp\left(-\frac{(t - \mathrm{E}(Z_1))^2}{2\mathrm{E}(Z_1)}\right) + \exp\left(-\frac{(t - \mathrm{E}(Z_2))^2}{3\mathrm{E}(Z_2)}\right) \\ \le &\exp\left(-\frac{(t - \mathrm{E}(Z_1))^2}{3\mathrm{E}(Z_1)}\right) + \exp\left(-\frac{(t - \mathrm{E}(Z_2))^2}{3\mathrm{E}(Z_2)}\right). \end{aligned}$$

Setting $t = \sqrt{\mathrm{E}(Z_1)\mathrm{E}(Z_2)}$, the probability is at most

$$2\exp\left(-\frac{\left(\sqrt{\mathrm{E}(Z_1)} - \sqrt{\mathrm{E}(Z_2)}\right)^2}{3}\right) \le 2\exp\left(-\frac{m\left(1 - \sqrt{\gamma(n)}\right)^2}{7n}\right) < 2e^{-cm/n},$$

by substituting the value bounds on the expectations of Z_1 and Z_2 from above. Here $0 < c < \left(1 - \sqrt{\gamma(n)}\right)^2/7$ is a positive constant following from the asymptotic bound on $\gamma(n)$.

Finally, by applying the union bound, the probability that any edge in $G'_{F,x^\star}$ is deceptive is at most $|E|2e^{-cm/n}$. The claim then follows from the fact that the number of edges in $G'_{F,x^\star}$ is at most $n2^{n-1}$. □

The uniform filtered 3-CNF distribution $\mathcal{U}_{n,m}^{\mathrm{SAT}}$ is the conditional distribution generated by conditioning $\mathcal{U}_{n,m}$ on satisfiability. For dense enough formulas, the uniform filtered distribution is statistically close to the planted distribution.

Theorem 4 (Ben-Sasson et al. [2]). *The 3-CNF distributions $\mathcal{P}_{n,m}$ and $\mathcal{U}_{n,m}^{\mathrm{SAT}}$ coincide in the regime $m/n = \Omega(\log n)$ in the following sense.*

There exists a constant $c > 0$ such that when $m \ge cn \ln n$, then with high probability, a formula constructed by the $\mathcal{P}_{n,m}$ or the $\mathcal{U}_{n,m}^{\mathrm{SAT}}$ model has exactly one satisfying assignment. Moreover, if F is an arbitrary formula with m clauses and n variables, such that F has a unique satisfying assignment, the probability of constructing F from $\mathcal{P}_{n,m}$ is equal to the probability of constructing F from $\mathcal{U}_{n,m}^{\mathrm{SAT}}$.

Hence for $m/n = \Omega(\log n)$, except for a set of measure that tends to zero, formulas constructed by the planted model or the filtered model have the same probability. It follows that the claim of Theorem 3 also applies to $\mathcal{U}_{n,m}^{\mathrm{SAT}}$ in the high-density regime ($m/n \ge cn$ for a constant $c > 0$ sufficiently large).

4 Runtime Analysis

The runtime analysis of randomized search heuristics on randomly constructed instances involves two sources of randomness. We must deal with random inputs, in this case, the random formula, and also with the random decisions of the algorithm at the same time. To handle this, we assume the formula has the well-structured property and derive tail bounds on the runtime conditioned on that property. We then use the results of the previous section to bound the probability that the formula is well-structured in a given density regime.

We analyze the runtime of the standard (1+1) EA (Algorithm 1) searching for a satisfying assignment to a formula F by optimizing the corresponding pseudo-Boolean function f that counts the satisfied clauses in F.

Algorithm 1. The (1+1) EA

```
choose x ∈ {0,1}^n uniformly at random;
repeat forever
    y ← x;
    flip each bit of y independently with prob. 1/n;
    if f(y) ≥ f(x) then x ← y
```

Following the typical approach to runtime analysis, we view each run of the (1+1) EA as an infinite stochastic process $(x^{(1)}, x^{(2)}, \ldots, x^{(t)}, \ldots)$, where $x^{(t)} \in \{0,1\}^n$ denotes the assignment generated in iteration t of the algorithm. The runtime T of an algorithm is the random variable $T = \inf\{t \in \mathbb{N} : x^{(t)} \text{ satisfies } F\}$. The main result of this section is stated in the following theorem.

Theorem 5. *Suppose F is a well-structured formula. Then with probability $1 - o(1)$, the time until the (1+1) EA finds a satisfying assignment for F is bounded by $O(n^2 \log n)$.*

To prove Theorem 5, we will rely on the favorable search space properties of well-structured formulas. In particular, we will show that, as long as the (1+1) EA remains in the good region, its drift towards the planted assignment can be bounded below by a positive term.

Lemma 2. *Suppose F is a well-structured formula and that $\varphi(x^{(t)}) \le (1/2 + \epsilon/2)n$. Then the probability that $x^{(t+1)}$ is a bad assignment is at most $e^{-\Omega(n \log n)}$.*

Moreover, if $\varphi(x^{(1)}) \le (1/2 + \epsilon/2)n$, then with probability $1 - o(1)$, after $t \le p(n)$ iterations, where p is a polynomial in n, the (1+1) EA never generates a bad assignment.

Proof. In each step, the probability that at least k bits are changed is at most

$$\binom{n}{k}\left(\frac{1}{n}\right)^k \le \frac{1}{k!} \le \left(\frac{e}{k}\right)^k = e^{-\Omega(k \log k)}.$$

The assignment $x^{(t)}$ is at Hamming distance at least $n\epsilon/2$ from any bad assignment. Thus, for $x^{(t+1)}$ to be bad, mutation must change at least $k = n\epsilon/2$ bits.

The second part of the claim follows from the fact that the probability of no bad assignment generated in $p(n)$ iterations is at least

$$\left(1 - e^{-\Omega(n \log n)}\right)^{p(n)} \geq 1 - p(n) \cdot e^{-\Omega(n \log n)} = 1 - o(1),$$

where we have applied Bernoulli's inequality. We remark here that even after any polynomial number of steps, the probability that the (1+1) EA never generates a bad assignment is going to one exponentially fast. □

Lemma 3. *We consider the execution of the (1+1) EA on a well-structured formula F. Define the sequence of random variables $\{X_t : t > 0\}$ as $X_t = \varphi(x^{(t)})$. We bound the drift of the stochastic process described by this sequence from below. Suppose that $\varphi(x^{(t)}) \leq (1/2 + \epsilon/2)n$, then* $\mathrm{E}\left(X_t - X_{t+1} \mid X_t\right) \geq cX_t/n^2$ *where c is a positive constant.*

Proof. Without loss of generality, let $x^\star = (1, 1, \dots, 1)$. We consider the contribution to the drift from different events. Let y be the intermediate offspring produced by mutating $x^{(t)}$. Note that by the dynamics of the (1+1) EA, $x^{(t+1)} = y$ if and only if $f(y) \geq f(x^{(t)})$.

Let A denote the event that $\varphi(y) > (1/2 + \epsilon)n$. In this event, the drift can be negative if $f(y)$ is no worse than $f(x^{(t)})$. By the law of total expectation, the drift can be written as

$$\mathrm{E}(X_t - X_{t+1} \mid X_t \cap \neg A)(1 - \Pr(A)) + \mathrm{E}(X_t - X_{t+1} \mid X_t \cap A) \Pr(A).$$

Moreover, we have assumed that $\varphi(x^{(t)}) \leq (1/2+\epsilon/2)n$ so $\Pr(A)$ can be bounded by Lemma 2, and we thus have

$$\mathrm{E}(X_t - X_{t+1} \mid X_t) \geq (1 - o(1))\mathrm{E}(X_t - X_{t+1} \mid X_t \cap \neg A) - ne^{-\Omega(n \log n)}. \quad (1)$$

For the remaining cases of the proof, we assume that the event $\neg A$ has occurred. This is equivalent to the assumption that y lies in the good region. Let B be the event that at least one of the $\varphi(x^{(t)})$ zero-bits flip. Since we assume that both $x^{(t)}$ and y are in the good region, we now argue that if the event $\neg B$ occurs, then either $y = x^{(t)}$, or $f(y) < f(x^{(t)})$. Under this event, if none of the $n - \varphi(x^{(t)})$ one-bits flip to zero, then obviously y and $x^{(t)}$ are equivalent. On the other hand, if some one-bits flip, by transitivity of non-deceptive edges in $G_{F,x^\star}$, y must satisfy strictly fewer clauses than $x^{(t)}$. In either case, after selection $x^{(t)} = x^{(t+1)}$. By the law of total probability we have

$$\mathrm{E}(X_t - X_{t+1} \mid X_t \cap \neg A) = \Pr(B) E(X_t - X_{t+1} \mid X_t \cap \neg A \cap B)$$

since the drift is zero under the event $\neg B$. Since each one-bit flips with probability $1/n$, by linearity of expectation,

$$E(X_t - X_{t+1} \mid X_t \cap \neg A \cap B) \geq \left(1 - \frac{n - \varphi(x^{(t)})}{n}\right) = \frac{X_t}{n}.$$

Finally, since $\varphi(x^{(t)}) \geq 1$ (otherwise, a satisfying assignment has been found) $\Pr(B) \geq 1/n$. The claim is then proved by applying Equation (1) and choosing a sufficiently small constant c. □

Proof of Theorem 5. By Theorem 1, with high probability $\varphi(x^{(1)}) \leq (1/2+\epsilon/2)n$. Lemma 3 ensures that the drift of the stochastic process defined by the potential function is multiplicative by a factor bounded by $\Omega(1/n^2)$. Applying the well-known Multiplicative Drift Theorem [4], as long as the (1+1) EA never jumps out of the good region, it has reduced the potential to zero in $O(n^2 \log n)$ steps. Furthermore, this bound holds with probability $1 - o(1)$ over the run [3].

Appealing to Lemma 2, after $O(n^2 \log n)$ iterations, the (1+1) EA generates any bad assignment only with probability $o(1)$ (and this term is even vanishing exponentially fast), hence the claim is proved. □

Corollary 1. *There exist positive constants c_1 and c_2 such that if F is a 3-CNF formula constructed from (1) the $\mathcal{P}_{n,p}$ model with $p \geq c_1/n$, or (2) the $\mathcal{P}_{n,m}$ model (and, due to Theorem 4, the $\mathcal{U}_{n,m}^{\mathrm{SAT}}$ model) with $m \geq c_2 n^2$, then the (1+1) EA has found a satisfying assignment in $O(n^2 \log n)$ steps with probability $1 - o(1)$.*

As we have already seen in the claim of Theorem 4, high density satisfiable random formulas are likely to have exactly one satisfying assignment. In such a case, it is straightforward to derive a lower bound on the expected runtime of the (1+1) EA. In particular, with probability 1/2, the randomly generated initial solution differs from the unique assignment in at least half the bits. Each such bit must flip at least once during the run until the satisfying assignment is found, and the expected number of steps before this event occurs is bounded below by $\Omega(n \log n)$. This bound is derived in Lemma 10 of the paper by Droste et al. [5] and immediately proves the following theorem.

Theorem 6. *If F is a random planted 3-CNF formula constructed as in Corollary 1, then with high probability F has exactly one satisfying assignment. In this case, the expected runtime of the (1+1) EA on F is bounded below by $\Omega(n \log n)$.*

5 Conclusion

In this paper, we have proved that all but a vanishing fraction of high-density random planted 3-CNF formulas can be solved efficiently by the (1+1) EA. We have shown that in the high-density regime, constraints impose favorable structure on the search space explored by such algorithms so that they run in polynomial time. In particular, we proved that the (1+1) EA finds a satisfying assignment in $O(n^2 \log n)$ iterations with probability $1 - o(1)$ on the $\mathcal{P}_{n,p}$ model when $p \geq c_1/n$ and on the $\mathcal{P}_{n,m}$ model when $m/n \geq c_2 n$ for sufficiently large positive constants c_1 and c_2. Since, at high densities, the $\mathcal{P}_{n,m}$ distribution is statistically close to the uniform filtered $\mathcal{U}_{n,m}^{\mathrm{SAT}}$ distribution, our results carry over to this case as well.

Additionally, we have presented a rigorous argument that the (1+1) EA takes at least $\Omega(n \log n)$ steps in expectation to solve all but a $o(1)$ fraction of random satisfiable 3-CNF formulas at high densities. We conjecture that the upper bound can be tightened to match this lower bound, and leave this as an open problem.

Acknowledgments. 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. Auger, A., Doerr, B.: Theory of Randomized Search Heuristics: Foundations and Recent Developments. World Scientific Publishing Company (2011)
2. Ben-Sasson, E., Bilu, Y., Gutfreund, D.: Finding a randomly planted assignment in a random 3-CNF (2002) (manuscript)
3. Doerr, B., Goldberg, L.A.: Drift analysis with tail bounds. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 174–183. Springer, Heidelberg (2010)
4. Doerr, B., Johannsen, D., Winzen, C.: Multiplicative drift analysis. Algorithmica 64(4), 673–697 (2012)
5. Droste, S., Jansen, T., Wegener, I.: On the analysis of the (1+1) evolutionary algorithm. Theoretical Computer Science 276(1-2), 51–81 (2002)
6. Gottlieb, J., Marchiori, E., Rossi, C.: Evolutionary algorithms for the satisfiability problem. Evolutionary Computation 10(1), 35–50 (2002)
7. Koutsoupias, E., Papadimitriou, C.H.: On the greedy algorithm for satisfiability. Information Processing Letters 43(1), 53–55 (1992)
8. Krivelevich, M., Sudakov, B., Vilenchik, D.: On the random satisfiability process. Combinatorics, Probability and Computing 18, 775–801 (2009)
9. Krivelevich, M., Vilenchik, D.: Solving random satisfiable 3CNF formulas in expected polynomial time. In: SODA, pp. 454–463 (2006)
10. Kroc, L., Sabharwal, A., Selman, B.: An empirical study of optimal noise and runtime distributions in local search. In: Strichman, O., Szeider, S. (eds.) SAT 2010. LNCS, vol. 6175, pp. 346–351. Springer, Heidelberg (2010)
11. Motwani, R., Raghavan, P.: Randomized algorithms. Cambridge University Press, New York (1995)
12. Neumann, F., Witt, C.: Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity. Springer (2010)
13. Seitz, S., Orponen, P.: An efficient local search method for random 3-satisfiability. Electronic Notes in Discrete Mathematics 16, 71–79 (2003)
14. Sutton, A.M., Howe, A.E., Whitley, L.D.: A theoretical analysis of the k-satisfiability search space. In: Stützle, T., Birattari, M., Hoos, H.H. (eds.) SLS 2009. LNCS, vol. 5752, pp. 46–60. Springer, Heidelberg (2009)
15. Sutton, A.M., Whitley, L.D., Howe, A.E.: A polynomial time computation of the exact correlation structure of k-satisfiability landscapes. In: GECCO (2009)
16. Witt, C.: Worst-case and average-case approximations by simple randomized search heuristics. In: Diekert, V., Durand, B. (eds.) STACS 2005. LNCS, vol. 3404, pp. 44–56. Springer, Heidelberg (2005)
17. Witt, C.: Tight bounds on the optimization time of a randomized search heuristic on linear functions. Combinatorics, Probability and Computing 22(2), 294–318 (2013)