Algorithmica (2016) 75:507–528
DOI 10.1007/s00453-015-0027-5

Superpolynomial Lower Bounds for the (1 + 1) EA on Some Easy Combinatorial Problems

Andrew M. Sutton[1]

Received: 18 October 2014 / Accepted: 2 July 2015 / Published online: 21 July 2015

Abstract The (1 + 1) EA is a simple evolutionary algorithm that is known to be efficient on linear functions and on some combinatorial optimization problems. In this paper, we rigorously study its behavior on three easy combinatorial problems: finding the 2-coloring of a class of bipartite graphs, solving a class of satisfiable 2-CNF formulas, and solving a class of satisfiable propositional Horn formulas. We prove that it is inefficient on all three problems in the sense that the number of iterations the algorithm needs to minimize the cost functions is superpolynomial with high probability. Our motivation is to better understand the influence of problem instance structure on the runtime character of a simple evolutionary algorithm. We are interested in what kind of structural features give rise to so-called metastable states at which, with probability $1 - o(1)$, the (1 + 1) EA becomes trapped and subsequently has difficulty leaving. Finally, we show how to modify the (1 + 1) EA slightly in order to obtain a polynomial-time performance guarantee on all three problems.

Keywords Runtime analysis · Lower bounds · (1 + 1) EA

1 Introduction

Randomized search heuristics such as evolutionary algorithms are general-purpose techniques for function optimization that are often applied to complex problems. These algorithms are usually deployed in situations in which problem-specific knowledge is incomplete or missing and the structure of the problem is not well-defined, changes

✉ Andrew M. Sutton
andrew.sutton@uni-jena.de

[1] Institut für Informatik, Theoretische Informatik I, Friedrich-Schiller-Universität Jena, 07743 Jena, Germany

dynamically, or is not completely accessible to the algorithm designer [18]. Still, it is often useful to study the behavior of randomized search heuristics on problems with well-understood structure in order to better understand their working principles. For evolutionary algorithms such as the $(1+1)$ EA, linear pseudo-Boolean functions have served this purpose well for obtaining positive theoretical results. For example, Witt [28] has recently shown that the $(1+1)$ EA requires only $en \ln n + O(n)$ steps in expectation and with high probability to optimize these functions. This bound is tight up to lower order terms, and is at most a polylogarithmic factor slower than the corresponding black-box complexity [8].

The use of computationally easy functions as an object of study has also illuminated several fundamental properties of randomized search heuristics and has cultivated the development of new proof techniques [7] and new efficient algorithms [5]. Inspired by these successes, we want to commute this approach into the realm of combinatorial problems with easily understood structure and study the influence of such structure on algorithm behavior. In particular, we are interested in natural, computationally easy problems for which the $(1+1)$ EA is provably inefficient. Our main focus is to better understand the structural properties of problems that pose challenges to evolutionary algorithms.

Our negative results are obtained by rigorously proving the existence of *metastable states* in the search space that arise from a particular condition called *spin-flip symmetry*, which is an invariance of the objective function under permutations on the alphabet [26]. Issues arising from spin-flip symmetry have been investigated for randomized search heuristics on the Ising model from statistical physics [3,13]. Simple variants of the Ising model are easy; the one-dimensional model can be optimized by the $(1+1)$ EA in expected time bounded by $O(n^3)$ [11], and the two-dimensional variant with positive unit weights can be optimized by the Metropolis algorithm in expected time bounded by $O(n^{4.5})$ [10]. Sudholt [24] proved that the expected runtime of the $(\mu+\lambda)$ EA applied to the Ising model on binary trees is bounded below by $2^{\Omega(n)}$, but that a simple genetic algorithm using crossover with a population size of two could solve the problem in expected polynomial time.

Inefficient behavior of the $(1+1)$ EA on easy combinatorial optimization problems has already been observed in a few other settings. Giel and Wegener [12] proved that the $(1+1)$ EA takes exponential time in expectation to find the maximum matching on a class of bipartite graphs. Neumann et al. [17] showed that the $(1+1)$ EA takes superpolynomial time with high probability to find the minimum cut in a weighted digraph and proved that a multi-objective evolutionary algorithm runs in polynomial time.

Insight into specific problem structure is often useful for the design of more efficient variation operators. This is common in applications, but has also been studied from a theoretical perspective. Doerr et al. [6] proved that tailored mutation operators for the Eulerian cycle problem on a graph of m edges can improve the efficiency of randomized local search and the $(1+1)$ EA resulting in an upper bound of $O(m^3)$. This beats the upper bound for the general version of the $(1+1)$ EA by a factor of m^2. Motivated similarly, Jansen and Sudholt [14] considered asymmetric mutation operators for binary strings on functions for which good solutions have low (or high) Hamming weight and give asymptotic speed-ups for the $(1+1)$ EA on certain pseudo-

Boolean functions with this property. Raidl et al. [22] showed that, when using a biased edge-exchange mutation for minimum spanning tree, the expected optimization time of the (1 + 1) EA solving minimum spanning tree is asymptotically equivalent to Kruskal's classical algorithm. In the current paper, we will develop a problem-specific operator that yields even a superpolynomial speed-up for the (1 + 1) EA on bipartite coloring and 2-SAT, and for a particular class of propositional Horn formulas.

A preliminary version of this paper appeared in the proceedings of GECCO 2014 [25]. The paper is organized as follows. In the remainder of this section, we introduce the algorithm and basic analytical framework. In Sect. 2, we analyze the runtime of the (1 + 1) EA searching for a 2-coloring of a bipartite graph and provide tail bounds that prove with high probability the evolutionary algorithm needs at least a superpolynomial number of iterations in the worst case.

In Sect. 3 we prove a similar result for the (1 + 1) EA searching for a satisfying assignment to satisfiable 2-CNF formulas and Horn formulas. Both problems are easy in the classical sense, but we show that there are instances for which, with high probability, the (1+1) EA does not find a satisfying assignment in time that is bounded above by any polynomial in the number of variables.

Finally, in Sect. 4 we show that the (1 + 1) EA can be adapted slightly by changing the fitness function and mutation operator to take advantage of domain-specific information, and that this adaptation results in a polynomial-time performance guarantee for every 2-coloring instance and every satisfiable 2-CNF formula. Furthermore, we prove that it can efficiently solve the class of pathological Horn satisfiability formulas presented in Sect. 3.2. We conclude the paper in Sect. 5.

1.1 Preliminaries

We consider in this paper the minimization of nonlinear pseudo-Boolean functions $f\colon \{0,1\}^n \to \mathbb{R}$ where f measures the cost of candidate solutions (encoded as length-n bitstrings) to a combinatorial problem. For a bitstring $x \in \{0,1\}^n$, we will denote the i-th element in x as $x[i]$. The (1 + 1) EA, defined in Algorithm 1 for minimizing f, is a basic randomized search heuristic for optimizing discrete functions. The (1 + 1) EA has been a prominent object of study in theoretical investigations of randomized search heuristics [2] since it is a *degenerate* evolutionary algorithm in the sense that it uses the smallest nontrivial population size (one), simple uniform mutation, no recombination, and a simple survival selection scheme. This natural simplicity facilitates theoretical analysis while still reflecting the behavior of more complex evolutionary approaches.

Algorithm 1: (1 + 1) EA.

```
Choose $x \in \{0,1\}^n$ uniformly at random;
repeat forever
    $y \leftarrow x$;
    Flip each bit $y[i]$ independently with probability $1/n$;
    if $f(y) \leq f(x)$ then $x \leftarrow y$
```

We regard a run of Algorithm 1 as an infinite stochastic process $\{x^{(t)} : t \in \mathbb{N}_0\}$, where $x^{(t)} \in \{0, 1\}^n$ denotes the candidate solution generated in iteration t of the algorithm. The *optimization time* T of an algorithm is the random variable

$$T := \inf \left\{ t \in \mathbb{N}_0 : \ f(x^{(t)}) \text{ is minimal} \right\}.$$

In the following, we will be primarily interested in tail bounds for T. In other words, we want to derive rigorous estimates on the probability that T exceeds some value $g(n)$.

We will need the following theorem that relates the drift of a stochastic process to bounds on its hitting time.

Theorem 1 (Negative Drift [19,20]) *Let* $\{X_t : t \in \mathbb{N}_0\}$ *be a sequence of random variables over* $\mathbb{R}_{\geq 0}$. *Suppose there exist an interval* $[a, b] \subseteq \mathbb{R}$, *two constants* $\delta, \varepsilon > 0$ *and, possibly depending on* $\ell := b - a$, *a function* $r(\ell)$ *satisfying* $1 \leq r(\ell) = o(\ell/\log(\ell))$ *such that for all* $t \geq 0$ *the following two conditions hold:*

1. $\mathrm{E}(X_{t+1} - X_t \mid X_t, a < X_t < b) \geq \varepsilon$;
2. *For all* $j \geq 0$, $\Pr(|X_{t+1} - X_t| \geq j \mid X_t, a < X_t) \leq \frac{r(\ell)}{(1+\delta)^j}$.

Then there is a constant $c > 0$ *such that for* $T := \min\{t \geq 0 : X_t \leq a \mid X_0 \geq b\}$ *it holds* $\Pr(T \leq 2^{c\ell/r(\ell)}) = 2^{-\Omega(\ell/r(\ell))}$.

2 Coloring Bipartite Graphs

Given a graph $G = (V, E)$ and a set $[k] = \{1, 2, \ldots, k\}$, a *coloring* of G is a mapping $\chi : V \rightarrow [k]$. We are interested in finding a coloring χ that minimizes the set of monochromatic edges $\{(u, v) \in E : \chi(u) = \chi(v)\}$. We say a graph G is k-colorable if there exists a k-coloring $\chi^\star$ such that

$$\left\{(u, v) \in E : \chi^\star(u) = \chi^\star(v)\right\} = \emptyset.$$

Note that a graph is 2-colorable if and only if it is bipartite, and in this case the problem of finding a 2-coloring with no monochromatic edges is in P. Obviously, a 2-coloring of a bipartite graph can be found in time linear to the size of the graph by depth-first search. In this section, we will introduce a class of pathological instances of bipartite graphs that the (1 + 1) EA fails to solve efficiently with high probability.

We construct a graph $G = (V, E)$ on $|V| = n$ vertices as follows. Let $0 < \epsilon < 1/2$ be an arbitrary real constant. Without loss of generality, let $m = n^{2\epsilon}/4 - 1/2$ and $k = \Theta(n^{1-2\epsilon})$ be integers. A *block* is a subgraph $G_i = (V_i, E_i)$ where $|V_i| = 4m + 2 = n^{2\epsilon}$ and $|E_i| = 2m^2 + 2m + 1$.

The set V_i is composed of five mutually disjoint sets $\{A_i, B_i, C_i, D_i, \{u_i, v_i\}\}$ where $|A_i| = |B_i| = |C_i| = |D_i| = m$. The subgraph induced by $A_i \cup B_i$ is isomorphic to the complete bipartite graph $K_{m,m}$. Similarly, the subgraph induced by $C_i \cup D_i$ is also isomorphic to $K_{m,m}$. We construct the remaining $2m + 1$ edges to form a cutset between these two complete bipartite subgraphs. We add (u_i, v_i) to E_i, and an edge

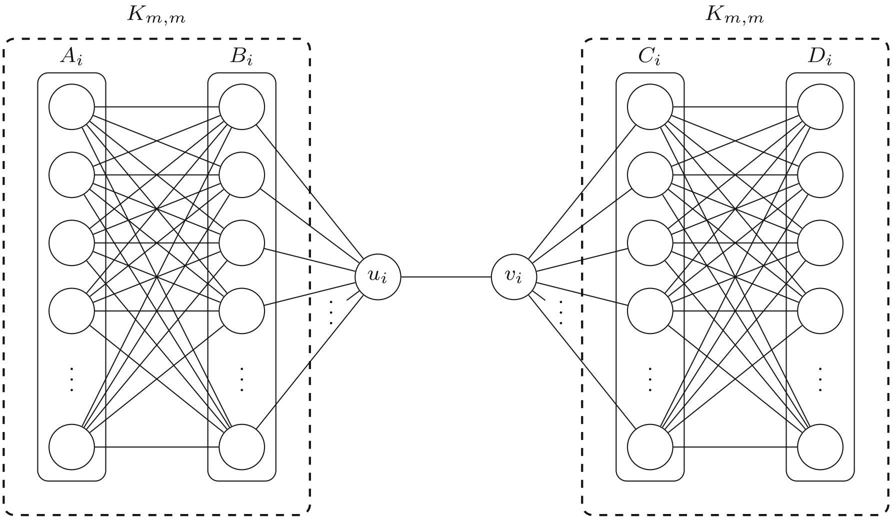

Fig. 1 Bipartite block subgraph $G_i = (V_i, E_i)$

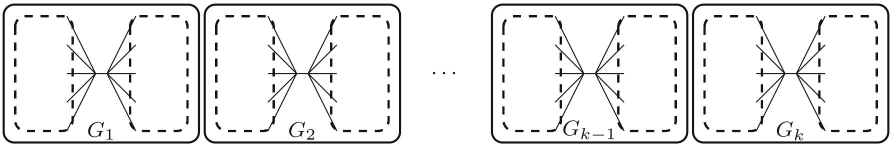

Fig. 2 A bipartite graph G constructed by connecting a sequence of $k = \Theta(n^{1-2\epsilon})$ blocks

between u_i and every vertex in B_i, as well as an edge between v_i and every vertex in C_i. Clearly G_i is bipartite (see Fig. 1).

We construct G by taking a sequence of $k = \Theta(n^{1-2\epsilon})$ blocks $G_1, G_2, \ldots, G_k$ (see Fig. 2) and attaching G_i to G_{i+1} for all $1 \leq i < k$ by adding arbitrary edges between vertices in D_i and A_{i+1} subject to the constraint that the maximum degree of the subgraph of G induced by $A_{i+1} \cup D_i$ is one.

The set of all 2-colorings of G is isomorphic to $\{0, 1\}^n$, and, as is typical with evolutionary algorithms, we consider each candidate solution as a bitstring $x \in \{0, 1\}^n$. The pseudo-Boolean fitness function $f : \{0, 1\}^n \to \mathbb{N}_0$ counts the number of monochromatic edges in a coloring, i.e.,

$$f(x) = |\{(u, v) \in E : x[u] = x[v]\}|. \tag{1}$$

Here $x[u]$ is the element of the bitstring corresponding to vertex u. Obviously, a 2-coloring corresponding to x is optimal if and only if $f(x) = 0$. Furthermore, the function f has the property of *spin-flip symmetry* since $f(x) = f(\sigma(x))$ where $\sigma : \{0, 1\}^n \to \{0, 1\}^n$ is the permutation that takes each bitstring to its complement. The presence of spin-flip symmetry can have a strong influence on the dynamics of a search algorithm [26], and we will shortly see that it can produce deceptive regions in the search space.

We want to analyze the performance of Algorithm 1 minimizing f as defined in (1) on the graph G. The main result of this section is that, with high probability, the (1 + 1) EA optimizing the pseudo-Boolean fitness function defined in Eq. (1) fails to find a 2-coloring for G in time bounded by any polynomial of n.

Theorem 2 *Let G be a bipartite graph as defined above for any real constant $0 < \epsilon < 1/2$. With probability $1 - o(1)$, the $(1+1)$ EA does not find a feasible 2-coloring of G in $2^{\Omega(n^\epsilon)}$ steps.*

Before proving Theorem 2, we will need a few definitions and technical lemmas. We will use the following result to obtain tight bounds on the probabilities of specific events.

Lemma 1 *Let $1 \leq h(n) < n$ be an integer function of n. Then*

$$\frac{h(n)}{en} \leq 1 - \left(1 - \frac{1}{n}\right)^{h(n)} \leq \frac{h(n)}{n}.$$

Proof For the lower bound, we have

$$\frac{1 - (1 - 1/n)^{h(n)}}{1/n} = \frac{1 - (1 - 1/n)^{h(n)}}{1 - (1 - 1/n)},$$

which is equal to the sum of the first $h(n)$ terms of a geometric series:

$$\sum_{j=0}^{h(n)-1} (1 - 1/n)^j \geq h(n)(1 - 1/n)^{n-1} \geq h(n)/e.$$

The upper bound follows from Bernoulli's inequality. □

Definition 1 For a block G_i, we say that a 2-coloring x of G is *deceptive for G_i* if the following five conditions hold.

1. $x[u_i] = x[v_i] = 0$.
2. $|\{v \in A_i : x[v] = 0\}| > m/2$,
3. $|\{v \in B_i : x[v] = 1\}| > m/2$,
4. $|\{v \in C_i : x[v] = 1\}| > m/2$,
5. $|\{v \in D_i : x[v] = 0\}| > m/2$,

For brevity, we will sometimes in this case say x is *deceptive* when G_i is clear from context.

Definition 2 Let $\delta \colon \{0,1\}^n \times 2^V \to \mathbb{N}_0$ count the deviation of the majority color from $|U|/2$ in a set $U \subseteq V$ under a particular coloring. That is, for a 2-coloring x and a subset $U \subseteq V$,

$$\delta(x, U) = \max_{z \in \{0,1\}} |\{v \in U : x[v] = z\}| - |U|/2.$$

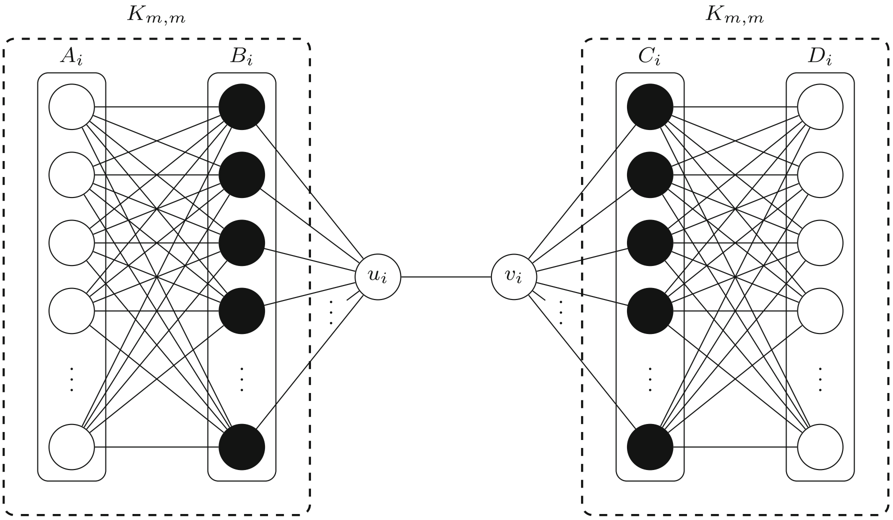

Fig. 3 A metastable state for a block G_i. Deceptive colorings for G_i lie in this state's basin of attraction, i.e., the stochastic process drifts toward this metastable state in the region of deceptive colorings for G_i

Let G_i be a block. For each $U \in \{A_i, B_i, C_i, D_i\}$ we define a function

$$\psi_U(x) = \begin{cases} \delta(x, U) & \text{if } x \text{ is deceptive for } G_i; \\ 0 & \text{otherwise.} \end{cases}$$

We define the following *block potential function* with respect to G_i.

$$\varphi_i(x) = \psi_{A_i}(x) + \psi_{B_i}(x) + \psi_{C_i}(x) + \psi_{D_i}(x).$$

Informally, a coloring is called deceptive for a block G_i because it lies in the basin of attraction of a so-called *metastable state* where the only monochromatic edge in the block is (u_i, v_i) (see Fig. 3). This metastable state is an attractor for the dynamics of the (1 + 1) EA, and we formalize this in the following lemma that ensures that the stochastic process described by the (1 + 1) EA has a positive drift toward this metastable state.

Lemma 2 *Let $x^{(t)}$ be the coloring found in iteration t of the (1 + 1) EA. Suppose $x^{(t)}$ is deceptive for a block G_i, and let P be the event that mutation changes the color of some vertices in G_i in iteration t. Suppose that*

$$\sqrt{m}/5 \leq \psi_U(x^{(t)}) \leq \sqrt{m}/4$$

for each $U \in \{A_i, B_i, C_i, D_i\}$. Then there exists a positive constant c such that

$$\mathrm{E}\left(\varphi_i(x^{(t+1)}) - \varphi_i(x^{(t)}) \;\middle|\; P\right) > 4c.$$

Proof We first show that under the claimed conditions there exists a positive constant c such that the expectation of $\psi_{A_i}(x^{(t+1)}) - \psi_{A_i}(x^{(t)})$ conditioned on P is at least c. Let y be the offspring created by uniform mutation in iteration t of the $(1+1)$ EA. By the selection process, $x^{(t+1)} = y$ if and only if $f(y) \le f(x^{(t)})$, otherwise $x^{(t+1)} = x^{(t)}$. Define the random variable $Z := \psi_{A_i}(x^{(t+1)}) - \psi_{A_i}(x^{(t)})$ and let I_P be the indicator random variable for event P.

In this proof, we will rely heavily on the law of total probability and partition the probability space based on the following events (along with their complements). Let Q_1 be the event that in iteration t, mutation does not change the color of any vertices in $V \setminus A_i$. Let Q_2 be the event that the block remains deceptive after mutation and selection.

We first consider the event $Q_1 \cap Q_2$. Since $x^{(t)}$ is deceptive for G_i, the majority of vertices in A_i are labeled 0 by $x^{(t)}$. Furthermore, each vertex in A_i labeled 1 is adjacent to at least $m/2 + \sqrt{m}/5$ vertices labeled 1 in B_i, and adjacent to at most $m/2 - \sqrt{m}/5$ vertices labeled 0 in B_i, and at most one vertex in D_{i-1} possibly labeled 0.

Under event Q_1, if the majority color in A_i is strictly reduced by mutation, the resulting coloring y has strictly more monochromatic edges, and thus y will not be accepted. Thus, $Q_1 \subseteq Q_2$ and so $Q_1 \cap Q_2 = Q_1$ and it suffices to bound $\mathrm{E}(Z \mid Q_1)$. There are $m-(m/2+\psi_{A_i}(x^{(t)}))$ vertices labeled 1 by $x^{(t)}$. The probability that exactly one of these vertices changes color under mutation is at least $1/(en)$. Each such event contributes to the expectation of Z under Q_1. Since a reduction in majority color will not be accepted under Q_1, by linearity of expectation and the assumed bounds on ψ_{A_i},

$$\mathrm{E}(Z \mid Q_1 \cap Q_2) = \mathrm{E}(Z \mid Q_1) \ge (m/2 - \sqrt{m}/4)/(en).$$

We now examine the event $\overline{Q_1} \cap Q_2$. In this case, vertices outside A_i are allowed to change, and the vertices of majority color in A_i might be lost after selection because a change in $V \setminus A_i$ is so favorable that it masks any loss of fitness coming from A_i. Before bounding the expectation of Z conditioned on $\overline{Q_1} \cap Q_2$, we consider the expected change in the majority color of A_i under *pure mutation*, that is, ignoring any effect of selection. Since each vertex changes color with probability $1/n$, by linearity of expectation, the expected change in majority color of A_i under pure mutation is:

$$\left(m - \left(\frac{m}{2} + \psi_{A_i}(x^{(t)})\right)\right)\Big/ n - \left(\frac{m}{2} + \psi_{A_i}(x^{(t)})\right)\Big/ n \ge -\frac{\sqrt{m}}{2n}. \tag{2}$$

The inequality is due to the assumption that $\psi_{A_i}(x^{(t)}) \le \sqrt{m}/4$. Under event $\overline{Q_1} \cap Q_2$, the expected loss in majority color under mutation and selection cannot be greater than the expected loss under pure mutation. By (2) we thus have

$$\mathrm{E}\left(Z \mid \overline{Q_1} \cap Q_2\right) \ge -\frac{\sqrt{m}}{2n}.$$

Applying the law of total probability, we can write

$$\begin{aligned}\mathrm{E}(Z \mid Q_2) &= \mathrm{E}(Z \mid Q_1 \cap Q_2)\Pr(Q_1 \mid Q_2) + \mathrm{E}(Z \mid \overline{Q_1} \cap Q_2)\,(1-\Pr(Q_1 \mid Q_2))\\ &\geq \frac{(m/2)-\sqrt{m}/4}{en}\Pr(Q_1 \mid Q_2) - \frac{\sqrt{m}}{2n}\,(1-\Pr(Q_1 \mid Q_2))\\ &\geq \frac{(m/2)-\sqrt{m}/4}{e^2 n} - \frac{\sqrt{m}}{2n}\end{aligned}$$

since $\Pr(Q_1 \mid Q_2) \geq \Pr(Q_1) = (1-1/n)^{n-m} \geq 1/e$. Again, by the law of total probability,

$$\mathrm{E}(Z) = \mathrm{E}(Z \mid Q_2)(1-\Pr(\overline{Q_2})) + \mathrm{E}(Z \mid \overline{Q_2})\Pr(\overline{Q_2}).$$

The block can only go non-deceptive after mutating at least $\sqrt{m}/5$ bits. The number of bits that change under mutation is distributed binomially, so the probability that at least $\sqrt{m}/5$ bits in block G_i change is at most $\binom{4m+2}{\sqrt{m}/5}(1/n)^{\sqrt{m}/5} \leq 1/(\sqrt{m}/5)!$ so we have $\Pr(\overline{Q_2}) \leq e^{-\Omega(\sqrt{m}\log m)}$. Thus,

$$\mathrm{E}(Z) \geq \frac{(m/2)-\sqrt{m}/4}{e^2 n} - \frac{\sqrt{m}}{2n} - e^{-\Omega(\sqrt{m}\log m)}.$$

To complete the proof, we must condition the expectation on the event P. If event P has not occurred, no change is possible inside the block and Z must be zero. Stated more formally, for an element ω of the probability space, $Z(\omega) = 0$ if $\omega \in \overline{P}$ and so we have $Z = ZI_P$. By definition,

$$\mathrm{E}(Z \mid P) = \mathrm{E}(ZI_P)/\Pr(P) = \mathrm{E}(Z)/\Pr(P).$$

Since $\Pr(P) = 1-\Pr(\overline{P}) = 1-(1-1/n)^{4m+2}$, by applying Lemma 1, we can tightly bound the probability of P,

$$(4m+2)/(en) \leq \Pr(P) \leq (4m+2)/n,$$

and thus

$$\mathrm{E}(Z \mid P) \geq \frac{2m-\sqrt{m}}{16e^2 m + 8e^2} - \frac{e\sqrt{m}}{8m+4} - e^{-\Omega(\sqrt{m}\log m)}.$$

It follows that, for n sufficiently large, there exists a positive constant c such that

$$\mathrm{E}(Z|P) = \mathrm{E}\left(\psi_{A_i}(x^{(t+1)}) - \psi_{A_i}(x^{(t)}) \,\middle|\, P\right) > c.$$

Using symmetric arguments, the expected differences for ψ_{B_i}, ψ_{C_i}, and ψ_{D_i} are also all bounded below by c. The claim for the block potential function φ_i then follows by linearity of expectation. □

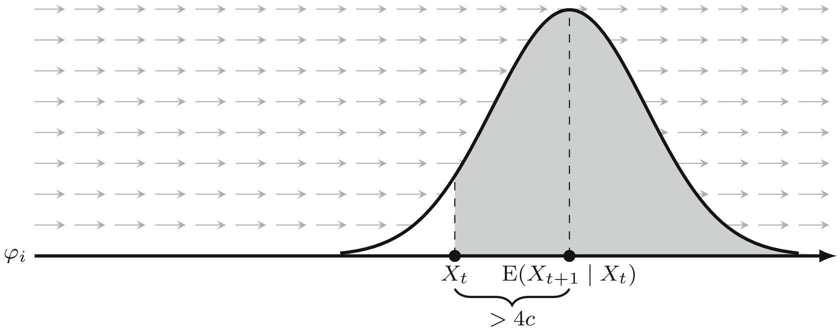

Fig. 4 Stochastic drift of the block potential function $X_t = \varphi_i(x^{(t)})$ on a coloring deceptive for G_i

We have therefore proved that the drift (conditioned on P) of the stochastic process $\{\varphi_i(x^{(t)}) : t \in \mathbb{N}_0\}$ is strictly positive for a set of colorings in the deceptive region for G_i. Figure 4 illustrates the situation. This lemma formalizes the concept that for colorings deceptive for a block G_i, in expectation the (1 + 1) EA is attracted to the metastable state for G_i.

Definition 3 For a block G_i, we say a deceptive 2-coloring x of G is *bad for G_i* if the following five conditions hold.

1. $x[u_i] = x[v_i] = 0$.
2. $|\{v \in A_i : x[v] = 0\}| \geq m/2 + \sqrt{m}/4$,
3. $|\{v \in B_i : x[v] = 1\}| \geq m/2 + \sqrt{m}/4$,
4. $|\{v \in C_i : x[v] = 1\}| \geq m/2 + \sqrt{m}/4$,
5. $|\{v \in D_i : x[v] = 0\}| \geq m/2 + \sqrt{m}/4$,

We sometimes say that such a coloring is *bad* when G_i is clear from context.

Lemma 3 *Let $U \subseteq V$ be a subset of m vertices where $m > 4$. In any uniformly random coloring of U, at least $m/2 + \sqrt{m}/4$ vertices in U are colored 0 (1) with probability at least $\frac{1}{15e} > 1/41$.*

Proof Let X be the random variable that counts the number of vertices in U colored 0 (1) in a uniformly random coloring of V. Since X is distributed binomially, we have the following lower bound for any $0 < t < m/8$ (derived explicitly in [15, Proposition 7.3.2 on page 46] and follows as a special case of a result due to Feller [9]).

$$\Pr(X \geq \mathrm{E}(X) + t) \geq \frac{1}{15} e^{-16t^2/m}.$$

The claim holds by substituting $\mathrm{E}(X) = m/2$ and $t = \sqrt{m}/4 < m/8$. □

Lemma 3 gives a lower bound on the probability that, in a uniformly random 2-coloring, there is a majority of vertices of a certain color in a subset of size m, and that this majority deviates from $m/2$ by at least $\sqrt{m}/4$. This immediately provides the following bound on the probability that the initial random 2-coloring is bad for any block.

Lemma 4 *Let $G_i = (V_i, E_i)$ be an arbitrary block of G. With at least constant probability, the initial uniformly random 2-coloring is bad for G_i.*

Proof Since $\{A_i, B_i, C_i, D_i, \{u_i, v_i\}\}$ form a partition of V_i, the five events in Definition 3 are mutually independent, and it suffices to show their joint probability is bounded below by a constant. Clearly, event (1), i.e., the event in which $x[u_i] = x[v_i] = 0$ in the initial coloring $x \in \{0,1\}^n$, occurs with probability $1/4$. By Lemma 3, the probability of each of events 2–5 is at least $1/41$. By independence, the joint probability of the five events is $\Omega(1)$. □

We are now ready to prove the main result of this section.

Proof of Theorem 2 A feasible 2-coloring of G must be non-deceptive for all blocks so a necessary condition for every feasible 2-coloring $x^\star$ is that $\varphi_i(x^\star) = 0$ for every $i \in \{1, 2, \ldots, k\}$. If at any time during the execution of the (1+1) EA a block is bad, the algorithm must change the color of at least $\Omega(\sqrt{m})$ vertices to make it non-deceptive.

Suppose that $x^{(0)}$ is bad with respect to block G_i. We consider the restriction of the stochastic process $\{x^{(t)} : t \in \mathbb{N}_0\}$ consisting only of steps in which there is a mutation that changes vertices in G_i. The drift of φ_i in this process is exactly the conditional drift from the claim of Lemma 2. We now appeal to the Negative Drift Theorem (Theorem 1). The first condition of Theorem 1 is given by Lemma 2. Since the (1 + 1) EA uses uniform mutation, the number of vertices that change under mutation is distributed binomially so the probability of a jump of size at least j is at most $\binom{n}{j}(1/n)^j \leq 2^{-j+1}$ and so the second condition holds with $\delta = 1$ and $r(\ell) = 2$. Thus, with probability $1 - o(1)$, the EA needs $2^{\Omega(\sqrt{m})} = 2^{\Omega(n^{\epsilon})}$ steps in this restricted process to reduce the potential of a bad block since $m = \Theta(n^{2\epsilon})$. Since the restricted process to solve the bad block can only be faster than the optimization time of the EA (in which there can only be additional steps where nothing occurs within the block), it follows that with high probability the EA needs at least superpolynomial time to solve any bad blocks.

Finally, by Lemma 4, with probability $p = \Omega(1)$ the initial uniformly random coloring is already bad for an arbitrary block. Consequently, with probability at least $1 - (1-p)^k = 1 - o(1)$ there is at least one bad block in the initial coloring. The asymptotic expression follows from the fact that $k = \Theta(n^{1-2\epsilon})$. Thus, with high probability, the EA must first fix a bad block, and this takes superpolynomial time with high probability. Since all bad blocks must be fixed before a feasible 2-coloring is found, the claim follows. □

3 Tractable Subclasses of Propositional Formulas

Propositional satisfiability is an important combinatorial problem with extensive theoretical and practical relevance. A propositional logic formula is an expression built from a set of Boolean variables, logical operators ($\wedge$, $\vee$, $\neg$) and parentheses. A propositional formula is said to be *satisfiable* if there is an assignment to its variables so that the entire formula evaluates to true. If no such assignment exists, it is *unsatisfiable*.

In general, deciding the satisfiability of a propositional formula is NP-hard. Schaefer's Dichotomy Theorem [23] identifies a set of properties that guarantee a formula is decidable in polynomial time: (1) formulas that are satisfied when all variables are set to false (respectively, true), (2) formulas formed by conjunctions of disjunctive clauses that each contain at most one positive (respectively, negative) literal (3) formulas in conjunctive normal form with at most two literals in each clause, and (4) affine formulas. On the other hand, the class of formulas for which none of these properties hold is NP-complete. In this section, we study the behavior of the (1 + 1) EA searching for satisfying assignments to tractable formulas with properties (2) and (3). Formulas satisfying property (2) are called Horn (respectively, reverse Horn) formulas, and formulas satisfying property (3) are called 2-CNF formulas.

In the context of propositional satisfiability, the (1 + 1) EA is obviously an *incomplete* heuristic because it is incapable of proving that a formula is unsatisfiable. We therefore restrict our analysis to its performance on *satisfiable* formulas. This restriction has the following theoretical motivation. Suppose T is the random variable that counts the number of iterations until the (1+1) EA finds a satisfying assignment to a satisfiable formula on n variables and that we can estimate tail bounds $\Pr(T \geq g(n)) \leq \varepsilon$ for some constant $0 < \varepsilon < 1$. Then we can construct a Monte Carlo algorithm with one-sided error as follows. Given an arbitrary formula on n variables, run the (1+1) EA for $g(n)$ steps and declare it unsatisfiable if no satisfying assignment is found during that time, otherwise return the found assignment. In this strategy, the (1 + 1) EA can only produce an incorrect result on satisfiable formulas, and the corresponding error probability is at most ε. On the other hand, bounds of the form $\Pr(T \geq g(n)) = 1-o(1)$ indicate that with high probability at least $g(n)$ steps are needed before any satisfying assignment is found, and a suitable error probability is not attained for $g(n)$ or fewer iterations.

3.1 Satisfiable 2-CNF Formulas

A propositional formula in k-conjunctive normal form (k-CNF) is a set of disjunctive clauses on a set of n Boolean variables in which each clause is a set of at most k literals (a variable or its negation). A formula is said to be *satisfiable* if there is an assignment to the variables such that all clauses in the formula evaluate to true. If there is no such assignment, the formula is said to be *unsatisfiable*.

For $k \geq 3$, the problem of deciding the satisfiability of k-CNF formulas is already NP-complete. However, when $k = 2$, there is a linear time deterministic algorithm [1] for solving the decision problem, and a randomized local search heuristic that can identify satisfiable formula in quadratic time [16,21].

In this section we study the performance of the (1 + 1) EA on satisfiable 2-CNF formulas. The set of assignments to n Boolean variables $\{v_1, v_2, \ldots, v_n\}$ is isomorphic to $\{0, 1\}^n$, the set of length-n bitstrings. Given a bitstring $x \in \{0, 1\}^n$, we assign variable v_i the value *true* if and only if $x[i] = 1$ and v_i the value *false* if and only if $x[i] = 0$. A literal ℓ is satisfied by an assignment $x \in \{0, 1\}^n$ if $\ell = v_i$ and $x[i] = 1$, or $\ell = \neg v_i$ and $x[i] = 0$.

We call a literal *negative* if it is an instance of a negated variable, otherwise, it is *positive*. A clause is satisfied by x if at least one of its literals is satisfied by x. Given an assignment x, we use the standard fitness function for propositional satisfiability that counts the number of unsatisfied clauses under x, that is,

$$f(x) = |\{C \in F : C \text{ is not satisfied by } x\}|. \tag{3}$$

Thus, the satisfying assignments of F are exactly the zeros of f. We now show by a straightforward reduction that the result in Sect. 2 also carries over to 2-CNF formulas.

Let $G = (V, E)$ be a graph on n vertices. We construct a 2-CNF formula consisting of n variables $v_1, v_2, \ldots v_n$ with $2|E|$ clauses as follows. For every edge $(v_i, v_j) \in E$ we add the clauses $(v_i \vee v_j)$ and $(\neg v_i \vee \neg v_j)$. For any assignment, at most one of these clauses is violated, and one is violated if and only if $x[i] = x[j]$. Thus the fitness of x calculated by the coloring fitness function defined in Eq. (1) is equivalent to the fitness defined by the number of unsatisfied clauses in Eq. (3) for the corresponding formula. The next Lemma follows as an immediate consequence.

Lemma 5 *Let $G = (V, E)$ be a bipartite graph. Then the 2-CNF formula constructed using the above process is satisfiable.*

Moreover, the equivalence immediately implies that there are also hard 2-CNF formulas for the (1 + 1) EA. In particular, this result entails the following theorem.

Theorem 3 *Let F be a propositional formula constructed by reduction from the bipartite graph G in Sect. 2. By Lemma 5, F is satisfiable. With probability $1 - o(1)$, the $(1 + 1)$ EA does not find a satisfying assignment to F in $2^{\Omega(n^{\epsilon})}$ steps.*

We will later revisit this problem in Sect. 4 as the groundwork for designing an efficient version of the (1 + 1) EA that incorporates the structure of the propositional formula under an assignment to the mutation and selection operators.

3.2 Satisfiable Horn Formulas

Another easy to solve subclass of propositional satisfiability is *Horn satisfiability* in which one is interested in determining the satisfiability of Horn formulas. A Horn formula is a propositional formula in conjunctive normal form such that each clause is a Horn clause. A Horn clause is a disjunctive clause containing at most one positive literal. Horn satisfiability is interesting from our perspective because it can be considered the hardest of all "easy problems" in the sense that it is complete for P [4].

Similar to the approach of Sect. 2, we construct a class of pathological Horn formulas on variables that are provably difficult to optimize by the (1 + 1) EA. This class of formulas is close to the set of 2-CNF formulas in the sense that they contain many clauses of length two. However, since they also contain a significant proportion of very long clauses, a Horn formula cannot be directly transformed to a 2-CNF formula in the same manner as with the transformation of bipartite graphs in Sect. 3.1. Nevertheless, we will show in Sect. 4 that a (1 + 1) EA that has been modified to be efficient on 2-CNF formulas will also perform well for this class of Horn formulas.

We define a *ring* formula $\mathcal{R}$ on a set of r variables $\{v_1, v_2, \ldots, v_r\}$ and $2r+1$ clauses as follows.

$$\begin{aligned}
\mathcal{R} &= \mathcal{R}_a \wedge \mathcal{R}_b \\
\mathcal{R}_a &= \bigwedge_{i=1}^{r} \Big((v_i \vee \neg v_{i+1 \pmod r}) \wedge (\neg v_i \vee v_{i+1 \pmod r}) \Big) \\
\mathcal{R}_b &= (\neg v_1 \vee \neg v_2 \vee \cdots \vee \neg v_r).
\end{aligned}$$

Denote $V[\mathcal{R}]$ to be the set of variables in $\mathcal{R}$. All $2r$ clauses of $\mathcal{R}_a$ are only satisfied when $v_i = v_j$ for all $v_i, v_j \in V[\mathcal{R}]$. The singleton clause $\mathcal{R}_b$ is satisfied when at least one variable $v_i \in V[\mathcal{R}]$ is set to false.

We construct a pathological formula F_{path} on n variables as follows. Without loss of generality, assume $r = \sqrt{n}$ is an integer and take F_{path} to be the conjunction of $\sqrt{n}$ rings on $\sqrt{n}$ mutually disjoint sets of variables. Clearly, F_{path} is a Horn formula on n variables and $2n + \sqrt{n}$ clauses.

Using the same approach as with 2-CNF formulas, we associate each distinct assignment to the n variables of F_{path} with an element of $\{0,1\}^n$ and consider the $(1+1)$ EA minimizing the pseudo-Boolean function defined in (3) that counts the number of unsatisfied clauses of F_{path} under any given assignment. The main result of this section is stated in the following theorem.

Theorem 4 *With probability superpolynomially close to one, the (*$1+1$*) EA needs* $\Omega(n^{\log n})$ *steps to solve* F_{path}.

To prove Theorem 4 we make use of the following definitions. Let $\mathcal{R}$ be an arbitrary ring of F_{path}. We say a state $x \in \{0,1\}$ is a *metastable state* for $\mathcal{R}$ if $x[v] = 1$ for all $v \in V[\mathcal{R}]$. Similarly, we say x is a *ground state* for $\mathcal{R}$ if $x[v] = 0$ for all $v \in V[\mathcal{R}]$. We call a state x a *low-energy state* for $\mathcal{R}$ if it is either a metastable state for $\mathcal{R}$ or it is a ground state for $\mathcal{R}$. In general, we say x is a *low-energy state* for F_{path} if it is a low-energy state for all its rings. We call a low-energy state for F_{path} the *ground state* for F_{path} if it is a ground state for all of its rings. All remaining low-energy states for F_{path} are called *metastable states* for F_{path}.

Clearly, if x is a ground state for F_{path}, then $f(x) = 0$ and x is the unique satisfying assignment for F_{path}. Our choice of terminology becomes clear from the claim of the following lemma, which formalizes the concept that metastable states are difficult for the $(1+1)$ EA to leave.

Lemma 6 *Consider the (*$1+1$*) EA working on* F_{path}*. If the current state* x *of the (*$1+1$*) EA is metastable for* F_{path}*, then with probability superpolynomially close to one, it needs* $\Omega(n^{\log n})$ *additional steps to reach the ground state.*

Proof If the current state is metastable, in order to make an improving move, the $(1+1)$ EA must change all the bits of at least one ring at once. The waiting time until any particular r bits flip under uniform mutation is geometrically distributed with success probability

$$p = \left(\frac{1}{n}\right)^r \left(1 - \frac{1}{n}\right)^{n-r} \le \frac{1}{n^r}.$$

The probability that the EA needs strictly fewer than t steps until a success is $1 - (1-p)^t \le t/n^r$ where we have applied Bernoulli's inequality. Thus the probability it takes at least $t = n^{\log n}$ steps is bounded below by $1 - n^{-r+\log n}$. Setting $r = \sqrt{n}$ completes the proof. □

Lemma 7 *Let $\mathcal{R}$ be an arbitrary ring in F_{path}. Let T be the first point in time such that $x^{(T)}$ is a low-energy state for F_{path}. Then $x^{(T)}$ is metastable for $\mathcal{R}$ with probability* 1/2.

Proof We identify the symmetries of the search space and show that the probability of first hitting a low-energy state for which $\mathcal{R}$ is metastable is equivalent to the probability of first hitting a low-energy state that is a ground state for $\mathcal{R}$. We define an absorbing Markov chain $\{X_t : t \in \mathbb{N}_0\}$ on the state space $\{0,1\}^n$. Let $A \subseteq \{0,1\}^n$ be the set of all states that are low-energy for F_{path}. We partition $A := B \cup C$ into the set B of all low-energy states that are metastable for $\mathcal{R}$ and the set C of low-energy states that are ground states for $\mathcal{R}$. We set A to be absorbing and for all $y \notin A$,

$$\Pr(X_t = x \mid X_{t-1} = y) := \left(\frac{1}{n}\right)^{d_H(x,y)} \left(1 - \frac{1}{n}\right)^{n-d_H(x,y)} [f(x) \le f(y)]. \quad (4)$$

where $d_H(x,y)$ denotes Hamming distance between x and y and $[\cdot]$ is the Iverson bracket. In other words, the dynamics of the Markov chain are equivalent to the (1 + 1) EA until an absorbing state is reached.

We now argue that the hitting probabilities of B and C are equal. Let σ be the unique involutory permutation $\sigma\colon \{0,1\}^n \to \{0,1\}^n$ such that for $y = \sigma(x)$,

$$y[v] = \begin{cases} 1 - x[v] & \text{if } v \in V[\mathcal{R}], \\ x[v] & \text{otherwise.} \end{cases}$$

We argue that σ is a symmetry of the Markov chain in the following sense.

$$x \in B \iff \sigma(x) \in C, \quad (5)$$
$$x \in C \iff \sigma(x) \in B, \quad (6)$$
$$\Pr(X_t = x \mid X_0 = y) = \Pr(X_t = \sigma(x) \mid X_0 = \sigma(y)). \quad (7)$$

Identities (5) and (6) are obvious from the definition of σ. To prove Identity (7), note that all variables that appear in the clauses of F_{path} other than $\mathcal{R}$ are fixed by σ. Moreover, if a clause is unsatisfied (satisfied) in $\mathcal{R}_a$, it remains unsatisfied (satisfied) under σ. Finally, as long as x and $\sigma(x)$ are not in A, then $\mathcal{R}_b$ is satisfied under both x and $\sigma(x)$ since both must contain at least one false variable and at least one true variable. Thus the count of unsatisfied clauses in $\mathcal{R}$ are invariant under σ so

$f(y) - f(x) = f(\sigma(y)) - f(\sigma(x))$. By this fact and the fact that σ preserves Hamming distances we see from the definition in (4),

$$\Pr(X_t = x \mid X_{t-1} = y) = \Pr(X_t = \sigma(x) \mid X_{t-1} = \sigma(y))$$

and (7) follows directly from this by induction on powers of the Markov transition matrix.

Let $y \in \{0, 1\}^n$ be arbitrary and let $h^B(y)$ and $h^C(y)$ be the absorption probabilities for B and C respectively. Since $X_t \in B \implies X_{t+1} \in B$,

$$\begin{aligned} h^B(y) &= \lim_{t\to\infty} \Pr(X_t \in B \mid X_0 = y) \\ &= \lim_{t\to\infty} \sum_{x\in B} \Pr(X_t = x \mid X_0 = y) \\ &= \lim_{t\to\infty} \sum_{\sigma(x)\in C} \Pr(X_t = \sigma(x) \mid X_0 = \sigma(y)) && \text{by (7)} \\ &= \lim_{t\to\infty} \Pr(X_t \in C \mid X_0 = \sigma(y)) \\ &= h^C(\sigma(y)). \end{aligned}$$

Since the (1+1) EA is initialized with a uniform random string, we need to calculate the hitting probability for B and C under the uniform distribution. In particular, for any $y \in \{0, 1\}^n$, we have $\Pr(X_0 = y) = 2^{-n}$ so the uniform distributed hitting probability for B is

$$\sum_{y\in S} 2^{-n} h^B(y) = \sum_{y\in S} 2^{-n} h^C(\sigma(y)),$$

and is therefore equivalent to the uniform distributed hitting probability for C. Because the Markov chain is absorbing (clearly every transient state can reach a state in A in a finite number of steps) it hits $A = B \cup C$ almost surely, and the proof is complete. □

The proof of the time bound for the (1 + 1) EA is now straightforward.

Proof of Theorem 4 Let T be the first point in time the (1 + 1) EA finds a low-energy state $x^{(T)}$ for F_{path}. By Lemma 7, $x^{(T)}$ is the ground state for F_{path} with probability $(1 - 1/2)^{\sqrt{n}}$. Thus with probability $1 - 2^{-\sqrt{n}}$, $x^{(T)}$ is a metastable state for F_{path} and the time bound follows from Lemma 6. □

4 Designing an Efficient EA

In this section we show how to slightly modify the evolutionary algorithm listed in Algorithm 1 to run in expected polynomial time on every bipartite graph, and every satisfiable 2-CNF formula. We also show that the class of satisfiable Horn formulas introduced in Sect. 3.2 can be solved in polynomial time by this modification. To achieve this, we leave the black-box setting and introduce problem-specific information into the runtime environment of the evolutionary algorithm. We focus on the case of satisfiable 2-CNF formulas and then implicitly rely on the equivalence stated in Sect. 3.1. We then prove the result extends also to the pathological formulas constructed in Sect. 3.2 that are not covered by the reduction.

We modify Algorithm 1 in two ways. First, we change the representation slightly so that in each iteration the algorithm works on a subformula induced by the unsatisfied clauses. Second, we employ a nonuniform mutation rate that takes into consideration the number of unsatisfied clauses in which each variable appears (cf. Algorithm 2). Given a complete assignment to all n variables of a satisfiable 2-CNF formula, if an unsatisfied clause C exists, choosing a variable uniformly at random from C and negating it results in an assignment that is closer to a satisfying assignment with probability at least $1/2$. This is the insight behind the conflict-directed random walk algorithm discovered independently by Papadimitriou [21] and McDiarmid [16]. This algorithm iteratively flips a variable uniformly at random from an arbitrary unsatisfied clause until a satisfying assignment is found. As long as a solution exists, the expected runtime of the conflict-directed random walk is quadratic in n.

The modifications to the (1 + 1) EA that result in Algorithm 2 yield a process that behaves closer to the Papadimitriou–McDiarmid algorithm and less like a traditional evolutionary algorithm (a close look reveals that the selection step is even redundant). Moreover, our runtime bound is a $\sqrt{m}$-factor slower than Papadimitriou–McDiarmid for reasons we discuss at the end of the section. Nevertheless, we consider these modifications in the context of tailored search operators [6,14,22]. Specifically, we want to show that even small modifications that incorporate problem-specific knowledge into the representation and variation operators can close a superpolynomial runtime gap.

Let F be a propositional formula with m clauses and n variables. For any assignment $x \in \{0,1\}^n$, we denote the unsatisfied subformula induced by x as

$$F[x] = \{C \in F : C \text{ is not satisfied under } x\}.$$

Let $\gamma_i(x)$ count the clauses in $F[x]$ that contain the i-th variable v_i. For the fitness function, we use a subscript to denote the formula, e.g., $f_{F[x]}(y)$ is the number of clauses in the induced subformula $F[x]$ that are not satisfied by y.

Algorithm 2: Modified (1 + 1) EA.

```
Choose $x \in \{0,1\}^n$ uniformly at random;
**repeat** *forever*
    $y \leftarrow x$;
    Flip each bit $y[i]$ with probability $\frac{\gamma_i(x)}{|F[x]|+1}$;
    **if** $f_{F[x]}(y) \leq f_{F[x]}(x)$ **then** $x \leftarrow y$
```

Theorem 5 *Let F be a satisfiable 2-CNF formula on n variables with m pairwise distinct clauses. Algorithm 2 finds a satisfying assignment for F in expected time bounded by $O(n^2\sqrt{m})$.*

Proof Let $x^\star$ be a satisfying assignment to F. We examine the stochastic process $\{X_t : t \in \mathbb{N}_0\}$ on $\{0, 1, \ldots, n\}$ where

$$X_t = \begin{cases} |\{x^{(t)}[i] \neq x^\star[i] : i \in \{1, \ldots, n\}\}| & \text{if } F[x^{(t)}] \neq \emptyset; \\ 0 & \text{otherwise.} \end{cases}$$

Thus X_t is zero only if $x^{(t)}$ is a satisfying assignment, otherwise X_t is the Hamming distance between $x^\star$ and the candidate solution of the (1 + 1) EA in iteration t. Let $T \in \{0, 1, 2, \ldots\} \cup \{\infty\}$ be a random variable defined as $T := \inf\{t \in \mathbb{N}_0 : X_t = 0\}$. T therefore measures the number of iterations until the (1 + 1) EA finds any satisfying assignment. In the following, we make the implicit assumption that no other satisfying assignment is found before $x^\star$, which could only result in a faster process.

Denote the filtration $\mathscr{F}_{t-1} := X_0, \ldots, X_{t-1}$. Let $\Delta_t = X_t - X_{t-1}$. We first show that there exists a $\sigma^2 = \Omega(1/\sqrt{m})$ such that $\mathrm{E}(\Delta_t^2 \mid \mathscr{F}_{t-1}) \geq \sigma^2$. We will later apply this fact to derive an asymptotic bound on the expectation of T. For any $t > 0$, let $x = x^{(t-1)}$ be the candidate solution during iteration $t - 1$ of the (1 + 1) EA. Note that the selection operation in Algorithm 2 is actually superfluous since it is impossible to generate a disimproving move with respect to $F[x]$. Thus Δ_t measures the change in Hamming distance to $x^\star$, provided that no other satisfying assignment has been found. Let $Z_i \in \{0, 1\}$ for $i \in \{1, \ldots, n\}$ be the set of Bernoulli random variables that take on the value 1 if and only if the i-th bit of x flips. Thus, letting n' denote the number of distinct variables in $F[x]$ and m' the number of clauses in $F[x]$,

$$\Delta_t = \sum_{i=1}^{n'} a_i Z_i, \qquad \text{where } a_i = \begin{cases} -1 & \text{if } x^{(t-1)}[i] \neq x^\star[i]; \\ 1 & \text{otherwise.} \end{cases}$$

Since each bit is flipped independently, we have

$$\mathrm{E}\left(\Delta_t^2 \mid \mathscr{F}_{t-1}\right) \geq \mathrm{Var}(\Delta_t) = \sum_{i=1}^{n'} \mathrm{Var}(a_i Z_i) = \sum_{i=1}^{n'} a_i^2 \mathrm{Var}(Z_i)\,.$$

In the final step, we use the fact that the a_i depend only on $x^{(t-1)}$ and $x^\star$, and so given $\mathscr{F}_{t-1}$, they are fixed (not random variables) in this context. Now since $a_i \in \{-1, 1\}$ for all $i \in \{1, \ldots, n'\}$,

$$\begin{aligned} \mathrm{Var}(\Delta_t) = \sum_{i=1}^{n'} \mathrm{Var}(Z_i) = \sum_{i=1}^{n'} \frac{\gamma_i(x)}{m'+1}\left(1 - \frac{\gamma_i(x)}{m'+1}\right) &\geq \frac{n'}{m'+1}\left(1 - \frac{1}{m'+1}\right) \\ &\geq \frac{n'}{2(m'+1)}. \end{aligned}$$

In any set of m' distinct clauses, there must be at least $\sqrt{m'}/2$ distinct variables. Setting $\sigma^2 = \sqrt{m'}/(4m' + 4) = \Omega(1/\sqrt{m})$ since $m' \leq m$, we have the claimed bound on $\mathrm{E}(\Delta_t^2 \mid \mathscr{F}_{t-1})$.

We now show that, using the modified mutation and selection operators of Algorithm 2,

$$\mathrm{E}\,(\Delta_t \mid \mathscr{F}_{t-1}) \leq 0. \tag{8}$$

Each unsatisfied clause C in $F[x]$ contains a literal ℓ that *must* be set incorrectly since C is satisfied by $x^\star$. Let S be the set of variables in $F[x]$ corresponding to such literals. Let R denote the set of remaining variables in $F[x]$. Flipping any variable in S moves the solution closer to $x^\star$, whereas flipping a variable in R moves the solution further from $x^\star$. Thus we have

$$\Delta_t \leq |\{\text{bits in } R \text{ that flip}\}| - |\{\text{bits in } S \text{ that flip}\}|.$$

By definition, each clause contains at least one variable from S and the sum of $\gamma_i(x)$ over $i \in S$ (resp., $i \in R$) counts at least half (resp., at most half) of the $2m'$ literals in $F[x]$. Thus,

$$\mathrm{E}\,(|\,\{\text{bits in } S \text{ that flip}\}\,|) = \sum_{i \in S} \gamma_i(x) \,/\, \left(m' + 1\right) \geq \frac{m'}{(m' + 1)}$$
$$\mathrm{E}\,(|\,\{\text{bits in } R \text{ that flip}\}\,|) = \sum_{i \in R} \gamma_i(x) \,/\, \left(m' + 1\right) \leq \frac{m'}{(m' + 1)},$$

and Inequality (8) holds. We define the random variable $Y_t = X_t^2 - 2nX_t - t\sigma^2$.

$$\mathrm{E}(Y_t \mid \mathscr{F}_{t-1}) = \mathrm{E}\left((X_{t-1} + \Delta_t)^2 - 2n\,(X_{t-1} + \Delta_t) - t\sigma^2 \mid \mathscr{F}_{t-1}\right).$$

Using the fact that any function g, $\mathrm{E}(g(X_{t-1}) \mid \mathscr{F}_{t-1}) = g(X_{t-1})$, and by linearity of expectation,

$$\begin{aligned}\mathrm{E}\,(Y_t \mid \mathscr{F}_{t-1}) &= X_{t-1}^2 - 2nX_{t-1} - t\sigma^2 + (2X_{t-1} - 2n)\mathrm{E}\,(\Delta_t \mid \mathscr{F}_{t-1}) \\ &\quad + \mathrm{E}\left(\Delta_t^2 \mid \mathscr{F}_{t-1}\right) \\ &= Y_{t-1} - \sigma^2 + (2X_{t-1} - 2n)\,\mathrm{E}\,(\Delta_t \mid \mathscr{F}_{t-1}) + \mathrm{E}\left(\Delta_t^2 \mid \mathscr{F}_{t-1}\right) \\ &\geq Y_{t-1}.\end{aligned}$$

The final inequality comes from the fact that $(2X_{t-1} - 2n)\mathrm{E}(\Delta_t \mid \mathscr{F}_{t-1}) \geq 0$, taking $X_{t-1} \leq n$ together with Inequality (8), and from $\mathrm{E}(\Delta_t^2 \mid \mathscr{F}_{t-1}) - \sigma^2 \geq 0$ by our previous bound on $\mathrm{E}(\Delta_t^2 \mid \mathscr{F}_{t-1})$. Therefore, the sequence $\{Y_t\}$ is a submartingale with respect to the sequence $\{X_t\}$, and by the Optional Stopping Theorem [27, Theorem 10.10] $\mathrm{E}(Y_T) \geq \mathrm{E}(Y_0)$. Therefore,

$$\mathrm{E}\left(X_T^2\right) - 2n\mathrm{E}(X_T) - \sigma^2 E(T) \geq \mathrm{E}\left(X_0^2 - 2nX_0\right)$$

and since $X_T^2 = X_T = 0$ and $0 \leq X_0 \leq n$,

$$\mathrm{E}(T) \leq \mathrm{E}\left(2nX_0 - X_0^2\right)/\sigma^2 \leq n^2/\sigma^2 = O\left(n^2\sqrt{m}\right),$$

which follows by the asymptotic bound on σ^2. □

Thus, the modified (1+1) EA can solve every satisfiable 2-CNF formula in expected polynomial time. Moreover, by Lemma 5, the modified algorithm can also find the 2-coloring of any bipartite graph in expected polynomial time. The following theorem extends the polynomial time bound of the modified (1 + 1) EA to also cover the pathological Horn formulas of the previous section.

Theorem 6 *Let F_{path} be a pathological Horn formula on n variables and $2n + \sqrt{n}$ clauses as constructed in Sect. 3.2. Algorithm 2 finds a satisfying assignment for F_{path} in expected time bounded by $O(n^3)$.*

Proof We first observe that at any time during the execution of the modified (1+1) EA, if $x^{(t)}$ is a ground state for a ring $\mathcal{R}$, then for all $s \geq t$, $x^{(s)}$ is also a ground state for $\mathcal{R}$. This follows since $\gamma_i\left(x^{(t)}\right) = 0$ for all $i \in V[\mathcal{R}]$, and therefore the probability of mutation flipping a bit corresponding to any variable in $\mathcal{R}$ is zero. Since this remains invariant for all times greater than t, it must be the case that once the (1 + 1) EA finds a ground state for a ring, it remains in a ground state indefinitely.

We therefore only need to estimate the expected time until a ground state is found for an arbitrary ring. Let $\mathcal{R}$ be an arbitrary ring in F_{path} and let $x^\star$ be a ground state for $\mathcal{R}$. We examine the stochastic process $\{X_t : t \in \mathbb{N}_0\}$ on $\{0, 1, \ldots, \sqrt{n}\}$ where

$$X_t = \left|\left\{x^{(t)}[v] \neq 0 : v \in V[\mathcal{R}]\right\}\right|.$$

Thus X_t is zero only if $x^{(t)}$ is a ground state for $\mathcal{R}$. As before, let $T \in \{0, 1, 2, \ldots\} \cup \{\infty\}$ be the random variable $T := \inf\{t \in \mathbb{N}_0 : X_t = 0\}$ and let $\mathscr{F}_{t-1} := X_0, \ldots, X_{t-1}$. Let $\Delta_t = X_t - X_{t-1}$.

Letting $F[x]$ denote the clauses of $\mathcal{R}$ that are not satisfied by x, using the same argumentation as in the proof of Theorem 5, it is clear that $\mathrm{E}(\Delta_t^2 \mid \mathscr{F}_{t-1}) = \Omega(1/\sqrt{m}) = \Omega(1/\sqrt{n})$.

Note that if $x = x^{(t-1)}$ is metastable for $\mathcal{R}$, $\mathrm{E}(\Delta_t \mid \mathscr{F}_{t-1}) \leq 0$, since $x[v] = 1$ for every $v \in V[\mathcal{R}]$. Therefore, we assume that x is not metastable. In this case, every unsatisfied clause of $\mathcal{R}$ is a 2-CNF clause in $\mathcal{R}_a$. Let S and R be the set of variables in $F[x]$ that are set to one (zero, respectively). Since x is not metastable, only clauses from $\mathcal{R}_a$ are unsatisfied by x and hence appear in $F[x]$. Moreover, each unsatisfied clause of $\mathcal{R}_a$ contains exactly one true variable and exactly one false variable. It follows that every variable in S appears exactly once in each clause of $F[x]$ and every variable in R appears in exactly one clause of $F[x]$. Thus we have

$$\sum_{i \in S} \gamma_i(x) = |F[x]| = \sum_{i \in R} \gamma_i(x),$$

and so in this case $\mathrm{E}(\Delta_t \mid \mathscr{F}_{t-1}) = 0$. The remainder of the proof is then identical to the proof of Theorem 5, and the bound on the expected time follows by setting $m = \Theta(n)$ and applying the argument sequentially to each of the $\sqrt{n}$ rings of F_{path} until they have all reached their ground state. □

We point out that the modified (1 + 1) EA is not necessarily a good practical algorithm. First, as mentioned in the proof of Theorem 5, the selection step is not even necessary since disimproving offspring technically cannot be created. Furthermore, the bound on the runtime is a factor of $\sqrt{m}$ worse than the Papadimitriou–McDiarmid conflict-directed random walk algorithm for 2-SAT. This factor arises because the probability of flipping a bit depends now on m so the variance of each mutation operation can no longer be bounded below by a positive constant. However, the goal of this section was simply to demonstrate that a slight modification to an evolutionary algorithm that allows it to use a small amount of domain knowledge can mean the difference between polynomial and superpolynomial performance.

5 Conclusion

Understanding the influence of problem structure on algorithm performance is important to the design and analysis of algorithms. In this paper, we have rigorously studied the performance of the (1 + 1) EA on three combinatorial problems that are known to be solvable in polynomial time. For each problem, we identified structural aspects that make it difficult to solve by the (1 + 1) EA. We showed there are problem instances in which simple symmetries lead to metastable states that lie in large basins of attraction for the dynamics of the algorithm. The trajectory of the (1 + 1) EA is very likely to enter these basins, but unlikely to escape them in reasonable time. Consequently, on such instances, with probability $1 - o(1)$, the evolutionary algorithm needs at least a superpolynomial number of iterations to find any optimal solution. We then showed how to modify the (1 + 1) EA to guarantee expected polynomial-time behavior on all of the presented problems.

Acknowledgments The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under Grant Agreement No. 618091 (SAGE), and from the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under Grant No. FA9550-11-1-0088. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

1. Apsvall, B., Plass, M.F., Tarjan, R.E.: A linear time algorithm for testing the truth of certain quantified Boolean formulas. Inf. Process. Lett. **8**(3), 121–123 (1979)
2. Auger, A., Doerr, B. (eds.): Theory of Randomized Search Heuristics: Foundations and Recent Developments. World Scientific, Singapore (2011)
3. Briest, P., Brockhoff, D., Degener, B., Englert, M., Gunia, C., Heering, O., Jansen, T., Leifhelm, M., Plociennik, K., Röglin, H., Schweer, A., Sudholt, D., Tannenbaum, S., Wegener, I.: The Ising model: simple evolutionary algorithms as adaptation schemes. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.P. (eds.) Parallel

Problem Solving from Nature (PPSN VIII). Lecture Notes in Computer Science, vol. 3242, pp. 31–40. Springer, Heidelberg (2004)

4. Cook, S., Nguyen, P.: Logical Foundations of Proof Complexity. Cambridge University Press, Cambridge (2010)
5. Doerr, B., Doerr, C., Ebel, F.: Lessons from the black-box: fast crossover-based genetic algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 781–788. ACM (2013)
6. Doerr, B., Hebbinghaus, N., Neumann, F.: Speeding up evolutionary algorithms through asymmetric mutation operators. Evol. Comput. **15**(4), 401–410 (2007)
7. Doerr, B., Johannsen, D., Winzen, C.: Multiplicative drift analysis. Algorithmica **64**(4), 673–697 (2012)
8. Droste, S., Jansen, T., Wegener, I.: Upper and lower bounds for randomized search heuristics in black-box optimization. Theory Comput. Syst. **39**(4), 525–544 (2006)
9. Feller, W.: Generalization of a probability limit theorem of Cramér. Trans. Am. Math. Soc. **54**(3), 361–372 (1943)
10. Fischer, S.: A polynomial upper bound for a mutation-based algorithm on the two-dimensional Ising model. In: Deb, K. (ed.) Proceedings of the Genetic and Evolutionary Computation (GECCO). Lecture Notes in Computer Science, vol. 3102, pp. 1100–1112. Springer, Heidelberg (2004)
11. Fischer, S., Wegener, I.: The one-dimensional Ising model: mutation versus recombination. Theor. Comput. Sci. **344**, 208–225 (2005)
12. Giel, O., Wegener, I.: Evolutionary algorithms and the maximum matching problem. In: Alt, H., Habib, M. (eds.) Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Computer Science, vol. 2607, pp. 415–426. Springer (2003)
13. Hoyweghen, C.V., Goldberg, D.E., Naudts, B.: From twomax to the Ising model: easy and hard symmetrical problems. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 626–633. Morgan Kaufmann Publishers Inc. (2002)
14. Jansen, T., Sudholt, D.: Analysis of an asymmetric mutation operator. Evol. Comput. **18**(1), 1–26 (2010)
15. Matoušek, J., Vondrák, J.: The probabilistic method (lecture notes) (2008). http://kam.mff.cuni.cz/~matousek/prob-ln.ps.gz
16. McDiarmid, C.: A random recolouring method for graphs and hypergraphs. Comb. Probab. Comput. **2**, 363–365 (1993)
17. Neumann, F., Reichel, J., Skutella, M.: Computing minimum cuts by randomized search heuristics. Algorithmica **59**(3), 323–342 (2011)
18. Neumann, F., Witt, C.: Bioinspired Computation in Combinatorial Optimization: Algorithms and their Computational Complexity. Springer, Berlin (2010)
19. Oliveto, P.S., Witt, C.: Simplified drift analysis for proving lower bounds in evolutionary computation. Algorithmica **59**(3), 369–386 (2011)
20. Oliveto, P.S., Witt, C.: Erratum: Simplified drift analysis for proving lower bounds in evolutionary computation (2012) arXiv:1211.7184 [cs.NE]
21. Papadimitriou, C.H., Schäffer, A.A., Yannakakis, M.: On the complexity of local search. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing (STOC), pp. 438–445. ACM (1990)
22. Raidl, G.R., Koller, G., Julstrom, B.A.: Biased mutation operators for subgraph-selection problems. IEEE Trans. Evol. Comput. **10**(2), 145–156 (2006)
23. Schaefer, T.J.: The complexity of satisfiability problems. In: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing (STOC), pp. 216–226. ACM (1978)
24. Sudholt, D.: Crossover is provably essential for the Ising model on trees. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 1161–1167. ACM (2005)
25. Sutton, A.M.: Superpolynomial lower bounds for the $(1+1)$ EA on some easy combinatorial problems. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 1431–1438. ACM (2014)
26. Van Hoyweghen, C., Naudts, B., Goldberg, D.E.: Spin-flip symmetry and synchronization. Evol. Comput. **10**(4), 317–344 (2002)
27. Williams, D.: Probability with Martingales. Cambridge Mathematical Textbooks. Cambridge University Press, Cambridge (1991)
28. Witt, C.: Tight bounds on the optimization time of a randomized search heuristic on linear functions. Comb. Probab. Comput. **22**(2), 294–318 (2013)