UNIVERSITY OF DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

COLLABORATIVE RESEARCH CENTER 531

Design and Management of Complex Technical Processes and Systems by means of Computational Intelligence Methods

Analysis of the (1+1) EA for a Dynamically Bitwise Changing OneMax

Stefan Droste

No. CI-150/03

Technical Report ISSN 1433-3325 April 2003

Secretary of the SFB 531 · University of Dortmund · Dept. of Computer Science/XI
44221 Dortmund · Germany

This work is a product of the Collaborative Research Center 531, "Computational Intelligence," at the University of Dortmund and was printed with financial support of the Deutsche Forschungsgemeinschaft.

Analysis of the (1+1) EA for a Dynamically Bitwise Changing OneMax

Stefan Droste*

LS Informatik 2, Universität Dortmund, 44221 Dortmund, Germany
stefan.droste@udo.edu

Abstract. Although evolutionary algorithms (EAs) are often successfully used for the optimization of dynamically changing objective function, there are only very few theoretical results for EAs in this scenario. In this paper we analyze the (1+1) EA for a dynamically changing OneMax, whose target bit string changes bitwise, i. e. possibly by more than one bit in a step. We compute the movement rate of the target bit string resulting in a polynomial expected first hitting time of the (1+1) EA asymptotically exactly. This strengthens a previous result, where the dynamically changing OneMax changed only at most one bit at a time.

1 Introduction

Evolutionary Algorithms (EAs) optimize objective functions heuristically by following principles of natural evolution (see [BFM97] for an in-depth coverage). A great variety of EAs is successfully used in practice and the experimental knowledge about EAs is immense. In comparison, theory of EAs is still in its beginnings. Many results are based on empirical observations, precluding generalizations. Other results, like the basic forms of the schema theorem ([Hol75]) are limited to the short-time behaviour of EAs. Such results cannot be carried over to other dimensions of the search space, even if the objective function is fixed. This can only be achieved by a rigorous mathematical analysis showing how the first hitting time of the EA (or any other measure of interest) depends on the dimension of the search space.

As there are no theoretical foundations, the EAs and objective functions rigorously analyzed so far are simple in comparison to those used in practice. One of the most thoroughly examined EAs is the (1+1) EA which was analyzed by many different researchers (e. g., see [Müh92], [Rud97], or [DJW02]). The techniques developed while analyzing this simple-structured EA have led to analyses of more complex EAs using crossover and populations (see [JW99] or [JW01]), making it a good starting point for the analysis of dynamic EAs.

So far theoretical analyses of EAs almost completely focus on static optimization problems, i. e. objective functions that do not change over time. Although different kinds of EAs are successfully used for a broad range of dynamical optimization problems (see [Bra01] for an overview), there are only very few theoretical results. Most investigations are limited to experimental work or heuristic

* This research was partly supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Computational Intelligence" (531).

approximations. A first more theoretical approach is [SD99], where the transition probabilities of a (1+1) EA for a dynamic variant of the OneMax-problem are numerically evaluated and compared to empirical data, but not analyzed mathematically.

In this paper, we strengthen the first rigorous mathematical analysis of an EA for a dynamic optimization problem presented in [Dro02]. There the standard (1+1) EA with a mutation probability of $1/n$ is analyzed on a dynamic variant of OneMax, where in every step with probability p' exactly one uniformly chosen bit of the target bit string changes. Hence, in contrast to the static case, the Hamming distance of the current search point to the target bit string can increase by one. It is shown in [Dro02] that the expected first hitting time of the (1+1) EA is polynomial if and only if p' is asymptotically of growth rate at most $\log(n)/n$, i. e. $O(\log(n)/n)$ (see [MR95] for the notation of growth rates).

Here, we look at the more interesting case of changing the target bit string by the same type of random process as the current search point of the (1+1) EA, i. e. each bit is flipped with probability p'. We prove that the expected first hitting time of the (1+1) EA is with high probability super-polynomial if p' grows faster than $\log(n)/n^2$, i. e. $p' = \omega(\log(n)/n^2)$. In the light of [Dro02] the result may not be surprising, but its analysis is significantly more complicated, because we have to consider a process where the Hamming distance between the current search point and the target bit string can increase by more than one in one step. By showing how such a process can be upper bounded by a random process where the distance increases by at most one in one step we prove the desired result. This bounding technique (although it should not be new, we have not found it in the literature) may be applicable in other settings and therefore be of more general interest.

In the next section we define the (1+1) EA for dynamic objective functions, the dynamic variant of OneMax, and the first hitting time, our measure of efficiency. In Sect. 3 we show that for $p' = \omega(\log(n)/n^2)$ the first hitting time of the (1+1) EA is polynomial only with super-polynomially small probability, implying a super-polynomial expected first hitting time. In Sect. 4 we show how stochastic processes that can move away from the target in one step by more than one can be related to processes where this difference can only increase by one in one step. Using this result we show that the expected first hitting time for $p' = O(\log(n)/n^2)$ is polynomial in n. The paper ends with some conclusions.

2 The (1+1) EA and a Dynamic OneMax

The (1+1) EA is probably the most simple EA for maximizing an objective function $f\colon \{0,1\}^n \to \mathbb{R}$ as it uses only one individual being effected by mutation only. The commonly used mutation operator for bit strings is the bitwise mutation, i. e. each bit is flipped independently with probability $p \in [0,1]$. In most cases p is $1/n$. Selection chooses between the old bit string and the mutant, where the mutant is only chosen if its f-value is at least as high as that of the old bit string. This mutation-selection scheme is repeated until a stopping criterion is fulfilled. Since we are interested in the number of steps until an optimum of f is evaluated for the first time, we omit this stopping criterion.

If f is not static but changes over time, we model it by a family $(f_t\colon \{0,1\}^n \to \mathbb{R})_{t\in\mathbb{N}_0}$ of functions, where f_t is the current objective function after t steps of the (1+1) EA. In this situation the (1+1) EA should evaluate even the old bit string after each step, as its fitness might have changed since the last evaluation. So, the (1+1) EA for dynamic optimization problems is defined as follows:

Definition 1 ((1+1) EA for dynamic optimization problems).

1. Set $t := 0$ and choose $x_t \in \{0,1\}^n$ randomly uniformly.
2. Set $x' := x_t$ and flip each bit of x' with probability $1/n$.
3. If $f_t(x') \geq f_t(x_t)$, set $x_{t+1} := x'$, else $x_{t+1} := x_t$.
4. Set $t := t+1$ and go to step 2.

There are a number of different measures for the performance of an algorithm for dynamic optimization, e. g. Hamming distance to the nearest optimum. Here we concentrate on the number T_f of steps until the (1+1) EA for the first time evaluates a point x' where this distance is zero:

Definition 2 (First hitting time of the (1+1) EA).
Let $f = (f_t\colon \{0,1\}^n \to \mathbb{R})_{t\in\mathbb{N}_0}$ be a dynamically changing objective function. The first hitting time T_f *of the (1+1) EA on f is defined as*

$$T_f := \min\{t \in \mathbb{N}_0 \mid f_t(x_t) = \max\{f_t(x) \mid x \in \{0,1\}^n\}\}.$$

One of the first functions the (1+1) EA has been analyzed for in the static case is OneMax. Here the expected first hitting time is $O(n\log(n))$ (see [Müh92] for an approximate analysis and [DJW98] for a rigorous one). OneMax simply counts the number of ones in its argument $x \in \{0,1\}^n$. This can also be interpreted as the number of *matching* bits with the *target bit string* $(1,\ldots,1)$. Hence, maximizing OneMax is equivalent to minimizing the Hamming distance to this target bit string. Using this interpretation, a dynamic OneMax-variant where the target bit string is changed by a random process is obvious. Let $M\colon \{0,1\}^n \to \{0,1\}^n$ represent this random process (formally it is a function $M\colon \{0,1\}^n \times \Omega \to \{0,1\}^n$, where Ω together with a function $P\colon \mathcal{P}(\Omega) \to [0,1]$ forms a probability space, but we omit these technicalities) and $H(x,y)$ denote the Hamming distance between bit strings x and y:

Definition 3 ($\text{OneMax}_{t,M}$).
The family $(\text{OneMax}_{t,M}\colon \{0,1\}^n \to \mathbb{N}_0)_{t\in\mathbb{N}_0}$ of functions is defined as

$$\text{OneMax}_{t,M}(x) := n - H(x, y_t),$$

where $y_0 := (1,\ldots,1)$ and $y_{t+1} := M(y_t)$ for $t \geq 0$.

(As the initial individual x_0 is chosen uniformly at random, initializing y_0 with any other bit string makes no difference.) In [Dro02] the random operator M_1 flips exactly one uniformly chosen bit with probability p' and it is shown that the expected first hitting time of the (1+1) EA is polynomial if and only if $p' = O(\log(n)/n)$. Here we look at a more natural random operator M_n by allowing M_n to flip each bit independently with probability p':

Definition 4 (Bitwise operator M_n).
Let $p' \in [0,1]$. Then the random operator $M_n \colon \{0,1\}^n \to \{0,1\}^n$ is defined for all $y, y' \in \{0,1\}^n$ by:

$$P(M_n(y) = y') := (p')^i \cdot (1-p')^{n-i} \text{ if } H(y,y') = i.$$

The parameter p' is called the *movement rate* of the target bit string as it influences the number of steps in which the target moves and also the number of bits that flip. In the following we will denote $(\text{OneMax}_{t,M_n})_{t\in\mathbb{N}_0}$ shortly by OneMax_D to indicate the dynamic variant of OneMax. To separate more easily between the changes made by the mutation of the (1+1) EA and those of the random operator M_n, we call bits flipped by mutation *mutating bits* and bits changed by M_n *moving bits*. In the next section we show that for $p' = \omega(\log(n)/n^2)$ the (1+1) EA has a super-polynomial expected first hitting time.

3 A Super-Polynomial Lower Bound for Large Movement Rates

In this section we assume that p' grows asymptotically faster than $\log(n)/n^2$, i. e. $p' = \omega(\log(n)/n^2)$. We show that the probability of the (1+1) EA reaching the optimum of OneMax_D in polynomially many steps is super-polynomially small, i. e. at most $1/\beta(n)$, where $\beta(n)$ grows faster than any polynomial in n.

First, we consider $p' = \Omega(1/n^{1-\varepsilon})$ for a constant $\varepsilon > 0$. Then the probability of any specific movement of the target bit string is super-polynomially small, implying that in the last step before reaching the optimum of OneMax_D a super-polynomially unlikely movement of the target bit string has to happen (here we assume that $p' \leq 1/2$ as otherwise more than half of the bits of the target would move on average in each step, making efficient optimization impossible):

Theorem 1. *Let $p' = \Omega(1/n^{1-\varepsilon})$ for a constant $\varepsilon \in]0,1[$. Then the first hitting time of the (1+1) EA on OneMax_D is $\exp(o(n^\varepsilon))$ only with exponentially small probability.*

Proof. We look at an arbitrary run of the (1+1) EA that reaches the optimum after exactly $T > 0$ steps. Then it is necessary that in the last step the movement of the target bit string from y_{T-1} to y_T leads to $y_T = x'$. For $H(y_{T-1}, x') = i$, the probability of such a movement equals $(p')^i \cdot (1-p')^{n-i}$.

If i grows linearly in n, the factor $(p')^i$ is $\exp(-O(n))$. If $n-i$ grows linearly in n, the second factor is asymptotically at most (where $n - i = \beta(n) = \Theta(n)$)

$$\left(1 - \frac{1}{n^{1-\varepsilon}}\right)^{\beta(n)} = \left(1 - \frac{n^\varepsilon}{n}\right)^{(n/n^\varepsilon)\cdot(n^\varepsilon/n)\cdot\beta(n)},$$

which is upper bounded by $\exp(-c \cdot n^\varepsilon)$ for a constant $c > 0$. Since i or $n-i$ grows linearly and the probability of initializing with $x_0 \neq y_0$ is exponentially close to 1, we know that in order to reach the optimum an event with probability at most $\exp(-c \cdot n^\varepsilon)$ has to happen. $\square$

We have to argue more carefully if p' is not that large but still $\omega(\log(n)/n^2)$. Although the proof follows the lines of the super-polynomial lower bound in [Dro02], it must be adjusted to take into account that the target bit string can move by more than one bit at a time step. First, we show the following lemma:

Lemma 1. *Let a Bernoulli experiment with success probability $p \in]0,1]$ be repeated independently $a+b$-times ($a, b \in \mathbb{N}$). The probability that in the first a experiments the number of successes is larger than the number of successes in the last b experiments is at most a/b.*

Proof. Let the random variables A resp. B denote the number of successes in the first a resp. last b experiments and define $Y := A/(B+1)$. Then $E(Y) = pa/(pb+1) \leq a/b$. If and only if the number of successes in the first a experiments is larger than in the last b experiments Y is at least 1. Using the Markov bound ([MR95]) we get $P(Y \geq 1) \leq E(Y) \leq a/b$. □

Theorem 2. *Let $p' = \omega(\log(n)/n^2)$. Then the first hitting time of the (1+1) EA on* OneMax$_D$ *is polynomial only with super-polynomially small probability.*

Proof. Because of Theorem 1 we concentrate on values of p' with $p' \leq 1/n^{6/7}$. Let $\alpha(n) = \min\left\{n/\log(n), p' \cdot n^2/\log(n)\right\}$ be the minimum of $n/\log(n)$ and the factor by which p' grows faster than $\log(n)/n^2$. Then $\lim_{n\to\infty} \alpha(n) = \infty$.

Our first assumption on the (1+1) EA is that the initial bit string contains at most $n - 2G$ matching bits, where G grows slightly faster than $\log(n)$, i. e. $G := \log(n) \cdot \alpha(n)^{4/7}$. As $G = o(n)$, Chernoff bounds ([MR95]) guarantee that initializing with a bit string with at most $n-2G$ matching bits has a probability super-polynomially close to one, i. e. this event is *almost sure*. We show that the (1+1) EA reaches the optimum in polynomially many steps only via a search point whose number of matching bits is between $n-2G$ and $n-G$ almost surely.

Since the mutation rate of the (1+1) EA is $1/n$, almost surely at most $\log(n)$ matching bits are mutated in one step. If the movement rate p' of the target bit string is $O(1/n)$, the number of moving bits of the target bit string in one step is also at most $\log(n)$ almost surely. Hence, in one step the OneMax$_D$-value of the current search point changes by at most $2\log(n)$ almost surely. As G is growing asymptotically faster than $\log(n)$, it is at least $2\log(n)$ for n large enough. Hence, any run of the (1+1) EA with polynomial many steps starting with at most $n-2G$ matching bits reaches the optimum only via a search point with at least $n-2G$ and at most $n-G$ matching bits almost surely.

If $p' = \omega(1/n)$, we have $\alpha(n) = n/\log(n)$. As p' is $O(1/n^{6/7})$ by assumption, Chernoff bounds yield that the number of moving bits in one step is at most $cn^{1/7}$ for a constant $c > 1$ almost surely. Since G is $n^{4/7} \cdot \log(n)^{3/7}$ in this case, i. e. asymptotically larger than the number of moving or mutating bits in one step, we again know that the (1+1) EA reaches the optimum in polynomial time only via a search point with at least $n-2G$ and at most $n-G$ bits almost surely.

Let I_t denote the current state of the (1+1) EA, i. e. the number of matching bits of x_t. The preceding argumentation shows that if state n is reached in polynomially many steps, almost surely two points in time $t_1 < t_2$ exist, where

$I_{t_1} \in \{n-2G,\dots,n-G\}$, $I_t \in \{n-G+1,\dots,n-1\}$ for all $t \in \{t_1+1,\dots,t_2-1\}$, and $I_{t_2} = n$. In the following we show that for every $t = t_2 - t_1 \in \mathbb{N}$ it is super-polynomially unlikely that such a sequence $I_{t_1},\dots,I_{t_2}$ exists (a similar technique is used in [RRS95]).

We exclude the possible, but very unlikely event that the (1+1) EA bridges the gap, i. e. comes from $n-G$ to n ones, using only few but far reaching steps. Therefore, we show that in one step at most $L := \alpha(n)^{1/7}$ non-matching bits flip almost surely. The probability of flipping at least $L+1$ of the at most $2G$ non-matching bits during one step by mutation and/or movement of the target bit string is at most

$$\binom{2G}{L+1} \cdot \left(\frac{1}{n} + p'\right)^{L+1} \leq \left(\frac{4G}{n^{6/7}}\right)^{L+1},$$

which is super-polynomially small as $G \leq n^{4/7}\log(n)^{3/7}$ and L grows with n to infinity. Hence, we can rule out mutations increasing the number of ones by more than L. The error introduced by this assumption has only super-polynomially small probability if we concentrate on polynomially many steps.

Now, we upper resp. lower bound the probability p_i^+ resp. p_i^-, that the number of matching bits increases resp. decreases during one step when the current number of matching bits is i. The number of matching bits decreases if one matching bit moves, but no non-matching bit moves and no bit mutates. Assuming that $p' \leq 1/n$ we can lower bound p_i^- by (where $c > 0$ is a constant):

$$\left(1-(1-p')^i\right) \cdot (1-p')^{n-i} \cdot \left(1-\frac{1}{n}\right)^n \geq i \cdot p' \cdot c \geq \frac{c}{2} \cdot \alpha(n) \cdot \frac{\log(n)}{n}.$$

If p' is larger than $1/n$, there is a constant probability that at least one of the i matching bits, but none of the $n-i$ non-matching bits moves, while no bit mutates. Hence, the inequality $p_i^- \geq (c/2)\cdot\alpha(n)\cdot(\log(n)/n)$ is still valid for an appropriate constant $c > 0$, because $\alpha(n) = n/\log(n)$ for all $p' \geq 1/n$.

For the upper bound p_i^+ we notice that a step increasing the number of matching bits implies that the mutation resp. the movement of the target bit string flips more non-matching bits than matching bits after the movement resp. mutation has occurred. According to Lemma 1 both events have probability at most $(n-i+L)/(i-L)$ because at most L matching bits change during movement resp. mutation by assumption. Consequently, $p_i^+ \leq 2(n-i+L)/(i-L)$. Since i is at least $n - 2G = n - 2\log(n)\alpha(n)^{4/7}$ and $L = o(G)$, we know that for all $n \geq n_0$ for a constant n_0 (i. e. n is *large enough*):

$$p_i^+ \leq 2 \cdot \frac{2G+L}{n-2G-L} \leq 2 \cdot \frac{3G}{n-3G} \leq 12 \cdot \frac{\alpha(n)^{4/7}\log(n)}{n}.$$

Since it is sufficient for us that decreasing the number of matching bits is by a factor $\alpha(n)^{2/7}$ more likely than increasing it, we lower resp. upper bound p_i^- resp. p_i^+ for n large enough by

$$\tilde{p}_i^- := \frac{c}{2} \cdot \frac{\alpha(n)\log(n)}{n} \quad \text{resp.} \quad \tilde{p}_i^+ := \frac{c}{2} \cdot \frac{\alpha(n)^{5/7}\log(n)}{n}.$$

Hence, the assumption that the probabilities of steps decreasing resp. increasing the number of matching bits are exactly $\tilde{p}_i^-$ resp. $\tilde{p}_i^+$ and that every improvement leads us L bits closer to the target while every deterioration only changes one bit, results in a "faster process" (a proof would follow the line of Lemma 2 in the next section). Then the probability that the next mutation changing the number of matching bits (called *effective mutation*) increases the number of matching bits equals

$$\frac{\frac{c}{2} \cdot \frac{\alpha(n)^{5/7} \log(n)}{n}}{\left(\alpha(n) + \alpha(n)^{5/7}\right) \cdot \frac{c}{2} \cdot \frac{\log(n)}{n}} = \frac{\alpha(n)^{5/7}}{\alpha(n) + \alpha(n)^{5/7}} \leq \alpha(n)^{-2/7}.$$

Consequently, the expected number of increasing mutations during t effective mutations is at most $t/\alpha(n)^{2/7}$ and we can follow the argumentation presented in [Dro02]. Again, by assuming that the expected number of increasing mutations is exactly $t/\alpha(n)^{2/7}$ the process can only become faster. To obtain a superpolynomially small upper bound by Chernoff bounds we show that

1. the expected number of increasing mutations grows significantly faster than $\log(n)$, i. e. $t/\alpha(n)^{2/7} = \omega(\log(n))$ and
2. the number of increasing mutations necessary to reach the optimum is by a constant factor larger than the expected number of increasing mutations.

To prove these claims we argue that t, the length of the sequence $I_{t_1}, \ldots, I_{t_2}$, is at least $G/L = \log(n)\alpha(n)^{3/7}$, because every increasing mutation raises the number of matching bits by at most $L = \alpha(n)^{1/7}$. Hence, $t/\alpha(n)^{2/7} = \omega(\log(n))$, i. e. only sequences of some minimum length are candidates for $I_{t_1}, \ldots, I_{t_2}$. The necessary number of increasing mutations can be lower bounded as follows: if t^+ denotes the number of increasing and t^- the number of decreasing mutations of $t = t^+ + t^-$ effective mutations, the number of matching bits increases by $t^+ \cdot L - t^-$. In order to reach the global optimum this value has to be at least G. This is equivalent to

$$t^+ \cdot L - t^- \geq G \iff t^+ \geq \frac{G + t - t^+}{L} \iff t^+ \geq \frac{G + t}{L + 1}.$$

For n large enough we have

$$\frac{G + t}{L + 1} = \frac{\log(n)\alpha(n)^{4/7} + t}{\alpha(n)^{1/7} + 1} \geq 2 \cdot \frac{t}{\alpha(n)^{2/7}}.$$

In other words, to reach the global optimum in t steps the number of increasing mutations necessary is at least by a factor 2 larger than the expected number $t/\alpha(n)^{2/7}$ of increasing mutations. Hence, using Chernoff bounds the probability of a sequence $I_{t_1}, \ldots, I_{t_2}$ of length t with the desired properties is bounded above by $\exp\left(-\frac{t}{\alpha(n)^{2/7}} \cdot \frac{4}{3}\right)$. Because t is at least $\alpha(n)^{3/7} \log(n)$, this is upper bounded by $\exp\left(-\Omega(\alpha(n)^{1/7} \log(n))\right) = n^{-\Omega(\alpha(n)^{1/7})}$.

All in all, we have shown that the (1+1) EA finds the optimum of $\textsc{OneMax}_D$ with $p' = \omega(\log(n)/n^2)$ in polynomially many steps only, if a super-polynomially unlikely event happens. □

4 A Polynomial Upper Bound for Small Movement Rates

We now show that the expected first hitting time of the (1+1) EA for OneMax is polynomial for $p' = O(\log(n)/n^2)$. While doing this we have to analyze a process where the distance from the target (the number of non-matching bits) can increase in one step by more than one. The crucial idea is to show how this process can be replaced by a slower one (i. e. how to *upper bound the process*), where the distance from the target in one step increases by at most one.

In the following we describe the (1+1) EA and other processes by their transition probabilities $p_{i,j}$ ($i, j \in \{0, \dots, n\}$) of changing from a point with i matching bits in one step to a point with j matching bits. As all considered processes are Markovian, i. e. time-independent, and operating on $\{0, \dots, n\}$ the transition probabilities describe them completely. Hence, we denote the process with transition probabilities $p_{i,j}$ shortly by $(p_{\cdot,\cdot})$. The random variable $T_{i,j}$ denotes the first hitting time to come from state i to state j. The names are sometimes adapted suitably, e. g. $\tilde{T}_{i,j}$ denotes the first hitting time of the process $(\tilde{p}_{\cdot,\cdot})$.

The outline of the proof is as follows: firstly, we "cut off" all transitions from states i to states $j \geq i+2$ (we will call processes with $p_{i,j} = 0$ for all i and $j \geq i+2$ briefly *≤1-processes* in the following). This ≤1-process is replaced by one with a stronger tendency towards lower states. Finally, we replace this process by one changing its state from i only to $i-1$, i, or $i+1$ and analyze it by standard methods.

To show that all these steps lead to processes having at least the same expected first hitting time, we need a number of lemmata:

Lemma 2. *Let $(p_{\cdot,\cdot})$ be a process. For the ≤1-process $(\tilde{p}_{\cdot,\cdot})$ defined by*

$$\tilde{p}_{i,j} := \begin{cases} p_{i,j} & \text{if } j < i \text{ or } j = i+1, \\ p_{i,i} + \sum_{j=i+2}^{n} p_{i,j} & \text{if } j = i, \end{cases}$$

we have for all $i \in \{0, \dots, n-1\}$, $k \geq i$ and all $t \in \mathbb{N}_0$: $P(\tilde{T}_{i,n} \geq t) \geq P(T_{k,n} \geq t)$.

Proof. By induction over $t \in \mathbb{N}_0$: for $t = 0$ the statement is trivial. For the step from t to $t+1$, we lower bound $P(\tilde{T}_{i,n} \geq t+1)$ according to two cases:

1. $$k > i: \sum_{j=0}^{i+1} \tilde{p}_{i,j} \cdot P(\tilde{T}_{j,n} \geq t) \geq \sum_{j=0}^{i+1} \tilde{p}_{i,j} \cdot P(T_{k,n} \geq t) \geq P(T_{k,n} \geq t+1).$$

2. $$\begin{aligned} k = i: \quad & \sum_{j=0}^{i-1} \tilde{p}_{i,j} \cdot P(\tilde{T}_{j,n} \geq t) + \tilde{p}_{i,i} \cdot P(\tilde{T}_{i,n} \geq t) + \tilde{p}_{i,i+1} \cdot P(\tilde{T}_{i+1,n} \geq t) \\ & \geq \sum_{j=0}^{i-1} p_{i,j} \cdot P(T_{j,n} \geq t) + p_{i,i} \cdot P(T_{i,n} \geq t) + \sum_{j=i+2}^{n} p_{i,j} \cdot P(T_{j,n} \geq t) \\ & \quad + p_{i,i+1} \cdot P(T_{i+1,n} \geq t) = P(T_{i,n} \geq t+1). \end{aligned}$$

□

Since $E(T_{i,n}) = \sum_{t=1}^{\infty} t \cdot P(T_{i,n} = t) = \sum_{t=1}^{\infty} P(T_{i,n} \geq t)$, Lemma 2 tells us that by "cutting off" all improvements by more than one, the expected first hitting time does not decrease. A $\leq$1-process has two useful properties (for the sake of brevity we omit the easy proofs by induction):

Lemma 3. *Let $(p_{\cdot,\cdot})$ be a $\leq$1-process. Then we have for all $i \in \{0, \ldots, n\}$ and $t \in \mathbb{N}_0$: $P(T_{i,n} \geq t) \geq P(T_{i+1,n} \geq t)$ (implying $E(T_{i,n}) \geq E(T_{i+1,n})$).*

Lemma 4. *Let $(p_{\cdot,\cdot})$ and $(\tilde{p}_{\cdot,\cdot})$ be $\leq$1-processes. If for all $i \in \{0, \ldots, n-1\}$ and all $j < i$ we have $\tilde{p}_{i,j} \geq p_{i,j}$ and $\tilde{p}_{i,i+1} \leq p_{i,i+1}$ then $P(\tilde{T}_{i,n} \geq t) \geq P(T_{i,n} \geq t)$ (implying $E(\tilde{T}_{i,n}) \geq E(T_{i,n})$) for all $i \in \{0, \ldots, n\}$ and $t \in \mathbb{N}_0$.*

In general the expected first hitting time of a $\leq$1-process is difficult to upper bound as it might jump to lower states with high probability. But if this is not too likely we can find a Markov process jumping only from i to $i-1$, i, or $i+1$ (called a $\{-1, 0, 1\}$-*process*) having at least the same expected first hitting time. The key idea is to interpret a number of steps of a given $\{-1, 0, 1\}$-process as a single step of a $\leq$1-process. If we do not "forget" any transitions of the $\{-1, 0, 1\}$-process we get a $\leq$1-process that lower bounds the original $\{-1, 0, 1\}$-process:

Lemma 5. *Let $(p_{\cdot,\cdot})$ resp. $(\tilde{p}_{\cdot,\cdot})$ be a $\leq$1- resp. a $\{-1, 0, 1\}$-process where*

1. $\forall i \in \{1, \ldots, n-1\} : p_{i,0} = \prod_{k=1}^{i} \tilde{p}_{k,k-1}$,
2. $\forall i \in \{1, \ldots, n-1\} : \forall j \in \{1, \ldots, i-1\} : p_{i,j} = \tilde{p}_{j,j} \prod_{k=j+1}^{i} \tilde{p}_{k,k-1}$,
3. $\forall i \in \{0, \ldots, n-1\} : p_{i,i+1} = \tilde{p}_{i,i+1}$.

Then for all $i \in \{0, \ldots, n-1\}$: $E(T_{i,i+1}) \leq E(\tilde{T}_{i,i+1})$.

Proof. Assume that the current state of the process $(\tilde{p}_{\cdot,\cdot})$ is i. Consider the next at most i transitions of $(\tilde{p}_{\cdot,\cdot})$. According to the following cases we split a prefix from this sequence and consider it as a one step of the process $(p_{\cdot,\cdot})$:

- All i transitions decrease the number of the current state by exactly one. This event has probability $\prod_{k=1}^{i} \tilde{p}_{k,k-1}$ and afterwards the process is in state 0.
- The first $i - j \in \{1, \ldots, i-1\}$ transitions decrease the number of the current state by exactly one, while the $(i - j + 1)$-th transition stays in the current state. This event has probability $\tilde{p}_{j,j} \prod_{k=j+1}^{i} \tilde{p}_{k,k-1}$ and afterwards the process is in state j.
- The first transition leads from state i to $i + 1$. This has probability $\tilde{p}_{i,i+1}$ and afterwards the process is in state $i + 1$.
- In all other cases, i. e. when the considered i transitions start with at least one but at most $i-1$ decreasing transitions directly followed by an increasing transition, the process is in state at most i. By assuming that the process is exactly in state i, the expected first hitting time does not increase according to Lemma 3, because the described process is a $\leq$1-process.

These cases show how the $\{-1, 0, 1\}$-process $(\tilde{p}_{\cdot,\cdot})$ can be considered as a $\leq$1-process $(p_{\cdot,\cdot})$ with exactly the transition probabilities given above. By interpreting the considered transition(s) as a single transition (observe that their number varies according to the respective case), we get the claimed lower bound. $\square$

Lemma 5 can hardly be applied directly as it only correlates processes where the transition probabilities fulfill some equalities. Using Lemma 4 we can weaken the conditions of Lemma 5 to inequalities:

Corollary 1. *Let* $(p_{\cdot,\cdot})$ *resp.* $(\tilde{p}_{\cdot,\cdot})$ *be a* $\leq$*1-process resp. a* $\{-1,0,1\}$*-process on* $\{0,\dots,n\}$*. If the following conditions hold*

1. $\forall i \in \{1,\dots,n-1\} : p_{i,0} \leq \prod_{k=1}^{i} \tilde{p}_{k,k-1}$,
2. $\forall i \in \{1,\dots,n-1\} : \forall j \in \{1,\dots,i-1\} : p_{i,j} \leq \tilde{p}_{j,j} \prod_{k=j+1}^{i} \tilde{p}_{k,k-1}$,
3. $\forall i \in \{0,\dots,n-1\} : p_{i,i+1} \geq \tilde{p}_{i,i+1}$,

then $E(T_{i,n}) \leq E(\tilde{T}_{i,n})$ *for all* $i \in \{0,\dots,n\}$*.*

Now we can upper bound the expected first hitting time of the (1+1) EA by repeatedly replacing the original process by simpler ones and using the preceding results to show that the new process is not faster than the former one. As we are interested in asymptotical results, the processes may be defined in a way that only makes sense for n large enough (e. g. some probabilities may be negative for small n). At the end we come up with a $\{-1,0,1\}$-process we can analyze by standard methods.

Theorem 3. *The expected first hitting time of the (1+1) EA on* ONEMAX$_D$ *with* $p' = O(\log(n)/n^2)$ *is polynomial.*

Proof. 1. Let $(p^1_{\cdot,\cdot})$ denote the random process exactly describing the (1+1) EA on ONEMAX$_D$. According to Lemma 2, for the $\leq$1-process defined by

$$p^2_{i,j} := \begin{cases} p^1_{i,j} & \text{if } j < i \text{ or } j = i+1, \\ p^1_{i,i} + \sum_{j=i+2}^{n} p^1_{i,j} & \text{if } j = i, \end{cases}$$

$E(T^2_{i,n}) \geq E(T^1_{i,n})$ for all $i \in \{0,\dots,n\}$. Hence, from now on we consider the simpler process $(p^2_{\cdot,\cdot})$ instead of $(p^1_{\cdot,\cdot})$.

2. The $\leq$1-process $(p^2_{\cdot,\cdot})$ is replaced by the $\leq$1-process $(p^3_{\cdot,\cdot})$, where for all $i \in \{0,\dots,n-1\}$ and $k \in \{1,\dots,i\}$ (remember that $p' = O(\log(n)/n^2)$):

$$p^3_{i,i-k} := \binom{i}{k} \cdot (p')^k \quad \text{and} \quad p^3_{i,i+1} := \frac{n-i}{n} \cdot \frac{2}{3\exp(1)}$$

($p^3_{i,i}$ is implicitly defined as the "remaining probability" to stay in state i). According to Lemma 4, this process $(p^3_{\cdot,\cdot})$ does not have smaller expected first hitting times than the process $(p^2_{\cdot,\cdot})$, because for all $i \in \{0,\dots,n-1\}$ and $k \in \{1,\dots,i\}$

$$p^2_{i,i-k} = p^1_{i,i-k} \leq \binom{i}{k} \cdot (p')^k = p^3_{i,i-k} \quad \text{and for } n \text{ large enough}$$

$$p^2_{i,i+1} = p^1_{i,i+1} \geq \binom{n-i}{1} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \cdot (1-p')^n \geq p^3_{i,i+1}$$

as $p' = O(\log(n)/n^2)$. Hence, from now on we consider the process $(p^3_{\cdot,\cdot})$.

3. The $\{-1,0,1\}$-process $(p^4_{\cdot,\cdot})$ is defined by (for all $i \in \{0,\dots,n-1\}$)

$$p^4_{i,i-1} := 2 \cdot i \cdot p' \quad \text{and} \quad p^4_{i,i+1} := \frac{n-i}{n} \cdot \frac{1}{2\exp(1)}.$$

(Note that $p^4_{i,i}$ is only implicitly defined as $1 - p^4_{i,i-1} - p^4_{i,i+1}$, which is at least $1/2$ for n large enough, since $p' = O(\log(n)/n^2)$.) Corollary 1 yields that $(p^4_{\cdot,\cdot})$ upper bounds $(p^3_{\cdot,\cdot})$: Condition 1 of Corollary 1 is fulfilled, since $p^3_{i,0} = (p')^i \le \prod_{k=1}^{i} 2kp' = \prod_{k=1}^{i} p^4_{k,k-1}$ for $i \in \{1,\dots,n-1\}$.
Condition 2 holds because for all $i \in \{1,\dots,n-1\}$ and $j \in \{1,\dots,i-1\}$:

$$p^3_{i,j} \le p^4_{j,j} \prod_{k=j+1}^{i} p^4_{k,k-1} \iff \binom{i}{i-j}(p')^{i-j} \le p^4_{j,j} \prod_{k=j+1}^{i} 2kp',$$

which is valid, since $p^4_{j,j} \ge 1/2$ for n large enough. Condition 3 follows directly by $p^3_{i,i+1} > p^4_{i,i+1}$ for all $i < n$. Now the $\{-1,0,1\}$-process $(p^4_{\cdot,\cdot})$ can be analyzed with standard methods. It is well-known (see e. g. [DJW00]) that

$$E(T^4_{i,i+1}) = \sum_{k=0}^{i} \frac{1}{p^4_{k,k+1}} \cdot \prod_{l=k+1}^{i} \frac{p^4_{l,l-1}}{p^4_{l,l+1}}.$$

Filling in our values for $p^4_{i,j}$, we get the following upper bound on $E(T^4_{i,n})$ for $p' \le c \cdot \log(n)/n^2$ (see [Dro02] for a similar calculation):

$$\begin{aligned}
&\sum_{k=0}^{i} \frac{2\exp(1)\cdot n}{n-k} \cdot \prod_{l=k+1}^{i} 2 \cdot l \cdot \frac{c \cdot \log(n)}{n^2} \cdot \frac{2\exp(1)\cdot n}{n-l} \\
&= \sum_{k=0}^{i} \frac{2\exp(1)\cdot n}{n-k} \cdot \left(\frac{4c\exp(1)\log(n)}{n}\right)^{i-k} \cdot \frac{i!}{k!} \cdot \frac{(n-i-1)!}{(n-k-1)!} \\
&\le \frac{2\exp(1)\cdot n}{n-i} \cdot \sum_{k=0}^{i} \left(\frac{4c\exp(1)\log(n)}{n}\right)^{i-k} \cdot \binom{i}{k} \Big/ \binom{n-k-1}{n-i-1} \\
&\le 2\exp(1) \cdot \frac{n}{n-i} \cdot \left(1 + \frac{4c\exp(1)\log(n)}{n}\right)^{i}
\end{aligned}$$

Using the estimation $(1+1/x)^x \le \exp(1)$, $E(T^4_{i,i+1})$ is bounded above by:

$$2\exp(1) \cdot \frac{n}{n-i} \cdot \exp\left(\frac{4c\log(n)\exp(1)}{n} \cdot i\right) \le 2\exp(1) \cdot \frac{n}{n-i} \cdot n^{\frac{4c\exp(1)}{\ln(2)}}.$$

Hence, by linearity of expectation and pessimistically assuming that we initialize in $(0,\dots,0)$,

$$E(T_{\textsc{OneMax}_D}) \le 2\exp(1) \cdot n^{\frac{4c\exp(1)}{\ln(2)}+1} \cdot \sum_{i=1}^{n} \frac{1}{i} = O\left(n^{4c\cdot\frac{\exp(1)}{\ln(2)}+1} \cdot \log(n)\right).$$

$\square$

5 Conclusions

We have analyzed the (1+1) EA on a dynamic variant of OneMax where every bit of the target flips with probability p'. The critical growth rate of p' where the expected first hitting time of the (1+1) EA changes from polynomial to superpolynomial was shown to be $\Theta(\log(n)/n^2)$. Besides this result, the techniques developed may be of more general interest. They show how a Markov process which can move away from the target by more than one in one step can be replaced by a slower Markov process that can move away from the target by at most one in one step (Lemma 5 resp. Corollary 1). This method might help in the analysis of other EAs or randomized processes in general.

Acknowledgements

I thank Jens Jägersküpper, Ingo Wegener, and Carsten Witt for their valuable advice and help while preparing this paper.

References

[BFM97] Th. Bäck, D.B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing, 1997.

[Bra01] J. Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic, 2001.

[DJW98] S. Droste, Th. Jansen, and I. Wegener. A rigorous complexity analysis of the $(1+1)$ EA for linear functions with Boolean inputs. In *Proceedings of ICEC 1998*, pages 499–504, 1998.

[DJW00] S. Droste, Th. Jansen, and I. Wegener. Dynamic parameter control in simple evolutionary algorithms. In *Proceedings of FOGA 2000*, pages 275–294, 2000.

[DJW02] S. Droste, Th. Jansen, and I. Wegener. On the analysis of the $(1+1)$ EA. *Theoretical Computer Science*, (276):51–81, 2002.

[Dro02] S. Droste. Analysis of the (1+1) EA for a dynamically changing OneMax-variant. In *Proceedings of CEC 2002*, pages 55–60, 2002.

[Hol75] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[JW99] Th. Jansen and I. Wegener. On the analysis of evolutionary algorithms – a proof that crossover really can help. In *Proceedings of ESA 1999*, pages 184–193, 1999. LNCS 1643.

[JW01] Th. Jansen and I. Wegener. Real royal road functions – where crossover is provably essential. In *Proceedings of GECCO 2001*, pages 1034–1041, 2001.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Müh92] H. Mühlenbein. How genetic algorithms really work: I. mutation and hill-climbing. In *Proceedings of PPSN 1992*, pages 15–26, 1992.

[RRS95] Y. Rabani, Y. Rabinovich, and A. Sinclair. A computational view of population genetics. In *Proceedings of STOC 1995*, pages 83–92, 1995.

[Rud97] G. Rudolph. *Convergence Properties of Evolutionary Algorithms*. Verlag Dr. Kovač, 1997.

[SD99] S.A. Stanhope and J.M. Daida. (1+1) GA fitness dynamics in a changing environment. In *Proceedings of the CEC 1999*, pages 1851–1858, 1999.