Yugoslav Journal of Operations Research
xx (20xx), Number xx, xxx–xxx
DOI: https://doi.org/10.2298/YJOR250715042E

Research Article

ON RUNTIME OF NON-ELITIST EVOLUTIONARY ALGORITHMS OPTIMIZING FITNESS FUNCTIONS WITH A PLATEAU

Anton EREMEEV
Novosibirsk State University, Sobolev Institute of Mathematics
eremeev@ofim.oscsbras.ru, ORCID: 0000-0001-5289-7874

Received: July 2025 / Accepted: November 2025

Abstract: The expected runtime of non-elitist evolutionary algorithms (EAs), when they are applied to a family of fitness functions Plateau_r on the search space of binary strings of length n is considered asymptotically with unbounded n. The functions of this family have a plateau of second-best fitness in a Hamming ball of a constant radius r around a unique global optimum, while below the plateau the fitness equals the number of one-bits. On one hand, using the level-based theorems, we obtain polynomial upper bounds on the expected runtime for non-elitist EAs with tournament selection, (μ, λ)-selection and proportionate selection. We assume that these EAs are based on an unbiased mutation, in particular, the bitwise mutation. In the case of proportionate selection, the mutation probability is of order $1/n^2$. On the other hand, we show that the EA with fitness proportionate selection is inefficient even to search for approximate solutions if the bitwise mutation is used with the mutation probability greater than $\ln(2)/n$.

Keywords: Evolutionary algorithm, selection, runtime, Plateau, unbiased mutation.

MSC: 35A01.

1. INTRODUCTION

In the present paper, we study the efficiency of non-elitist evolutionary algorithms (EAs) without recombination, applied to optimization problems with a single plateau of constant values of objective function around the unique global optimum. Significance of plateaus analysis is associated with several reasons. Plateaus often occur in combinatorial optimization problems, especially in the unweighted problems [1, 2, 3], in fitness landscapes associated with genetic programming [4], feature selection in machine learning [5] etc. Fitness plateaus caused by gene duplication are one of the key elements of the well-known theory of neutrality in evolutionary biology [6]. A number of artificial benchmarks with built-in plateaus have been introduced in the theory of EAs to demonstrate and compare performance of EAs in different situations [7, 8, 9, 10].

As a measure of efficiency, we consider the expected runtime, i.e., the expected number of objective (or fitness) function evaluations until the optimal solution is reached. This approach is standard for the area of runtime analysis in evolutionary computation. In this area, the number of fitness evaluations is used more frequently than the time complexity in terms of elementary operations because in practical applications of the EAs, the fitness evaluations usually make up the main CPU cost. From the theoretical point of view, the number of fitness evaluations is also convenient because it equals the number of queries to an oracle, if the fitness function is given by an oracle. This assumption is natural if one is interested in analysis of general-purpose heuristics, applicable to a broad range of problems. As alternatives to the expectation of the runtime, one can consider tail bounds for the distribution of EA runtime (see e.g. [11]), or fixed-budget performance (see e.g. [12]), but these approaches are less popular in the runtime analysis of EAs.

We study the EAs without elite individuals, based on the bitwise mutation (with a specified bit flip probability), and more generally, based on the unbiased mutation [13], when the EAs are applied to optimize fitness functions with plateaus of constant fitness. To this end, we consider the Plateau_r function with a plateau of second-best fitness in a ball of radius r around the unique optimum. The goal of this paper is to study the expected runtime of non-elitist EAs, optimizing Plateau_r, asymptotically for growing number of binary variables n, assuming constant parameter r.

It is shown in [7] that the (1 + 1) EA, which is one of the simplest mutation-based evolutionary algorithms, using an unbiased mutation operator (e.g., the bitwise mutation or the one-point mutation) optimizes the Plateau_r function with expected runtime $\frac{n^r(1+o(1))}{r!\Pr(1\leq\xi\leq r)}$, where ξ is a random variable, equal to the number of bits flipped in an application of the mutation operator. This is proved under the condition that mutation flips exactly one bit with probability $\omega(n^{-\frac{1}{2r-2}})$. The most natural special case when this condition is satisfied is when exactly one bit is flipped with probability $\Omega(1)$.

The authors of [14] analyze the runtime of the (1 + 1) EA on Royal Staircase fitness functions – see e.g., [10] – consisting of n/ℓ plateaus, which have to be optimized sequentially. Using their method based on discrete Fourier analysis, the authors determine precise expected runtime both for static and fitness-dependent mutation rates and find the asymptotically optimal static and fitness-dependent mutation rates.

Studies of EAs with non-trivial populations on problems with plateaus are seldom. In the case of Cliff function [15], the non-elitist $(1, \lambda)$ EA was shown to be able to overcome a plateau of local optima in polynomial time, while the elitist $(1 + \lambda)$ EA is unable to do this. Here a plateau consists of solutions being locally optimal, all of which have the same number of ones. However the Cliff function presents a specific kind of fitness function with a large plateau of local optima, such that even a random walk ignoring the fitness, reaches this plateau with high probability, as it was noted in [16].

In the present paper, we consider the performance of non-elitist EAs optimizing the

Plateau_r fitness under different conditions on mutation and selection operators, including the similar conditions on unbiased mutation as in [7]. We obtain polynomial upper bounds on the expected runtime of non-elitist EAs, using tournament selection or (μ, λ)-selection (Theorems 7-10) and, in the case of bitwise mutation with low mutation probability of order $1/n^2$, using the fitness proportionate selection (Theorem 11). These upper bounds are obtained by means of the level-based theorems [17], [18] and [19].

Taking into account the similarity of function Plateau_r to the well-known OneMax function, in Proposition 15 we derive an exponential lower bound on the expected runtime of the EAs with the proportionate selection and with mutation probability χ/n, where χ is a constant greater than ln 2. Given these conditions, we also show in Theorem 16 that finding an approximate solution within certain constant approximation ratio also requires an exponential time in expectation. The lower bounds for the case of proportionate selection are based on the negative drift theorem from [20], and a stochastic domination argument suggested in [21] for analysis of the proportionate selection. The obtained lower bounds generalize the results of [21] and [22] known for OneMax.

From the methodological perspective, for upper bounding the expected runtime this paper suggests a new combination of typical application of a level-based theorem with the analysis of expected proportions of individuals from different level sets in the EA population [23]. A level-based theorem here gives an upper bound on the expected time until a constant fraction of a population will reach the plateau. Then the analysis of expected proportions of individuals from different level sets allows to compare the behaviour of the EA population to the behaviour of a single individual in a mutation-driven random walk on the plateau. This analysis will appear as Theorem 8 in the paper.

In contrast to [24], here we improve the upper bound of Theorem 8 by a factor of n, relax the constraint on the population size in Theorem 16 and turn the lower bound of this theorem from super-polynomial into exponential. Theorem 7 is given with the full proof.

2. PRELIMINARIES

We use the same notation as in [17, 25, 22]. For any $n \in \mathbb{N}$, define $[n] := \{1, 2, \ldots, n\}$. The natural logarithm and logarithm to the base 2 are denoted by $\ln(\cdot)$ and $\log(\cdot)$ respectively. By $\text{poly}(n)$ we will denote any polynomially bounded function, i.e. a function from $\Theta(n^k)$ for some constant $k > 0$. For $x \in \{0, 1\}^n$, we write x_i for the ith bit value. The Hamming distance is denoted by $H(\cdot, \cdot)$ and the Iverson bracket by $[\cdot]$. Throughout the paper the maximisation of a *fitness function* $f : \mathcal{X} \to \mathbb{R}$ over a *search space* $\mathcal{X} := \{0, 1\}^n$ is considered. Given a partition of $\mathcal{X}$ into m ordered subsets, called *levels* $(A_1, \ldots, A_m)$, let $A_{\geq j} := \cup_{i=j}^{m} A_i$. Note that by this definition, $A_{\geq 1} = \mathcal{X}$. A *population* is a vector $P \in \mathcal{X}^\lambda$, where the ith element $P(i)$ is called the ith *individual*. For $A \subseteq \mathcal{X}$, define $|P \cap A| := |\{i \; : \; P(i) \in A\}|$, i.e., the count of individuals of P in A.

2.1. The Objective Function

We are specifically interested in two fitness functions defined on $\mathcal{X} = \{0, 1\}^n$:

- The well-known benchmark function

$$\text{OneMax}(x) := \sum_{i=1}^{n} x_i,$$

- and a function from [7] with a single plateau of the second-best fitness in a ball of radius r around the unique global optimum

$$\text{Plateau}_r := \begin{cases} \text{OneMax}(x) & \text{if } \text{OneMax}(x) \le n - r, \\ n - r & \text{if } n - r < \text{OneMax}(x) < n, \\ n & \text{if } \text{OneMax}(x) = n, \end{cases}$$

parametrized by an integer $r \ge 1$.

Note that results of the paper will also hold for the generalised classes of such functions, see e.g. [26], where the meaning of 0-bit and 1-bit in each position can be exchanged, and/or x is rearranged according to a fixed permutation before each evaluation.

2.2. Non-Elitist Evolutionary Algorithm and Its Operators

The non-elitist EAs considered in this paper fall into the framework of Algorithm 1, see, e.g., [25, 22]. Starting with some P_0 which is sampled uniformly from $\mathcal{X}^\lambda$, in each iteration t of the outer loop a new population P_{t+1} is generated by independently sampling λ individuals from the existing population P_t using two operators: *selection* `select`: $\mathcal{X}^\lambda \to [\lambda]$ and *mutation* `mutate`: $\mathcal{X} \to \mathcal{X}$. Here, `select` takes a vector of λ individuals as input, then implicitly makes use of the function f, i. e., through *fitness evaluations*, to return the index of the individual to be selected. We assume that fitness evaluations are "cached" and the entire population is not re-evaluated to sample each individual. So it is sufficient to make λ fitness evaluations per iteration with any t.

Algorithm 1 Non-Elitist Evolutionary Algorithm

Require: Finite state space $\mathcal{X}$, population size $\lambda \in \mathbb{N}$,
$P_0 \sim \text{Unif}(\mathcal{X}^\lambda)$
for $t = 0, 1, 2, \ldots$ until termination condition met
  for $i = 1, 2, \ldots, \lambda$
    Sample $I_t(i) :=$ `select`(P_t), and set $x := P_t(I_t(i))$
    Sample $P_{t+1}(i) :=$ `mutate`(x)

The fitness function is considered optimised when an optimal solution appears in P_t for the first time, i. e., the operator `mutate` samples a solution x^*, such that $f(x^*) = \max_{x \in \mathcal{X}}\{f(x)\}$. The *optimisation time (or runtime)* is defined as the number of fitness evaluations made until the fitness function is optimized.

In this paper, we assume that the termination condition is never satisfied and the algorithm produces an infinite sequence of iterations. This simplifying assumption is frequently used in the theoretical analysis of EAs.

Formally, `select` is represented by a probability distribution over $[\lambda]$, and we use $p_{\text{sel}}(i \mid P)$ to denote the probability of selecting the ith individual $P(i)$ of P. The well-known *fitness-proportionate selection* is an implementation of `select` with

$$\forall P \in \mathcal{X}^\lambda, \forall i \in [\lambda] \colon p_{\text{sel}}(i \mid P) = \frac{f(P(i))}{\sum_{j=1}^{\lambda} f(P(j))}$$

(if $\sum_{j=1}^{\lambda} f(P(j)) = 0$, then one can assume that `select` has the uniform distribution). By definition, in the *k-tournament selection*, k individuals are sampled uniformly at random with replacement from the population, and a fittest of these individuals is returned. In *(μ, λ)-selection*, parents are sampled uniformly at random among the fittest μ individuals in the population. The ties in terms of fitness function are resolved arbitrarily.

The definitions presented in the rest of this subsection are standard for runtime analysis of non-elitist EAs and may be found in [17, 20] or similar sources.

We say that `select` is *f-monotone* if for all $P \in \mathcal{X}^\lambda$ and all $i, j \in [\lambda]$ it holds that $p_{\text{sel}}(i \mid P) \geq p_{\text{sel}}(j \mid P) \Leftrightarrow f(P(i)) \geq f(P(j))$. It is easy to see that all three selection mechanisms mentioned above are f-monotone.

The *cumulative selection probability* β of `select`(P) for any $\gamma \in (0, 1]$ is

$$\beta(\gamma, P) := \sum_{i=1}^{\lambda} p_{\text{sel}}(i \mid P) \cdot \left[f(P(i)) \geq f_{\lceil \gamma\lambda \rceil}\right], \text{ where } P \in \mathcal{X}^\lambda, \tag{1}$$

assuming a sorting $(f_1, \cdots, f_\lambda)$ of the fitnesses of P in descending order. For the sake of completeness, we assume $\beta(0, P) := 0$. In essence, $\beta(\gamma, P)$ is the probability of selecting an individual at least as good as the $\lceil \gamma\lambda \rceil$-ranked individual of P. Clearly, for any monotone selection mechanism and any $P \in \mathcal{X}^\lambda$ holds $\beta(\gamma, P) \geq \gamma$ at any $\gamma \in [0, 1]$.

When sampling λ times with `select`(P_t) and recording the outcomes as a vector $I_t \in [\lambda]^\lambda$, the *reproductive rate* of $P_t(i)$ is

$$\alpha_t(i) := \mathbf{E}\left[R_t(i) \mid P_t\right] \text{ where } R_t(i) := \sum_{j=1}^{\lambda} [I_t(j) = i].$$

Thus $\alpha_t(i)$ is the expected number of times that $P_t(i)$ is selected. The reproductive rate α_0 of Algorithm 1 is defined as $\alpha_0 := \sup_{t \geq 0} \max_{i \in [\lambda]} \{\alpha_t(i)\}$.

2.3. Properties of Mutation

The operator `mutate` is represented by a transition matrix $p_{\text{mut}} \colon \mathcal{X} \times \mathcal{X} \to [0, 1]$, and we use $p_{\text{mut}}(y \mid x)$ to denote the probability to mutate an individual x into y. We consider the *unbiased* mutation operators [13], i.e. those symmetric w.r.t. the bit-positions $1, \ldots, n$ and w.r.t. the bit-values 0 and 1:

1. For any $x, y, z, \in \mathcal{X}$ holds

$$p_{\text{mut}}(y \mid x) = p_{\text{mut}}(y \oplus z \mid x \oplus z),$$

   where $\oplus$ denotes the XOR operation.

2. For any $x, y \in \mathcal{X}$ and any permutation of bits $\sigma_{bits} : \mathcal{X} \to \mathcal{X}$

$$p_{\text{mut}}(y \mid x) = p_{\text{mut}}(\sigma_{bits}(y) \mid \sigma_{bits}(x)).$$

One of the most frequently used unbiased mutation operators, the *bitwise mutation* (also known as the *standard bit mutation*), changes each bit of a given solution with a fixed mutation probability p_{mut}. Usually it is assumed that $p_{mut} = \chi/n$ for some parameter $\chi > 0$. For the bitwise mutation with mutation probability χ/n we have

$$\forall x, y \in \{0,1\}^n : p_{\text{mut}}(y \mid x) = \left(\frac{\chi}{n}\right)^{H(x,y)} \left(1 - \frac{\chi}{n}\right)^{n-H(x,y)}.$$

Another well-known mutation operator, the *point mutation operator,* keeps the input string x unchanged with some given probability p_0 or flips one bit x_i in randomly chosen position $i \in [n]$ with probability $1 - p_0$. Note that both of these mutation operators treat the bit values 0 and 1 indifferently, as well as the bit positions, and therefore satisfy the conditions of unbiasedness.

Suppose a level partition $(A_1, \ldots, A_m)$ of the search space is given, then we will call a mutation operator *monotone* with respect to this level partition if for any $i, i', j \in [m]$, such that $i \geq i'$, for any $x \in A_i$, $x' \in A_{i'}$ the following inequality holds

$$p_{\text{mut}}(A_{\geq j} \mid x) \geq p_{\text{mut}}(A_{\geq j} \mid x'),$$

where $p_{\text{mut}}(A_{\geq j} \mid x) = \Pr(\texttt{mutate}(x) \in A_{\geq j})$ and $p_{\text{mut}}(A_{\geq j} \mid x') = \Pr(\texttt{mutate}(x') \in A_{\geq j})$. It was proved in [27] that the standard mutation with $p_{mut} \leq 1/2$ is monotone w.r.t. the level partition which consists of the subsets $A_i = \{x : \text{OneMax}(x) = i - 1\},\ i = 1, \ldots, n+1$. Another example of monotone mutation operator w.r.t. this level partition is the point mutation which keeps the input string x unchanged with some given probability $p_0 \geq 1/(n+1)$, see e.g. [23].

Analogously to Propositions 1-3 [23] for the case of tournament selection or Proposition 2 [28] for the case of (μ, λ)-selection, we can formulate the following

Theorem 1. *Suppose the EA from Algorithm 1 uses a mutation operator* `mutate` *and a monotone selection mechanism* `select`*, besides that suppose that a mutation operator* $\texttt{mutate}_0$*, is monotone w.r.t. a level partition* $(A_1, \ldots, A_m)$*, and for any* $i, j \in [m]$ *it satisfies*

$$\Pr(\texttt{mutate}(x) \in A_{\geq j}) \geq \Pr(\texttt{mutate}_0(x') \in A_{\geq j}) \textit{ for any } x, x' \in A_i.$$

Then, the random string x^{t+1}*, obtained as* $x^{t+1} = \texttt{mutate}_0(x^t)$ *from a random string* x^t*, such that* $\Pr(P_t(1) \in A_{\geq j}) \geq \Pr(x^t \in A_{\geq j})$ *for all* $j \in [m]$*, satisfies the inequalities*

$$\Pr(P_{t+1}(1) \in A_{\geq j}) \geq \Pr(x^{t+1} \in A_{\geq j}) \ \textit{for all}\ j \in [m]. \tag{2}$$

For the sake of completeness, a full proof of Theorem 1 is provided in Appendix A.

Clearly, $P_t(1)$ may be replaced by $P_t(i)$ with any $i \in [\lambda]$ in the formulation of Theorem 1 because all individuals of a population are identically distributed. Note that the inequality (2) may be applied recursively up to any given iteration number t. In particular, if we start the random sequence $x^0, x^1, \ldots$ with $x^{t_0} = P_{t_0}(1)$ at some iteration t_0, then iterative application of Theorem 1 implies that $\Pr(P_t(1) \in A_{\geq j}) \geq \Pr(x^t \in A_{\geq j})$ for any j and $t \geq t_0$.

2.4. Runtime Bounds from the Literature

In this paper, a number of results on runtime analysis are employed from other works. This subsection contains the formulations of these results. Some of the formulations are given with slight modifications, which do not require a special proof.

Our lower bound is based on the *negative drift theorem for populations* [20].

Theorem 2. *Consider the EA from Algorithm 1 on* $\mathcal{X} = \{0,1\}^n$ *with bitwise mutation rate* χ/n *and population size* $\lambda = \text{poly}(n)$*, let* $a(n)$ *and* $b(n)$ *be positive integers such that* $b(n) \leq n/\chi$ *and* $d(n) = b(n) - a(n) = \omega(\ln n)$*. Given* $x^* \in \{0,1\}^n$*, define*

$$T(n) := \min\left(t : P_t \cap \{x \in \mathcal{X} : H(x, x^*) \leq a(n)\}| \neq \emptyset\right).$$

If there exist constants $\alpha > 1$*,* $\delta > 0$ *such that*

1) $\forall t \geq 0$*,* $\forall i \in [\lambda]$*: if* $a(n) < H(P_t(i), x^*) < b(n)$ *then* $\alpha_t(i) \leq \alpha$*,*
2) $\psi := \ln(\alpha)/\chi + \delta < 1$*,*
3) $b(n)/n < \min\left\{1/5, 1/2 - \sqrt{\psi(2-\psi)/4}\right\}$*,*

then $\Pr\left(T(n) \leq e^{cd(n)}\right) = e^{-\Omega(d(n))}$ *for some constant* $c > 0$*.*

We also use Corollary 1 from [20], which follows from the negative drift theorem for populations, focused on achieving the global optimum:

Corollary 3. *The probability that an EA from Algorithm 1 with population size* $\lambda = \text{poly}(n)$*, bitwise mutation probability* χ/n*, and maximal reproductive rate bounded by* $\alpha_0 \leq \alpha < e^{\chi} - \delta$*, for a constant* $\delta > 0$*, optimises any function with a polynomial number of optima within* e^{cn} *generations is* $e^{-\Omega(n)}$*, for some constant* $c > 0$*.*

To estimate the expected optimization time of Algorithm 1 from above, we use the *level-based analysis* [17]. The following theorem is a re-formulation of Corollary 7 from [17], tailored to the case of no recombination.

Theorem 4. *Given a partition* $(A_1, \dots, A_m)$ *of* $\mathcal{X}$*, if there exist* $s_1, \dots, s_{m-1}, p_0, \delta \in (0,1]$*,* $\gamma_0 \in (0,1)$ *such that*

(M1) $\forall j \in [m-1]\colon\ p_{\text{mut}}\left(y \in A_{\geq j+1} \mid x \in A_j\right) \geq s_j$,
(M2) $\forall j \in [m]\colon\ p_{\text{mut}}\left(y \in A_{\geq j} \mid x \in A_j\right) \geq p_0$,
(M3) $\forall P \in (\mathcal{X})^{\lambda}, \forall \gamma \in (0, \gamma_0]\colon\ \beta(\gamma, P) \geq (1+\delta)\gamma/p_0$,
(M4) *population size* $\lambda \geq \dfrac{4}{\gamma_0 \delta^2} \ln\left(\dfrac{128m}{\gamma_0 s_* \delta^2}\right)$*, where* $s_* := \min_{j \in [m-1]}\{s_j\}$*,*

then

$$\mathbf{E}[T_0] < \left(\frac{8}{\delta^2}\right) \sum_{j=1}^{m-1} \left(\ln\left(\frac{6\delta\lambda}{4 + \gamma_0 s_j \delta\lambda}\right) + \frac{1}{\gamma_0 s_j \lambda}\right), \tag{3}$$

where $T_0 := \min\{t : |P_t \cap A_m| \geq \gamma_0 \lambda\}$.

Note that literally the formulation of Corollary 7 in [17] gives the bound (3) only for the expected runtime, but it is easy to see from the proof therein that the bound actually holds for the expected number T_0 of the first population that contains at least $\gamma_0\lambda$ individuals in level A_m as we put it in Theorem 4. This slight modification is useful for analysis in Section 3.

As an alternative to Theorem 4 we also use the following level-based theorem which was proved in [29] using the *multiplicative up-drift* [30].

Theorem 5. *Given a partition* $(A_1, \ldots, A_m)$ *of* $\mathcal{X}$, *define* $T := \min\{t\lambda : |P_t \cap A_m| > 0\}$. *If there exist* $s_1, \ldots, s_{m-1}, p_0, \delta \in (0, 1]$, $\gamma_0 \in (0, 1)$, *such that conditions (M1)–(M3) of Theorem 4 hold and*

(M4′) *for some constant* $C > 0$, *the population size* λ *satisfies*

$$\lambda \geq \frac{8}{\gamma_0 \delta^2} \log\left(\frac{Cm}{\delta}\left(\log\lambda + \frac{1}{\gamma_0 s_* \lambda}\right)\right), \text{ where } s_* := \min_{j\in[m-1]}\{s_j\},$$

then

$$\mathbf{E}[T] = \mathcal{O}\left(\frac{\lambda m \log(\gamma_0\lambda)}{\delta} + \frac{1}{\delta}\sum_{j=1}^{m-1}\frac{1}{\gamma_0 s_j}\right).$$

Theorem 5 improves on Theorem 4 in terms of dependence of the runtime bound denominator on δ, but only gives an asymptotical bound. The proof of Theorem 5 is analogous to that of Theorem 4 and may be found in the appendix to [29].

In the case of very high selection pressure, non-elitist EAs tend to choose the best found solutions as parents, similar to the elitist EAs. The upper bound on expected runtime, formulated as Theorem 6 below, is a slight modification of Theorem 1 [18].

A level partition $(A_1, \ldots, A_m)$ is said to be an f*-based partition* if $f(x) < f(y)$ for all $x \in A_j$, $y \in A_{j+1}$ and all $j \in [m-1]$.

Theorem 6. *Given an* f*-based partition* $(A_1, \ldots, A_m)$ *of* $\mathcal{X}$, *if the EA from Algorithm 1 uses*

1) *such a mutation that* $\Pr(\texttt{mutate}(x) \in A_{\geq j+1}) \geq s_*$ *for any* $x \in A_j$, $j \in [m-1]$ *and*
2) *a* k*-tournament selection,* $k \geq c\lambda$ *with a population of size* $\lambda \geq (1 + \ln m)/(s_*(1 - e^{-c}))$ *for some* $c > 0$, *or* (μ, λ)*-selection and* $\lambda \geq \mu(1 + \ln m)/s_*$,

then an element from A_m *is found after at most* m *generations with probability not less than* $1/e$, *where* $e = 2.71828...$ *is the base of the natural logarithm.*

For the sake of completeness, a full proof of Theorem 6 is provided in the appendix.

3. UPPER BOUNDS FOR EXPECTED RUNTIME

3.1. Tournament and (μ, λ)-Selection

First of all, due to similarity of function Plateau_r to the well-known function Jump_r,

$$\text{Jump}_r(x) := \begin{cases} n+1 & \text{if OneMax}(x) = n \\ r + \text{OneMax}(x) & \text{if OneMax}(x) \leq n - r \\ n - \text{OneMax}(x) & \text{otherwise} \end{cases},$$

analogously to the proof of Theorem 11 [17] (its Jump_r case), we obtain the following theorem.

Theorem 7. *Let the EA from Algorithm 1 be applied to* Plateau_r, $r = \mathcal{O}(1)$, *using*

1) *a bitwise mutation given a mutation rate* $p_{mut} = \chi/n$ *for any fixed constant* $\chi > 0$,

2) *a* k*-tournament selection or a* (μ, λ)*-selection with their parameters* k *or* λ/μ *(respectively) being set to no less than* $(1+\delta')e^{\chi}$, *where* $\delta' \in (0, 1]$ *is a constant, and*

3) *a population of size* $\lambda \geq c \ln n$, *for a sufficiently large constant* c.

Then the EA has an expected runtime $\mathcal{O}(n^r + n\lambda)$.

Proof. We apply Theorem 4, partitioning $\mathcal{X}$ into $m = n - r + 2$ subsets, where

$$A_i := \{x : \text{OneMax}(x) = i - 1\},\ i \leq m - 2,$$

$$A_{m-1} := \{x : n - r < \text{OneMax}(x) < n\},\ \ A_m = \{(1, \ldots, 1)\}.$$

The choices of s_j and s_* to satisfy (M1) are the following. For $j = 0, \ldots, m-2$

$$s_j := \frac{\chi}{n}(n - j + 1)\left(1 - \frac{\chi}{n}\right)^{n-1} = \Omega\left(\frac{n - j + 1}{n}\right),$$

but $s_{m-1} := \left(\frac{\chi}{n}\right)^r \left(1 - \frac{\chi}{n}\right)^{n-r} = \Omega\left(\left(\frac{\chi}{n}\right)^r\right)$, i.e. the probability of flipping the r remained 0-bits, so $s_* := \Omega\left(\left(\frac{1}{n}\right)^r\right)$.

Lemma 3 from [31] (see Lemma 17, assuming $\varepsilon = (\delta'/2)(1 + \delta'/2)$ in Appendix B below) implies that the probability of not flipping any bit position by mutation is

$$(1 - \chi/n)^n \geq \left(1 - \frac{\delta'/2}{1 + \delta'/2}\right) e^{-\chi} = \frac{e^{-\chi}}{1 + \delta'/2}$$

for n sufficiently large. Thus choosing $p_0 := \frac{e^{-\chi}}{1+\delta'/2}$ satisfies (M2).

We now look at (M3). For k-tournament selection, we have

$$k \geq (1 + \delta')e^{\chi} = \left(1 + \frac{\delta'/2}{1 + \delta'/2}\right) \frac{1}{p_0}$$

due to our choice of p_0. Hence, Lemma 5 from [22] (see Lemma 18, assuming $\Delta = (\delta'/2)/(1 + \delta'/2)$ in Appendix B below) implies that (M3) is satisfied for any $\gamma \in (0, \gamma_0]$, $\gamma_0 = \theta(1)$ with some constant $\delta \in (0, 1]$. The same conclusion can be drawn for the other selection mechanism.

In (M4), since γ_0 and δ are constants, there should exist a constant $c > 0$ such that the condition (M4) is satisfied.

Since all conditions are satisfied, Theorem 4 may be applied. Let us analyse the resulting upper bound on the optimisation time. For the first $m-2$ terms of the sum

$$\sum_{j=1}^{m-1} \ln\left(\frac{6\delta\lambda}{4+\gamma_0 s_j \delta\lambda}\right), \tag{4}$$

bounding each term by $\ln\left(\frac{6}{\gamma_0 s_j}\right)$, we get $\mathcal{O}(n)$ as in the case of OneMax in the proof of Theorem 11 from [17]. For the last term we have the upper bound $\ln(6/(\gamma_0 s_{m-1})) = \mathcal{O}(\log n)$ since $s_{m-1} = \Omega(1/n^r)$. In total, we have the upper bound $\mathcal{O}(n)$ for the sum (4). So $\mathbf{E}[T_0] = \mathcal{O}(n + n^r/\lambda)$, and multiplication by the population size λ gives the required upper bound on $\mathbf{E}[T]$. □

Note that by the similar modification of the last step in the proof of Theorem 11 [17], one can also obtain the $\mathcal{O}(n^r + n\lambda)$ upper bound on the expected EA runtime in the case of Jump function.

In the more general case of unbiased mutation we prove the following

Theorem 8. *Let the EA from Algorithm 1 be applied to* Plateau$_r$, $r = \mathcal{O}(1)$, *using*

1) *an unbiased mutation* `mutate` *with* $\Pr(\xi = 0) \geq p_0 = \Omega(1)$ *and* $\Pr(\xi = 1) \geq p_1 = \Omega(1)$, *where ξ is the random variable equal to the number of bits flipped in mutation,*

2) *a k-tournament selection or a (μ, λ)-selection with their parameters k or λ/μ (respectively) being set to no less than $(1+\delta')/p_0$, where $\delta' \in (0,1]$ is a constant, and*

3) *population size $\lambda \geq c \ln n$ for sufficiently large constant c, independent of r.*

Then the expected runtime of this EA is $\mathcal{O}(\lambda n^r)$.

Before the proof of this theorem we formulate the following lemma, which resembles one of the steps in the proof of the level-based theorem [17], but applies to level m.

Lemma 9. *Assuming conditions of Theorem 4 are satisfied, if $|P_t \cap A_m| \geq \gamma_0\lambda$ on iteration t, then for the next iteration $t+1$ we have*

$$\Pr(|P_{t+1} \cap A_m| < \gamma_0\lambda) \leq \frac{\delta^2 \gamma_0 s_*}{128m}. \tag{5}$$

Proof of Lemma 9. To create an individual in A_m it suffices to pick an $x \in P \cap A_m$ and mutate it to an individual in A_m, the probability of such an event according to (M2) and (M3) is at least $\beta(\gamma_0, P)p_0 \geq (1+\delta)\gamma_0$. So, we obtain the following by the multiplicative Chernoff bound

$$\Pr(|P_{t+1} \cap A_m| < \gamma_0\lambda) = \Pr\left(|P_{t+1} \cap A_m| < \left(1 - \frac{\delta}{1+\delta}\right)(1+\delta)\gamma_0\lambda\right)$$
$$\leq \exp\left(-\frac{\delta^2\gamma_0\lambda}{2(1+\delta)}\right) \leq \frac{\delta^2 s_*\gamma_0}{128m}, \quad (6)$$

where the last inequality takes into account the condition (M4) and $\delta \leq 1$. $\square$

Proof of Theorem 8. We apply Theorem 4, partitioning $\mathcal{X}$ into $m = n-r+1$ subsets, where $A_i := \{x : |x| = i-1\},\ i \in [m-1], A_m := \{x : |x| \geq n-r\}$. The choices for s_j and s_* to satisfy (M1) condition are the following. For $j = 1, \dots, m-1$

$$s_j := \frac{\chi}{n}(n-j+1)\left(1 - \frac{\chi}{n}\right)^{n-1} = \Omega\left(\frac{n-j+1}{n}\right),$$

so $s_* := \Omega\left(\frac{1}{n}\right)$.

Conditions (M2)-(M4) are verified exactly as in the proof of Theorem 7 for some constant $\delta \in (0, 1]$ defined by δ'.

Since all conditions are satisfied, Theorem 4 may be applied. Each term in the sum

$$\sum_{j=1}^{m-1} \ln\left(\frac{6\delta\lambda}{4 + \gamma_0 s_j \delta\lambda}\right),$$

may be bounded by $\ln\left(\frac{6}{\gamma_0 s_j}\right)$, which implies that $\mathbf{E}[T_0] = \mathcal{O}(n)$, i.e. on average after at most Cn iterations the EA will produce a population with at least $\gamma_0\lambda$ individuals in A_m, where C and γ_0 are positive constants. By the Markov's inequality, this implies that with probability at least 1/2, starting from any population, within $2Cn$ iterations, the EA produces a population with at least $\gamma_0\lambda$ individuals in A_m. By Lemma 9, with $s_* = \Theta(1/n)$ and $m = \Theta(n)$, in case of constant δ and γ_0, if $|P_t \cap A_m| \geq \gamma_0\lambda$ then the probability to get less than $\gamma_0\lambda$ plateau individuals in the next iteration is at most K/n^2 for some constant K. Thus, by the union bound, the probability to get less than $\gamma_0\lambda$ plateau individuals at least once during the following $n^2/(2K)$ iterations is at most $1/2$. In what follows, a sequence of $2Cn + n^2/(2K)$ iterations we will call a *series* of the EA iterations. Clearly, the last $n^2/(2K)$ iterations of any given series contain at least $\gamma_0\lambda$ plateau individuals with probability not less than $1/4$.

In what follows, let us use the more detailed level partition $A'_i := \{x : |x| = i-1\},\ i = 1, \dots, n+1$ with $m' = n+1$. Let us suppose that the optimum is not found until the following scenario of reaching the optimum is performed (otherwise the runtime will be just shorter).

Suppose that P_{t_0} contains at least $\gamma_0\lambda$ individuals on the plateau. Then to create an individual in $A'_{\geq n-r+1}$ in iteration $t_0 + 1$ it suffices to pick an individual in $P \cap A'_{\geq n-r}$ and mutate it to an individual in $A'_{\geq n-r+1}$. The probability of such an event is at least $\beta(\gamma_0, P)p_1/n \geq (1+\delta)\gamma_0 p_1/n = \Omega(1/n)$.

To obtain a lower bound on the probability to reach the global optimum, having started in some point on the plateau, let us consider a simplified mutation operator $\texttt{mutate}_0(x)$ such that for any $x \in \mathcal{X}$:

- in the case when $|\texttt{mutate}(x)| = |x|$, the new mutation has identical output to $\texttt{mutate}(x)$;
- in the case when $|\texttt{mutate}(x)| = |x| + 1$, a random choice is made: with probability p_0 the operator $\texttt{mutate}_0$ outputs the same string as $\texttt{mutate}(x)$, and with probability $1 - p_0$ it outputs a random point from A_1;
- in all other cases $\texttt{mutate}_0(x)$ outputs a random point from A_1.

Let us check that $\texttt{mutate}_0$ is a monotone mutation operator. Indeed, all strings within the same level set $A'_i,\ i \in [m']$ have equal probability to produce an offspring in any target set $A'_{\geq j},\ j \in [m']$, therefore we can consider only cumulative transition probabilities $\alpha_{ij} := \Pr\{\texttt{mutate}_0(x) \in A'_{\geq j}\}$ where $x \in A'_i,\ i, j \in [m']$ to prove the monotonicity of $\texttt{mutate}_0$. By the definition of the transition probabilities, $\alpha_{i1} = 1$ for all $i \in [m']$. By the definition of $\texttt{mutate}_0$ we have $\alpha_{ij} = 0$ for all $i = 2, \dots, m',\ j = 2, \dots, i-1, i+2, \dots, m'$, and since $\texttt{mutate}$ is unbiased by assumption 1), so $\alpha_{i,i+1} = p_0 \cdot \Pr\left(H(\texttt{mutate}(x), x) = 1\right)(n-i+1)/n$ is the probability that $|\texttt{mutate}(x)| = |x|+1$ for $x \in A'_i,\ i \in [m'-1]$. Note that for proving monotonicity for operator $\texttt{mutate}_0$ it is sufficient to show that $\alpha_{i+1,j} - \alpha_{i,j} \geq 0, i \in [m'-1], j \in [m']$. For all $i \neq j-1$ this inequality is trivial in view of the definition of $\texttt{mutate}_0$. For the case of $i = j-1,\ j = 2, \dots, m'$ we have

$$\alpha_{i+1,j} - \alpha_{i,j} = \alpha_{jj} - \alpha_{j-1,j} \geq p_0 - p_0 \cdot \Pr\left(H(\texttt{mutate}(x), x) = 1\right)(n - j + 2)/n$$

which is nonnegative. Therefore $\texttt{mutate}_0$ is monotone.

The conditions of Theorem 1 are satisfied for the pair of operators $\texttt{mutate}$ and $\texttt{mutate}_0$. Therefore, Theorem 1 allows to estimate the distribution of an individual $P_t(1)$ (or any other individual $P_t(j)$ for a fixed $j \in [\lambda]$) over the subsets $A'_{\geq i},\ i \in [m]$ by the distribution of the random state x^t.

Consider a random state x^{t_0+1}, distributed identically with $P_{t_0+1}(1)$, which implies that $|x^{t_0+1}| \geq n-r+1$ with probability $\Omega(1/n)$, because the probability to choose a parent from $A_{\geq n-r}$ is $\beta(\gamma_0, P_{t_0}) \geq \gamma_0 = \Omega(1)$, the mutation is unbiased, and $\Pr(\xi = 1) \geq p_1 = \Omega(1)$. Using this as a basis for induction argument, let us assume that $|x^{t_0+i}| \geq n - r + i$ with probability $\Omega(1/n^i)$. At iteration $t = t_0 + 2, \dots, t_0 + r$ the operator $\texttt{mutate}_0(x^t)$ with probability $p_1 = \Omega(1)$ mutates a uniformly chosen single bit from x^t, so for the next iteration x^{t+1}, the probability to increment the number of ones is $\Omega(1/n)$. Finally, $\Pr(x^{t_0+r} \in A_n) = \Omega(1/n^r)$. Then an $(r-1)$-fold application of Theorem 1 implies that $\Pr(P_{t_0+r}(1) \in A_n) = \Omega(1/n^r)$ as well.

Now we can consider a sequence of series of the EA iterations, where the length of each series is $2Cn + n^2/(2K)$ iterations. In each series, following the first $2Cn$ iterations of a series, there are $\frac{n^2}{2Kr} = \Omega(n^2)$ *stages* of r iterations in each stage. As it was shown in the beginning of the proof, with probability 1/4 each stage starts with $\gamma_0\lambda$ individuals on the plateau. Suppose, $D_j,\ j = 1, 2, \dots,$ denotes an event of absence of the optimal individuals in the EA population throughout the jth stage. In view of the above consideration, the probability of each event $D_j,\ j = 1, 2, \dots,$ is at most $1 - \Omega(1/n^r)$, so the probability to reach the optimum for the first time in more than j stages is at most $(1 - C'/n^r)^j$ for some constant C'.

Let Y denote the random variable equal to the number of the first stage when the optimal solution is obtained. By the properties of expectation, see e.g. [32],

$$\mathbf{E}[Y] = \sum_{j=0}^{\infty} \Pr(Y > j) = 1 + \sum_{j=1}^{\infty} \Pr(D_1 \& \ldots \& D_j) \le 1 + \sum_{j=1}^{\infty} (1 - C'/n^r)^j = \mathcal{O}(n^r). \tag{7}$$

Since each series contains $2Cn$ iterations to reach the plateau and $n^2/(2K)$ iterations that cover $\frac{n^2}{2Kr}$ stages, so the total number of iterations till finding the optimum differs from the total length of all stages that count till finding the optimum by the factor of $1+\mathcal{O}(1/n)$, so by (7) the expected runtime, measured in fitness evaluations, is $\mathcal{O}(\lambda n^r)$. □

Note that the runtime bound of Theorem 8 is by a factor n tighter than the bound from Theorem 2 [24], besides that, the proof of Theorem 2 in [24] contained an error.

The following theorem shows that the requirement of a positive constant lower bound on probability to mutate none of the bits $\Pr(\xi = 0) = \Omega(1)$ may be avoided at the expense of very high selection pressure and a factor of λr longer runtime.

Theorem 10. *Let the EA from Algorithm 1 be applied to* Plateau_r, $r = \mathcal{O}(1)$, *using*

1) *an unbiased mutation with* $\Pr(\xi = 1) = \Omega(1)$, *where* ξ *is the random variable equal to the number of bits flipped in mutation, and*

2) *a* k*-tournament selection,* $k \ge c\lambda$ *with a population of size* $\lambda \ge \frac{n(1+\ln n)}{\Pr(\xi=1)(1-e^{-c})}$ *for some* $c > 0$, *or* (μ, λ)*-selection with* $\lambda/\mu \ge \frac{n(1+\ln n)}{\Pr(\xi=1)}$.

Then the expected runtime of the EA is $\mathcal{O}(\lambda^2 n^{r+1})$.

Proof. The proof is analogous to that of Theorem 8, but now by the means of Theorem 6 we only ensure that with probability at least $1/e$, at least one individual from A_m will be present in the EA population after at most $n - r$ iterations at the beginning of each series. Suppose this event has happened in some iteration P_t. The probability to choose a parent from A_m for constructing P_{t+1} is lower-bounded only by $1/\lambda$ because the best-fit individual in tournament and in (μ, λ)-selection mechanisms is chosen with probability at least $1/\lambda$. Then any offspring in the population P_{t+1} with the probability $\Omega(1/(n\lambda))$ will have at least $n - r + 1$ ones. So, by the inductive argument, analogous to that in the proof of Theorem 8, we conclude that $\Pr\left(f(P_{t+r}(1)) = f^* | P_t \cap A_m \ne \emptyset\right) = \Omega(1/(n^r\lambda))$.

Consider a sequence of series of the EA iterations, where the length of each series is n iterations. Suppose that $D_j,\ j = 1, 2, \ldots,$ denotes an event of absence of the optimal individuals in the population throughout the jth series. Let Y denote the random variable equal to the number of the first series when the optimal solution is obtained. In view of the above consideration, the probability of each event $D_j,\ j = 1, 2, \ldots,$ is $1 - \Omega(1/(n^r\lambda))$, so the probability to reach the optimum in at most j series is lower bounded by $(1 - C''/n^r)^j$ for some constant C'', and $\mathbf{E}[Y] = \mathcal{O}(n^r\lambda)$. The length of each series is n iterations, so the total runtime is $\mathcal{O}(\lambda^2 n^{r+1})$. □

We see two main reasons for the difference of the runtime bounds obtained in Theorem 8 and Theorem 10 by the factor of $n\lambda$:

(i) In Theorem 8, we rely on r consecutive successes of the (1,1) EA (there is no selection here), while in Theorem 10 the proof relies on r consecutive successes involving the selection step of the EA, which imposes an extra factor of λ. It is not clear how to relate the non-elitist EA from Theorem 10 to the (1,1) EA because based on the assumptions of the latter theorem we can only assign $p_0 := 0$, which does not allow to define a suitable mutation operator $\texttt{mutate}_0$ for (1,1) EA.

(ii) Lemma 9 can not be applied either in the proof of Theorem 10 because this lemma requires a positive lower bound p_0 (condition M2), which is not relevant in the case of Theorem 10. But Lemma 9 is used in Theorem 8 to avoid complete restarts from the first level whenever the EA has failed to reach the global optimum on its journey through the plateau. This leads to an extra factor of n in the runtime bound of Theorem 10.

The assumptions of Theorem 10 require very high selection pressure, so it is likely that the runtime bound in this case may be significantly improved, relying on the similarity with the (1+1) EA.

3.2. Fitness-Proportionate Selection and Low Mutation Rate

In this subsection, we analyse the use of the fitness proportionate selection. This selection mechanism, also known as *roulette-wheel* selection, was the main selection used in the early development of genetic algorithms (GA) and their applications [33]. Unlike the rank-based selection mechanisms considered above, this selection is sensitive to the absolute values of fitness function $f(x)$. This is often seen as a weakness from the theoretical view point, but at the same time there exists a large body of literature reporting successful applications of the the fitness proportionate selection.

The results of [22], [25] and [19] make use of the level-based analysis for proving polynomial upper bounds on the expected optimisation time on the OneMax, given small mutation rate $p_{mut} = \mathcal{O}(n^{-2})$. In Theorem 11 we will employ this technique for the case of Plateau.

Theorem 11. *The expected runtime of the EA from Algorithm 1 on* Plateau_r, $r = \mathcal{O}(1)$, *using*

1) fitness-proportionate selection,

2) bitwise mutation with mutation probability χ/n, $\chi = (1-c)/n$, *for any constant* $c \in (0,1)$

3) population size $\lambda \geq c' n^2 \ln(n)$, $\lambda = \mathcal{O}\left(n^K\right)$, *where* c' *and* K *are positive sufficiently large constants*

is $\mathcal{O}\left(\lambda n^2 \log n + n^{2r+1}\right)$.

Proof. We will apply Theorem 5. To this end, we use a partition of $\mathcal{X}$ into $m = n+1$ subsets, where $A_i := \{x : |x| = i-1\},\ i \in [m-1], A_m := \{x : |x| = n\}$.

Let us choose the lower bounds s_j that satisfy (M1). Given $x \in A_j$ for any $j < m-1$, among the first $j+1$ bits, there must be at least one 0-bit, thus it suffices to flip the first 0-bit on the left while keeping all the other bits unchanged to produce a search point at a higher level. The probability of this event is

$$\frac{\chi}{n}\left(1-\frac{\chi}{n}\right)^{n-1} > \frac{\chi}{n}\left(1-\frac{1}{n}\right)^{n-1} \geq \frac{1-c}{en^2} =: s_j, j \in [m-2].$$

For s_{m-1} we have $s_* := s_{m-1} = \left(\frac{\chi}{n}\right)^r \cdot \left(1-\frac{1}{n}\right)^{n-r} = \Omega(n^{-2r})$.

To satisfy (M2), we pick $p_0 := (1-\chi/n)^n$, i.e., the probability of not flipping any bit position by mutation.

In (M3), we choose $\gamma_0 := c/4$ and for any $\gamma \leq \gamma_0$, let f_γ be the fitness of the $\lceil\gamma\lambda\rceil$-ranked individual of any given $P \in \mathcal{X}^\lambda$. Thus there are at least $k \geq \lceil\gamma\lambda\rceil \geq \gamma\lambda$ individuals with fitness at least f_γ and let $s \geq kf_\gamma \geq \gamma\lambda f_\gamma$ be their sum of fitness. We can pessimistically assume that individuals with fitness less than f_γ have fitness $f_\gamma - 1$, therefore

$$\begin{aligned}\beta(\gamma, P) &\geq \frac{s}{s+(\lambda-k)(f_\gamma-1)} \geq \frac{s}{s+(\lambda-\gamma\lambda)(f_\gamma-1)} \\ &\geq \frac{\gamma\lambda f_\gamma}{\gamma\lambda f_\gamma+(\lambda-\gamma\lambda)(f_\gamma-1)} = \frac{\gamma}{1-(1-\gamma)/f_\gamma} \\ &\geq \frac{\gamma}{1-(1-c/4)/f^*} \geq \gamma e^{(1-c/4)/f^*},\end{aligned}$$

where $f^* := n$ and in the last line we apply the inequality $e^{-x} \geq 1-x$. Note that $p_0 = (1-\chi/n)^n \geq e^{-\chi/(1-\varepsilon)}$ for any constant $\varepsilon \in (0,1)$ and sufficiently large n. Indeed, by Taylor theorem, $e^{-z} = 1-z+z\alpha(z)$, where $\alpha(z) \to 0$ as $z \to 0$. So given any $\varepsilon > 0$, for all sufficiently small $z > 0$ it holds that $e^{-z} \leq 1-(1-\varepsilon)z$. For any $\varepsilon \in (0,1)$ we can assume that $z = \chi/(n(1-\varepsilon))$, then for all sufficiently large n it holds that $(1-\chi/n)^n \geq e^{-zn} = e^{-\chi/(1-\varepsilon)}$. So we conclude that

$$\beta(\gamma, P)p_0 \geq \gamma e^{(1-c/4)/f^*} e^{-\chi/(1-\varepsilon)} \geq \gamma\left(1+\frac{1-c/4-\chi f^*/(1-\varepsilon)}{f^*}\right).$$

Now $\chi f^* = \chi n = 1-c$, thus choosing $\varepsilon := 1-\frac{1-c}{1-c/2} \in (0,1)$ implies $\chi f^*/(1-\varepsilon) \leq 1-c/2$. Condition (M3) then holds for $\delta := c/(4n)$ because

$$\beta(\gamma, P)p_0 \geq \gamma\left(1+\frac{1-c/4-(1-c/2)}{f^*}\right) \geq \gamma\left(1+\frac{c}{4n}\right).$$

To verify condition (M4′), we assume $C = 1$ and note that

$$\frac{8}{\gamma_0\delta^2}\log\left(\frac{Cm}{\delta}\left(\log\lambda+\frac{1}{\gamma_0 s_*\lambda}\right)\right) \leq c'' rn^2 \ln(n),$$

for some constant $c'' > 0$, since $\lambda \le n^K$ and $s^* = \Omega(n^{-2r})$. So (M4′) holds if c' is large enough. By Theorem 5, we conclude that on average after at most $O(\lambda n^2 \log n + n^{2r+1})$ fitness evaluations the EA will produce the optimum. □

In the case of $r = 1$ (i.e. the OneMax fitness), application of Theorem 11 for $\lambda = \Theta(n^2 \log n)$ gives $\mathbf{E}[T] = \mathcal{O}\left(n^4 (\log n)^2\right)$, the same as the upper bound in Theorem 4.1 in [30]. A refined analysis of [19] improves the expected runtime bound for this special case, yielding $\mathbf{E}[T] = \mathcal{O}(\lambda n^2 \log n + n \log(n)/p_{mut})$, which, given appropriate $\lambda = \Theta(n \log n)$ turns into $\mathcal{O}(n^3 (\log n)^2)$. However for $r > 1$ the last term n^{2r+1} plays the main role and the approach from [19] loses its advantage.

4. INEFFICIENCY OF FITNESS-PROPORTIONATE SELECTION WITH STANDARD SETTINGS

In this section, we consider Algorithm 1 with fitness-proportionate selection and the bitwise mutation, given a constant value of the parameter $\chi > \ln 2$. This algorithm turns out to be inefficient on Plateau_r for any constant $r \ge 1$. For the proof we will use the *Negative Drift Theorem* from [22]. In order to obtain an upper bound on the reproductive rate, required for this theorem, in Lemma 12 below we modify Lemma 11 of [21], extending it from the case of $p_{\text{mut}} = 1/n$ to the case of arbitrary $p_{\text{mut}} \le 1/2$. Here we also extend Lemma 11 [21] from the case of OneMax to Plateau_r, $r \ge 1$.

In what follows, we use the standard notion of stochastic domination. For two random variables X and Y, it is said that variable Y stochastically dominates X, denoted by $X \preceq Y$, if for all $r \in \mathbb{R}$ holds $\Pr(Y \le r) \le \Pr(X \le r)$.

Lemma 12. *Consider a run of Algorithm 1 with* $p_{\text{mut}} \le 1/2$*, applied to optimize a fitness function* Plateau_r, $r \ge 1$. *Then for each generation* $t \ge 0$ *and each* $i \in [\lambda]$*, we have* $Bin(n, 1/2) \preceq \text{OneMax}(P_t(i))$.

Lemma 12 is proved using Lemma 13 below. Let us define a supplementary probability distribution $\text{fp}(\cdot)$. Given λ non-negative values $f_1, \dots, f_\lambda$, a random variable v is distributed according to $\text{fp}(f_1, \dots, f_\lambda)$, if in the case that $f_i > 0$ for at least one $i \in [\lambda]$, we have $\Pr(v = i) = f_i / \sum_{j=1}^{\lambda} f_j$ for all $i \in [\lambda]$, and otherwise (i.e. if $f_i = 0$ for all $i \in [\lambda]$), we have $\Pr(v = i) = 1/\lambda$ for all $i \in [\lambda]$.

Lemma 13. *Suppose,* $\text{Om}_1 := \text{Om}(P_t(1))$, ..., $\text{Om}_\lambda := \text{Om}(P_t(\lambda))$, *and*

$$f_1 := \text{Plateau}_r(P_t(1)), \dots, f_\lambda := \text{Plateau}_r(P_t(\lambda)).$$

Let $u \in \{1, \dots, \lambda\}$ *be uniformly chosen and* $U = \text{Om}_u$. *Let* $v \sim \text{fp}(f_1, \dots, f_\lambda)$ *and* $V = \text{Om}_v$. *Then* $U \preceq V$.

Lemma 13 is analogous to Lemma 12 in [21], but it has a significant difference because the proportionate selection probability distribution is now defined according to Plateau_r, where individuals with different OneMax value may have equal fitness. This adds some technical issues to the proof.

Proof of Lemma 13. When $f_i = 0$ for all $i \in [\lambda]$, the claim follows immediately from the definition of fp$(\cdot)$. In what follows let us assume that $f_i > 0$ for at least one $i \in [\lambda]$. Without loss of generality we assume a lexicographic ordering of the individuals, such that $f_1 \leq f_2 \leq \cdots \leq f_\lambda$ and if the equality $f_i = f_{i+1}$ holds for some i then $\text{Om}_i \leq \text{Om}_{i+1}$. Then

$$\frac{1}{i}\sum_{j=1}^{i} f_j \leq \frac{1}{\lambda}\sum_{j=1}^{\lambda} f_j, \quad i = 1, \dots, \lambda. \tag{8}$$

Let $i(1), \dots, i(K)$ be the sequence of all *points of growth* for the sequence $f_1, f_2, \dots, f_\lambda$, defined as the increasing sequence of all such i that $f_i < f_{i+1}$. In view of the lexicographic ordering, this implies that for each $k \in [K]$ if $f_j = f_{i(k)+1}$ then $\text{Om}_j \geq \text{Om}_{i(k)+1}$. In view of (8), for any point of growth $i(k),\ k \in [K-1]$ we have

$$\Pr\left(\text{Om}_v < \text{Om}_{i(k)+1}\right) \leq \Pr\left(f_v < f_{i(k)+1}\right) = \Pr\left(f_v \leq f_{i(k)}\right) = \frac{\sum_{j=1}^{i(k)} f_j}{\sum_{j=1}^{\lambda} f_j} \leq \frac{i(k)}{\lambda}$$

$$= \Pr\left(f_u \leq f_{i(k)}\right) \leq \Pr\left(\text{Om}_u \leq \text{Om}_{i(k)}\right) = \Pr\left(\text{Om}_u < \text{Om}_{i(k)+1}\right). \tag{9}$$

Here the first equality holds because if the number of ones is less than $\text{Om}_{i(k)+1}$ then the value of Plateau_r is less than $f_{i(k)+1}$. (Otherwise for some j with $\text{Om}_j < \text{Om}_{i(k)+1}$ we would have $f_j \geq f_{i(k)+1}$ where the inequality can not be strict because by the definition of Plateau this would imply $\text{Om}_j > \text{Om}_{i(k)+1}$, but if the equality $f_j = f_{i(k)+1}$ holds then $\text{Om}_j \geq \text{Om}_{i(k)+1}$.) The last inequality $\Pr\left(f_u \leq f_{i(k)}\right) \leq \Pr\left(\text{Om}_u \leq \text{Om}_{i(k)}\right)$ holds because if the value of Plateau_r is less or equal to $f_{i(k)}$ then the number of ones is less or equal to $\text{Om}_{i(k)}$. (Otherwise for some j with $\text{Plateau}_r(P(j)) \leq f_{i(k)}$ we would have $\text{Om}(P(j)) > \text{Om}_{i(k)}$ but according to the lexicographical ordering of individuals, $P(i(k))$ has the highest number of ones among the individuals with fitness $f_{i(k)}$.)

Now let us prove that $\Pr(\text{Om}_v < \text{Om}_j) \leq \Pr(\text{Om}_u < \text{Om}_j)$ for the rest of the indices $j = i(k)+2, \dots, i(k+1)$, such that $i(k)+2 \leq i(k+1)$ and $k \leq K-1$. Here and below we assume that $i(0) = 0$ and $i(K) = n$. Note that for any such j we have

$$\Pr(\text{Om}_v < \text{Om}_j) = \Pr(\text{Om}_v \leq \text{Om}_{i(k)}) + \Pr(\text{Om}_{i(k)} < \text{Om}_v < \text{Om}_j), \tag{10}$$

$$\Pr(\text{Om}_u < \text{Om}_j) = \Pr(\text{Om}_u \leq \text{Om}_{i(k)}) + \Pr(\text{Om}_{i(k)} < \text{Om}_u < \text{Om}_j). \tag{11}$$

Also note that all individuals x, such that $\text{Om}_{i(k)+1} \leq |x| \leq \text{Om}_{i(k+1)}$, have equal fitness $f_{i(k)}$, and are equally probable under proportionate selection. Therefore

$$\Pr\left(\text{Om}_{i(k)} < \text{Om}_v < \text{Om}_j\right) = \frac{(j - i(k) - 2)\Pr(\text{Om}_{i(k)} < \text{Om}_v < \text{Om}_{i(k+1)})}{i(k+1) - i(k) - 2}.$$

For the uniform selection we also have

$$\Pr(\text{Om}_{i(k)} < \text{Om}_u < \text{Om}_j) = \frac{(j - i(k) - 2)\Pr(\text{Om}_{i(k)} < \text{Om}_u < \text{Om}_{i(k+1)})}{i(k+1) - i(k) - 2}.$$

Therefore, in view of (10) and (11),

$$\Pr(\mathrm{Om}_u < \mathrm{Om}_j) - \Pr(\mathrm{Om}_v < \mathrm{Om}_j) = \Pr(\mathrm{Om}_u \le \mathrm{Om}_{i(k)}) - \Pr(\mathrm{Om}_v \le \mathrm{Om}_{i(k)})$$

$$+\frac{(j-i(k)-2)\left(\Pr(\mathrm{Om}_{i(k)} < \mathrm{Om}_u < \mathrm{Om}_{i(k+1)}) - \Pr(\mathrm{Om}_{i(k)} < \mathrm{Om}_v < \mathrm{Om}_{i(k+1)})\right)}{i(k+1)-i(k)-2}$$

$$\ge \left(\Pr(\mathrm{Om}_u \le \mathrm{Om}_{i(k)}) - \Pr(\mathrm{Om}_v \le \mathrm{Om}_{i(k)})\right)\left(1 - \frac{j-i(k)-2}{i(k+1)-i(k)-2}\right) \ge 0. \quad (12)$$

The first inequality in (12) follows from (10) and (11) with $j = i(k+1)$. Indeed,

$$\begin{aligned}
&\Pr(\mathrm{Om}_{i(k)} < \mathrm{Om}_u < \mathrm{Om}_{i(k+1)}) - \Pr(\mathrm{Om}_{i(k)} < \mathrm{Om}_v < \mathrm{Om}_{i(k+1)})\\
&\quad = \Pr(\mathrm{Om}_u < \mathrm{Om}_{i(k+1)}) - \Pr(\mathrm{Om}_u \le \mathrm{Om}_{i(k)})\\
&\qquad - \Pr(\mathrm{Om}_v < \mathrm{Om}_{i(k+1)}) + \Pr(\mathrm{Om}_v \le \mathrm{Om}_{i(k)})\\
&\qquad\quad \ge \Pr(\mathrm{Om}_v \le \mathrm{Om}_{i(k)}) - \Pr(\mathrm{Om}_u \le \mathrm{Om}_{i(k)}),
\end{aligned}$$

because $\Pr(\mathrm{Om}_u < \mathrm{Om}_{i(k+1)}) \ge \Pr(\mathrm{Om}_v < \mathrm{Om}_{i(k+1)})$ according to (9).

The last inequality in (12) also follows from (9), since $\Pr(\mathrm{Om}_u \le \mathrm{Om}_{i(k)}) = \Pr(\mathrm{Om}_u < \mathrm{Om}_{i(k)+1})$ and $\Pr(\mathrm{Om}_v \le \mathrm{Om}_{i(k)}) = \Pr(\mathrm{Om}_v < \mathrm{Om}_{i(k)+1})$.

The required stochastic domination property holds since both U and V can take only the values from $\mathrm{Om}_1, \mathrm{Om}_2, \dots, \mathrm{Om}_\lambda$ and this whole range is considered. □

The proof of Lemma 12 almost literally coincides with the proof of Lemma 11 in [21], except that the distribution of the random variable v is now $v \sim \mathrm{fp}(\mathrm{Plateau}_r(P_t(1)), \dots, \mathrm{Plateau}_r(P_t(\lambda))$. Extension from $\chi = 1$ as it was in [21] to $\chi \le n/2$ is correct because Lemma 2 in [21] holds for this whole range of the mutation parameter. Alternatively we can refer to Proposition 6 in [27], or to Lemma 6.1 in [34] where this property of the standard mutation was independently proved.

Lemma 14. *Suppose Algorithm 1 with population size $\lambda = \mathrm{poly}(n)$, proportionate selection, bitwise mutation with mutation probability χ/n for a constant $\chi > \ln 2$, and fitness-proportionate selection is applied to $\mathrm{Plateau}_r(x)$ with constant $r \ge 1$. Then for any $\varepsilon \in (0,1)$, there exists a constant $c' > 0$ such that during $e^{c'n}$ generations with probability at least $1 - e^{-\Omega(n)}$ the following inequality holds*

$$\sum_{j=1}^{\lambda} \mathrm{Plateau}_r(P_t(j)) \ge \frac{\lambda(1-\varepsilon)(n-r)}{2}.$$

Proof. It follows by Lemma 12 that for any $i \in [\lambda]$ and any $t \ge 0$ the fitness of a particular solution $P_t(i)$ stochastically dominates a sum of n independent random variables, uniformly distributed on $\{0,1\}$. Hence using the multiplicative form of Chernoff bound, we see that $\mathrm{Om}(P_t(i)) \le (1-\varepsilon)n/2$ with probability at most $\exp\left(-(\varepsilon)^2 n/2\right)$ for any constant $\varepsilon \in (0,1)$. So with probability at least $1 - \lambda \exp\left(-(\varepsilon)^2 n/2\right) = 1 - e^{-\Omega(n)}$, by the union bound we have

$$\text{Om}(P_t(i)) \leq (1-\varepsilon)n/2 \quad \text{for all } \ i \in [\lambda]. \tag{13}$$

Define T to be the smallest t such that $\sum_{j=1}^{\lambda} |P_t(j)| \leq \lambda(n/2)(1-\varepsilon)$. Then for all $t \geq 0$, in view of (13), Pr $(T = t+1 \mid T > t) \leq e^{-c'n}$ for a constant $c' > 0$, which by the union bound implies that Pr $\left(T < e^{dn}\right) \leq e^{dn-c'n}$, which is just $e^{-\Omega(n)}$ for any constant $d < c'$. Therefore with probability $1 - e^{-\Omega(n)}$

$$\sum_{j=1}^{\lambda} \text{Plateau}_r(P_t(j)) \geq \sum_{j=1}^{\lambda} \frac{\text{Om}(P_t(j))(n-r)}{n} \geq \frac{\lambda(1-\varepsilon)(n-r)}{2}$$

during the first $e^{c'n}$ iterations for some constant $c' > 0$. □

Let us start with a lower bound for the time to reach the optimal solution.

Proposition 15. *There exists a constant $c > 0$ such that Algorithm 1 using population of size $\lambda = \text{poly}(n)$, proportionate selection, and the bitwise mutation given a constant mutation parameter $\chi > \ln 2$, obtains the optimum of* $\text{Plateau}_r(x)$ *during e^{cn} generations with probability at most $e^{-\Omega(n)}$.*

Proof. Let us take a sufficiently small constant ε such that $2/(1-\varepsilon) < e^{\chi}$. It follows by Lemma 14 that with probability at least $1 - e^{-\Omega(n)}$ the reproductive rate α_0 satisfies

$$\alpha_0 \leq \frac{\lambda n}{\sum_{j=1}^{\lambda} \text{Plateau}_r(P_t(j))} \leq \frac{2n}{(1-\varepsilon)(n-r)} \tag{14}$$

during $e^{c'n}$ iterations for some constant $c' > 0$. Thus we can choose $\varepsilon' = \varepsilon/2$ and the inequality $\alpha_0 \leq 2/(1-\varepsilon)$ will be satisfied for all sufficiently large n. Otherwise, with probability $e^{-\Omega(n)}$ we can pessimistically assume that the optimum is found before iteration $e^{c'n}$.

Conditioned that the pessimistic scenario does not happen, by Corollary 3, we conclude that the probability to optimize a function Plateau_r within $e^{c''n}$ generations is $\lambda e^{-\Omega(n)}$ for some constant $c'' > 0$. Therefore with $c := \min\{c', c''\}$, the proposition follows. □

In combinatorial optimization, a feasible solution y to a maximization problem is called a *ρ-approximate* solution if it satisfies the inequality $f(y) \geq \rho f^*$, where $0 < \rho < 1$ and f^* is the optimal objective function value. The main result of this section is Theorem 16 which establishes an exponential lower bound for the expected time till finding an approximate solution with a constant approximation ratio ρ specified therein. The proof of this bound is based on the negative drift theorem for populations [20] (reproduced here as Theorem 2) and Lemma 12.

Theorem 16. *Suppose that the EA from Algorithm 1 with population size* $\lambda = \text{poly}(n)$, *using the fitness-proportionate selection and bitwise mutation with any constant parameter* χ, $\ln 2 < \chi$, *is applied to* Plateau_r, *given a constant* $r \geq 1$. *Then there exists a constant* $C > 0$ *such that during the first* e^{Cn} *generations the EA obtains a* ρ*-approximate solution for a constant*

$$\rho > \max\left(0.5\exp\left(\chi - 2\chi^2 + 2\chi\sqrt{\chi^2 - \chi + \ln 2}\right), 1 - \frac{1}{\chi}\right)$$

only with probability $e^{-\Omega(n)}$.

Proof. Suppose that $T(n)$ is defined as in Theorem 2. (The specific function $a(n)$ for this definition will be chosen below.) Then until the iteration $T(n)$, all individuals in the EA population have a fitness less than $n - a(n)$. Now in view of Lemma 14, with probability at least $1 - e^{-\Omega(n)}$ the reproductive rate α_0 satisfies the inequality $\alpha_0 \leq 2(n - a(n))/(1-\varepsilon)n$ during the first $\min(T(n), e^{c'n})$ iterations for some constant $c' > 0$. Otherwise, with probability $e^{-\Omega(n)}$ we pessimistically assume that the optimum is found before iteration $e^{c'n}$.

When the pessimistic scenario does not take place, the upper bound $\alpha := \frac{2(n-a(n))}{(1-\varepsilon)n}$ satisfies condition 1) of Theorem 2 for any $a(n)$ and $b(n)$. Note that this α also satisfies the inequality $\ln\alpha = \ln 2 + \ln(1 - a(n)/n) - \ln(1-\varepsilon) < \ln 2 + \ln(1 - a(n)/n) + \varepsilon e$ for any $\varepsilon \in (0, 1/e)$, which follows from the Taylor expansion of e^{-x} with $x = \varepsilon e$.

Condition 2) of Theorem 2 requires that $\ln(\alpha)/\chi + \delta < 1$ for a constant $\delta > 0$. This condition is satisfied, given sufficiently small ε and sufficiently small $a(n) \leq n/3$ (to be defined later), such that $\ln(\alpha)/\chi < (\ln 2 + \ln(1 - a(n)/n) + \varepsilon e)/\chi < 1$, because $\chi > \ln(2)$.

To verify Condition 3) of Theorem 2, we define $\psi := (\ln 2 + \ln(1 - a(n)/n) + \varepsilon e)/\chi$ and consider

$$M(\chi) := \frac{1 - \sqrt{\psi(2-\psi)}}{2}.$$

Now we define $a(n)$ and $b(n)$ so that $0 < a(n)/n < b(n)/n < \min(1/5, M(\chi))$ are some constants to be defined later. Note that $a(n)/n < -\ln(1 - a(n)/n)$ and this inequality is strict because $a(n)/n > 0$. So it suffices to ensure that $-\ln(1 - a(n)/n) - \varepsilon e = \ln 2 - \psi\chi \leq \min(1/5, M(\chi))$ and set ε sufficiently small. Using the fact that $\sqrt{1-x} \leq 1 - x/2$, we can bound

$$\frac{1 - \psi(2-\psi)}{4} \leq M(\chi), \tag{15}$$

so in the case of $M(\chi) \leq 1/5$, it suffices to have $\ln 2 - \psi\chi = \frac{1-\psi(2-\psi)}{4}$. To ensure the latter equality we take $\psi = \psi_1(\chi) = 1 - 2\chi + 2\sqrt{\chi^2 - \chi + \ln 2}$, which implies our choice

$$\begin{aligned} a(n) &:= n(1-\varepsilon')\min\left(\frac{1}{5}, 1 - e^{\psi_1(\chi)\chi - \ln 2 - \varepsilon e}, \frac{1}{\chi}\right) \\ &= n(1-\varepsilon'/2)\min\left(1 - e^{\psi_1(\chi)\chi - \ln 2}, \frac{1}{\chi}\right), \end{aligned}$$

$$b(n) := n(1 - \varepsilon'/3) \min \left(1 - e^{\psi_1(\chi)\chi - \ln 2}, \frac{1}{\chi}\right),$$

where ε' is any constant from $(0,1)$ and the constant ε is chosen sufficiently small. Note that $a(n)/n$ and $b(n)/n$ are constants, independent of n and r, besides that, $b(n) \le n/\chi$. Now Condition 3) is satisfied, if $\varepsilon > 0$ is small enough.

Application of Theorem 2 implies that there exists a constant $c > 0$, such that

$$\Pr\left(T(n) \le e^{c(b(n)-a(n))}\right) = e^{-\Omega(b(n)-a(n))},$$

meaning that during the first $e^{c(b(n)-a(n))}$ iterations the EA obtains a search point with less than

$$z(\varepsilon') := n\left(1 - \frac{\varepsilon'}{2}\right) \min\left(1 - e^{\chi - 2\chi^2 + 2\chi\sqrt{\chi^2 - \chi + \ln 2} - \ln 2}, \frac{1}{\chi}\right) \tag{16}$$

zero-bits with probability at most $e^{-\Omega(n)}$. The same applies to finding a $(1-\omega)$-approximate solution with $\omega = z(\varepsilon')/n$ because any $(1-\omega)$-approximate solution x has $n - |x| \le \omega n$ zeros due to the inequality $|x| \ge \text{Plateau}_r(x) \ge (1-\omega)n$. To finish the proof, we can define the constant $C := 0.5c\varepsilon' \min(1 - e^{\psi_1(\chi)\chi - \ln 2}, 1/\chi)$, implying that $e^{c(b(n)-a(n))} \ge e^{Cn}$. □

For the typical setting of $\chi = 1$ Theorem 16 gives the inapproximability bound with $\rho > 0.97234$. This bound could be slightly improved if instead of the linear upper bound on $\sqrt{1-x}$ in (15) we used the precise expression. In particular, for $\chi = 1$ it would give the inapproximability bound with $\rho > 0.97137$. However the expression for ρ would become much more cumbersome.

Note that Theorem 16 applies to the fitness function OneMax as a special case of Plateau_r with $r = 1$. So in the context of OneMax, this proposition extends Theorem 10 from [21] in terms of the range of parameter χ. Besides that, Theorem 16 has a weaker population size requirement, compared to Corollary 13 from [22].

5. DISCUSSION

It is shown in [7] that under very general conditions, mentioned in the introduction, the (1+1) EA easily (in expected $\mathcal{O}(n \log n)$ time) reaches the plateau and then performs a random walk on it, quickly approaching to a "nearly-uniform" distribution. A similar behaviour may be expected from the elitist EAs like $(\mu+\lambda)$ EA, where the best incumbent, once having reached the plateau, will travel on it, until the optimum is found. One can expect that in the case of non-elitist EAs, if the selection is strong enough, the population will stick to the plateau and spread on it as well. In the present paper, we identify and use such behaviour only in Theorem 8.

In the case of bitwise mutation, our Theorem 7 relies on a scenario, where the EA quickly reaches the edge of the plateau and most of the remaining time (with seldom possible retreats from the plateau) spends on the attempts to hit the optimum by "large" mutations, inverting up to $n - r$ zero-bits. Theorem 8, applicable to a wider class of mutation operators, relies on a more graduate scenario, where the search may consist of multiple stages, each one starting from an "arbitrary bad" population, then reaches the edge of the plateau in expected $\mathcal{O}(n \log n)$ time and tries to reach the optimum by making r sequential single-bit mutations, reducing the Hamming distance to the optimum by 1 in each EA iteration. A quadratic number of such journeys on the plateau are possible with a constant probability. Theorem 10 is even less demanding to the properties of mutation operators, but demanding very high selection pressure. It is likely that the runtime bound in this case may be significantly improved, since with such a high selection pressure the non-elitist EA becomes close to the (1+1) EA. While Theorems 7–10 deal with the tournament or (μ, λ)-selection, Theorem 11 shows that a similar situation may be observed in the case of fitness proportionate selection, if the mutation probability is reduced to $\Theta(1/n^2)$.

If the mutation probability is set to $\chi/n, \chi > \ln 2$, the EA with the fitness proportionate selection is likely to spend exponential time on the way to the plateau, as it is shown in Proposition 15. So there remains a significant gap with p_{mut} greater than $\Theta(1/n^2)$ but not more than $\ln 2$ where the possibility for expected polynomial runtime is unclear. This gap is present in the special case of $r = 1$, i.e. the fitness OneMax, which is also known from the literature – see the gap between the positive results from Theorem 24 in [19] and the negative result from Theorem 14 in [22].

6. CONCLUSIONS

This paper demonstrates the results, which are accessible by the available tools of runtime analysis. It also naturally leads to several questions for further research, some of which may require to develop principially new tools for EA analysis:

- What are the leading constants in the obtained upper bounds?
- What lower bounds can complement the obtained upper bounds?
- Under what conditions on selection pressure is it possible to transfer the tight results on (1+1) EA from [7] to the non-elitist EAs?
- How to extend the detailed runtime analysis to the Royal Road and Royal Staircase fitness functions [10] which have multiple plateaus?
- Would the genetic algorithms, which use the crossover operators, have any advantage over the mutation-based EAs considered in this paper?

Funding: This research was funded by the Mathematical Center in Akademgorodok under the agreement № 075-15-2025-349 with the Ministry of Science and Higher Education of the Russian Federation.

REFERENCES

[1] P. A. Borisovsky and M. S. Zavolovskaya, "Experimental comparison of two evolutionary algorithms for the independent set problem," in *Applications of Evolutionary Computing*, S. Cagnoni, C. G. Johnson, J. J. R. Cardalda, E. Marchiori, D. W. Corne, J.-A. Meyer, J. Gottlieb, M. Middendorf, A. Guillot, G. R. Raidl, and E. Hart, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 154–164.

[2] A. M. Sutton, A. E. Howe, and L. D. Whitley, “Directed plateau search for Max-k-Sat,” in *Proceedings of the Third Annual Symposium on Combinatorial Search, SOCS 2010, Stone Mountain, Atlanta, Georgia, USA, July 8-10, 2010*, 2010.

[3] D. Whitley, A. Howe, and D. Hains, “Greedy or not? best improving versus first improving stochastic local search for MaxSat,” in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, ser. AAAI’13. AAAI Press, 2013, pp. 940–946.

[4] E. F. de Souza, C. Le Goues, and C. G. Camilo-Junior, “A novel fitness function for automated program repair based on source code checkpoints,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO’18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1443–1450.

[5] I. O. Oduntan, M. Toulouse, R. Baumgartner, C. Bowman, R. L. Somorjai, and T. G. Crainic, “A multilevel tabu search algorithm for the feature selection problem in biomedical data,” *Comput. Math. Appl.*, vol. 55, pp. 1019–1033, 2008.

[6] F. A. Kondrashov, I. B. Rogozin, Y. I. Wolf, and E. V. Koonin, “Selection in the evolution of gene duplications,” *Genome biology*, vol. 3, no. 2, pp. 427–427, 2002.

[7] D. Antipov and B. Doerr, “Precise runtime analysis for plateau functions,” *ACM Trans. Evol. Learn. Optim.*, vol. 1, no. 4, Oct. 2021. doi: 10.1145/3469800

[8] T. Jansen and I. Wegener, “Evolutionary algorithms - how to cope with plateaus of constant fitness and when to reject strings of the same fitness,” *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 6, pp. 589–599, 2001. doi: 10.1109/4235.974841

[9] A. Lissovoi, D. Sudholt, M. Wagner, and C. Zarges, “Theoretical results on bet-and-run as an initialisation strategy,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO ’17. New York, NY, USA: Association for Computing Machinery, 2017. doi: 10.1145/3071178.3071329 pp. 857–864.

[10] E. van Nimwegen and J. Crutchfield, “Optimizing epochal evolutionary search population-size independent theory,” *Computer Methods in Applied Mech. and Engineering*, vol. 186, no. 2-4, pp. 171–194, 2000.

[11] P. K. Lehre and C. Witt, “Tail bounds on hitting times of randomized search heuristics using variable drift analysis,” *Combinatorics, Probability and Computing*, vol. 30, no. 4, pp. 550–569, 2021. doi: 10.1017/S0963548320000565

[12] T. Jansen and C. Zarges, “Performance analysis of randomised search heuristics operating with a fixed budget,” *Theoretical Computer Science*, vol. 545, pp. 39–58, 2014. doi: https://doi.org/10.1016/j.tcs.2013.06.007

[13] P. K. Lehre and C. Witt, “Black-box search by unbiased variation,” *Algorithmica*, vol. 64, pp. 623–642, 2012.

[14] B. Doerr and A. J. Kelley, “Fourier analysis meets runtime analysis: Precise runtimes on plateaus,” *Algorithmica*, vol. 86, no. 8, pp. 2479–2518, 2024. doi: 10.1007/S00453-024-01232-5

[15] J. Jägerskupper and T. Storch, “When the plus strategy outperforms the comma strategy and when not,” in *2007 IEEE Symposium on Foundations of Computational Intelligence*, 2007, pp. 25–32.

[16] J. Jorritsma, J. Lengler, and D. Sudholt, “Comma selection outperforms plus selection on OneMax with randomly planted optima,” *Algorithmica*, 2025. doi: 10.1007/s00453-025-01330-y Early access.

[17] D. Corus, D. Dang, A. V. Eremeev, and P. K. Lehre, “Level-based analysis of genetic algorithms and other search processes,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 707–719, Oct 2018.

[18] A. Eremeev, “Hitting times of local and global optima in genetic algorithms with very high selection pressure,” *Yugosl. J. Oper. Res.*, vol. 27, no. 3, pp. 323–339, 2017.

[19] B. Doerr and T. Kötzing, "Multiplicative up-drift," *Algorithmica*, vol. 83, pp. 1–42, 10 2021. doi: 10.1007/s00453-020-00775-7

[20] P. K. Lehre, "Negative drift in populations," in *Proc. of PPSN'10*, 2010, pp. 244–253.

[21] B. Doerr, "Lower bounds for non-elitist evolutionary algorithms via negative multiplicative drift," *Evolutionary Computation*, vol. 29, no. 2, pp. 305–329, 06 2021.

[22] P. K. Lehre, "Fitness-levels for non-elitist populations," in *Proc. of GECCO'11*, 2011, pp. 2075–2082.

[23] A. Eremeev, "On proportions of fit individuals in population of mutation-based evolutionary algorithm with tournament selection," *Evolutionary Computation*, vol. 26, no. 2, pp. 269–297, 2018.

[24] A. V. Eremeev, "On non-elitist evolutionary algorithms optimizing fitness functions with a plateau," in *Mathematical Optimization Theory and Operations Research*, A. Kononov, M. Khachay, V. A. Kalyagin, and P. Pardalos, Eds. Cham: Springer International Publishing, 2020, pp. 329–342.

[25] D.-C. Dang and P. K. Lehre, "Runtime analysis of non-elitist populations: From classical optimisation to partial information," *Algorithmica*, vol. 75, no. 3, pp. 428–461, 2016.

[26] S. Droste, T. Jansen, and I. Wegener, "Upper and lower bounds for randomized search heuristics in black-box optimization," *Theory of Computing Systems*, vol. 39, no. 4, pp. 525–544, Jul. 2006. doi: 10.1007/s00224-004-1177-z

[27] P. Borisovsky and A. Eremeev, "Comparing evolutionary algorithms to the (1+1)-EA," *Theoretical Computer Science*, vol. 403, no. 1, pp. 33–41, 2008.

[28] A. V. Eremeev and A. V. Spirov, "Modeling SELEX for regulatory regions using royal road and royal staircase fitness functions," *Biosystems*, vol. 200, p. 104312, 2021. doi: 10.1016/j.biosystems.2020.104312

[29] D.-C. Dang, A. Eremeev, and P. K. Lehre, "Runtime analysis of fitness-proportionate selection on linear functions," *ArXiv*, vol. 1908.08686 [cs.NE], 2019.

[30] B. Doerr and T. Kötzing, "Multiplicative up-drift," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, 2019, pp. 1470–1478.

[31] D.-C. Dang and P. K. Lehre, "Self-adaptation of mutation rates in non-elitist populations," in *Proc. of PPSN XIV*. Springer, 2016, pp. 803–813, (arXiv:1606.05551).

[32] B. V. Gnedenko, *Theory of probability*. Gordon and Breach, 1997.

[33] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[34] C. Witt, "Tight bounds on the optimization time of a randomized search heuristic on linear functions," *Combinatorics, Probability and Computing*, vol. 22, pp. 294–318, 3 2013.

APPENDIX A

Proof of Theorem 1. The proof consists in comparing the expected number of individuals of P_{t+1} in each cumulative set of levels $A_{\geq j}$ to the probability of hitting a subset $A_{\geq j}$ by the mutant x^{t+1}, scaled by the factor λ.

Consider the sequence of identically distributed indicator variables $I_1^i, I_2^i, ..., I_\lambda^i$, where $I_l^i = 1$ if the l-th individual in the population P_t belongs to $A_{\geq i}$, otherwise $I_l^i = 0$. Denote $z_i^{(t)} := \sum_{l=1}^{\lambda} \xi_l^i/\lambda$, $i \in [m]$, the proportion of genotypes from $A_{\geq i}$ in P_t. Then

$$\mathbf{E}\left[z_i^{(t)}\right] = \sum_{l=1}^{\lambda} \mathbf{E}\left[I_l^i\right]/\lambda = \sum_{l=1}^{\lambda} \Pr\{P_t(1) \in A_{\geq i}\}/\lambda = \Pr\{P_t(1) \in A_{\geq i}\}. \quad (17)$$

Therefore, in what follows, we can consider the expected proportions of individuals at certain levels instead of the probabilities of obtaining individuals of these levels in the population of Algorithm 1, if this is more convenient for us. Also, it will be convenient for us to define $z^{(t)}_{m+1} := 0$ and $z^{(t)}_0 := 1$ and $\mathbf{z}^{(t)} = (z^{(t)}_1, \dots, z^{(t)}_m)$.

Note that if the current population is described by a vector $\mathbf{z}^{(t)} = \mathbf{z}$, then an individual obtained by selection and mutation would belong to $A_{\geq j}$ with a conditional probability

$$\Pr\{P_{t+1}(1) \in A_{\geq j} | \mathbf{z}^{(t)} = \mathbf{z}\}$$

$$= \sum_{i=1}^{m} \sum_{x \in A_i} \Pr(\mathtt{mutate}(x) \in A_{\geq j}) \Pr(P_t(\mathtt{select}(P_t)) = x | \mathbf{z}^{(t)} = \mathbf{z}). \quad (18)$$

The alternative mutation operator $\mathtt{mutate}_0$ gives us a family of cumulative transition probabilities $\alpha_{ij} := \Pr\{\mathtt{mutate}_0(x) \in A_{\geq j}\}$ for any $x \in A_i,\ i, j \in [m]$. Parameters α_{ij} are well-defined because $\Pr\{\mathtt{mutate}_0(x) \in A_{\geq j}\}$ is the same for any $x \in A_i,\ i, j \in [m]$ due to the monotonicity of the mutation operator $\mathtt{mutate}_0$.

Expression (18) and the definitions of α_{ij}, together with the inequality (2), imply for all $j \in [m]$:

$$\Pr\{P_{t+1}(1) \in A_{\geq j} | \mathbf{z}^{(t)} = \mathbf{z}\} \geq \sum_{i=1}^{m} \alpha_{ij} \sum_{x \in A_i} \Pr\{P_t(\mathtt{select}(P_t)) = x | \mathbf{z}^{(t)} = \mathbf{z}\}$$

$$= \sum_{i=1}^{m} \alpha_{ij} \Pr\{P_t(\mathtt{select}(P_t)) \in A_i | \mathbf{z}^{(t)} = \mathbf{z}\}. \quad (19)$$

Note that if operator $\mathtt{mutate}$ is identical with $\mathtt{mutate}_0$, then (19) holds as equality. This leads to the equality:

$$\Pr\{P_{t+1}(1) \in A_{\geq j} | \mathbf{z}^{(t)} = \mathbf{z}\} \geq \sum_{i=1}^{m} \alpha_{ij} (\beta(z^{(t)}_i) - \beta(z^{(t)}_{i+1})).$$

Here and below, we denote $\beta(\gamma) := \beta(\gamma, P_t)$ to simplify the notation from Equation (1). Besides that, let us denote the set of all possible population vectors **z** by

$$Z_\lambda := \{\mathbf{z} \in \{0, 1/\lambda, 2/\lambda, \dots, 1\}^m : z_i \geq z_{i+1}, i = 1 \dots, m-1\}.$$

Using the total probability formula we bound the unconditional probability

$$\Pr\{P_{t+1}(1) \in A_{\geq j}\} = \sum_{\mathbf{z} \in Z_\lambda} \Pr\{P_{t+1}(1) \in A_{\geq j} | \mathbf{z}^{(t)} = \mathbf{z}\} \Pr\{\mathbf{z}^{(t)} = \mathbf{z}\} \quad (20)$$

$$\geq \sum_{\mathbf{z} \in Z_\lambda} \sum_{i=1}^{m} \alpha_{ij} (\beta(z^{(t)}_i) - \beta(z^{(t)}_{i+1})) \Pr\{\mathbf{z}^{(t)} = \mathbf{z}\} = \sum_{i=1}^{m} \alpha_{ij} \mathbf{E}\left[\beta(z^{(t)}_i) - \beta(z^{(t)}_{i+1})\right]$$

$$= \alpha_{1j} \mathbf{E}\left[\beta(z^{(t)}_0)\right] - \alpha_{mj} \mathbf{E}\left[\beta(z^{(t)}_{m+1})\right] + \sum_{i=1}^{m} (\alpha_{ij} - \alpha_{i-1,j}) \mathbf{E}\left[\beta(z^{(t)}_i)\right], \quad (21)$$

where the last equality is obtained by regrouping the summation terms. Equation (17) implies that $\mathbf{E}\left[z_j^{(t+1)}\right] = \Pr\{P_{t+1}(1) \in A_{\geq j}\}$. Consequently, since $\beta(z_{m+1}^{(t)}) = \beta(0) = 0$ and $\beta(z_0^{(t)}) = \beta(1) = 1$, we have

$$\Pr\{P_{t+1}(1) \in A_{\geq j}\} \geq \alpha_{0j} + \sum_{i=1}^{m}(\alpha_{i,j} - \alpha_{i-1,j})\,\mathbf{E}\left[\beta(z_i^{(t)})\right]. \tag{22}$$

Again, if `mutate` is identical with $\texttt{mutate}_0$, then (20)-(22) hold as equality.

On one hand, the random offspring x^{t+1} may be considered as a result of one iteration of Algorithm 1 with $\lambda = 1$ and $\texttt{mutate} = \texttt{mutate}_0$. Clearly, under any selection mechanism, in case of $\lambda = 1$ we have $\texttt{select}(P) \equiv 1$, $z_i^{(t)} \in \{0,1\}$, depending whether $x^t \in A_{\geq i}$ or not. By the definition, $\beta(0) = 0$ and $\beta(1) = 1$ so $\mathbf{E}\left[\beta(z_i^{(t)})\right] = \Pr\{x^t \in A_{\geq i}\}$ and

$$\Pr\{x^{t+1} \in A_{\geq j}\} = \alpha_{0j} + \sum_{i=1}^{m}(\alpha_{i,j} - \alpha_{i-1,j})\Pr(x^t \in A_{\geq i}), \tag{23}$$

since Equation 22 holds for any selection mechanism and any λ.

On the other hand, for a monotone selection mechanism we have $\beta(\gamma) \geq \gamma$ at any $\gamma \in [0,1]$, besides that, in view of monotonicity of mutation operator $\texttt{mutate}_0$, we have $\alpha_{i,j} - \alpha_{i-1,j} \geq 0$ for all i, j. So substitution of $\mathbf{E}\left[\beta(z_i^{(t)})\right]$ by a smaller or equal value $\mathbf{E}\left[z_i^{(t)}\right]$ in Equation (22) gives a valid inequality

$$\Pr\{P_{t+1}(1) \in A_{\geq j}\} \geq \alpha_{0j} + \sum_{i=1}^{m}(\alpha_{i,j} - \alpha_{i-1,j})\,\mathbf{E}\left[z_i^{(t)}\right]. \tag{24}$$

Equality (17) allows us to replace $\mathbf{E}\left[z_i^{(t)}\right]$ by $\Pr\{P_t(1) \in A_{\geq i}\}$ in (24). After that, comparison of (23) and (24) gives the required inequality for each $j \in [m]$. $\square$

The proof of Theorem 6 For any $t = 0, 1, \dots$ let the event E_i^t, $i = 1, \dots, \lambda$, consist in fulfilment of the following two conditions when the i-th offspring is being produced:

1. The parent x is chosen from the highest level A_j occupied by population P_t.
2. The mutation operator applied to x produces a genotype in $A_{\geq j+1}$.

Let p denote the probability of the union of events E_i^t, $i = 1, \dots, \lambda$. In what follows, we will construct a lower bound $\ell \leq p$, which holds for any population P_t. According to the outline of the EA, $\Pr(E_1^t) = \cdots = \Pr(E_\lambda^t)$. Let us denote this probability by q. Note that q is bounded from below by $s_*\beta_0$, where β_0 is a lower bound for the probability of selecting an individual from the highest level subset A_j, occupied by the individuals of a population P_t.

Given a population P_t, the events $E^t_j,\ j = 1, \dots, \lambda$, are independent, so $p \geq 1 - (1 - q)^\lambda \geq 1 - e^{-q\lambda}$. In what follows we shall use the fact that the assumption on λ in the formulation of this theorem implies that

$$\lambda \geq \frac{1}{s_* \beta_0} \geq 1/q. \tag{25}$$

To bound probability p from below, we first note that for any $z \in [0, 1]$ holds $1 - \frac{z}{e} \geq e^{-z}$. Assume $z = e^{-q\lambda+1}$. Then in view of inequality (25), $z \leq 1$, and consequently,

$$p \geq \exp\left\{-e^{1-q\lambda}\right\} \geq \exp\left\{-e^{1-s_*\beta_0\lambda}\right\} =: \ell. \tag{26}$$

In what follows, we will use the right-hand side ℓ defined here as a lower bound for p.

For any $t = 0, 1, \dots$ let us define the event $\mathcal{E}_t := E^t_1 + \dots + E^t_\lambda$. Then the probability to reach the target level A_m in a series of at most m iterations is lower bounded by $\Pr(\mathcal{E}_0 \& \dots \& \mathcal{E}_{m-1})$ and

$$\Pr(\mathcal{E}_0 \& \dots \& \mathcal{E}_{m-1}) = \Pr(\mathcal{E}_0) \prod_{t=1}^{m-1} \Pr(\mathcal{E}_t | \mathcal{E}_0 \& \dots \& \mathcal{E}_{t-1}) \geq \ell^m = \exp\left\{-me^{1-s_*\beta_0\lambda}\right\}. \tag{27}$$

In the case of (μ, λ)-selection, it is easy to see that $\beta_0 = 1/\mu$ is a valid lower bound for the probability of selecting an individual from the highest level subset A_j, occupied by the individuals of a population P_t. In this case we have $1 - s_*\beta_0\lambda \leq -\ln m$ as follows from the assumption on population size.

In the case of tournament selection with $k \geq c\lambda$, the probability of selecting an individual from the highest level subset A_j, occupied by the individuals of a population P_t is not less than $1-(1 - 1/\lambda)^k \geq 1-(1-1/\lambda)^{c\lambda} \geq 1-e^{-c}$. So we can assume $\beta_0 = 1 - e^{-c}$. In this case again we have $1-s_*\beta_0\lambda \leq -\ln m$ as follows from the assumption on population size $\lambda \geq (1 + \ln m)/(s_*(1 - e^{-c}))$.

Therefore, for both selection mechanisms we have

$$\Pr(\mathcal{E}_0 \& \dots \& \mathcal{E}_{m-1}) \geq \exp\left\{-me^{-\ln m}\right\} = 1/e. \ \square \tag{28}$$

APPENDIX B

This appendix contains formulations of two technical results from the literature, which are used in the paper.

Lemma 17. (Lemma 3 in [31]) *For any $\varepsilon \in (0, 1)$ and $\chi > 0$, if $n \geq (\chi + \varepsilon)(\chi/\varepsilon)$ then*

$$(1 - \varepsilon)e^{-\chi} \leq \left(1 - \frac{\chi}{n}\right)^n \leq e^{-\chi}.$$

Lemma 18. (Lemma 5 in [22]) *The reproductive rate of k-tournament selection is at most k. If $k \geq (1 + \Delta)/p_0$ for any constants $p_0 \in (0, 1)$ and $\Delta > 0$, then there exist constants $\gamma_0, \delta > 0$ such that $p_0\beta(\gamma, P) > \gamma_0(1 + \delta)$ for any $\gamma \in (0, \gamma_0]$ and any $P \in (\mathcal{X})^\lambda$.*