



Lower Bounds for Non-elitist Evolutionary Algorithms via Negative Multiplicative Drift

Benjamin Doerr^(✉)

Laboratoire d'Informatique (LIX), CNRS, École Polytechnique,
Institut Polytechnique de Paris, Palaiseau, France
doerr@lix.polytechnique.fr

Abstract. A decent number of lower bounds for non-elitist population-based evolutionary algorithms has been shown by now. Most of them are technically demanding due to the (hard to avoid) use of negative drift theorems – general results which translate an expected progress away from the target into a high hitting time.

We propose a simple negative drift theorem for multiplicative drift scenarios and show that it can simplify existing analyses. We discuss in more detail Lehre's (PPSN 2010) *negative drift in populations* method, one of the most general tools to prove lower bounds on the runtime of non-elitist mutation-based evolutionary algorithms for discrete search spaces. Together with other arguments, we obtain an alternative and simpler proof, which also strengthens and simplifies this method. In particular, now only three of the five technical conditions of the previous result have to be verified. The lower bounds we obtain are explicit instead of only asymptotic. This allows to compute concrete lower bounds for concrete algorithms, but also enables us to show that super-polynomial runtimes appear already when the reproduction rate is only a $(1 - \omega(n^{-1/2}))$ factor below the threshold. As one particular result, we apply this method and a novel domination argument to show an exponential lower bound for the runtime of the mutation-only simple GA on ONEMAX for arbitrary population size.

Keywords: Runtime analysis · Drift analysis · Lower bounds · Population-based algorithms · Theory · Discrete optimization

1 Introduction

Lower bounds for the runtimes of evolutionary algorithms are important as they can warn the algorithm user that certain algorithms or certain parameter settings will not lead to good solutions in acceptable time. Unfortunately, the existing results in this direction, for non-elitist algorithms in particular, are very technical. In the case of Lehre's powerful *negative drift in populations* method [24], this also renders the method difficult to use.

One reason for this high complexity is the use of drift analysis, which seems hard to circumvent. Drift analysis [26] is a set of tools that all try to derive useful information on a hitting time (e.g., the first time a solution of a certain quality is found) from information on the expected progress in one iteration. The hope is that the progress in a single iteration can be analyzed with only moderate difficulty and then the drift theorem does the remaining work. While more direct analysis methods exist and have been successfully used for simple algorithms, for population-based algorithms and in particular non-elitist ones, it is hard to imagine that the complicated population dynamics can be captured in proofs not using more advanced tools such as drift analysis.

Drift analysis has been used with great success to prove upper bounds on runtimes of evolutionary algorithms. Tools such as the additive [19], multiplicative [13], and variable drift theorem [22, 28] all allow to easily obtain an upper bound on a hitting time solely from the expected progress in one iteration. Unfortunately, proving matching lower bounds is much harder since here the drift theorems also require additional technical assumptions on the distribution of the progress in one iteration. This is even more true in the case of so-called *negative drift*, where the drift is away from the target and we aim at proving a high lower bound on the hitting time.

In this work, we propose a very simple negative drift theorem for the case of multiplicative drift (Lemma 1). We briefly show that this result can simplify two classic lower bound analyses (Sect. 2).

In more detail, we use the new drift theorem (and some more arguments) to rework Lehre's *negative drift in populations* method [24]. This highly general analysis method allows to show exponential lower bounds on the runtime of a large class of evolutionary algorithms solely by comparing the reproduction rate of individuals in the population with a threshold that depends only on the mutation rate.

The downside of Lehre's method is that both the result and its proof is very technical. To apply the general result (and not the specialization to algorithms using standard bit mutation), five technical conditions need to be verified, which requires the user to choose suitable values for six different constants; these have an influence on the lower bound one obtains. This renders the method of Lehre hard to use. Among the 54 citations to [24] (according to Google scholar on June 9, 2020), only the two works [6, 25] apply this method. To hopefully ease future analyses of negative drift in populations, we revisit this method and obtain the following improvements.

A Simpler Result: We manage to show essentially the same lower bounds by only verifying three of the five conditions Lehre was using (Theorem 2 and 3). This also reduces the number of constants one needs to choose from six to four.

A Non-asymptotic Result: Our general tool proves explicit lower bounds, that is, free from asymptotic notation or unspecified constants. Consequently, our specialization to algorithms using standard bit mutation (Theorem 4) also gives explicit bounds. This allows one to prove concrete bounds for specific situations

(e.g., that the (μ, λ) EA with $\lambda = 2\mu$ needs more than 13 million fitness evaluations to find a unique optimum of problem defined over bit strings of length $n = 500$, see the example following Theorem 4) and gives more fine-grained theoretical results (by choosing Lehre's constant δ as suitable function of the problems size, we show that a super-polynomial runtime behavior is observed already when the reproduction rate is only a $(1 - \omega(n^{1/2}))$ factor below the threshold, see Corollary 5).

A Simple Proof: Besides the important aspect that a proof guarantees the result to be mathematically correct, an understandable proof can also tell us why a result is correct and give further insights into working principles of algorithms. While every reader will have a different view on how the ideal proof looks like, we felt that Lehre's proof, combining several deep and abstract tools such as multi-type branching processes, eigenvalue arguments, and Hajek's drift theorem [17], does not easily give a broader understanding of the proof mechanics and the working principles of the algorithms analyzed. Our proof, based on a simple potential function argument together with our negative drift theorem, hopefully is more accessible.

Finally, we analyze an algorithm using fitness proportionate selection. The negative drift in populations method is not immediately applicable to such algorithms since it is hard to provide a general unconditional upper bound on the reproduction rate. We show that at all times all search points are at least as good (in the stochastic domination sense) as random search points. This gives a simple proof of an exponential lower bound for the mutation-only simple genetic algorithm with arbitrary population size optimizing the simple ONEMAX benchmark, improving over the mildly sub-exponential lower bound in [29] and the exponential lower bound only for large population sizes in [25].

1.1 Related Works

A number of different drift theorems dealing with negative drift have been proven so far, among other, in [18, 23, 27, 31, 32, 34, 35, 39]. They all require some additional assumptions on the distribution of the one-step progress, which makes them non-trivial to use. We refer to [26, Section 2.4.3] for more details. Another approach to negative drift was used in [2, 8, 9]. There the original process was transformed suitably (via an exponential function), but in a way that the drift of the new process still is negative or at most very slowly approaches the target. To this transformed process the lower bound version of the additive drift theorem [19] was applied, which gave large lower bounds since the target, due to the exponential rescaling, now was far from the starting point of the process.

In terms of lower bounds for non-elitist algorithms, besides Lehre's general result [24], the following results for particular algorithms exist (always, n is the problem size, ε can be any positive constant, and $e \approx 2.718$ is the base of the natural logarithm). Jägerskupper and Storch [21, Theorem 1] showed that the $(1, \lambda)$ EA with $\lambda \leq \frac{1}{14} \ln(n)$ is inefficient on any pseudo-Boolean function with a unique optimum. The asymptotically tight condition $\lambda \leq (1 - \varepsilon) \log_{\frac{e}{e-1}} n$ to

yield a super-polynomial runtime was given by Rowe and Sudholt [35]. Happ, Johannsen, Klein, and Neumann [18] showed that two simple (1+1)-type hill-climbers with fitness proportionate selection cannot optimize efficiently any linear function with positive weights. Neumann, Oliveto, and Witt [29] showed that a mutation-only variant of the simple genetic algorithm (simple GA) with fitness proportionate selection is inefficient on the ONEMAX function when the population size μ is at most polynomial, and it is inefficient on any pseudo-Boolean function with unique global optimum when $\mu \leq \frac{1}{4} \ln(n)$. The mildly subexponential lower bound for ONEMAX was improved to an exponential lower bound by Lehre [25], but only for $\mu \geq n^3$. In a series of remarkable works up to [34], Oliveto and Witt showed that the true simple GA using crossover cannot optimize ONEMAX efficiently when $\mu \leq n^{\frac{1}{4} - \epsilon}$. None of these results gives an explicit lower bound or specifies the base of the exponential function. In [2], an explicit lower bound for the runtime of the (μ, λ) EA is proven (but stated only in the proof of Theorem 3.1 in [2]). Section 3 of [2] bears some similarity with ours, in fact, one can argue that our work extends [2, Section 3] from a particular algorithm to the general class of population-based processes regarded by Lehre [24] (where, naturally, [2] did not have the negative multiplicative drift result and therefore did not obtain bounds that hold with high probability).

2 Negative Multiplicative Drift

The following elementary result allows to prove lower bounds on the time to reach a target in the presence of multiplicative drift away from the target. While looking innocent, it has the potential to replace more the complicated lower bound arguments previously used in analyses of non-elitist algorithms such as *simplified drift theorems* ([29, Theorem 1], [33, Theorem 22], [34, Theorem 2]). We discuss this briefly at the end of this section.

Lemma 1 (Negative multiplicative drift theorem). *Let X_0, X_1, \dots be a random process in a finite subset of $\mathbb{R}_{\geq 0}$. Assume that there are $\Delta, \delta > 0$ such that for each $t \geq 0$, the following multiplicative drift condition with additive disturbance holds:*

$$E[X_{t+1}] \leq (1 - \delta)E[X_t] + \Delta. \tag{1}$$

Assume further that $E[X_0] \leq \frac{\Delta}{\delta}$. Then the following two assertions hold.

- For all $t \geq 0$, $E[X_t] \leq \frac{\Delta}{\delta}$.
- Let $M > \frac{\Delta}{\delta}$ and $T = \min\{t \geq 0 \mid X_t \geq M\}$. Then for all integers $L \geq 0$,

$$\Pr[T \geq L] \geq 1 - L \frac{\Delta}{\delta M},$$

and $E[T] \geq \frac{\delta M}{2\Delta} - \frac{1}{2}$.

The proof is an easy computation of expectations and an application of Markov’s inequality similar to the direct proof of the multiplicative drift theorem in [12]. We do not see a reason why the result should not also hold for processes taking more than a finite number of values, but since we are only interested in the finite setting, we spare us the more complicated world of continuous probability spaces.

Proof (of Lemma 1). If $E[X_t] \leq \frac{\Delta}{\delta}$, then $E[X_{t+1}] \leq (1 - \delta)E[X_t] + \Delta \leq (1 - \delta)\frac{\Delta}{\delta} = \frac{\Delta}{\delta}$ by (1). Hence the first claim follows by induction. To prove the second claim, we compute

$$\Pr[T < L] \leq \Pr[X_0 + \dots + X_{L-1} \geq M] \leq \frac{E[X_0 + \dots + X_{L-1}]}{M} \leq \frac{L\Delta}{\delta M},$$

where the middle inequality follows from Markov’s inequality and the fact that the X_t by assumption are all non-negative. From this estimate, using the shorthand $s = \lfloor \frac{\delta M}{\Delta} \rfloor$, we compute $E[T] = \sum_{t=1}^{\infty} \Pr[T \geq t] \geq \sum_{t=1}^s (1 - \frac{t\Delta}{\delta M}) = s - \frac{1}{2}s(s+1)\frac{\Delta}{\delta M} \geq \frac{\delta M}{2\Delta} - \frac{1}{2}$, where the first equality is a standard way to express the expectation of a random variable taking non-negative integral values and the last inequality is an elementary computation omitted here. \square

We note that in the typical application of this result (as in the proof of Theorem 2 below), we expect to see the condition that for all $t \geq 0$,

$$E[X_{t+1} \mid X_t] \leq (1 - \delta)X_t + \Delta. \tag{2}$$

Clearly, this condition implies (1) by the law of total expectation.

We now argue that our negative multiplicative drift theorem is likely to find applications beyond ours to the negative drift in populations method in the following section. To this aim, we regard two classic lower bound analyses of non-elitist algorithms and point out where our drift theorem would have eased the analysis.

In [29], Neumann, Oliveto, and Witt show that the variant of the simple genetic algorithm (simple GA) not using crossover needs time $2^{n^{1-O(1/\log \log n)}}$ to optimize the simple ONEMAX benchmark. The key argument in [29] is as follows. The potential X_t of the population $P^{(t)}$ in iteration t is defined as $X_t = \sum_{x \in P^{(t)}} 8^{\text{ONEMAX}(x)}$. For this potential, it is shown [29, Lemma 7] that if $X_t \geq 8^{0.996n}$, then $E[X_{t+1}] \leq (1 - \delta)X_t$ for some constant $\delta > 0$. By bluntly estimating $E[X_{t+1}]$ in the case that $X_t < 8^{0.996n}$, this bound could easily be extended to $E[X_{t+1} \mid X_t] \leq (1 - \delta)X_t + \Delta$ for some number Δ . This suffices to employ our negative drift theorem and obtain the desired lower bound. Without our drift theorem at hand, in [29] the potential $Y_t = \log_8(X_t)$ was considered, it was argued that it displays an additive drift away from the target and that Y_t satisfies certain concentration statements necessary for the subsequent use of a negative drift theorem for additive drift.

A second example using similar techniques, and thus most likely profiting from our drift theorem, is the work of Oliveto and Witt [33, 34] analyzing the

simple GA with crossover optimizing ONEMAX. Due to the use of crossover, this work is much more involved, so without much detail we point the reader interested in the details to the location where we feel that our drift theorem would have eased the analysis. In Lemma 19 of [34], again a multiplicative drift statement (away from the target) is proven. To use a negative drift theorem for additive drift (Theorem 2 in [34]), in the proof of Lemma 20 the logarithm of the original process is regarded. So here again, we feel that a direct application of our drift theorem would have eased the analysis.

3 Negative Drift in Populations Revisited

In this section, we use our negative multiplicative drift result and some more arguments to rework Lehre’s negative drift in populations method [24] and obtain Theorem 2 further below. This method allows to analyze a broad class of evolutionary algorithms, namely all that give rise to the following *population selection-mutation (PSM) process* (identical to the one defined in [24] even though we use a slightly more algorithmic language). Let Ω be a finite set. We call Ω the *search space* and its elements *solution candidates* or *individuals*. Let $\lambda \in \mathbb{N}$ be called the *population size* of the process. An ordered multi-set of cardinality λ , in other words, a λ -tuple, over the search space Ω is called a *population*. Let $\mathcal{P} = \Omega^\lambda$ be the set of all populations. For $P \in \mathcal{P}$, we write P_1, \dots, P_λ to denote the elements of P . We also write $x \in P$ to denote that there is an $i \in [1..\lambda]$ such that $x = P_i$.

A PSM process starts with some, possibly random, population $P^{(0)}$. In each iteration $t = 1, 2, \dots$, a new population $P^{(t)}$ is sampled from the previous one $P^{(t-1)}$ as follows. Via a (possibly) randomized *selection operator* $\text{sel}(\cdot)$, a λ -tuple of individuals is selected and then each of them creates an offspring through the application of a randomized *mutation operator* $\text{mut}(\cdot)$.

The *selection operator* can be arbitrary except that it only selects individuals from $P^{(t-1)}$. In particular, we do not assume that the selected individuals are independent. Formally speaking, the outcome of the selection process is a random λ -tuple $Q = \text{sel}(P) \in [1..\lambda]^\lambda$ such that $P_{Q_1}^{(t-1)}, \dots, P_{Q_\lambda}^{(t-1)}$ are the selected parents.

From each selected parent $P_{Q_i}^{(t-1)}$, a single offspring $P_i^{(t)}$ is generated via a randomized *mutation operator* $P_i^{(t)} = \text{mut}(P_{Q_i}^{(t-1)})$. Formally speaking, for each $x \in \Omega$, $\text{mut}(x)$ is a probability distribution on Ω and we write $y = \text{mut}(x)$ to indicate that y is sampled from this distribution. We assume that each sample, that is, each call of a mutation operator, uses independent randomness. With this notation, we can write the new population as $P^{(t)} = (\text{mut}(P_{\text{sel}(P)_1}^{(t-1)}), \dots, \text{mut}(P_{\text{sel}(P)_\lambda}^{(t-1)}))$. From the definition it is clear that a PSM process is a Markov process with state space \mathcal{P} .

The following characteristic of the selection operator was found to be crucial for the analysis of PSM processes in [24]. Let $P \in \mathcal{P}$ and $i \in [1..\lambda]$. Then the random variable $R(i, P) = |\{j \in [1..\lambda] \mid \text{sel}(P)_j = P_i\}|$, called *reproduction number of the i -th individual in P* , denotes the number of times P_i was selected from P as parent. Its expectation $E[R(i, P)]$ is called *reproduction rate*.

Our version of the negative drift in populations method now is the following.

Theorem 2. *Consider a PSM process $(P^{(t)})_{t \geq 0}$ as described above. Let $g : \Omega \rightarrow \mathbb{Z}_{\geq 0}$, called potential function, and $a, b \in \mathbb{Z}_{\geq 0}$ with $a \leq b$. Assume that for all $x \in P^{(0)}$ we have $g(x) \geq b$. Let $T = \min\{t \geq 0 \mid \exists i \in [1.. \lambda] : g(P_i^{(t)}) \leq a\}$ the first time we have a search point with potential a or less in the population. Assume that the following three conditions are satisfied.*

- (i) *There is an $\alpha \geq 1$ such that for all populations $P \in \mathcal{P}$ with $\min\{g(P_i) \mid i \in [1.. \lambda]\} > a$ and all $i \in [1.. \lambda]$ with $g(P_i) < b$, we have $E[R(i, P)] \leq \alpha$.*
- (ii) *There is a $\kappa > 0$ and a $0 < \delta < 1$ such that for all $x \in \Omega$ with $a < g(x) < b$ we have*

$$E[\exp(-\kappa g(\text{mut}(x)))] \leq \frac{1}{\alpha}(1 - \delta) \exp(-\kappa g(x)).$$

- (iii) *There is a $D \geq \delta$ such that for all $x \in \Omega$ with $g(x) \geq b$, we have*

$$E[\exp(-\kappa g(\text{mut}(x)))] \leq D \exp(-\kappa b).$$

Then

- $E[T] \geq \frac{\delta}{2D\lambda} \exp(\kappa(b - a)) - \frac{1}{2}$, and
- for all $L \geq 1$, we have $\Pr[T < L] \leq L\lambda \frac{D}{\delta} \exp(-\kappa(b - a))$.

Before proceeding with the proof, we compare our result with Theorem 1 of [24]. We first note that, apart from a technicality which we discuss toward the end of this comparison, the assumptions of our result are weaker than the ones on [24] since we do not need the technical fourth and fifth assumption of [24], which in our notation would read as follows.

- There is a $\delta_2 > 0$ such that for all $i \in [a..b]$ and all $k, \ell \in \mathbb{Z}$ with $1 \leq k + \ell$ and all $x, y \in \Omega$ with $g(x) = i$ and $g(y) = i - \ell$ we have

$$\begin{aligned} \Pr[g(\text{mut}(x)) = i - \ell \wedge g(\text{mut}(y)) = i - \ell - k] \\ \leq \exp(\kappa(1 - \delta_2)(b - a)) \Pr[g(\text{mut}(x)) = i - k - \ell]. \end{aligned}$$

- There is a $\delta_3 > 0$ such that for all $i, j, k, \ell \in \mathbb{Z}$ with $a \leq i \leq b$ and $1 \leq k + \ell \leq j$ and all $x, y \in \Omega$ with $g(x) = i$ and $g(y) = i - k$ we have

$$\Pr[g(\text{mut}(x)) = i - j] \leq \delta_3 \Pr[g(\text{mut}(y)) = i - k - \ell].$$

The assertion of our result is of the same type as in [24], but stronger in terms of numbers. For the probability $\Pr[T < L]$ to find a potential of at most a in time less than L , a bound of

$$O(\lambda L^2 D (b - a) \exp(-\kappa \delta_2 (b - a)))$$

is shown in [24]. Hence our result is smaller by a factor of $\Omega(L(b - a) \exp(-\kappa(1 - \delta_2)(b - a)))$. In addition, our result is non-asymptotic, that is, the lower bound contains no asymptotic notation or unspecified constants.

The one point where Lehre’s [24] result potentially is stronger is that it needs assumptions only on the average drift, whereas we require the same assertion on the point-wise drift. More concretely, Lehre uses the notation $(X_t)_{t \geq 0}$ to denote the Markov process on Ω associated with the mutation operator (it is not said in [24] what is X_0 , that is, how this process is started). Then $\Delta_t(i) = (g(X_{t+1}) - g(X_t) \mid g(X_t) = i)$ defines the potential gain in step t when the current state has potential i . With this notation, instead of our second and third condition, Lehre [24] requires only the weaker conditions (here again translated into our notation).

- (ii’) For all $t \geq 0$ and all $a < i < b$, $E[\exp(-\kappa \Delta_t(i))] < \frac{1}{\alpha}(1 - \delta)$.
- (iii’) For all $t \geq 0$, $E[\exp(-\kappa(g(X_{t+1}) - b)) \mid g(X_t) \geq b] < D$.

So Lehre only requires that the random individual at time t , conditional on having a certain potential, gives rise to a certain drift, whereas we require that each particular individual with this potential gives rise to this drift. On the formal level, Lehre’s condition is much weaker than ours (assuming that the unclear point of what is X_0 can be fixed). That said, to exploit such weaker conditions, one would need to be able to compute such average drifts and they would need to be smaller than the worst-case point-wise drift. We are not aware of many examples where average drift was successfully used in drift analysis (one is Jägersküpfer’s remarkable analysis of the linear functions problem [20]) despite the fact that many classic drift theorems only require conditions on the average drift to hold.

We now prove Theorem 2. Before stating the formal proof, we describe on a high level its main ingredients and how it differs from Lehre’s proof.

The main challenge when using drift analysis is designing a potential function that suitably measures the progress. For simple hillclimbers and optimization problems, the fitness of the current solution may suffice, but already the analysis of the $(1 + 1)$ EA on linear functions resisted such easy approaches [13, 16, 19, 38]. For population-based algorithms, the additional challenge is to capture the quality of the whole population in a single number. We note at this point that the notion of “negative drift in populations” was used in Lehre to informally describe the characteristic of the population processes regarded, but drift analysis as a mathematical tool was employed only on the level of single individuals and the resulting findings were lifted to the whole population via advanced tools like branching processes and eigenvalue arguments.

To prove upper bounds, in [1, 3–5, 14, 25, 37], implicitly or explicitly potential functions were used that build on the fitness of the best individual in the population and the number of individuals having this fitness. Regarding only the current-best individuals, these potential functions might not be suitable for lower bound proofs.

The lower bound proofs in [2, 29, 33, 34] all define a natural potential for single individuals, namely the Hamming distance to the optimum, and then lift this potential to populations by summing over all individuals an exponential transformation of their base potential (this ingenious definition was, to the best of our knowledge, not known in the theory of evolutionary algorithms before

the work of Neumann, Oliveto, and Witt [29]). This is the type of potential we shall use as well, and given the assumptions of Theorem 2, it is not surprising that $\sum_{x \in P} \exp(-\kappa g(x))$ is a good choice. For this potential, we shall then show with only mild effort that it satisfies the assumptions of our drift theorem, which yields the desired lower bounds on the runtime (using that a single good solution in the population already requires a very high potential due to the exponential scaling). We now give the details of this proof idea.

Proof (of Theorem 2). We consider the process (X_t) defined by $X_t = \sum_{i=1}^\lambda \exp(-\kappa g(P_i^{(t)}))$. To apply drift arguments, we first analyze the expected state after one iteration, that is, $E[X_t \mid X_{t-1}]$. To this end, let us consider a fixed parent population $P = P^{(t-1)}$ in iteration t . Let $Q = \text{sel}(P)$ be the indices of the individuals selected for generating offspring.

We first condition on Q (and as always on P), that is, we regard only the probability space defined via the mutation operator, and compute

$$\begin{aligned} E[X_t \mid Q] &= E \left[\sum_{j=1}^\lambda \exp(-\kappa g(\text{mut}(P_{Q_j})) \right) \\ &= \sum_{i=1}^\lambda (R(i, P) \mid Q) E[\exp(-\kappa g(\text{mut}(P_i)))]. \end{aligned}$$

Using that $\sum_{i=1}^\lambda R(i, P) = \lambda$ and not anymore conditioning on Q , by the law of total expectation, we have

$$\begin{aligned} E[X_t] &= E_Q[E[X_t \mid Q]] \\ &= \sum_{i=1}^\lambda E[R(i, P)] E[\exp(-\kappa g(\text{mut}(P_i)))] \\ &= \sum_{P_i: g(P_i) < b} \alpha E[\exp(-\kappa g(\text{mut}(P_i)))] + \sum_{P_i: g(P_i) \geq b} E[R(i, P)] D \exp(-\kappa b) \\ &\leq \sum_{P_i: g(P_i) < b} \alpha \cdot \frac{1}{\alpha} (1 - \delta) \exp(-\kappa g(P_i)) + \lambda \cdot D \exp(-\kappa b) \\ &\leq (1 - \delta) X_{t-1} + \lambda D \exp(-\kappa b) \end{aligned}$$

and recall that this is conditional on $P^{(t-1)}$, hence also on X_{t-1} .

Let $\Delta = \lambda D \exp(-\kappa b)$. Since $P^{(0)}$ contains no individual with potential below b , we have $X_0 \leq \lambda \exp(-\kappa b) = \frac{\Delta}{D} \leq \frac{\Delta}{\delta}$. Hence also the assumption $E[X_0] \leq \frac{\Delta}{\delta}$ of Lemma 1 is fulfilled.

Let $M = \exp(-\kappa a)$ and $T' := \min\{t \geq 0 \mid X_t \geq M\}$. Note that T , the first time to have an individual with potential at least a in the population, is at least T' . Now the negative multiplicative drift theorem (Lemma 1) gives $\Pr[T < L] \leq \Pr[T' < L] \leq \frac{L\Delta}{M\delta} = L\lambda D \frac{\exp(-\kappa(b-a))}{\delta}$ and $E[T] \geq E[T'] \geq \frac{\delta M}{2\Delta} - \frac{1}{2} = \frac{\delta}{2D\lambda} \exp(\kappa(b-a)) - \frac{1}{2}$. \square

We note that the proof above actually shows the following slightly stronger statement, which might be useful when working with random initial populations.

Theorem 3. *Theorem 2 remains valid when the assumption that all initial individuals have potential at least b is replaced by the assumption $\sum_{i=1}^{\lambda} E[\exp(-\kappa g(P_i^{(0)}))] \leq \frac{\lambda D \exp(-\kappa b)}{\delta}$.*

4 Processes Using Standard Bit Mutation

Since many EAs use standard bit mutation, as in [24] we now simplify our main result for processes using standard bit mutation and for g being the Hamming distance to a target solution. Hence in this section, we have $\Omega = \{0, 1\}^n$ and $y = \text{mut}(x)$ is obtained from x by flipping each bit of x independently with probability p . Since our results are non-asymptotic, we can work with any $p \leq \frac{1}{2}$.

Theorem 4. *Consider a PSM process with search space $\Omega = \{0, 1\}^n$, using standard bit mutation with mutation rate $p \in [0, \frac{1}{2}]$ as mutation operator, and such that $P_i^{(0)}$ is uniformly distributed in Ω for each $i \in [1.. \lambda]$. Let $x^* \in \Omega$ be the target of the process. For all $x \in \Omega$, let $g(x) := H(x, x^*)$ denote the Hamming distance from the target.*

Let $\alpha > 1$ and $0 < \delta < 1$ such that $\ln(\frac{\alpha}{1-\delta}) < pn$, that is, such that $1 - \frac{1}{pn} \ln(\frac{\alpha}{1-\delta}) =: \varepsilon > 0$. Let $B = \frac{2}{\varepsilon}$. Let a, b be integers such that $0 \leq a < b$ and $b \leq \tilde{b} := n \frac{1}{B^2 - 1}$.

Selection condition: Assume that for all populations $P \in \mathcal{P}$ with $\min\{g(P_i) \mid i \in [1.. \lambda]\} > a$ and all $i \in [1.. \lambda]$ with $g(P_i) < b$, we have $E[R(i, P)] \leq \alpha$.

Then the first time $T := \min\{t \geq 0 \mid \exists i \in [1.. \lambda] : g(P_i^{(t)}) \leq a\}$ that the population contains an individual in distance a or less from x^ satisfies*

$$E[T] \geq \frac{1}{2\lambda} \min \left\{ \frac{\delta\alpha}{(1-\delta)}, 1 \right\} \exp \left(\ln \left(\frac{2}{1 - \frac{1}{pn} \ln(\frac{\alpha}{1-\delta})} \right) (b-a) \right) - \frac{1}{2},$$

$$\Pr[T < L] \leq L\lambda \max \left\{ \frac{(1-\delta)}{\delta\alpha}, 1 \right\} \exp \left(-\ln \left(\frac{2}{1 - \frac{1}{pn} \ln(\frac{\alpha}{1-\delta})} \right) (b-a) \right).$$

We have to defer the elementary proof, a reduction to Theorem 2, to the extended version [10] for reasons of space. To show that the second and third condition of Theorem 2 are satisfied, one has to estimate $E[\exp(-\kappa(g(\text{mut}(x)) - g(x)))]$, which is not difficult since $g(\text{mut}(x)) - g(x)$ can be written as sum of independent random variables. With a similar computation, we show that the weaker starting condition of Theorem 3 is satisfied.

As a simple **example** for an application of this result, let us consider the classic (μ, λ) EA (with uniform selection for variation, truncation selection for inclusion into the next generation, and mutation rate $p = \frac{1}{n}$) with $\lambda = 2\mu$ optimizing some function $f : \{0, 1\}^n \rightarrow \mathbb{R}$, $n = 500$, with unique global optimum. For simplicity, let us take as performance measure λT , that is, the number of

fitness evaluations in all iterations up to the one in which the optimum was found. Since $\lambda = 2\mu$, we have $\alpha = 2$. By taking $\delta = 0.01$, we obtain a concrete lower bound of more than 13 million fitness evaluations until the optimum is found (regardless of μ and f).

Since the result above is slightly technical, we now formulate the following corollary, which removes the variable δ without significantly weakening the result. We note that the proof of this result applies Theorem 4 with a non-constant δ , so we do not see how such a result could have been proven from Lehre’s result [24].

Corollary 5. *Consider a PSM process as in Theorem 4. Let $x^* \in \Omega$ be the target of the process. For all $x \in \Omega$, let $g(x) := H(x, x^*)$ denote the Hamming distance from the target. Assume that there is an $\alpha > 1$ such that*

- $\ln(\alpha) \leq p(n - 1)$, which is equivalent to $\gamma := 1 - \frac{\ln \alpha}{pn} \geq \frac{1}{n}$;
- there is an $a \leq b := \lfloor (1 - \frac{a}{n})n^{\frac{1}{\gamma^2} - 1} \rfloor$ such that for all populations $P \in \mathcal{P}$ with $\min\{g(P_i) \mid i \in [1.. \lambda]\} > a$ and for all $i \in [1.. \lambda]$, we have $E[R(i, P)] \leq \alpha$.

Then the first time $T := \min\{t \geq 0 \mid \exists i \in [1.. \lambda] : g(P_i^{(t)}) \leq a\}$ that the population contains an individual in distance a or less from x^* satisfies

$$E[T] \geq \frac{p\alpha}{4\lambda n} \min\left\{1, \frac{2n}{p\alpha}\right\} \exp\left(\ln\left(\frac{2}{\gamma}\right)(b - a)\right) - \frac{1}{2},$$

$$\Pr[T < L] \leq \frac{2L\lambda n}{p\alpha} \max\left\{1, \frac{p\alpha}{2n}\right\} \exp\left(-\ln\left(\frac{2}{\gamma}\right)(b - a)\right).$$

In particular, if $a \leq (1 - \varepsilon)b$ for some constant $\varepsilon > 0$, then T is super-polynomial in n (in expectation and with high probability) when $\gamma = \omega(n^{-1/2})$ and exponential when $\gamma = \Omega(1)$.

We omit the proof for reasons of space. It can be found in [10]. The main argument is employing Theorem 4 with the $\delta = \frac{p}{2n}$ and computing that this small δ has no significant influence on the exponential term of the bounds.

5 Fitness Proportionate Selection

In this section, we apply our method to a mutation-only version of the simple genetic algorithm (simple GA). This algorithm starts with a population of μ random bit strings of length n . In each iteration, it computes a new population by μ times independently selecting an individual from the existing population via fitness proportionate selection and mutating it via standard bit mutation with mutation rate $p = \frac{1}{n}$.

The first work [29, Theorem 8] analyzing this algorithm showed that with $\mu \leq \text{poly}(n)$ it needs with high probability more than $2^{n^{1 - O(1/\log \log n)}}$ iterations to find the optimum of the ONEMAX function or any search point in Hamming distance at most $0.003n$ from it. Hence this is only a subexponential lower bound. In [25,

Corollary 13], building on the lower bound method from [24], a truly exponential lower bound is shown for the task of finding a search point in Hamming distance at most $0.029n$ from the optimum, but only for a relatively large population size of $\mu \geq n^3$ (and again $\mu \leq \text{poly}(n)$).

We now extend this result to arbitrary μ , that is, we remove the conditions $\mu \geq n^3$ and $\mu \leq \text{poly}(n)$. To obtain the constant 0.029, we have to compromise with the constants in the runtime, which consequently are only of a theoretical interest. We therefore do not specify the base of the exponential function or the leading constant. We note that this would have been easily possible since we only use a simple additive Chernoff bound and Corollary 5. We further note that Lehre [25] also shows lower bounds for a scaled version of fitness proportionate selection and a general $\Theta(1/n)$ mutation rate. This would also be possible with our approach and would again remove the conditions on λ , but we do not see that the additional effort is justified here.

Theorem 6. *There is a $T = \exp(\Omega(n))$ such that the mutation-only simple GA optimizing ONEMAX with any population size μ with probability $1 - \exp(-\Omega(n))$ does not find any solution x with $\text{ONEMAX}(x) \geq 0.971n$ within T fitness evaluations.*

The main difficulty for proving lower bounds for algorithms using fitness proportionate selection (and maybe the reason why [24] does not show such bounds) is that the reproduction number is non-trivial to estimate. If all but one individual have a fitness of zero, then this individual is selected μ times. Hence μ is the only general upper bound for the reproduction number. The previous works and ours overcome this difficulty by arguing that the average fitness in the population cannot significantly drop below the initial value of $n/2$, which immediately yields that an individual with fitness k has a reproduction number of roughly at most $\frac{k}{n/2}$.

While it is natural that the typical fitness of an individual should not drop far below $n/2$, informally arguing that the individuals should be at least as good as random individuals, making this argument precise is not completely trivial. In [29, Lemma 6], it is informally argued that the situation with fitness proportionate selection cannot be worse than with uniform selection and for the latter situation a union bound over all lineages of individuals is employed and a negative-drift analysis from [30, Section 3] is used for a single lineage. The analysis in [25, Lemma 9] builds on the (positive) drift stemming from standard bit mutation when the fitness is below $n/2$ (this argument needs a mutation rate of at least $\Omega(1/n)$) and the independence of the offspring (here the lower bound $\lambda \geq n^3$ is needed to allow the desired Chernoff bound estimates).

Our proof relies on a natural domination argument that shows that at all times all individuals are at least as good as random individuals in the sense of stochastic domination (see, e.g., [7]) of their fitness. This allows to use a simple Chernoff bound to argue that with high probability, for a long time all individuals have a fitness of at least $(\frac{1}{2} - \varepsilon)n$. The remainder of the proof is an application of Corollary 5. Clearly, Lehre's lower bound [24, Theorem 4] would

have been applicable as well with the main difference being that one has to deal with the constant δ , which does not exist in Corollary 5. The full proof can again be found in [10].

6 Conclusion and Outlook

In this work, we have proven two technical tools which might ease future lower bound proofs in discrete evolutionary optimization. The negative multiplicative drift theorem has the potential to replace the more technical negative drift theorems used so far in different contexts. Our strengthening and simplification of the negative drift in populations method should help increasing our not very developed understanding of population-based algorithms in the future. Clearly, it is restricted to mutation-based algorithms – providing such a tool for crossover-based algorithms and extending our understanding how to prove lower bounds for these beyond the few results [11, 15, 34, 36] would be a great progress.

References

1. Antipov, D., Doerr, B., Fang, J., Hetet, T.: Runtime analysis for the $(\mu + \lambda)$ EA optimizing OneMax. In: Genetic and Evolutionary Computation Conference, GECCO 2018, pp. 1459–1466. ACM (2018)
2. Antipov, D., Doerr, B., Yang, Q.: The efficiency threshold for the offspring population size of the (μ, λ) EA. In: Genetic and Evolutionary Computation Conference, GECCO 2019, pp. 1461–1469. ACM (2019)
3. Chen, T., He, J., Sun, G., Chen, G., Yao, X.: A new approach for analyzing average time complexity of population-based evolutionary algorithms on unimodal problems. *IEEE Trans. Syst. Man Cybern. Part B* **39**, 1092–1106 (2009)
4. Corus, D., Dang, D., Eremeev, A.V., Lehre, P.K.: Level-based analysis of genetic algorithms and other search processes. *IEEE Trans. Evol. Comput.* **22**, 707–719 (2018)
5. Dang, D., Lehre, P.K.: Runtime analysis of non-elitist populations: from classical optimisation to partial information. *Algorithmica* **75**, 428–461 (2016)
6. Dang, D.-C., Lehre, P.K.: Self-adaptation of mutation rates in non-elitist populations. In: Handl, J., Hart, E., Lewis, P.R., López-Ibáñez, M., Ochoa, G., Paechter, B. (eds.) PPSN 2016. LNCS, vol. 9921, pp. 803–813. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45823-6_75
7. Doerr, B.: Analyzing randomized search heuristics via stochastic domination. *Theoret. Comput. Sci.* **773**, 115–137 (2019)
8. Doerr, B.: An exponential lower bound for the runtime of the compact genetic algorithm on jump functions. In: Foundations of Genetic Algorithms, FOGA 2019, pp. 25–33. ACM (2019)
9. Doerr, B.: Does comma selection help to cope with local optima? In: Genetic and Evolutionary Computation Conference, GECCO 2020. ACM (2020, to appear)
10. Doerr, B.: Lower bounds for non-elitist evolutionary algorithms via negative multiplicative drift. *CoRR abs/2004.01274* (2020)
11. Doerr, B.: Runtime analysis of evolutionary algorithms via symmetry arguments. *CoRR abs/2006.04663* (2020)

12. Doerr, B., Goldberg, L.A.: Adaptive drift analysis. *Algorithmica* **65**, 224–250 (2013)
13. Doerr, B., Johannsen, D., Winzen, C.: Multiplicative drift analysis. *Algorithmica* **64**, 673–697 (2012)
14. Doerr, B., Kötzing, T.: Multiplicative up-drift. In: Genetic and Evolutionary Computation Conference, GECCO 2019, pp. 1470–1478. ACM (2019)
15. Doerr, B., Theile, M.: Improved analysis methods for crossover-based algorithms. In: Genetic and Evolutionary Computation Conference, GECCO 2009, pp. 247–254. ACM (2009)
16. Droste, S., Jansen, T., Wegener, I.: On the analysis of the (1+1) evolutionary algorithm. *Theoret. Comput. Sci.* **276**, 51–81 (2002)
17. Hajek, B.: Hitting-time and occupation-time bounds implied by drift analysis with applications. *Adv. Appl. Probab.* **13**, 502–525 (1982)
18. Happ, E., Johannsen, D., Klein, C., Neumann, F.: Rigorous analyses of fitness-proportional selection for optimizing linear functions. In: Genetic and Evolutionary Computation Conference, GECCO 2008, pp. 953–960. ACM (2008)
19. He, J., Yao, X.: Drift analysis and average time complexity of evolutionary algorithms. *Artif. Intell.* **127**, 51–81 (2001)
20. Jägersküpper, J.: A blend of Markov-chain and drift analysis. In: Rudolph, G., Jansen, T., Beume, N., Lucas, S., Poloni, C. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 41–51. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87700-4_5
21. Jägersküpper, J., Storch, T.: When the plus strategy outperforms the comma strategy and when not. In: Foundations of Computational Intelligence, FOCI 2007, pp. 25–32. IEEE (2007)
22. Johannsen, D.: Random combinatorial structures and randomized search heuristics. Ph.D. thesis, Universität des Saarlandes (2010)
23. Kötzing, T.: Concentration of first hitting times under additive drift. *Algorithmica* **75**, 490–506 (2016)
24. Lehre, P.K.: Negative drift in populations. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN 2010. LNCS, vol. 6238, pp. 244–253. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15844-5_25
25. Lehre, P.K.: Fitness-levels for non-elitist populations. In: Genetic and Evolutionary Computation Conference, GECCO 2011, pp. 2075–2082. ACM (2011)
26. Lengler, J.: Drift analysis. In: Doerr, B., Neumann, F. (eds.) Theory of Evolutionary Computation: Recent Developments in Discrete Optimization, pp. 89–131. Springer, Cham (2020). <https://arxiv.org/abs/1712.00964>
27. Lengler, J., Steger, A.: Drift analysis and evolutionary algorithms revisited. *Comb. Probab. Comput.* **27**, 643–666 (2018)
28. Mitavskiy, B., Rowe, J.E., Cannings, C.: Theoretical analysis of local search strategies to optimize network communication subject to preserving the total number of links. *Int. J. Intell. Comput. Cybern.* **2**, 243–284 (2009)
29. Neumann, F., Oliveto, P.S., Witt, C.: Theoretical analysis of fitness-proportional selection: landscapes and efficiency. In: Genetic and Evolutionary Computation Conference, GECCO 2009, pp. 835–842. ACM (2009)
30. Oliveto, P.S., Witt, C.: Simplified drift analysis for proving lower bounds in evolutionary computation. In: Rudolph, G., Jansen, T., Beume, N., Lucas, S., Poloni, C. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 82–91. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87700-4_9
31. Oliveto, P.S., Witt, C.: Simplified drift analysis for proving lower bounds in evolutionary computation. *Algorithmica* **59**, 369–386 (2011)

32. Oliveto, P.S., Witt, C.: Erratum: simplified drift analysis for proving lower bounds in evolutionary computation. CoRR abs/1211.7184 (2012)
33. Oliveto, P.S., Witt, C.: On the runtime analysis of the simple genetic algorithm. *Theoret. Comput. Sci.* **545**, 2–19 (2014)
34. Oliveto, P.S., Witt, C.: Improved time complexity analysis of the simple genetic algorithm. *Theoret. Comput. Sci.* **605**, 21–41 (2015)
35. Rowe, J.E., Sudholt, D.: The choice of the offspring population size in the $(1, \lambda)$ evolutionary algorithm. *Theoret. Comput. Sci.* **545**, 20–38 (2014)
36. Sutton, A.M., Witt, C.: Lower bounds on the runtime of crossover-based algorithms via decoupling and family graphs. In: Genetic and Evolutionary Computation Conference, GECCO 2019, pp. 1515–1522. ACM (2019)
37. Witt, C.: Runtime analysis of the $(\mu + 1)$ EA on simple pseudo-Boolean functions. *Evol. Comput.* **14**, 65–86 (2006)
38. Witt, C.: Tight bounds on the optimization time of a randomized search heuristic on linear functions. *Comb. Probab. Comput.* **22**, 294–318 (2013)
39. Witt, C.: Upper bounds on the running time of the univariate marginal distribution algorithm on OneMax. *Algorithmica* **81**, 632–667 (2019)