Multiplicative Up-Drift

Benjamin Doerr
École Polytechnique, CNRS
Laboratoire d'Informatique (LIX)
Palaiseau, France

Timo Kötzing
Hasso Plattner Institute
Potsdam, Germany

ABSTRACT

Drift analysis aims at translating the expected progress of an evolutionary algorithm (or more generally, a random process) into a probabilistic guarantee on its run time (hitting time). So far, drift arguments have been successfully employed in the rigorous analysis of evolutionary algorithms, however, only for the situation that the progress is constant or becomes weaker when approaching the target.

Motivated by questions like how fast fit individuals take over a population, we analyze random processes exhibiting a multiplicative growth in expectation. We prove a drift theorem translating this expected progress into a hitting time. This drift theorem gives a simple and insightful proof of the level-based theorem first proposed by Lehre (2011). Our version of this theorem has, for the first time, the best-possible linear dependence on the growth parameter δ (the previous-best was quadratic). This gives immediately stronger run time guarantees for a number of applications.

CCS CONCEPTS

• **Theory of computation → Theory of randomized search heuristics**;

KEYWORDS

drift theory, monotone decreasing drift, run time analysis

ACM Reference Format:
Benjamin Doerr and Timo Kötzing. 2019. Multiplicative Up-Drift. In *Genetic and Evolutionary Computation Conference (GECCO '19), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3321707.3321819

1 INTRODUCTION

In a typical situation in evolutionary search, an algorithm first makes good progress while far away from the target, since a lot can still be improved. As the search focuses more and more on the fine details, progress slows and finding improving moves becomes rarer. Thus, the expected progress is typically an increasing function in the distance from the optimum. However, there are also many processes where this situation is reversed. For example, for heuristics involving a population, once a superior individual is found, this improvement needs to be spread over the population. This process gains speed when more individuals exist with the improvement.

Turning expected progress into an expected first hitting time is the purpose of drift theorems. For example, the additive drift theorem [15, 16] requires a uniform lower bound δ on the expected progress (the expected *drift*) and gives an expected first hitting time of at most n/δ, where n is the initial distance from the optimum. This theorem can also be applied when the drift is changing during the process, but since a uniform δ is used in the argument, the additive drift theorem cannot be used to exploit a stronger drift later in the process.

A first step towards profiting from a changing drift behavior was the multiplicative drift theorem [8, 9]. It assumes that the expected drift is at least δx when the distance from the optimum is x, for some factor $\delta < 1$. The first hitting time can then be bounded by $O(\log(n)/\delta)$, where n is again the initial distance from the optimum. Apparently, this gives a much better bound than what could be shown via the additive drift in this setting. Multiplicative drift can be found in many optimization processes, making the multiplicative drift theorem one of the most useful drift theorems.

To cope with a broader variety of changing drift patterns, the variable drift theorem [18, 23] has been developed. However, while there are several variants of this drift theorem, they all require that the strength of the drift is a monotone *increasing* function in the distance from the optimum (the farther away from the optimum, the easier it is to make progress).

In this paper we are concerned with the reverse setting where drift is a *decreasing* function it the distance from the optimum. While many drift theorems are phrased such that the aim is to reach the point zero, for our setting it is more natural to consider the case of reaching some target value n starting at a value of 1, and to suppose that the drift is δx *going up* (for the multiplicative drift theorem, we had a drift of δx *going down*). Thus, we call our resulting drift theorem the *multiplicative up-drift theorem*.

Making things more formal, consider a random process $(X_t)_{t \in \mathbb{N}}$ over positive reals starting at $X_0 = 1$ and with target $n > 1$. We speak of *multiplicative up-drift* if there is a $\delta > 0$ such that, for all $t \geq 0$, we have the drift condition

(D) $E[X_{t+1} - X_t \mid X_t] \geq \delta X_t$.

Note that this is equivalent to

(D') $E[X_{t+1} \mid X_t] \geq (1+\delta) X_t$.

One trivial case of any drift process is the deterministic process with the desired gain per iteration. We quickly regard this case now as it gives the right impression of what should be a natural expected first hitting time for a well-behaved process exhibiting multiplicative up-drift.

Example 1.1. Let $\delta > 0$. Suppose $X_0 = 1$ and, for all t, $X_{t+1} = (1+\delta)X_t$ with probability 1. Then this process satisfies the drift

GECCO '19, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6111-8/19/07…$15.00
https://doi.org/10.1145/3321707.3321819

condition **(D)** with equality. Clearly, the time to reach a value of at least n is $\lceil \log_{1+\delta}(n) \rceil$. For small δ, this is approximately $\log(n)/\delta$, for large δ, it is approximately $\log(n)/\log(\delta)$. We note here already that we will be mostly concerned with the case where δ is small. This case is the harder one since the progress is weaker, and thus there is a greater need for stronger analyses tools in this case.

Unfortunately, not all processes with multiplicative up-drift have a hitting time of $O(\log(n)/\delta)$, as the following example shows.

Example 1.2. Let $\delta > 0$. Suppose $X_0 = 1$ and, for all t, $X_{t+1} = 2(1+\delta)X_t - 1$ with probability 0.5 (which we term a *success*) and $X_{t+1} = 1$ otherwise. Again, the drift condition **(D)** is satisfied with equality. A straightforward induction shows that, after k successes, the process has a value of $1 + 2\delta \sum_{i=0}^{k-1} (2+2\delta)^i$. Thus, we require a sequence of about $\log_{2+2\delta}(n/\delta)$ consecutive successes to reach a value of n which, for values of $\delta = o(1/\log n)$, has a probability of about $2^{-\log_{2+2\delta}(n/\delta)} \approx \delta/n$. Therefore we expect to need $\Omega(n/\delta)$ iterations, significantly more than the $O(\log(n)/\delta)$ seen in the deterministic process.

Note that for this process the additive drift theorem immediately gives the upper bound of $O(n/\delta)$ since we always have a drift of at least δ towards the target. Hence Example 1.2 describes a process where the stronger assumption of multiplicative up-drift does not lead to a better hitting time.

Our first main result (Theorem 2.1) shows that the targeted bound of $O(\log(n)/\delta)$, which as we saw is optimal when we want to cover the deterministic process given in Example 1.1, can be obtained when strengthening condition **(D)** by assuming (i) that, given X_t, the next state X_{t+1} is binomially distributed with expectation $(1+\delta)X_t$, and (ii) that the process never reaches the state 0. The first condition is very natural. When generating offspring independently, the number of offspring satisfying a particular desired property is binomially distributed. The second condition is technical necessity. From the up-drift condition alone, we cannot infer any progress from state 0. Consequently, 0 could well be an absorbing state, resulting in an infinite hitting time if this state can be reached with positive probability.

In quite some applications, however, we cannot rule out that the random process reaches the state 0. For example, when regarding the subpopulation of individuals having some desired property, then in an algorithm using comma selection, this might die out completely in one iterations (though often with small probability only). To cover also such processes, in our second drift theorem (Theorem 2.2) we extend our Theorem 2.1 to include that the state 0 is reached with at most the probability that can be deduced from the up-drift and the binomial distribution conditions. To avoid that the the state 0 is absorbing, we add an additional condition for this state 0: we assume a minimum probability of ε to leave state 0. Since it allows for stronger run time guarantees in some cases, we also consider lower bounds E_0 on the expected state reached when leaving zero (see Theorem 2.1 for the precise statement of this aspect). We note that this is a feature, but not a restriction, since we can always take $E_0 = \varepsilon$.

As mentioned before, a main application for multiplicatively increasing drift towards the optimum is the analysis of how fit individuals spread in a population. This particular setting was previously analyzed as the *level-based theorem* [4, 5, 20], modeled after the method of fitness-based partitions [26]. Essentially, the search space is partitioned into an ordered sequence of *levels*. The ongoing search process increases the probability that a newly-created individual is *at least* on a given level and, once this probability is sufficiently high, that there is a good chance that the individual is on an even higher level. We restate the details of this theorem in the version from [4] in Theorem 3.1 below. The level-based theorem was originally intended for the analysis of non-elitist population-based algorithms [5], but has since also been applied to EDAs, namely to the UMDA in [21] and, with some additional arguments, to PBIL in [22].

We use our second multiplicative up-drift theorem (Theorem 2.2) to prove a new version of the level-based theorem (Theorem 3.2). This new theorem allows to derive better bounds under essentially the same conditions, in particular, improving the dependence on the parameter δ from quadratic to linear. Our upper bounds almost match the lower-bound example given in [4] and, in particular, match the asymptotic dependence on δ displayed by this example.

Our version of the level-based theorem directly leads to better bounds in all settings where the level-based theorem was used previously. In Section 4, we show how it improves the run time guarantees of two previous analyses of non-elitist evolutionary algorithms. (i) We prove that the (λ, λ) EA with fitness-proportionate selection and suitable parameters can optimize the OneMax and LeadingOnes functions in time $O(n^4 \log^2 n)$, improving over the previous-best published bound of $O(n^8 \log n)$. (ii) We prove that the (λ, λ) EA with 2-tournament selection and suitable parameters in the restricted setting that only a constant fraction of the bits of the search points are evaluated finds the optimum of OneMax in $O(n^{2.5} \log^2 n)$ iterations. The previous-best published bound here is $O(n^{4.5} \log n)$.

Beyond these particular results, our modular proof (first analyzing the multiplicative up-drift excluding 0, then including 0, then applying it in the context of the level-based theorem) shows the level-based theorem in a way that is more accessible than the previous versions and that gives more insight into population-based optimization processes.

In particular, our proof suggests that the behavior of the process under the named conditions is as follows.

- Once a critical mass in a level is reached, this level is never again abandoned. Thus, we can focus in our analysis on having a critical mass of individuals in one level and analyze the time it takes to gain a critical mass in the next level.
- Reaching a critical mass in the next level consists of two steps.
  1. When few elements are in the next level, then these elements go extinct regularly and need to be respawned until this initial population on this level via a mostly unbiased random walk gains a moderate amount of elements.
  2. With this moderate amount of elements, the bias of the random walk is large enough to make a significant decrease of the population unlikely, but instead the number

of elements increases steadily, as can be shown using a concentration bound for supermartingales, so that we quickly gain a critical mass in the next level.

We are optimistic that this increased understanding of population-based processes helps in the future design and analysis of such processes.

2 MULTIPLICATIVE UP-DRIFT THEOREMS

In this section we prove two versions of the multiplicative up-drift theorem. The first is concerned with processes that cannot reach the value 0; the second one extends the first theorem to include also the possibility of going down to 0.

2.1 Processes on the Positive Integers

As discussed in the introduction, an expected multiplicative increase as described by **(D)** is not enough to ensure the run time we aim at. For this reason, we assume that there is a number k such that, conditional on X_t, the next state X_{t+1} is binomially distributed with parameters k and $(1+\delta)X_t/k$. Note that this implies **(D)**. Since often precise distributions are hard to specify, we only require that X_{t+1} is at least as large as this binomial distribution, that is, we require that X_{t+1} stochastically dominates $\text{Bin}(k, (1+\delta)X_t/k)$. See [6] for an introduction to stochastic domination and its use in run time analysis. To avoid that the process reaches the possible absorbing state 0, we explicitly forbid this, that is, we require that all X_t take values only in the positive integers.

Under these conditions, we analyze the time the process takes to reach or overshoot a given state n. For technical reasons, we require that n is not too close to k, that is, that there is a constant $\gamma_0 < 1$ such that $n \le \gamma_0 k$. For the trivial reason that the condition $X_{t+1} \ge \text{Bin}(k, (1+\delta)X_t/k)$ does not make sense for $X_t > (1+\delta)^{-1}k$, we also require $n \le (1+\delta)^{-1}k$. For all such n, we show that an expected number of $O(\log(n)/\delta)$ iterations suffices to reach n.

Theorem 2.1 (First Multiplicative Up-Drift Theorem). *Let $\gamma_0 < 1$. Let $(X_t)_{t\in\mathbb{N}}$ be a stochastic process over the positive integers. Assume that there are $n, k \in \mathbb{Z}_{\ge 1}$ and $\delta \in (0, 1]$ such that $n \le \min\{\gamma_0 k, (1+\delta)^{-1}k\}$ and for all $t \ge 0$ and all $x \in [1..n]$ with $\Pr[X_t = x] > 0$ we have the binomial condition*

(Bin) $(X_{t+1} \mid X_t = x) \succeq \text{Bin}(k, (1+\delta)x/k)$.

Let $T := \min\{t \ge 0 \mid X_t \ge n\}$. Further, suppose that δ is bounded from above by a polynomial in n. Then there is a constant C, depending on γ_0, but independent of δ, k and n, such that

$$E[T] \le C\,\frac{\log(n)}{\delta}.$$

In addition, once the process has reached state $100/\delta$ or higher, the probability to ever return to a state $50/\delta$ or lower, is at most $\frac{1}{e-1}$.

Before proving this result, let us give a simple example of a possible application. Consider the following elitist (μ, λ) EA. It starts with a parent population of μ individuals chosen uniformly and independently from $\{0,1\}^n$. In each iteration, it generates λ offspring, each by independently and uniformly choosing a parent individual and mutating it via standard bit mutation (with the usual mutation rate $1/n$). If the offspring population contains at least one individual that is at least as good as the best (in terms of fitness) parent, then the new parent population is chosen by selecting μ best offspring (breaking ties arbitrarily). If all offspring are worse than the best parent, then the new parent population is composed of a best individual from the old parent population and $\mu - 1$ best offspring (again, breaking all ties randomly).

We now use the above theorem to analyze the spread of fit individuals in the parent population. Let us assume that at some time, the parent population contains at least one individual of at least a certain fitness. We shall call such individuals *fit* in the following. Recall that standard bit mutation creates a copy of the parent individual with probability $1/e_n := (1-1/n)^n \approx 1/e$. Hence if the parent population contains x fit individuals, the number of fit individuals in the offspring population is at least (in the domination sense) $\text{Bin}(\lambda, \frac{x}{\mu e_n})$. Due to the elitist selection mechanism, it is also always at least one. If $(1+\delta) := \frac{\lambda}{\mu e_n}$ is greater than one, and let us for simplicity assume that $\delta \le 1$ as well, then we can apply the first up-drift theorem and observe that after an expected number of $O(\log(\mu)/\delta)$ iterations, the parent population consists of only fit individuals.

We now prove the theorem. The formal proof is omitted for reasons of space (in can be found in [10]), so we only outline the main difficulties and solutions in a high-level language. One of the main difficulties is that the drift towards the target is negligibly weak in the early stages of the process. To demonstrate this, assume that $\delta = o(1)$ and that $X_t = o(1/\delta)$. Then the up-drift condition **(D)** only ensures a drift of $E[X_{t+1}-X_t \mid X_t] \ge \delta X_t = o(X_t)$. At the same time, the binomial condition **(Bin)** allows a variance $\text{Var}[X_{t+1} \mid X_t]$ of order X_t. For this reason, in this regime we do not progress because of the drift, but rather because of the random fluctuations of the process.

It is well-known that random fluctuations are enough to reach a target, with a classical example being the unbiased random walk (W_t) on the line $[0..n] := \{0, 1, \ldots, n\}$. This walk, when started in 0, still reaches n in an expected number of $O(n^2)$ iterations despite the complete absence of any drift in $[1..n-1]$. The key to the analysis is to not regard the drift $E[W_{t+1} - W_t \mid W_t]$ of the process, but instead the drift of the process (W_t^2). Then an easy calculation gives $E[W_{t+1}^2 - W_t^2 \mid W_t = x] = \frac{1}{2}(x+1)^2 + \frac{1}{2}(x-1)^2 - x^2 = 1$ for all $x \in [1..n-1]$ (see [12, Section 5] for an extensive discussion). Consequently, by regarding the drift with respect to a different potential function, we obtained an additive drift of 1, and from this an expected time of $O(n^2)$ to reach the state n.

Very similar arguments have been used in the analysis of unbiased processes arising from running evolutionary algorithms. [17] turned an area with small drift into an area with significantly more drift by employing the concave potential function $x \mapsto \sqrt{x}$, stating that any other function $x \mapsto x^{\varepsilon}$ with $\varepsilon < 1$ would be equally suitable to obtain the same tight upper bound; the same argument was used in a more general setting in [3].

In [11, Theorem 5] a negative drift in a (small) part of the search space was overcome by considering random changes which make it possible for the algorithm to pass through the area of negative drift by chance. This was formalized by using a tailored potential function turning negative drift into positive drift by excessively rewarding changes towards the target, as opposed to steps away from the target. This ad-hoc argument was made formal and cast into a *Headwind Drift Theorem* in [19, Theorem 4].

In abstract terms, the art here is finding a potential function $g : \mathbb{Z}_{\geq 0} \to \mathbb{R}$ that transforms the unbiased process (X_t) into a process $(g(X_t))$ with constant drift. Such a potential function has to be increasing and convex. Since the resulting bound for the expected hitting time is (roughly) the potential $g(n)$ of the target state n, at the same time the potential function should increase as slowly as possible.

For our situation, it turns out that g defined by $g(x) = x\ln(x)$ is a good choice as this again gives a constant drift and thus an expected time of roughly $O(\log(1/\delta)/\delta)$ to reach a state $\Omega(1/\delta)$, from where on we will observe that also the original process has sufficient drift. We are not aware of this potential function being used so far in the theory of evolutionary algorithms (apart from a similar function being used in [2], a work done in parallel to ours).

A technical annoyance in the analysis of the time taken to reach $\Omega(1/\delta)$ is that the additive drift theorem, for good reason, does not allow that the process overshoots the target. In the classical formulation, this follows from the target being 0 and the process living in the non-negative numbers. For this reason, we cannot just show that the process $(g(X_t))$ has a constant drift, but we need to show this drift for a version of this process that is suitably restricted to the range $[1..O(1/\delta)]$.

Once the process has reached a value of $\Theta(1/\delta)$, the drift is strong enough to rely on making progress from the drift (and not the random fluctuations around the expectation). This is easy when the process is above $X_t = \omega(1/\delta^2)$, since we then have an expected progress of at least $\Omega(\sqrt{X_t})$ and thus a simple Chernoff bound is enough to guarantee that each single round gives a progress of $X_{t+1} \geq (1-o(1))(1+\delta)X_t$. When X_t is smaller, say only $\Theta(1/\delta)$, only the combined result of $\Theta(1/\delta)$ rounds gives an expected progress large enough to admit such a strong concentration. Since the rounds are not independent, we need some careful martingale concentration arguments for this phase.

2.2 Processes That Can Reach Zero

We now extend the multiplicative up-drift theorem to include state 0. Since the subprocess consisting only of states greater than 0 satisfies the assumptions of the first up-drift theorem, we obtain from the latter an upper bound on the time spend above 0. It therefore remains to estimate the time spent on state 0, which in particular means estimating how often the process reaches this state. Since the process is a submartingale we can employ the optional stopping theorem to estimate that with probability $1 - \Omega(\delta)$ the process reaches 0 before $D_0 = \min\{100/\delta, n\}$. Consequently, after an expected number of $O(\delta)$ attempts, the process reaches D_0, and from there with constant probability never goes back to zero.

Theorem 2.2 (Second Multiplicative Up-Drift Theorem). *Let $\gamma_0 < 1$. Let $(X_t)_{t\in\mathbb{N}}$ be a stochastic process over $\mathbb{Z}_{\geq 0}$. Assume that there are $n, k \in \mathbb{Z}_{\geq 1}$, $E_0, \varepsilon > 0$, and $\delta \in (0, 1]$ such that $n \leq \min\{\gamma_0 k, (1+\delta)^{-1}k\}$ and for all $t \geq 0$ and all $x \in [0..n]$ with $\Pr[X_t = x] > 0$, the following two properties hold.*

(Bin) *If $x \geq 1$, then $(X_{t+1} \mid X_t = x) \succeq \mathrm{Bin}(k, (1+\delta)x/k)$.*

(0) *If $x = 0$, then*

- *$\Pr[X_{t+1} \geq 1 \mid X_t = x] \geq \varepsilon$;*
- *$E[\min\{100/\delta, n, X_{t+1}\} \mid X_t = x \wedge X_{t+1} \geq 1] \geq E_0$.*

Let $n \in \mathbb{Z}_{\geq 1}$ and $T := \min\{t \geq 0 \mid X_t \geq n\}$. Then

$$E[T] = O\left(\frac{1}{E_0\varepsilon\delta} + \frac{\log(n)}{\delta}\right).$$

To prove this theorem, we again need to cope with the problem that a process may overshoot a target and that this overshooting, while irrelevant for the time to reach the target, may interfere with our mathematical arguments. The following lemma solves this problem for us here, as it states, roughly, that we can replace a binomial random variable with expectation E with a random variable that is identically distributed in $[0..E]$ and takes values only in $[0..4E]$ such that the expectation is not lowered. We suspect that this result may be convenient in many other such situations, e.g., when using additive drift in processes that may overshoot the target.

Lemma 2.3. *Let Y be a random variable taking values in the non-negative integers such that $Y \succeq \mathrm{Bin}(k, p)$ for some $k \in \mathbb{N}$ and $p \in [0, 1]$ with $kp \geq 1$. Let $E = kp$ denote the expectation of $\mathrm{Bin}(k, p)$. Then there is a random variable Z such that*

- *$\Pr[Z = i] = \Pr[Y = i]$ for all $i \in [0..E]$,*
- *$\Pr[Z = i] = 0$ for all $i \geq 4E + 1$,*
- *$E[Z] \geq E$.*

Proof. Let Z be defined by $\Pr[Z = i] = \Pr[Y = i]$ for all $i \in [0..E]$ and $\Pr[Z = \lceil 4E \rceil] = 1 - \Pr[Y \in [0..E]]$. Then it remains to show that $E[Z] \geq E$. If $X \sim \mathrm{Bin}(k, p)$, and hence $E = E[X]$, then $\Pr[X > E] \geq \frac{1}{4}$ by [7]. Since $Y \succeq X$, we have $\Pr[Y > E] \geq \Pr[X > E] \geq \frac{1}{4}$. By definition, $\Pr[Y > E] = \Pr[Y = \lceil 4E \rceil]$ and thus $E[Y] = \sum_{i=0}^{\lceil 4E \rceil} i \Pr[Y = i] \geq \lceil 4E \rceil \Pr[Y = \lceil 4E \rceil] = \lceil 4E \rceil \cdot \frac{1}{4} \geq E$. □

We now prove Theorem 2.2.

Proof. We first analyze the time spend on all states different from 0. To this aim, let $\tilde{X}_t$, $t = 0, 1, \ldots$, be the subprocess where we are above zero. Formally speaking, $\tilde{X}$ is the subsequence of (X_t) consisting of all X_t that are greater than 0. Viewed as a random process, this means that we sample the next state according to the same rules as for the X-process; however, if this is zero, then immediately and without counting this as step we sample the new state from the distribution described in **(0)** conditional on being positive (which is the same as saying that we resample until we obtain a positive result). With this, the distribution describing one step of the process is a distribution on the positive integers such that $(\tilde{X}_{t+1} \mid \tilde{X}_t) \succeq \mathrm{Bin}(k, (1+\delta)\tilde{X}_t/k)$. We may thus apply Theorem 2.1 and obtain that after an expected total number of $O(\log(n)/\delta)$ steps, the process $\tilde{X}$ reaches or exceeds n.

It remains to analyze how many steps the process X spends on state 0. Clearly, by **(0)**, the expected time to leave the state of 0 is $1/\varepsilon$. Hence it suffices to analyze the number of times the process reaches zero. Let $D_0 = \min\{100/\delta, n\}$. Since we know from Theorem 2.1 that the process goes below $D_0/2$ only with constant probability, and hence only constantly many times, after reaching or exceeding D_0, it suffices to show that the process falls back to being 0 only an expected number of $O(1/E_0\delta)$ times before reaching or exceeding a value of D_0.

Let t_0 be a time where $X_{t_0} = 0$ and let t_1 be the first time after t_0 where $X_{t_1} > 0$. Let $Y_t = X_{t_1+t}$ for all $t \geq 0$. We are interested in the first time that (Y_t) reaches or exceeds D_0. Since it does not change this hitting time, we can replace Y_0 by D_0 in case that $Y_0 > D_0$. Then by **(0)**, $E[Y_0] = E_0$.

Let R be the first time that Y exceeds D_0 or hits 0. This is a stopping time. To ease the following argument, we regard the following process Z, which equals Y until the stopping time (and hence has the same stopping time). We define Z recursively. We start by setting $Z_0 := Y_0$. Assume that Z_t is defined and $Z_t \cdot \mathbf{1}_{Z_t \leq D_0} = Y_t \cdot \mathbf{1}_{Y_t \leq D_0}$. If $Z_t > D_0$, then we set $Z_{t+1} = Z_t$. Otherwise, that is, when $Z_t = Y_t = x \leq D_0$ for some x, then we recall that $Y_{t+1} \succeq \mathrm{Bin}(k, (1+\delta)x/k) \succeq \mathrm{Bin}(k, x/k)$. In this case, we let Z_{t+1} be the random variable constructed in Lemma 2.3 (w.r.t. Y_{t+1}, k, and $p = x/k$). By this lemma, we have $Z_{t+1} \cdot \mathbf{1}_{Z_{t+1} \leq D_0} = Y_{t+1} \cdot \mathbf{1}_{Y_{t+1} \leq D_0}$, allowing us to continue our recursive definition of Z, and $E[Z_{t+1} \mid Z_t] \geq E[Z_t]$, showing that (Z_t) is a submartingale. We can thus use the optional stopping theorem to see that $E[Z_R] \geq E[Z_0]$. Furthermore,

$$\begin{aligned} E[Z_R] &= \Pr[Z_R \geq D_0] E[Z_R \mid Z_R \geq D_0] + \Pr[Z_R = 0] E[Z_R \mid Z_R = 0] \\ &= \Pr[Z_R \geq D_0] E[Z_R \mid Z_R \geq D_0] \leq \Pr[Z_R \geq D_0] \lceil 4D_0 \rceil, \end{aligned}$$

the latter again due to Lemma 2.3. Consequently

$$\Pr[Y_R \geq D_0] = \Pr[Z_R \geq D_0] \geq \frac{E[Z_0]}{\lceil 4D_0 \rceil} = \frac{E_0}{\lceil 4D_0 \rceil}.$$

This shows that, in expectation, the process will restart at 0 at most $\lceil 4D_0 \rceil$ times before exceeding D_0. □

3 THE LEVEL-BASED THEOREM

We start by restating the best known version of the level-based theorem from [4]. Afterwards we will give our new version and its derivation.

3.1 Previous-Best Level-Based Theorem

The following version of the level-based theorem has a quadratic dependence on δ.

Theorem 3.1 ([4]). *Let D mapping populations to distributions over $\mathcal{X}$ be given. Given a partition $(A_1, \dots, A_m)$ of $\mathcal{X}$, define $T := \min\{\lambda t \mid P_t \cap A_m \neq \emptyset\}$ where, for all t, P_{t+1} is the population gained by sampling λ individuals from $D(P_t)$. If there are $z_1, \dots, z_{m-1}, \delta \in (0, 1]$ and $\gamma_0 \in (0, 1)$ such that, for any population $P \in \mathcal{X}^\lambda$, the following three conditions are satisfied.*

(G1) *For each level $j \in [m-1]$, if $|P \cap A_{\geq j}| \geq \gamma_0 \lambda$, then*

$$\Pr_{y \sim D(P)}[y \in A_{\geq j+1}] \geq z_j.$$

(G2) *For each level $j \in [m-2]$ and all $\gamma \in (0, \gamma_0]$, if $|P \cap A_{\geq j}| \geq \gamma_0 \lambda$ and $|P \cap A_{\geq j+1}| \geq \gamma \lambda$, then*

$$\Pr_{y \sim D(P)}[y \in A_{\geq j+1}] \geq (1+\delta)\gamma.$$

(G3) *The population size λ satisfies*

$$\lambda \geq \frac{4}{\gamma_0 \delta^2} \ln\left(\frac{128m}{z^* \delta^2}\right), \text{ where } z^* = \min_{j \in [m-1]} z_j.$$

Then we have

$$E[T] \leq \frac{8}{\delta^2} \sum_{j=1}^{m-1} \left(\lambda \ln\left(\frac{6\delta\lambda}{4 + z_j \delta \lambda}\right) + \frac{1}{z_j} \right).$$

The proof (given in [4]) used drift theory with an intricate potential function.

3.2 A Tighter Level-Based Theorem

We now derive from our multiplicative up-drift theorems a version of the level-based theorem with (tight) linear dependence on δ.

Theorem 3.2 (Level-Based Theorem). *Consider a population-based process as follows. For each possible population P of λ individuals from the search space $\mathcal{X}$, there is a distribution $D(P)$ on $\mathcal{X}$. Starting with an arbitrary population P_0, we iteratively define P_{t+1} by sampling λ times independently from $D(P_t)$.*

Assume that there are a partition $(A_1, \dots, A_m)$ of $\mathcal{X}$, a constant C, numbers $z_1, \dots, z_{m-1}, \delta \in (0, 1]$, and $\gamma_0 \in (0, 1)$ such that for any population $P \in \mathcal{X}^\lambda$ the following three conditions are satisfied, where we use the short-hand $A_{\geq j} := \bigcup_{k=j}^m A_k$.

(G1) *For each level $j \in [m-1]$, if $|P \cap A_{\geq j}| \geq \gamma_0 \lambda$, then*

$$\Pr_{y \sim D(P)}[y \in A_{\geq j+1}] \geq z_j.$$

(G2) *For each level $j \in [m-2]$ and all $\gamma \in (0, \gamma_0]$, if $|P \cap A_{\geq j}| \geq \gamma_0 \lambda$ and $|P \cap A_{\geq j+1}| \geq \gamma \lambda$, then*

$$\Pr_{y \sim D(P)}[y \in A_{\geq j+1}] \geq (1+\delta)\gamma.$$

(G3) *The population size λ satisfies*

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

Then $T := \min\{\lambda t \mid P_t \cap A_m \neq \emptyset\}$ satisfies

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

Besides the linear dependence on δ, the new level-based theorem differs from the previous one in a few minor technical details. One is the minimum value for λ in **(G3)**. Here we first note that our leading constant is larger by a factor of two, but as said before, we did not aim to optimize the constants in this first work giving the right dependence on δ. We then observe that in our bound on λ, the logarithmic factor differs from the corresponding one in the previous theorem. A precise and complete comparison is not totally trivial, but for almost all reasonable values of the variables our $O(\log \lambda + \frac{1}{z^* \lambda})$ expression should be significantly smaller than the corresponding expression $\frac{128}{z^* \delta}$ in the previous theorem. Minimally annoying in our version is that λ appears also in the right-hand side of **(G3)**, but since it only appears inside a logarithm, an optimal solution of this inequality is not important.

For the resulting time bound, as said, the main difference is the dependence on δ in the term reflecting the times needed to gain fitness levels, which is $O(\frac{1}{\delta} \sum_{j=1}^{m-1} \frac{1}{z_j})$ in our version and $\frac{8}{\delta^2} \sum_{j=1}^{m-1} \frac{1}{z_j}$ in the previous version. The remaining term, which in our theorem is the total time taken to fill the levels up to a fraction of γ_0, is again harder to compare. This is even more true since for the previous theorem, it is not totally clear where this term $\frac{8}{\delta^2}\sum_{j=1}^{m-1}\lambda\ln(\frac{6\delta\lambda}{4+z_j\delta\lambda})$ comes from (apart from a drift analysis with a complicated potential function). In particular, since this term also depends on the z_i, it seems to encompass not only the times to fill the levels. In any case, it seems to us that in most applications (i) these two terms will be of a similar order of magnitude or, when δ is small, the term in our theorem will be smaller, and (ii) when the parameters of the algorithm under investigation are chosen wisely, the times for finding improving solutions should clearly dominate the times for filling up the levels; hence the precise comparison of these terms is less important.

We now prove the new level-based theorem.

Proof. Our proof proceeds as follows. First we will show that we have multiplicative up-drift for the number of individuals on the lowest level which does not have at least $\gamma_0\lambda$ individuals, conditional on never losing a level. A simple induction allows us to go up level by level. Then we show that any level which has at least $\gamma_0\lambda$ individuals will not get lost until the optimization ends, with sufficiently high probability.

Since we are only interested in the time until we have the first individual in A_m, we may assume that condition **(G2)** also holds for $j = m-1$. Let a level $j \le m-1$ be given such that $|P \cap A_{\ge j}| \ge \gamma_0\lambda$; we now condition on never losing level j, that is, on never having less than $\gamma_0\lambda$ individuals on level j or higher. We let (X_t) be the random process describing the number of individuals on level $j+1$ or higher, that is, we have $X_t = |P_t \cap A_{\ge j+1}|$ for all t.

From **(G1)** we have that if $X_t = 0$, then the number $Y := X_{t+1}$ of individuals sampled in $A_{\ge j+1}$ follows a binomial law with parameters λ and success probability $p \ge z_j$. Consequently, the probability to sample at least one individual on a higher level is at least $\Pr[Y \ge 1] \ge 1-(1-z_j)^\lambda \ge \frac{1}{1+\frac{1}{z_j\lambda}} \ge \frac{1}{2}\min\{z_j\lambda, 1\} =: \varepsilon$, using the elementary, but very convenient estimate from [25, Lemma 9].

We now estimate $E_0 := E[\min\{100/\delta, \gamma_0\lambda, Y\} \mid Y \ge 1]$. Obviously, $E_0 \ge 1$. Assume that $\lambda z_j \ge 1$ and hence $E[Y] \ge 1$. Since a binomial random variable with expectation at least 1 is at least its expectation with probability at least $\frac{1}{4}$ [7, 13] and since, trivially, $(Y \mid Y \ge 1) \ge Y$, we have $E_0 \ge \frac{1}{4}\min\{100/\delta, \gamma_0\lambda, E[Y]\} = \frac{1}{4}\min\{100/\delta, \gamma_0\lambda, \lambda z_j\}$. Consequently, in any case $E_0\varepsilon \ge \frac{1}{8}\min\{100/\delta, \gamma_0\lambda, \lambda z_j\}$.

From **(G2)** we see that when $X_t > 0$, then the number X_{t+1} of individuals sampled on level $j+1$ or higher stochastically dominates a binomial law with parameters λ and $(1+\delta)X_t/\lambda$. Consequently, we can apply Theorem 2.2 and estimate that the expected number of generations until there are at least $\gamma_0\lambda$ individuals on level $j+1$ or higher is

$$O\left(\frac{1}{E_0\varepsilon\delta} + \frac{\log(\gamma_0\lambda)}{\delta}\right) = O\left(\frac{1}{\lambda z_j\delta} + \frac{\log(\gamma_0\lambda)}{\delta}\right).$$

Summing over all levels, we obtain the desired bound on the number of steps to reach a search point in A_m:

$$O\left(\sum_{i=1}^{m-1}\left(\frac{1}{\lambda z_j\delta} + \frac{\log(\gamma_0\lambda)}{\delta}\right)\right) = O\left(\frac{m\log(\gamma_0\lambda)}{\delta} + \frac{1}{\lambda\delta}\sum_{i=1}^{m-1}\frac{1}{z_j}\right).$$

We now argue that we indeed do not lose a level with at least $\gamma_0\lambda$ individual (except for a small failure probability). We let k be four times the implicit constant in the run time bound just computed.

Consider now a level $j \le m-1$ such that $|P \cap A_{\ge j}| \ge \gamma_0\lambda$. We use **(G2)** to see that the probability of any generated individual to be at least on level j is

$$\Pr_{y\sim D(P)}[y \in A_{\ge j}] \ge (1+\delta)\gamma_0.$$

Thus, the expected number of generated individuals on level j is at least $(1+\delta)\gamma_0\lambda$. We now want to determine the probability of undershooting this expected value by a factor of $1-\delta/2$; for this we use a multiplicative Chernoff bound and see that this probability is at most

$$\exp(-\delta^2\gamma_0\lambda/8) \overset{\textbf{(G3)}}{\le} \left(km\frac{\log(\lambda)+(z^*)^{-1}/\lambda}{\delta}\right)^{-1}.$$

Then, after $km\frac{\log(\lambda)+(z^*)^{-1}/\lambda}{\delta}/2$ generations, the process is done with probability at least $1/2$, if no level was ever lost. We bound the probability of ever losing a level in this time by a union bound with $1/2$. Since, conditional on never losing a level, we succeed in this time with probability at least $1/2$, we succeed overall with probability at least $1/4$. A simple restart argument concludes the proof. □

4 APPLICATIONS

With the improved level-based theorem, we easily obtain the following results. Both improve previous results that were obtained via the first level-based theorem.

4.1 Fitness-Proportionate Selection

Dang and Lehre [5] showed that fitness-proportionate selection can be efficient when the mutation rate is very small; in contrast to previous results that show, for the standard mutation rate $1/n$, that fitness-proportionate selection can lead to exponential runtimes [14, 24]. More precisely, Dang and Lehre regard the (λ, λ) EA with fitness-proportionate selection for variation and standard-bit mutation as variation operator (Algorithm 1). Here fitness-proportionate selection (with respect to a non-negative fitness function f) means that from a given population $x_1, \ldots, x_\lambda$ we choose a random element such that x_i is chosen with probability $\frac{f(x_i)}{\sum_{j=1}^{\lambda} f(x_j)}$. When $\sum_{j=1}^{\lambda} f(x_j)$ is zero, we choose an individual uniformly at random.

Dang and Lehre show that this algorithm with mutation rate $p_{\text{mut}} = \frac{1}{6n^2}$ and population size $\lambda = bn^2\ln n$ for some constant $b > 0$ optimizes the OneMax and LeadingOnes benchmark functions in an expected number of $O(n^8\log n)$ fitness evaluations. With our tighter version of the level-based theorem, we obtain the following result. We note that the previous improved level-based theorem (Theorem 3.1) would give a bound of $O(n^5\log^2 n)$ for the smallest-possible choice of λ.

Theorem 4.1. *Consider the (λ, λ) EA with fitness-proportionate selection,*

- *with population size $\lambda \ge cn^2\ln(n)$ with c sufficiently large and $\lambda = O(n^K)$ for some constant K, and*

Algorithm 1: The (λ, λ) EA with fitness-proportionate selection and mutation rate p_{mut} to maximize a function $f : \{0,1\}^n \to \mathbb{R}_{\geq 0}$.

Initialize P_0 as multi-set of λ individuals chosen independently and uniformly at random from $\{0,1\}^n$;
for $t = 1, 2, 3, \ldots$ **do**
  $P_t \leftarrow \emptyset$;
  for $i = 1$ **to** λ **do**
    select $x \in P_{t-1}$ via fitness-proportional selection;
    generate y from x by flipping each bit independently with probability p_{mut};
    $P_t \leftarrow P_t \cup \{y\}$;

- *mutation rate* $p_{\text{mut}} \leq \frac{1}{4n^2}$ *and* $p_{\text{mut}} = \Omega(\frac{1}{\lambda \log n})$.

Then in an expected number of $O(n^2 \lambda \log n)$ *fitness evaluations, this algorithm optimizes both the* OneMax *and the* LeadingOnes *function. For the best-possible choice of* λ*, this bound is* $O(n^4 \log^2 n)$.

Proof. Let f be either of the two functions OneMax or LeadingOnes. We apply Theorem 3.2 with $\gamma_0 = \frac{1}{2}$, $C = 1$, and the partition formed by the sets $A_i := \{x \in \{0,1\}^n \mid f(x) = i - 1\}$ with $i = 1, 2, \ldots, n+1 =: m$.

To show **(G1)**, assume that we have at least $\gamma_0 \lambda$ individuals with fitness at least j for some $j \in [0..n-1]$. Since the selection operator favors individuals with higher fitness, the probability that the parent of a particular offspring has fitness at least j, is at least γ_0. Given that such a parent was chosen (and that this does not have fitness n since we would be done then anyway), the probability to generate a strictly better search point is at least $p_{\text{mut}}(1-p_{\text{mut}})^{n-1} \geq p_{\text{mut}}(1-(n-1)p_{\text{mut}}) = p_{\text{mut}}(1-o(1))$ since $p_{\text{mut}} = o(\frac{1}{n})$. Hence we have **(G1)** satisfied with $z^* = z_j = \gamma_0 p_{\text{mut}}(1-o(1))$.

To show **(G2)**, let $j \in [0..n-2]$, $\gamma \in (0, \gamma_0]$ and P be a population such that at least $\gamma\lambda$ individuals have a fitness of at least $j+1$. Let F^+ be the sum of the fitness values of the individuals of fitness at least $j+1$ and let $F^- = \sum_{x \in P} f(x) - F^+$ be the sum of the remaining fitness values. By our assumption, $F^+ \geq \gamma\lambda(j+1)$. The probability that an individual of fitness $j+1$ or more is chosen as parent of a particular offspring is

$$\begin{aligned}\frac{F^+}{\sum_{x \in P} f(x)} &= \frac{F^+}{F^+ + F^-} \\ &\geq \frac{\gamma\lambda(j+1)}{\gamma\lambda(j+1) + F^-} \\ &\geq \frac{\gamma\lambda(j+1)}{\gamma\lambda(j+1) + (1-\gamma)\lambda j} \\ &= \gamma\left(1 + \frac{1-\gamma}{j+\gamma}\right) \geq \gamma\left(1 + \frac{\frac{1}{2}}{j + \frac{1}{2}}\right) \geq \gamma\left(1 + \frac{1}{2n}\right).\end{aligned}$$

The probability that a parent creates an identical offspring is $(1-p_{\text{mut}})^n \geq 1 - np_{\text{mut}}$. Consequently, the probability that an offspring has fitness at least $j+1$ is at least γ times $(1+\frac{1}{2n})(1-np_{\text{mut}}) \geq 1 + \frac{1}{2n} - np_{\text{mut}} - O(n^{-2}) \geq 1 + \frac{1}{4n} - O(n^{-2}) =: 1 + \delta$. With this δ, we have satisfied **(G2)**.

Finally, we observe that

$$\begin{aligned}\frac{8}{\gamma_0 \delta^2} \log\left(\frac{Cm}{\delta}\left(\log\lambda + \frac{1}{z^*\lambda}\right)\right) &= O\left(n^2 \log\left(O(n^2)\left(\log n^K + O(\log^{-1} n)\right)\right)\right) \\ &= O\left(n^2 \log n\right) = O(\lambda),\end{aligned}$$

that is, we have **(G3)**.

Consequently, we can employ Theorem 3.2 and derive an expected optimization time of

$$E[T] = O\left(\frac{m\lambda \log(\gamma_0\lambda)}{\delta} + \frac{m}{z^*\delta}\right) = O(n^2 \lambda \log n),$$

which is $O(n^4 \log^2 n)$ for the smallest-possible choice of λ. □

4.2 Partial Evaluation

Also in Dang and Lehre [5] a different parent selection algorithm was considered, 2-tournament selection, where a parent is chosen by picking two individuals uniformly at random and the fitter one is allowed to produce one offspring (see Algorithm 2).

Algorithm 2: The (λ, λ) EA with 2-tournament selection and mutation rate p_{mut} to optimize a function $f : \{0,1\}^n \to \mathbb{R}_{\geq 0}$.

Initialize P_0 as multi-set of λ individuals chosen independently and uniformly at random from $\{0,1\}^n$;
for $t = 1, 2, 3, \ldots$ **do**
  $P_t \leftarrow \emptyset$;
  for $i = 1$ **to** λ **do**
    select $x_0, x_1 \in P_{t-1}$ uniformly at random;
    select $x \in \{x_0, x_1\}$ with maximal fitness (breaking ties uniformly);
    generate y from x by flipping each bit independently with probability p_{mut};
    $P_t \leftarrow P_t \cup \{y\}$;

The test functions they considered were OneMax and LeadingOnes under partial evaluation (a scheme for randomizing a given function), which we here define only for OneMax. Given a parameter $c \in (0,1)$, we use n i.i.d. random variables $(R_i)_{i \leq n}$, each Bernoulli-distributed with parameter c. OneMax_c is defined such that, for all bit strings $x \in \{0,1\}^n$, $\text{OneMax}_c(x) = \sum_{i=1}^n R_i x_i$. With other words, a bit string has a value equal to the number of 1s in it, where each 1 only counts with probability c.

Dang and Lehre [5] showed the following statement as part of their core proof [5, proof of Theorem 21] regarding the performance of Algorithm 2 on $\text{OneMax}_c(x)$.

Lemma 4.2. *Let n be large and $c \in (1/n, 1)$. Then there is an a such that, for all $\gamma \in (0, 1/2)$, the probability to produce an offspring of at least the quality of the $\gamma\lambda$-ranked individual of the current population is at least $\gamma(1 + a\sqrt{c/n})$.*

Using their old level-based theorem (with a dependence on δ of order 5) and the best possible choice for λ, they obtain a bound for the expected number of fitness evaluations until optimizing OneMax with partial evaluation with parameter $c \geq 1/n$ of

$$O\left(\frac{n^{4.5}\log n}{c^{3.5}}\right).$$

Using the more refined level-based theorem from [4], see Theorem 3.1 (with a quadratic dependence on δ), one can find a run time bound of

$$O\left(\frac{n^{3}\log n}{c^{2}}\right).$$

With our level-based theorem given in Theorem 3.2 (with a linear dependence on δ), one can prove a run time bound of

$$O\left(\frac{n^{2.5}(\log(n))^2}{c^{1.5}}\right).$$

For this we chose analogously to [5]: $\delta = a\sqrt{c/n}$ as given in Lemma 4.2, $p_{\mathrm{mut}} = \delta/3$, $m = n+1$ (with the partitioning based on fitness), $\gamma_0 = 1/2$, $z_j = (1 - j/n)(\delta/9)$ and $\lambda = bn\ln(n)/c$ for some constant b.

Analogous improvements can be found in the case of LeadingOnes.

5 CONCLUSION

In this work, we prove two drift results for multiplicatively increasing drift. Since the desired hitting time bound of order $\log(n)/\delta$, which implies that the process behaves similarly to the deterministic process, can only be obtained under additional assumptions, we formulate our results for processes in which each state X_{t+1} is distributed according to a binomial distribution with expectation $(1+\delta)X_t$ (or better in the domination sense).

As main application for our drift results, we prove a stronger version of the level-based theorem. It in particular has the asymptotically right dependence on $1/\delta$, which is linear. Previous level-based theorems only shows a dependence of order δ^{-5} [5] or δ^{-2} [4]. This difference can be significant in applications with small δ, e.g., the result on fitness-proportionate selection [5], which has $\delta = \Theta(1/n)$.

An equally interesting progress from our new level-based theorem is that its relatively elementary proof gives more insight in the actual development of such processes. It thus tells us in a more informative manner how certain population-based algorithms optimize certain problems. Such additional information can be useful to detect bottlenecks and improve algorithms.

The fact that our proof is more explanatory might also help further extending this level-based theorem. For example, at the moment the assumptions are formulated in a way that the run time is estimated by the sum of the times needed to fill each level up to a fraction of γ_0 plus the times taken to find improvements from such levels. It is known from other analyzes like [1, 27] that often the best run times can be shown by waiting for different occupation ratios for different levels. If an improvement is easy to find from a certain level, then it pays off less to fill it up to a constant fraction. For our version of the level-based theorem, we are optimistic that an extension to different occupation ratios can be obtained with similar methods. Such an extension of the theorem could be useful to prove an asymptotically tight bound for the run time of the (μ, λ) EA on OneMax. The current bounds, obtainable from any of the level-based theorems (since δ is constant) most likely are not tight. Note that for λ sufficiently larger than μ and μ large enough, the (μ, λ) EA should have a similar run time as the $(\mu + \lambda)$ EA since good individuals are not lost due to producing sufficiently many copies, but the current bounds for the (μ, λ) EA only show larger run times.

In terms of future work, we also note that there are processes showing multiplicative up-drift where the next state is not described by a binomial distribution. One example are population-based algorithms using plus-selection, where, roughly speaking, $X_{t+1} \sim X_t + \mathrm{Bin}(\lambda, X_t/\lambda)$. We are optimistic that such processes can be handled with our methods as well. We did not do this in this first work on multiplicative up-drift since such processes can also be analyzed with elementary methods, e.g., exploiting that the process is non-decreasing and with constant probability attains the expected progress. Nevertheless, extending our drift theorems to such processes should give better constants and a more elegant analysis, so we feel that this is also an interesting goal for future work.

REFERENCES

[1] Denis Antipov, Benjamin Doerr, Jiefeng Fang, and Tangi Hetet. Runtime analysis for the $(\mu + \lambda)$ EA optimizing OneMax. In *Genetic and Evolutionary Computation Conference, GECCO 2018*, pages 1459–1466. ACM, 2018.

[2] Denis Antipov, Benjamin Doerr, and Quentin Yang. The efficiency threshold for the offspring population size of the (μ, λ) EA. In *Genetic and Evolutionary Computation Conference, GECCO 2019*. ACM, 2019. To appear.

[3] Sylvain Colin, Benjamin Doerr, and Gaspard Férey. Monotonic functions in EC: anything but monotone! In *Genetic and Evolutionary Computation Conference, GECCO 2014*, pages 753–760. ACM, 2014.

[4] Dogan Corus, Duc-Cuong Dang, Anton V. Eremeev, and Per Kristian Lehre. Level-based analysis of genetic algorithms and other search processes. *IEEE Transactions on Evolutionary Computation*, 22:707–719, 2018.

[5] Duc-Cuong Dang and Per Kristian Lehre. Runtime analysis of non-elitist populations: from classical optimisation to partial information. *Algorithmica*, 75:428–461, 2016.

[6] Benjamin Doerr. Better runtime guarantees via stochastic domination. In *Evolutionary Computation in Combinatorial Optimization, EvoCOP 2018*, pages 1–17. Springer, 2018.

[7] Benjamin Doerr. An elementary analysis of the probability that a binomial random variable exceeds its expectation. *Statistics and Probability Letters*, 139:67–74, 2018.

[8] Benjamin Doerr and Leslie A. Goldberg. Adaptive drift analysis. *Algorithmica*, 65:224–250, 2013.

[9] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. *Algorithmica*, 64:673–697, 2012.

[10] Benjamin Doerr and Timo Kötzing. Multiplicative up-drift. *CoRR*, abs/1904.05682, 2019.

[11] Christian Gießen and Timo Kötzing. Robustness of populations in stochastic environments. In *Genetic and Evolutionary Computation Conference, GECCO 2014*, pages 1383–1390. ACM, 2014.

[12] Andreas Göbel, Timo Kötzing, and Martin S. Krejca. Intuitive analyses via drift theory. *CoRR*, arxiv:1806.01919, 2018.

[13] Spencer Greenberg and Mehryar Mohri. Tight lower bound on the probability of a binomial exceeding its expectation. *Statistics and Probability Letters*, 86:91–98, 2014.

[14] Edda Happ, Daniel Johannsen, Christian Klein, and Frank Neumann. Rigorous analyses of fitness-proportional selection for optimizing linear functions. In *Genetic and Evolutionary Computation Conference, GECCO 2008*, pages 953–960. ACM, 2008.

[15] Jun He and Xin Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127:57–85, 2001.

[16] Jun He and Xin Yao. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3:21–35, 2004.

[17] Thomas Jansen. On the brittleness of evolutionary algorithms. In *Foundations of Genetic Algorithms, FOGA 2007*, pages 54–69. Springer, 2007.

[18] Daniel Johannsen. *Random Combinatorial Structures and Randomized Search Heuristics*. PhD thesis, Universität des Saarlandes, 2010. Available online at http://scidok.sulb.uni-saarland.de/volltexte/2011/3529/pdf/Dissertation_3166_Joha_Dani_2010.pdf.

[19] Timo Kötzing, Andrei Lissovoi, and Carsten Witt. (1+1) EA on generalized dynamic OneMax. In *Foundations of Genetic Algorithms, FOGA 2015*, pages 40–51. ACM, 2015.
[20] Per Kristian Lehre. Fitness-levels for non-elitist populations. In *Genetic and Evolutionary Computation Conference, GECCO 2011*, pages 2075–2082. ACM, 2011.
[21] Per Kristian Lehre and Phan Trung Hai Nguyen. Improved runtime bounds for the univariate marginal distribution algorithm via anti-concentration. In *Genetic and Evolutionary Computation Conference, GECCO 2017*, pages 1383–1390. ACM, 2017.
[22] Per Kristian Lehre and Phan Trung Hai Nguyen. Level-based analysis of the population-based incremental learning algorithm. In *Parallel Problems Solving From Nature, PPSN 2018*, pages 105–116. Springer, 2018.
[23] Boris Mitavskiy, Jonathan E. Rowe, and Chris Cannings. Preliminary theoretical analysis of a local search algorithm to optimize network communication subject to preserving the total number of links. In *Congress on Evolutionary Computation, CEC 2008*, pages 1484–1491. IEEE, 2008.
[24] Frank Neumann, Pietro Simone Oliveto, and Carsten Witt. Theoretical analysis of fitness-proportional selection: landscapes and efficiency. In *Genetic and Evolutionary Computation Conference, GECCO 2009*, pages 835–842. ACM, 2009.
[25] Jonathan E. Rowe and Dirk Sudholt. The choice of the offspring population size in the (1, λ) evolutionary algorithm. *Theoretical Computer Science*, 545:20–38, 2014.
[26] Ingo Wegener. Theoretical aspects of evolutionary algorithms. In *Automata, Languages and Programming, ICALP 2001*, pages 64–78. Springer, 2001.
[27] Carsten Witt. Runtime analysis of the (μ + 1) EA on simple pseudo-Boolean functions. *Evolutionary Computation*, 14:65–86, 2006.