Self-adjusting population sizes for the (1,λ)-EA on monotone functions

Journal Article

Author(s):
Kaufmann, Marc; Larcher, Maxime; Lengler, Johannes; Zou, Xun

Publication date:
2023-11-10

Permanent link:
https://doi.org/10.3929/ethz-b-000637220

Originally published in:
Theoretical Computer Science 979, https://doi.org/10.1016/j.tcs.2023.114181

Funding acknowledgement:
192079 - DynaGIRG (SNF)
173721 - Temporal Information Integration in Neural Networks (SNF)

Theoretical Computer Science 979 (2023) 114181

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Self-adjusting population sizes for the $(1, \lambda)$-EA on monotone functions [☆]

Marc Kaufmann [*,1], Maxime Larcher, Johannes Lengler, Xun Zou [2]

Department of Computer Science, ETH Zürich, Zürich, Switzerland

ARTICLE INFO

Article history:
Received 26 September 2022
Received in revised form 13 July 2023
Accepted 7 September 2023
Available online 19 September 2023
Communicated by C. Witt

Keywords:
Parameter control
Self-adaptation
$(1, \lambda)$-EA
One-fifth rule
Monotone functions
Dynamic environments
Evolutionary algorithm

ABSTRACT

We study the $(1, \lambda)$-EA with mutation rate c/n for $c \leq 1$, where the population size is adaptively controlled with the $(1 : s + 1)$-success rule. Recently, Hevia Fajardo and Sudholt have shown that this setup with $c = 1$ is efficient on OneMax for $s < 1$, but inefficient if $s \geq 18$. Surprisingly, the hardest part is not close to the optimum, but rather at linear distance. We show that this behaviour is not specific to OneMax. If s is small, then the algorithm is efficient on all monotone functions, and if s is large, then it needs superpolynomial time on all monotone functions. In the former case, for $c < 1$ we show a $O(n)$ upper bound for the number of generations and $O(n \log n)$ for the number of function evaluations, and for $c = 1$ we show $O(n \log n)$ generations and $O(n^2 \log\log n)$ evaluations. We also show formally that optimization is always fast, regardless of s, if the algorithm starts in proximity of the optimum. All results also hold in a dynamic environment where the fitness function changes in each generation.

An extended abstract, containing only the results without proofs, has been published at the PPSN conference [1].

1. Introduction

1.1. Background

Randomized Optimization Heuristics (ROHs) like evolutionary algorithms (EAs) are simple general-purpose optimizers. One of their strengths is that they can often be applied with little adaptation to the problem at hand. However, ROHs usually come with parameters, and their efficiency often depends on the parameter settings. Therefore, *parameter control* is a fundamental topic in the design and analysis of ROHs [2]. It aims at providing methods to automatically tune parameters over the course of optimization. The goal is not to remove parameters altogether; the parameter control mechanisms themselves introduce new meta-parameters. Nevertheless, there are two objectives that can sometimes be achieved with parameter control mechanisms.

[☆] This article belongs to Section C: Theory of natural computing, Edited by Lila Kari.

[*] Corresponding author.

E-mail addresses: marc.kaufmann@inf.ethz.ch (M. Kaufmann), larcherm@inf.ethz.ch (M. Larcher), johannes.lengler@inf.ethz.ch (J. Lengler), xun.zou@inf.ethz.ch (X. Zou).

[1] The author was supported by the Swiss National Science Foundation [grant number 192079].

[2] The author was supported by the Swiss National Science Foundation [grant number CR-SII5_173721].

https://doi.org/10.1016/j.tcs.2023.114181

Firstly, some ROHs are rather sensitive to small changes in the parameters, and inadequate setting can slow down or even prevent success. Two examples that are relevant for this paper are the $(1, \lambda)$-EA, which fails to optimize even the easy OneMax benchmark if λ is too small [3–5], and the $(1 + 1)$-EA, which fails on monotone functions if the mutation rate is too large [6–8]. In both cases, changing the parameters just by a constant factor makes all the difference between finding the optimum in time $O(n \log n)$, and not even finding an ε-approximation of the optimal solution in polynomial time. So these algorithms are extremely sensitive to small changes of parameters. In such cases, one hopes that performance is *more robust* with respect to the meta-parameters, i.e., that the parameter control mechanism manages to find a decent parameter setting regardless of its meta-parameters.

Secondly, often there is no single parameter setting that is optimal throughout the course of optimization. Instead, different phases of the optimization process profit from different parameter settings, and the overall performance with *dynamically adapted* parameters is better than for any static parameters [9–13]. This topic, which has always been studied in continuous optimization, has taken longer to gain traction in discrete domains [14–16,11] but has attracted increasing interest over the last years [17–30]. Instead of a detailed discussion we refer the reader to the book chapter [31] for an overview over theoretical results, and to [32] for a discussion of some recent developments.

One of the most traditional and influential methods for parameter control is the $(1 : s + 1)$-success rule [33], independently developed several times [34–36] and traditionally used with $s = 4$ as one-fifth rule in continuous domains, e.g. [37]. This rule has been used for controlling the offspring population size in discrete domains [11,16], in particular for the $(1, \lambda)$-EA [32,38], where it yields the so-called *self-adjusting* $(1, \lambda)$*-EA* or *SA-*$(1, \lambda)$*-EA*, also called $(1, \{\lambda/F, F^{1/s}\lambda\})$-EA. As in the basic $(1, \lambda)$-EA, in each generation the algorithm produces λ offspring, and selects the fittest of them as the unique parent for the next generation. The difference to the basic $(1, \lambda)$-EA is that the parameter λ is replaced by λ/F if the fittest offspring is fitter than the parent, and by $\lambda \cdot F^{1/s}$ otherwise. We will give a more thorough discussion of this algorithm in Section 2 below. Thus, the $(1 : s + 1)$-success rule replaces the parameter λ by two parameters s and F. As outlined above, there are two hopes associated with this scheme:

(i) that the performance is *more robust* with respect to F and s than with respect to λ;
(ii) that the scheme can adaptively find the *locally optimal* value of λ throughout the course of optimization.

1.2. Prior work

Recently, Hevia Fajardo and Sudholt have investigated both hypotheses on the OneMax benchmark [38,32]. They found a negative result for (i), and a (partial) positive result for (ii). The negative result says that performance is at least as fragile with respect to the parameters as before: if $s < 1$, then the SA-$(1, \lambda)$-EA finds the optimum of OneMax in $O(n)$ generations, but if $s \geq 18$ and $F \leq 1.5$ the runtime becomes exponential with overwhelming probability. Experimentally, they find that the range of bad parameter values even seems to include the standard choice $s = 4$, which corresponds to the 1/5-rule. On the other hand, they show that for $s < 1$, the algorithm successfully achieves (ii): they show that the expected number of function evaluations is $O(n \log n)$, which is optimal among all unary unbiased black-box algorithms [12,39]. Moreover, they show that the algorithm makes steady progress over the course of optimization, needing $O(b - a)$ generations to increase the fitness from a to b whenever $b - a \geq C \log n$ for a suitable constant C. The crucial point is that this is independent of a and b, so independent of the current state of the algorithm. It implies that the algorithm chooses $\lambda = O(1)$ in early stages when progress is easy, and (almost) linear values $\lambda = \Omega(n)$ in the end when progress is hard. Thus, it achieves (ii) conditional on having appropriate parameter settings.

Interestingly, it is shown in [32] that for $s \geq 18$, the SA-$(1, \lambda)$-EA fails in a region far away from the optimum, more precisely in the region with 85% one-bits. Consequently, it also fails for every other function that is identical with OneMax in the range of $[0.84n, 0.85n]$ one-bits, which includes other common benchmarks like Jump, Cliff, and Ridge. It is implicit that the algorithm would be efficient in regions that are closer to the optimum. This is remarkable, since usually optimization is harder close to the optimum. Such a reversed failure profile has previously only been observed in very few situations. One is the $(\mu + 1)$-EA with mutation rate c/n for an arbitrary constant $c > 0$ on certain monotone functions. This algorithm is efficient close to the optimum, but fails to cross some region in linear distance of the optimum if $\mu > \mu_0$ for some μ_0 that depends on c [40]. A similar phenomenon has been shown for $\mu = 2$ and a specific value of c in the dynamic environment Dynamic BinVal [41,42]. These are the only examples for this phenomenon that the authors are aware of.

1.3. Our results[3]

A limitation of [32] is that it studies only a single benchmark, the OneMax function. Although the negative result also holds for functions that are identical to OneMax in some range, the agreement with OneMax in this range must be perfect, and the positive result does not extend to other functions in such a way. This leaves the question on what happens for larger classes of benchmarks:

[3] An extended abstract, containing only the results without proofs, has been published at the PPSN conference [1].

(a) Is there a safe choice for s that makes the algorithm efficient for a whole class of functions?
(b) Does the positive result (ii) extend to other benchmarks than OneMax?

In this paper, we will answer both questions with *Yes* for the set of all (strictly) monotone pseudo-Boolean functions, i.e., functions where flipping a zero-bit into a one-bit always increases the fitness.[4] This is a very large class; for example, it contains all linear functions. In fact, all our results hold in an even more general *dynamic* setting: the fitness function may be different in each generation, as long as it is a monotone function every time and therefore shares the same global optimum $(1, \dots, 1)$. We show an upper bound of $O(n)$ generations (Theorem 18) and $O(n \log n)$ function evaluations (Theorem 28) if the mutation rate is c/n for some $c < 1$, which is a very natural assumption for monotone functions as many algorithms become inefficient for large values of c [7,8,45,46]. Those results are as strong as the positive results in [32], except that we replace the constant "1" in the condition $s < 1$ by a different constant that may depend on c. For $c = 1$ we still show that a bound of $O(n \log n)$ generations (Theorem 19) and $O(n^2 \log\log n)$ evaluations (Theorem 29). It is in line with general frameworks for elitist algorithms that the number of function evaluations stops being quasi-linear [47,7], although the bounds in other contexts are better than quadratic [48].

Both parts of the answer are encouraging news for the SA-$(1, \lambda)$-EA. It means that, at least for this class of benchmarks, there is a universal parameter setting that works in all situations. This resembles the role of the mutation rate c/n for the $(1+1)$-EA on monotone functions: If $c < 1 + \varepsilon$, then the $(1+1)$-EA is efficient on all monotone functions [7,8,45], and for $c < 1$ this is known for many other algorithms as well [46]. On the other hand, the $(\mu + 1)$-EA is an example where such a safe parameter choice for c does not exist: for any $c > 0$ there is μ such that the $(\mu + 1)$-EA with mutation rate c/n needs super-polynomial time to find the optimum of some monotone functions.

We do not just strengthen the positive result, but we show that the negative result generalizes in a surprisingly strong sense, too: for any arbitrary mutation rate c/n where $c < 1$, if s is sufficiently large, then the SA-$(1, \lambda)$-EA needs exponential time on *every* monotone function, Theorem 40. Thus, the failure mode for large s is not specific to OneMax. On the other hand, we also generalize the result (implicit in [32]) that the only hard region is in linear distance from the optimum: for any value of s, if the algorithm starts close enough to the optimum (but still in linear distance), then with high probability it optimizes every monotone function efficiently, Theorem 35. Finally, we complement the theoretical analysis with simulations in Section 7. These simulations show another interesting aspect: in a 'middle region' of s, it seems to depend on F whether the algorithm is efficient or not. Thus, we conjecture that there does *not* exist an efficiency threshold s_0 such that all parameters $s < s_0$ and $F > 1$ are efficient, while all $s > s_0$ and $F > 1$ are inefficient. Note that our results show that this dependency on F can only appear in a 'middle region', since our results for small s and large s are independent of F (which improves the negative result in [32]). A similar effect was observed for the self-adjusting $(1 + (\lambda, \lambda))$-GA in [49], but for different reasons. There, the effect was caused by a universal upper bound on the success probability of $\approx 0.31 < 1$, independent of λ. This can cause problems if the target success rate is larger than 0.31, and thus unachievable, see [49, Section 6.4] for a full discussion. In our setting, the success probability approaches one as λ grows, so the problem does not exist, and the reason for the impact of F seems different.

Our proofs build on ideas from [32]. In particular, we use a potential function of the form $g(x^t, \lambda^t) = \text{Zm}(x^t) + h(\lambda^t)$, where $\text{Zm}(x^t)$ is the number of zero-bits in x^t and $h(\lambda^t)$ is a penalty term for small values of λ^t. Similar decompositions have been used before [50]. The exact form of h depends on the situation; sometimes it is very similar to the choices in [32] (positive result for $c < 1$, negative result), but some cases are completely different (positive results for $c = 1$ and close to the optimum). With these potential functions, we obtain a positive or negative drift and upper or lower bounds on the number of generations, depending on the situation. When translating the number of generations into the number of function evaluations, while some themes from [32] reappear (e.g., to consider the best-so-far Zm value), the overall argument is different. In particular, we do not use the ratchet argument from [32], see Remark 31 for a discussion of the reasons.

1.4. Discussion of the SA-$(1, \lambda)$-EA

Let us give a short explanation of the concept of the $(1 : s+1)$-success rule (or $(1 : s+1)$-rule for short). For given λ and given position x in the search space, the algorithm has some *success probability* p, where success means that $f(y) > f(x)$ for the fittest of λ offspring y of x. For simplicity we will ignore the rounding effect coming from $\lambda \in \mathbb{N}$ and will assume that $p(\lambda) \le 1/(s+1)$ for $\lambda = 1$, and that $0 < p < 1$. The success probability $p = p(\lambda)$ is obviously an increasing function in λ, since additional offspring can only increase the chances of finding an improvement. Moreover, it is strictly increasing due to $0 < p < 1$. Hence there is a value λ^* such that $p(\lambda) < 1/(s+1)$ for $\lambda < \lambda^*$ and $p(\lambda) > 1/(s+1)$ for $\lambda > \lambda^*$. Now consider the potential $\log_F \lambda$. This potential decreases by 1 with probability p and increases by $1/s$ with probability $1 - p$. So in

[4] Shortly after our work, Hevia Fajardo and Sudholt also provided such a class in [43]. This is the class of *everywhere hard functions*, for which the chance of creating a strict improvement does not exceed $n^{-\varepsilon}$ anywhere in the search space. This includes the popular LeadingOnes benchmark, but not OneMax. In fact, it does not include any monotone function, since from the all-zero string (and from all other strings with $\Omega(n)$ zero-bits) the probability of an improvement is $\Omega(1)$ on monotone functions. Hence, their class is disjoint from ours. In [43] it was shown that for any constant s the SA-$(1, \lambda)$-EA imitates the elitist SA-$(1 + \lambda)$-EA on everywhere hard functions, which by design can never lose fitness. This arguably makes the comma variant a bit pointless for everywhere hard functions, since its potential benefit of escaping from local optima [44] is suppressed in this case.

expectation it changes by $-p + (1-p)/s = (1-(s+1)p)/s$. Hence, the expected change is positive if $\lambda < \lambda^*$ and negative if $\lambda > \lambda^*$. Therefore, λ has a drift towards λ^* from both sides (in a logarithmic scaling). So the rule implicitly has a *target population size* λ^*, and this population size λ^* corresponds to the *target success rate* $p = 1/(s+1)$.

Note that a drift towards λ^* does not necessarily imply that λ always stays close to λ^*. Firstly, p depends on the current state x of the algorithm, and might vary rapidly as the algorithm progresses (though this does not seem a very typical situation). In this case, the target value λ^* also varies. Secondly, even if λ^* remains constant, there may be random fluctuations around this value, see [51,52] for treatments on when drift towards a target guarantees concentration. However, we note that the $(1:s+1)$-rule for controlling λ gives stronger guarantees than the same rule for controlling other parameters like step size or mutation rate. The difference is that other parameters do not necessarily influence p in a monotone way, and therefore we cannot generally guarantee that there is a drift towards success probability $1/(s+1)$ when the $(1:s+1)$-rule is used to control them. Only when controlling λ we are guaranteed a drift in the right direction.

2. Preliminaries and definitions

Throughout the paper we will assume that $c > 0$, $s > 0$ and $F > 1$ are constants independent of n while $n \to \infty$. Note that s need not be an integer. Our search space is always $\{0,1\}^n$, and we denote by $\text{supp}\{x\} := \{i \in [n] \mid x_i = 1\}$ the *support* of a bit string $x \in \{0,1\}^n$. We say that an event $\mathcal{E} = \mathcal{E}(n)$ holds *with high probability* or *whp* if $\Pr[\mathcal{E}] \to 1$ for $n \to \infty$. We denote the negation of an event $\mathcal{E}$ by $\overline{\mathcal{E}}$, and by $\mathbf{1}_{\mathcal{E}}$ the indicator of $\mathcal{E}$, i.e., $\mathbf{1}_{\mathcal{E}} = 1$ if $\mathcal{E}$ holds and $\mathbf{1}_{\mathcal{E}} = 0$ otherwise.

2.1. The algorithm: SA-$(1,\lambda)$-EA

We will consider the self-adjusting $(1,\lambda)$-EA with $(1:s+1)$-success rate, with mutation rate c/n, success ratio s and update strength F, and we denote this algorithm by SA-$(1,\lambda)$-EA. It is given by the following pseudocode. Note that the parameter λ may take non-integral values during the execution of the algorithm, however the number of children generated at each step is chosen to be the closest integer $\lfloor \lambda \rceil$ to λ.

Algorithm 1 SA-$(1,\lambda)$-EA with success rate s, update strength F and mutation rate c/n for maximizing a fitness function $f : \{0,1\}^n \to \mathbb{R}$.

```
Initialization: Choose $x^0 \in \{0,1\}^n$ uniformly at random and $\lambda^0 := 1$
Optimization: for $t = 0, 1, \dots$ do
    Mutation: for $j \in \{1, \dots, \lfloor \lambda^t \rceil\}$ do
        $y^{t,j} \leftarrow$ mutate $(x^t)$ by flipping each bit independently with prob. $c/n$
    Selection: Choose $y^t = \arg\max_i f(y^{t,i})$, breaking ties randomly;
    Update: if $f(y^t) > f(x^t)$ then $\lambda^{t+1} \leftarrow \max\{1, \lambda^t/F\}$;
            else $\lambda^{t+1} \leftarrow F^{1/s}\lambda^t$;
            $x^{t+1} \leftarrow y^t$;
```

We will often omit the index t if it is clear from the context.

2.2. The benchmark: dynamic monotone functions

Whenever we speak of "monotone" functions in this paper, we mean strictly monotone pseudo-Boolean functions, defined as follows.

Definition 1. We call $f : \{0,1\}^n \to \mathbb{R}$ *monotone* if $f(x) > f(y)$ for every pair $x, y \in \{0,1\}^n$ with $x \neq y$ and $x_i \geq y_i$ for all $1 \leq i \leq n$.

In this paper we will consider the following set of benchmarks. For each $t \in \mathbb{N}$, let $f^t : \{0,1\}^n \to \mathbb{R}$ be a monotone function that may change at each step depending on x^t. Then the selection step in the t-th generation of Algorithm 1 is performed with respect to f^t. By slight abuse of notation we will still speak of *a* dynamic monotone function f.

All our results (positive and negative) hold in this dynamic setup. This set of benchmarks is quite general. Of course, it contains the static setup in which we only have a single monotone function to optimize, which includes linear functions and OneMax as special cases. It also contains the setup of Dynamic Linear Functions (originally introduced as Noisy Linear Functions in [53]) and Dynamic BinVal [41,42]. On the other hand, all monotone functions share the same global optimum $(1\dots1)$, have no local optima, and flipping a zero-bit into a one-bit strictly improves the fitness. In the dynamic setup, these properties still hold "locally", within each selection step. Thus, the setup falls into the general framework by Jansen [47], which was extended to the *partially ordered EA* (PO-EA) by Colin, Doerr, Férey [48]. This implies that the $(1+1)$-EA with mutation rate c/n finds the optimum of every such Dynamic Monotone Function in expected time $O(n \log n)$ if $c < 1$, and in time $O(n^{3/2})$ if $c = 1$.

2.3. Drift analysis and potential functions

Drift analysis is a key instrument in the theory of EAs. To apply it, one must define a *potential function* and compute the expected change of this potential. A common potential for simple problems in EAs are the OneMax and ZeroMax potential of the current state x^t, which assign to each search point $x \in \{0,1\}^n$ the number of one-bits and zero-bits, respectively:

$$\textsc{Om}(x) = \textsc{OneMax}(x) = \sum\nolimits_i x_i \text{ and } \textsc{Zm}(x) = \textsc{ZeroMax}(x) = n - \textsc{Om}(x).$$

Note that for two bit strings x and y, $\textsc{Om}(|x-y|)$ computes their Hamming distance, where the difference and absolute value are taken component-wise.

For our purposes, this potential function will not be sufficient since there is an intricate interplay between progress and the value of λ. Following [38,32], we use a composite potential function of the form $g(x^t, \lambda^t) = \textsc{Zm}(x^t) + h(\lambda^t)$, where $h(\lambda^t)$ varies from application to application (Definitions 20, 24, 36, 41). We will write $Z^t := \textsc{Zm}(x^t)$, $H^t := h(\lambda^t)$ and $G^t := g(x^t, \lambda^t)$ throughout the paper.

Once the drift is established, the positive and negative statements about generations then follow from standard drift analysis [52]. In particular, we will use the Additive, Multiplicative, and Negative Drift Theorem, given below.[5]

Theorem 2 *(Additive Drift Theorem [52]). Let $(X^t)_{t\geq 0}$ be a sequence of non-negative random variables over a bounded state space $S \subset \mathbb{R}_0^+$ containing the origin and let $T := \inf\{t \geq 0 \mid X^t = 0\}$ denote the hitting time of 0. Assume there exists $\delta > 0$ such that for all $t < T$,*

$$\mathbf{E}\left[X^t - X^{t+1} \mid X^t\right] \geq \delta,$$

then

$$\mathbf{E}[T \mid X^0] \leq \frac{X^0}{\delta}.$$

Theorem 3 *(Multiplicative Drift Theorem [54]). Let $(X^t)_{t\geq 0}$ be a sequence of non-negative random variables over a bounded state space $S \subset \mathbb{R}_0^+$ containing the origin and such that $x_{\min} := \min\{x \in S : x > 0\}$ is well defined. Let $T := \inf\{t \geq 0 \mid X^t \leq 0\}$ denote the hitting time of 0. Suppose that there exists a constant $\delta > 0$ such that for all $t < T$,*

$$\mathbf{E}[X^t - X^{t+1} \mid X^t] \geq \delta X^t.$$

Then,

$$\mathbf{E}[T \mid X^0] \leq \frac{1 + \log(X^0/x_{\min})}{\delta}.$$

Theorem 4 *(Negative Drift Theorem With Scaling [55]). Let $(X^t)_{t\geq 0}$ be a sequence of random variables over a state space $S \subset \mathbb{R}$. Suppose there exists an interval $[a,b] \subseteq \mathbb{R}$ and, possibly depending on $\ell := b - a$, a drift bound $\varepsilon := \varepsilon(\ell) > 0$ as well as a scaling factor $r := r(\ell)$ such that for all $t \geq 0$ the following three conditions hold:*

1. $\mathbf{E}[X^{t+1} - X^t \mid X^0, \ldots, X^t; a < X_t < b] \geq \varepsilon$.
2. $\Pr[|X^{t+1} - X^t| \geq jr \mid X_0, \ldots, X^t; a < X^t] \leq e^{-j}$ *for* $j \in \mathbb{N}_0$.
3. $1 \leq r^2 \leq \varepsilon\ell/(132\log(r/\varepsilon))$.

Then for all $X^0 \geq b$, the first time $T^* := \min\{t \geq 0\colon X^t < a \mid X^0, \ldots, X^t\}$ when X drops below a satisfies

$$\Pr[T^* \leq e^{\varepsilon\ell/(132r^2)}] = O(e^{-\varepsilon\ell/(132r^2)}).$$

2.4. Concentration of hitting times

In our analysis, we will prove concentration of the number of steps needed to improve the number of 1-bits. We will use the following results of Kötzing [51], which we slightly reformulate for convenience.

Definition 5 *(Sub-Gaussian [51]).* Let $\left(X^t\right)_{t\geq 0}$ be a sequence of random variables and $\mathcal{F} = \left(\mathcal{F}^t\right)_{t\geq 0}$ an adapted filtration. We say that $\left(X^t\right)_{t\geq 0}$ is (γ, δ)-sub-Gaussian if

[5] Theorem 2 and 3 are stated in the references for *finite* state spaces S but continue to hold as long as S is bounded, see the remark on p.6 in [52].

$$\mathbf{E}\left[e^{zX^{t+1}} \mid \mathcal{F}^t\right] \le e^{\gamma z^2/2},$$

for all t and all $z \in [0, \delta]$.

Theorem 6 *(Tail Bounds Imply Sub-Gaussian [51]). For every $0 < \alpha$, $0 < \beta < 1$, there exists $\gamma, \delta > 0$ such that the following holds.*

Let $\left(X^t\right)_{t\ge 0}$ be a random sequence and $\mathcal{F}$ an adapted filtration. Assume that $\mathbf{E}[X^{t+1} \mid \mathcal{F}^t] \le 0$ and for all times t and that for all $x \ge 0$ we have

$$\Pr\left[|X^{t+1}| \ge x \mid \mathcal{F}^t\right] \le \frac{\alpha}{(1+\beta)^x}.$$

Then $\left(X^t\right)_{t\ge 0}$ is (γ, δ)-sub-Gaussian.

Theorem 7 *(Concentration of Hitting Times [51]). For every $\gamma, \delta, \varepsilon > 0$ there exists a $D > 0$ such that the following holds. Let $\left(X^t\right)_{t\ge 0}$ be a random sequence and $\mathcal{F}$ an adapted filtration satisfying the following properties*

(i) $\mathbf{E}[X^{t+1} - X^t \mid \mathcal{F}^t] \ge \varepsilon$.
(ii) $\left(\varepsilon - X^t\right)_{t\ge 0}$ is (γ, δ)-sub-Gaussian;

Let T denote the first point in time when $\sum_{t=1}^{T} X^t \ge N$, then for all $\tau \ge 2N/\varepsilon$,

$$\Pr[T > \tau] \le \exp\left(-D\tau\right).$$

To prove concentration of the number of steps spent improving the fitness under multiplicative drift, we will use the following theorem.

Theorem 8 *(Multiplicative Drift, Tail Bound [56]). Let $(X^t)_{t\ge 0}$ be non-negative random variables over a state space $S \subset \mathbb{R}_0^+$. Assume that $X^0 \le b$ and let T be the random variable that denotes the first point in time $t \in \mathbb{N}$ for which $X^t \le a$, for some $a \le b$. Suppose that there exists $\delta > 0$ such that for all $t < T$,*

$$\mathbf{E}[X^t - X^{t+1} \mid X^t] \ge \delta X^t.$$

Then,

$$\Pr\left[T > \frac{t + \log(b/a)}{\delta}\right] \le e^{-t}.$$

2.5. Further tools

We will use the FKG inequality (Fortuin–Kasteleyn–Ginibre inequality), which is a standard tool in percolation theory, but less commonly used in the theory of EAs. We only give a special case of what is known as *Harris inequality*.

Theorem 9 *(FKG inequality [57, Section 2.2]). Let I be a finite set, and consider a product probability space $\Omega = \prod_{i\in I} \Omega_i$, where all Ω_i have binary sample space $\{0, 1\}$. A real-valued random variable X is called* increasing *if $X(\omega) \le X(\omega')$ holds for all elementary events ω, ω' in Ω with $\omega_i \le \omega'_i$ for all $i \in I$. It is called* decreasing *if $-X$ is increasing.*

1. *If two random variables X, Y are both increasing or are both decreasing, then*

$$\mathbf{E}[XY] \ge \mathbf{E}[X] \cdot \mathbf{E}[Y].$$

2. *If X is increasing and Y is decreasing, or vice versa, then*

$$\mathbf{E}[XY] \le \mathbf{E}[X] \cdot \mathbf{E}[Y].$$

We also say that X and Y are *positively correlated* and *negatively correlated* in the first and second case respectively. Note that the FKG inequality also applies to probabilities. E.g., if A, B are increasing events (which just means that their indicators $\mathbf{1}_A$ and $\mathbf{1}_B$ are increasing), then $\Pr[A] = \mathbf{E}[\mathbf{1}_A]$, $\Pr[B] = \mathbf{E}[\mathbf{1}_B]$, and $\mathbf{1}_A\mathbf{1}_B = \mathbf{1}_{A\cap B}$, hence $\Pr[A \cap B] \ge \Pr[A] \cdot \Pr[B]$.

To switch between differences and exponentials, we will frequently make use of the following estimates, taken from Lemma 1.4.2 – Corollary 1.4.6 in [58].

Lemma 10.

1. *For all* $r \geq 1$ *and* $0 \leq s \leq r$,

$$(1-1/r)^r \leq 1/e \leq (1-1/r)^{r-1}$$

and

$$(1-s/r)^r \leq e^{-s} \leq (1-s/r)^{r-s}.$$

2. *For all* $0 \leq x \leq 1$,

$$1-e^{-x} \geq x/2.$$

3. *For all* $0 \leq x \leq 1$ *and all* $y > 0$,

$$1-(1-x)^y \leq \tfrac{xy}{1+xy}.$$

Finally, we will also use standard Chernoff bounds.

Theorem 11 *(Chernoff Bound [58, Section 1.10]). Let* $X_1, \ldots, X_n$ *be independent random variables taking values in* $[0,1]$*. Let* $X = \sum_{i=1}^n X_i$ *and let* $1 \geq \delta \geq 0$*. Then*

$$\Pr[X \geq (1+\delta)E[X]] \leq \exp\left(-\frac{\delta^2 E[X]}{3}\right). \tag{1}$$

3. Technical definitions and results

As mentioned previously, our techniques resemble those of Hevia Fajardo and Sudholt [38,32]. The key is analysing a suitable potential function $g(x,\lambda) = \textsc{Zm}(x) + h(\lambda)$ which combines the distance $\textsc{Zm}(x)$ to the optimum (as defined in Section 2) with a penalty term for small λ. When this function has strong positive drift, we can establish that the optimum is reached fast; conversely, when g has (strong) negative drift, the optimisation takes super-polynomial time. In some cases, we use a similar penalty term $h(\lambda)$ (and thus potential function) as [32], in other cases very different ones. However, the potential always contains the number of zero-bits $Z^t = \textsc{Zm}(x^t)$ at time t as an additive term, so the drift of Z enters the drift of the potential in all cases. The goal of this section is to compute this drift in Lemma 13. Moreover, the definitions and results for showing this lemma are also used at other places in the paper.

Finally, we show that if the penalty term $h(\lambda)$ is 'reasonable', then the truncated change of g at each step $\min\{C, G^t - G^{t+1}\}$ has exponential tail bounds and is thus sub-Gaussian. This allows us to apply the concentration results from Section 2.4 to establish concentration from above of the optimisation time; see Remark 14 for more details.

3.1. Definition and properties of basic events

Because our analysis deals with an entire class of functions, we will not be able to precisely compute the probability of finding a fitness improvement. However, since we study (dynamic) monotone functions we can relate that probability to the probability of 1) having a child that flips no 1-bit of the parent, and 2) having a child that flips at least a 0-bit of the parent. Understanding those two events, which we respectively denote by A and B and formally define below, is the backbone of our approach and this subsection is devoted to their analysis.

Recall that x^t and λ^t are the search point and the offspring population size at time t respectively, and that $y^{t,j}$ denotes the j-th offspring at time t. For all times t we define

$$A^{t,j} := \{\mathrm{supp}(x^t) \subseteq \mathrm{supp}(y^{t,j})\} \quad \text{and} \quad A^t = \bigcup_j A^{t,j}.$$

In words, $A^{t,j}$ is the event that the j-th offspring at time t does not flip any one-bit of the parent, and A^t is the event that such a child exists at time t. We also define

$$B^{t,j} = \{\exists i : x_i^t < y_i^{t,j}\} \quad \text{and} \quad B^t = \bigcup_j B^{t,j},$$

respectively as the event that the j-th child *does* flip a *zero*-bit of the parent, and the event that such a child exists. We drop the superscript t when the time is clear from context, and just write x, y^j and λ for parent, offspring, and population size at time t, and A^j, A, B^j, B for the events defined above. We also observe that all the events $\{A^j\}_j \cup \{B^j\}_j$ are independent and in particular A and B are independent.

In the lemma below we estimate the probability of A^t and B^t in terms of Z^t and λ^t. We also provide a bound on the probability of *not* finding a fitness improvement.

Lemma 12. *For any mutation rate $c \le 1$, there exist constants $b_1, b_2, b_3 > 0$ depending only on c such that at all times t with $Z^t \ge 1$ we have*

$$\Pr[\bar{A}] \le e^{-b_1\lambda} \qquad \textit{and} \qquad e^{-b_2\lambda Z^t/n} \le \Pr[\bar{B}] \le e^{-b_3\lambda Z^t/n}.$$

Moreover, $\Pr[f(x^{t+1}) \le f(x^t)] \le e^{-\frac{1}{2}ce^{-c}\lfloor\lambda\rceil Z^t/n}$.

Proof. Let us start with the first inequality. The event A^j happens with probability $(1-c/n)^{n-Z^t} \ge e^{-c}$ by Lemma 10, so $\bar{A} = \bigcap_j \overline{A^j}$ has probability

$$\Pr[\bar{A}] \le \left(1 - e^{-c}\right)^{\lfloor\lambda\rceil} \le e^{-e^{-c}\lfloor\lambda\rceil},$$

again using Lemma 10. We conclude the first proof by observing that $\lfloor\lambda\rceil \ge \lambda/1.5$.

The event $\bar{B}$ happens if none of the $\lfloor\lambda\rceil$ offspring flips a zero-bit of the parent. This happens with probability

$$\Pr[\bar{B}] = (1-c/n)^{Z^t\lfloor\lambda\rceil}.$$

The upper bound is obtained as above: $(1-c/n) \le e^{-c/n}$ by Lemma 10 and $\lfloor\lambda\rceil \ge \lambda/1.5$. For the lower bound, we see that $(1-c/n) \ge e^{-2c/n}$ for $c/n \le 1/2$ by Lemma 10 and $\lfloor\lambda\rceil \le 2\lambda$ so that

$$\Pr[\bar{B}] \ge e^{-4cZ^t\lambda/n}.$$

For the last inequality, for every j the event $A^j \cap B^j$ implies $f(y^j) > f(x)$, since an offspring y^j is always an improvement if it is obtained by flipping a single zero-bit and no one-bit. Since all A^j and B^j are independent and $\Pr[B^j] = 1-(1-c/n)^{Z^t} \ge 1 - e^{-cZ^t/n} \ge \frac{1}{2}cZ^t/n$ by Lemma 10,

$$\begin{aligned}\Pr[f(x^{t+1}) \le f(x^t)] = \Pr[\forall j: f(y^{t,j}) \le f(x^t)] &\le \prod_{j=1}^{\lfloor\lambda\rceil}\left(1 - \Pr[A^j]\cdot\Pr[B^j]\right)\\ &\le \left(1 - e^{-c}\cdot\tfrac{1}{2}cZ^t/n\right)^{\lfloor\lambda\rceil} \le e^{-\frac{1}{2}ce^{-c}\lfloor\lambda\rceil Z^t/n}. \quad \square\end{aligned}$$

3.2. The drift of Z^t

With the definitions introduced in the previous subsection, we may now state and prove the key result of this section, that is, we compute the drift of Z in terms of $\Pr[A], \Pr[B]$, Z and λ.

Lemma 13. *Consider the SA-$(1,\lambda)$-EA with mutation rate $0 < c \le 1$. There exist constants $a_1, a_2, b > 0$ depending only on c such that at all times t with $Z^t > 0$ we have*

$$\mathbf{E}\left[Z^t - Z^{t+1} \mid x^t, \lambda^t\right] \ge \Pr[B]\cdot a_1\left(1 - c(1 - Z^t/n)\right) - a_2e^{-b\lambda^t}.$$

This also holds if we replace $Z^t - Z^{t+1}$ by $\min\{1, Z^t - Z^{t+1}\}$.

Remark 14. Theorems 6 and 7 which we use to prove concentration of hitting times require that the probability of having large jumps is small. This is not true in general: when we generate many children λ there is an increased probability of flipping many bits.

In order to still be able to prove concentration, we consider the situation in which the number of 0-bits may decrease by at most 1 at each step, i.e., this is why we cap the difference $Z^t - Z^{t+1}$ at 1. Even under this pessimistic assumption, we prove (in Sections 4 and 5) that the drift is positive and the optimum is reached fast.

The proof of this will be obtained using the following claims.

Claim 15. *At all times $t \ge 0$ with $Z^t > 0$ we have*

$$\mathbf{E}\left[Z^t - Z^{t+1} \mid A, B\right] \ge e^{-c}\left(1 - c\left(1 - \tfrac{Z^t}{n}\right)\right).$$

This also holds if we replace $Z^t - Z^{t+1}$ by $\min\{1, Z^t - Z^{t+1}\}$.

Claim 16. *At all times $t \geq 0$ with $Z^t > 0$ we have*

$$\mathbf{E}\left[Z^t - Z^{t+1} \mid \bar{A}\right] \geq -\frac{c}{1-e^{-c}}.$$

This also holds if we replace $Z^t - Z^{t+1}$ by $\min\{1, Z^t - Z^{t+1}\}$.

Proof of Claim 15. First, let us define K to be the index of the fittest offspring, i.e. $y^K = x^{t+1}$. A first step in proving the claim will be to show that

$$\Pr\left[y_i^K = 0 \mid A, B^K\right] \leq c/n, \tag{2}$$

for all $i \in \mathrm{supp}(x)$. Note that (2) would hold with equality if we replaced K by a fixed $j \in [\lfloor\lambda\rceil]$ and omitted conditioning on A, so the task is to show that conditioning on A and conditioning on the offspring being selected can only decrease the probability. To show this, we use a multiple exposure of the randomness: we let $u^1, \dots, u^{\lfloor\lambda\rceil}$ respectively be obtained from x by only revealing the flips (or non-flips) of the 0-bits of x in each of the $\lfloor\lambda\rceil$ children (where we abbreviate $x = x^t$ and $\lambda = \lambda^t$). The child y^j, $j \in [\lfloor\lambda\rceil]$ may then be obtained from u^j by revealing the rest of the bits, i.e. the flips of the 1-bits of x.

Consider an index i such that $x_i = 1$, and decompose

$$\begin{aligned}\Pr&\left[y_i^K = 0, A, B^K \mid (u^\ell)_{\ell\in[\lfloor\lambda\rceil]}\right] \\ &= \sum_{j=1}^{\lfloor\lambda\rceil} \Pr\left[K = j, y_i^j = 0, A, B^j \mid (u^\ell)_{\ell\in[\lfloor\lambda\rceil]}\right] \\ &= \sum_{j=1}^{\lfloor\lambda\rceil} \mathbf{1}_{B^j} \cdot \Pr\left[K = j, y_i^j = 0, A \mid (u^\ell)_{\ell\in[\lfloor\lambda\rceil]}\right].\end{aligned} \tag{3}$$

For a given j, we observe that if we additionally condition on the 1-bit flips in other children $(y^\ell)_{\ell\neq j}$, then $\mathbf{1}_{K=j,A}$ is a decreasing function of the 1-bit flips in the j-th child, while $\mathbf{1}_{y_i^j=0}$ is an increasing function of those flips. The FKG inequality, Theorem 9, thus gives

$$\begin{aligned}\Pr&\left[K = j, y_i^j = 0, A \mid (u^\ell)_{\ell\in[\lfloor\lambda\rceil]}, (y^\ell)_{\ell\in[\lfloor\lambda\rceil],\ell\neq j}\right] \\ &\leq \Pr\left[y_i^j = 0 \mid (u^\ell), (y^\ell)_{\ell\neq j}\right] \cdot \Pr\left[K = j, A \mid (u^\ell), (y^\ell)_{\ell\neq j}\right] \\ &= \frac{c}{n} \cdot \Pr\left[K = j, A \mid (u^\ell), (y^\ell)_{\ell\neq j}\right],\end{aligned}$$

where the last line simply comes from the fact that the i-th bit flip in y^j is independent of what happens in the other children and in the 0-bit flips of y^j. Using the law of total probability over $(y^\ell)_{\ell\neq j}$ gives

$$\Pr\left[K = j, y_i^j = 0, A \mid (u^\ell)\right] \leq \frac{c}{n} \cdot \Pr\left[K = j, A \mid (u^\ell)\right],$$

and plugging this into (3) gives

$$\Pr\left[y_i^K = 0, A, B^K \mid (u^\ell)_{\ell\in[\lfloor\lambda\rceil]}\right] \leq \sum_{j=1}^{\lfloor\lambda\rceil} \mathbf{1}_{B^j} \frac{c}{n} \Pr\left[K = j, A \mid (u^\ell)_{\ell\in[\lfloor\lambda\rceil]}\right]. \tag{4}$$

A similar decomposition gives

$$\Pr[A, B^K \mid (u^\ell)_{\ell\in[\lfloor\lambda\rceil]}] = \sum_{j=1}^{\lfloor\lambda\rceil} \mathbf{1}_{B^j} \cdot \Pr[K = j, A \mid (u^\ell)_{\ell\in[\lfloor\lambda\rceil]}], \tag{5}$$

and combining (4) and (5) gives $\Pr[y_i^K = 0 \mid A, B^K, (u^\ell)_{\ell\in[\lfloor\lambda\rceil]}] = (4)/(5) \leq c/n$. To obtain (2) it suffices to apply the law of total probability over $(u^\ell)_\ell$. We can now compute the drift conditioned on A, B^K: since B^K implies that y^K turns (at least) one 0-bit into a 1-bit, we obtain

$$\begin{aligned}\mathbf{E}[Z^t - Z^{t+1} \mid A, B^K] &\geq 1 - \sum_{i\in\mathrm{supp}(x)} \Pr[y_i^K = 0 \mid A, B^K] \\ &\overset{(2)}{\geq} 1 - (n - Z^t) \cdot \frac{c}{n} = 1 - c\left(1 - Z^t/n\right).\end{aligned} \tag{6}$$

Note that the first step in (6) remains correct if we replace $Z^t - Z^{t+1}$ with $\min\{1, Z^t - Z^{t+1}\}$, and this difference does not play a role in any other parts of the proof. To continue the proof, we decompose

$$\begin{aligned}\mathbf{E}[Z^t - Z^{t+1} \mid A, B] = \Pr\left[B^K \mid A, B\right] \cdot \mathbf{E}\left[Z^t - Z^{t+1} \mid A, B, B^K\right] \\ + \Pr\left[\overline{B^K} \mid A, B\right] \cdot \mathbf{E}\left[Z^t - Z^{t+1} \mid A, B, \overline{B^K}\right].\end{aligned}$$

Since $B \cap B^K = B^K$, the first conditional expectation of the RHS is exactly $\mathbf{E}[Z^t - Z^{t+1} \mid A, B^K] \geq 1 - c(1 - Z^t/n)$ by (6). To conclude the proof, it thus suffices to show

$$\Pr[B^K \mid A, B] \geq e^{-c}, \tag{7}$$

and

$$\mathbf{E}\left[Z^t - Z^{t+1} \mid A, B, \overline{B^K}\right] = 0. \tag{8}$$

We start by proving (8): we will argue that if both A and $\overline{B^K}$ hold, then $x^{t+1} = x^t$. Indeed, $\overline{B^K}$ implies that $f(x^{t+1}) \leq f(x^t)$ since no bit is flipped from 0 to 1. Additionally, the equality holds if and only if $x^{t+1} = x^t$, since flipping any 1-bit to 0 would decrease f by strict monotonicity. On the other hand, one easily observes that A implies $f(x^t) \leq f(x^{t+1})$. Consequently, $A \cap \overline{B^K}$ implies that $x^t = x^{t+1}$, which proves (8).

Finally, we prove (7). Once $(u^\ell)_{\ell \in [\lfloor\lambda\rceil]}$ is revealed, we can define J as the set of those indices j which maximise $f(u^j)$. We observe that if B and $\cup_{j \in J} A^j$ hold, then B^K also does. Indeed, if we let $J' \subseteq J$ be the set of indices for which A^j holds, and if $J' \neq \emptyset$, then the set of children which maximise $f(y^j)$ is exactly J'. Hence we have

$$\Pr\left[B^K \mid A, B\right] \geq \Pr\left[\cup_{j \in J} A^j \mid A, B\right] \geq (1 - c/n)^{n - Z^t} \geq e^{-c}.$$

The second inequality is simply obtained by noting that, under the assumption that B holds and since J is not empty, the probability of $\cup_{j \in J} A^j$ is at least that of A^j for a single (arbitrary) j in J. The event A^j has probability $(1 - c/n)^{n - Z^t}$, is independent of B and positively correlated with A. The last inequality follows from Lemma 10 since $Z^t \geq 1$. □

Proof of Claim 16. Recall that $\bar{A}$ is the event that every offspring flips at least one one-bit. Let K be the index of the fittest child and let N^j be the number of one-bits flipped in the j-th offspring, we want to show that $\mathbf{E}[N^K \mid \bar{A}] \leq \mathbf{E}[N^j \mid \bar{A}]$ holds for all $j \in [\lfloor\lambda\rceil]$, i.e. the fittest offspring does not flip more one-bits than an arbitrary offspring in expectation.

Note that conditioning on $\bar{A}$ leads to dependent bit flips within each individual offspring, but once we know that a specific one-bit is flipped, the remaining one-bit flips are independent. Therefore, we can couple the one-bits flips given $\bar{A}$ with the following procedure. Assume there are m one-bits in x, we first sample the position of the *first* (= left-most) one-bit l to be flipped. Afterwards, we still flip each bit *to the right of* l independently with probability c/n. This gives the usual distribution of one-bit flips, conditioned on $\bar{A}$. To make this formal, we sample $l \in [m]$ with probability $p_l(m) = (1 - (1 - c/n)^m)^{-1}(c/n)(1 - c/n)^{l-1}$ and flip the l-th one-bit. It is easy to verify that $\sum_{l=1}^m p_l(m) = 1$ since p_l is a geometric sequence. Then for each $l' \in [m] \setminus [l]$, we flip the l'-th one-bit independently with probability c/n. The probability that a specific one-bit is flipped given $\bar{A}$ is $(c/n)(1 - (1 - c/n)^m)^{-1}$. By our procedure, this probability is

$$\begin{aligned}\Pr[\text{the } l\text{-th one-bit is flipped}] &= p_l(m) + \sum_{i=1}^{l-1} p_i(m) \frac{c}{n} \\ &= p_l(m) + \frac{p_1(m)(1 - (1 - c/n)^{l-1})}{1 - (1 - c/n)} \frac{c}{n} \\ &= \frac{c}{n} \frac{1}{1 - (1 - c/n)^m},\end{aligned}$$

which is exactly the desired conditional probability.

Therefore, we can get rid of $\bar{A}$ as follows. Let $N^j_{a:b}$ be the random number of bit flips when we flip the a-th to the b-th one-bit independently with probability c/n for offspring j and l^j be the index l sampled for offspring j. Then

$$\begin{aligned}\mathbf{E}[N^K \mid \bar{A}] &= \sum_{j=1}^{\lfloor\lambda\rceil} \mathbf{E}[N^j \cdot \mathbf{1}_{K=j} \mid \bar{A}] \\ &= \sum_{j=1}^{\lfloor\lambda\rceil} \sum_{i=1}^{m} \mathbf{E}[(1 + N^j_{i+1:m}) \cdot \mathbf{1}_{K=j} \mid l^j = i] \Pr[l^j = i]\end{aligned}$$

$$= \sum_{j=1}^{\lfloor\lambda\rceil} \sum_{i=1}^{m} \mathbf{E}[(1 + N^j_{i+1:m}) \cdot \mathbf{1}_{K=j}] \cdot p_i(m).$$

Now we may use the FKG inequality to show

$$\mathbf{E}[N^j_{i+1:m} \cdot \mathbf{1}_{K=j}] \le \mathbf{E}[N^j_{i+1:m}] \cdot \mathbf{E}[\mathbf{1}_{K=j}]. \tag{9}$$

The proof is similar to that used in the previous claim: one conditions on (u^ℓ), $(y^\ell)_{\ell\neq j}$ in order to have a product space on which $N^j_{i+1:m}$ is increasing and $\mathbf{1}_{K=j}$ decreasing. One then applies the FKG inequality. Observing that $N^j_{i+1:m}$ is independent of $(u^\ell), (y^\ell)_{\ell\neq j}$ and using the law of total probability over $(u^\ell), (y^\ell)_{\ell\neq j}$ gives (9). Continuing the previous derivation,

$$\begin{aligned}\mathbf{E}[N^K \mid \bar{A}] &\le \sum_{j=1}^{\lfloor\lambda\rceil} \sum_{i=1}^{m} \mathbf{E}[1 + N^j_{i+1:m}]\mathbf{E}[\mathbf{1}_{K=j}] \cdot p_i(m) \\ &= \sum_{j=1}^{\lfloor\lambda\rceil} \mathbf{E}[\mathbf{1}_{K=j}] \sum_{i=1}^{m} \mathbf{E}[1 + N^j_{i+1:m}] \cdot p_i(m) \\ &= \sum_{i=1}^{m} \mathbf{E}[1 + N^j_{i+1:m} \mid l^j = i] \Pr[l^j = i] = \mathbf{E}[N^j \mid \bar{A}],\end{aligned}$$

where we use the fact that $\sum_{j=1}^{\lfloor\lambda\rceil} \mathbf{1}_{K=j} = 1$ and that $\mathbf{E}[(1 + N^j_{i+1:m})]$ is invariant with respect to the index j.

The term $\mathbf{E}[N^j \mid \bar{A}]$ is maximized for $m = n$, hence

$$\begin{aligned}\mathbf{E}[Z^t - Z^{t+1} \mid \bar{A}] &\ge -\mathbf{E}[N^K \mid \bar{A}] \ge -\mathbf{E}[N^j \mid \bar{A}] \\ &\ge -\frac{c}{n}\frac{1}{1-(1-c/n)^n} n \ge -\frac{c}{1-e^{-c}},\end{aligned}$$

which proves Claim 16. □

We now combine the two claims above to obtain Lemma 13.

Proof of Lemma 13. The drift of $Z^t = \text{ZM}(x^t)$ may be decomposed as follows,

$$\begin{aligned}\mathbf{E}[Z^t - Z^{t+1} \mid x, \lambda] = \Pr[A, B] \cdot \mathbf{E}[Z^t - Z^{t+1} \mid A, B] \\ + \Pr[A, \bar{B}] \cdot \mathbf{E}[Z^t - Z^{t+1} \mid A, \bar{B}] \\ + \Pr[\bar{A}] \cdot \mathbf{E}[Z^t - Z^{t+1} \mid \bar{A}],\end{aligned} \tag{10}$$

where we omitted the conditioning on x, λ on the right-hand side for brevity. As observed above, A, B are independent so we get $\Pr[A, B] = \Pr[A]\Pr[B]$. Also, we observe that the second conditional expectation in (10) must be 0: if $\bar{B}$ holds then no child is a strict improvement of the parent, but A guarantees that some children are at least as good. Hence, if $A, \bar{B}$ hold, we must have $x^t = x^{t+1}$. Combining those remarks with the bounds of Claims 15 and 16 gives

$$\mathbf{E}[Z^t - Z^{t+1} \mid x, \lambda] \ge \Pr[A]\Pr[B] \cdot \frac{1 - c(1 - Z^t/n)}{e^c} - \Pr[\bar{A}] \cdot \frac{c}{1 - e^{-c}}.$$

Lemma 12 guarantees that $\Pr[A]$ is at least a positive constant C when $\lambda \ge 1$ and that $\Pr[\bar{A}] \le e^{-b_1\lambda}$. Choosing $a_1 = Ce^{-c}$, $a_2 = c/(1 - e^{-c})$ and $b = b_1$ gives the result. □

3.3. Improvements are sub-Gaussian

In Sections 4.2 and 5 we will prove that the number of time steps needed to optimise a function is tightly concentrated. We provide the following result, based on Theorems 6 and 7, which allows us to relate strong positive drift and concentration of hitting times.

Lemma 17. *Consider the SA-$(1, \lambda)$-EA with parameters $0 < c < 1 < F$ and with an arbitrary success rate $s > 0$, and let $\varepsilon, C_1, C_2 > 0$ be constants. Then there exist $\gamma, \delta > 0$ which only depend on $c, F, s, \varepsilon, C_1, C_2$ such that the following holds.*

Let h be a decreasing function, $g(x, \lambda) = \text{ZM}(x) + h(\lambda)$ and define

$$\Gamma^t := \min\{1 + C_1, g(x^t, \lambda^t) - g(x^{t+1}, \lambda^{t+1})\}.$$

Assume the following two properties are satisfied

(i) $\mathbf{E}\left[\Gamma^t \mid x^t, \lambda^t\right] \geq \varepsilon$;
(ii) $h(\lambda) - h(\lambda F) \leq C_2$ *for all* $\lambda \in [1, \infty)$.

Then $\left(\varepsilon - \Gamma^t\right)_{t \geq 0}$ *is* (γ, δ)*-sub-Gaussian.*

Proof. We will use Theorem 6 to prove that $(\varepsilon - \Gamma^t)$ is sub-Gaussian. To be able to apply this theorem, we must prove that $\mathbf{E}\left[\varepsilon - \Gamma^t\right] \leq 0$ and that there exist $0 < \alpha$ and $0 < \beta < 1$ such that $\Pr[|\varepsilon - \Gamma^t| > w] \leq \alpha/(1+\beta)^w$ for all $w \geq 0$. The first immediately holds by i, so we focus on the second.

Let $y^1, \ldots, y^{\lfloor \lambda \rceil}$ be the children at step t and let $K = \arg\max_j f(y^j)$ the index of the fittest child. We also let $N^1, \ldots, N^{\lfloor \lambda \rceil}$ be the number of 1-bit flips in $y^1, \ldots, y^{\lfloor \lambda \rceil}$ and we define $N = N^K$. Clearly $N \geq Z^{t+1} - Z^t$.

Let $w > C_2 + \varepsilon$ and $w' = \lceil w - \varepsilon - C_2 \rceil$. Since the value of h may only decrease by C_2 at each step by ii, to have $\left(\varepsilon - \Gamma^t\right) \geq w$ it must be that $Z^{t+1} - Z^t \geq w - \varepsilon - C_2$, and in particular we must have $N \geq w'$. We compute

$$\begin{aligned}
\Pr[N \geq w'] &\leq \sum_j \Pr[N^j \geq w', K = j] \leq \sum_j \Pr[N^j \geq w'] \Pr[K = j] \\
&= \sum_j \binom{n - Z^t}{w'} \left(\frac{c}{n}\right)^{w'} \left(1 - \frac{c}{n}\right)^{n - Z^t - w'} \Pr[K = j] \\
&\leq 1/w'! \leq 2^{-w'} \leq 2^{\varepsilon + C_2 - w}.
\end{aligned}$$

Above, the second inequality is obtained using the FKG inequality in the same fashion as in the proof of Claim 15.

The above implies that for all $w > C_2 + \varepsilon$, the probability of having $\varepsilon - \Gamma^t \geq w$ is bounded by $\alpha/(1+\beta)^w$ for $\alpha = 2^{\varepsilon + C_2}$ and $\beta = 1$. Up to possibly increasing α to be a large constant, the same relation holds for $w \in [0, C_2 + \varepsilon]$. Since Γ^t is upper bounded by the constant $1 + C_1$, the quantity of interest $\varepsilon - \Gamma^t$ is lower-bounded by $\varepsilon - 1 - C_1$ so we can also trivially achieve

$$\Pr[\Gamma^t - \varepsilon \geq w \mid x^t, \lambda^t] \leq \frac{\alpha}{(1+\beta)^w},$$

by possibly increasing α again.

Theorem 6 may now be applied: $\varepsilon - \Gamma^t$ is (γ, δ)-sub-Gaussian for some γ, δ depending on ε, C_1, C_2. □

4. Monotone functions are efficiently optimized for small success rates

In this section we analyse the SA-$(1, \lambda)$-EA when the success rate s is small, and the mutation rate is c/n for a constant $0 < c \leq 1$. We show that if s is sufficiently small then for *any* strictly monotone fitness function, the optimum is found efficiently both in the number of generations and evaluations. We distinguish between the cases $c < 1$ and $c = 1$.

4.1. Bound on the number of generations

In this subsection, we study the number of generations required to reach the optimum and show that for $c \leq 1$, the SA-$(1, \lambda)$-EA finds the optimum efficiently. We start with the case $c < 1$.

Theorem 18. *Let* $0 < c < 1 < F$ *be constants. Then there exist* $C, s_0 > 0$ *such that for all* $0 < s \leq s_0$ *and for every dynamic monotone function the expected number of generations of the SA-*$(1, \lambda)$*-EA with success rate s, update strength F and mutation probability* c/n *is at most* Cn.

For $c = 1$, we additionally need to assume that the update strength F is bounded from above by a suitable constant $F_0 > 1$. As we will show experimentally, the update strength can have a notable impact on performance, but it remains open whether this effect vanishes for sufficiently small s.

Theorem 19. *There exist constants* $F_0 > 1$, $s_0 > 0$ *and* $C > 0$ *such that for all* $1 < F < F_0$, *all* $0 < s \leq s_0$, *and for all dynamic monotone functions the expected number of generations of the SA-*$(1, \lambda)$*-EA with success rate s, update strength F and mutation probability* $1/n$ *is at most* $Cn \log n$.

Our approach will be essentially the same for both theorems and will follow the ideas of Hevia Fajardo and Sudholt [32, 38]. We prove them in the following two subsections.

4.2. Expected number of generations when $c < 1$

We will prove Theorem 18 in this section; for the remainder of this section, we assume that $0 < c < 1 < F$ and the dynamic monotone function f are all given and we will show the existence of a desired s_0 independent of f. Recall that x^t is the search point at time t, its children are $y^{t,1}, \dots, y^{t,\lfloor\lambda_t\rceil}$, and λ^t is the value of λ at time t. In particular, the latter does not need to be an integer, and the actual number of offspring at time t is the closest integer $\lfloor\lambda^t\rceil$. Whenever the time t is clear from the context, we will remove it from the superscript.

We show that for an appropriate function g, the drift $\mathbf{E}[g(x^t,\lambda^t) - g(x^{t+1},\lambda^{t+1})]$ is positive. Our choice of g will guarantee that $g(x,\cdot) = 0$ implies $x = (1,\cdots,1)$, and the Additive Drift Theorem 2 will allow us to bound the time until this happens. For this section, we use the following $g = g_1$.

Definition 20 *(Potential function for positive result).* Let

$$h_1(\lambda) := K_1 \cdot \max\{0, \log_F \frac{\lambda_{\max}}{\lambda}\},$$

where K_1 is a constant to be chosen later, and $\lambda_{\max} := F^{1/s}n$. Then for all $x \in \{0,1\}^n$ and $\lambda \in [1,\infty)$ we define

$$g_1(x,\lambda) := \textsc{Zm}(x) + h_1(\lambda).$$

Recall that $Z^t = \textsc{Zm}(x^t)$, and denote $H_1^t := h_1(\lambda^t)$ and $G_1^t := g_1(x^t,\lambda^t)$.

Our first lemma states that $g_1(x,\lambda)$ does not deviate much from $\textsc{Zm}(x)$ for all x,λ, and that it suffices to show that g_1 reaches 0, since then the optimum is found.

Lemma 21 *("Sandwich" inequalities relating the potential function and the fitness). For all $x \in \{0,1\}^n$ and $\lambda \in [1,\infty)$ we have*

$$g_1(x,\lambda) - K_1 \log_F \lambda_{\max} \le \textsc{Zm}(x) \le g_1(x,\lambda).$$

In particular $g_1(x,\lambda) = 0$ implies that $\textsc{Zm}(x) = 0$.

Proof. The lemma follows trivially from the fact that

$$h_1(\lambda) = K_1 \max\left\{0, \log_F(\lambda_{\max}/\lambda)\right\} \in [0, K_1 \log \lambda_{\max}]. \quad \square$$

We will now compute the drift of G_1^t. The drift of Z^t was already computed in the previous section, so it suffices to compute that of H_1^t.

Claim 22. *At all times $t \ge 0$ we have*

$$\mathbf{E}\left[H_1^t - H_1^{t+1} \mid x^t, \lambda^t\right] \ge -K_1 \cdot \Pr[B] + \frac{K_1}{s} \cdot \Pr\left[\bar{B}\right] \cdot \mathbf{1}_{\lambda_t < n}.$$

Proof of Claim 22. We first give a general bound that we will use for the case that the fitness increases. We have $\lambda^{t+1} \ge \lambda^t/F$ and thus $\log_F(\lambda_{\max}/\lambda^{t+1}) \le 1 + \log_F(\lambda_{\max}/\lambda^t)$ and $H_1^{t+1} \le K_1 + H_1^t$. In particular,

$$\mathbf{E}\left[H_1^t - H_1^{t+1} \mid x^t, \lambda^t, f(x^{t+1}) > f(x^t)\right] \ge -K_1.$$

For $f(x^{t+1}) \le f(x^t)$, we have $\lambda^{t+1} \le \lambda^t$, and thus $H_1^t - H_1^{t+1} \ge 0$. If additionally $\lambda^t < \lambda_{\max} F^{-1/s} = n$, then $\lambda^t < F^{1/s}\lambda^t = \lambda^{t+1} \le \lambda_{\max}$, and hence $H_1^t - H_1^{t+1} = K_1/s$. Summarizing,

$$\mathbf{E}\left[H_1^t - H_1^{t+1} \mid x^t, \lambda^t, f(x^{t+1}) \le f(x^t)\right] \ge \begin{cases} K_1/s & \text{if } 1 \le \lambda^t < n; \\ 0 & \text{if } \lambda^t \ge n. \end{cases}$$

This gives

$$\mathbf{E}\left[H_1^t - H_1^{t+1}\right] \ge -K_1 \Pr[f(x^{t+1}) > f(x^t)] + \frac{K_1}{s} \Pr[f(x^{t+1}) \le f(x^t)] \mathbf{1}_{\lambda^t < n}.$$

Observe that $f(x^{t+1}) > f(x^t)$ only if B holds: indeed for the fitness to increase at time t, at least one child needs to mutate a 0-bit of x^t into a 1-bit. This gives $\Pr[f(x^{t+1}) > f(x^t)] \le \Pr[B]$, and Claim 22 follows. $\square$

Claims 15, 16 and 22 may now be combined to obtain the following drift of G_1^t. We again drop the index t from x^t and λ^t.

Corollary 23. *There exists a constant $s_0 > 0$ such that for all $0 < s \leq s_0$ the following holds. There is a constant δ and a choice of K_1 such that for all t with $Z^t > 0$,*

$$\mathbf{E}\left[G_1^t - G_1^{t+1} \mid x, \lambda\right] \geq \delta.$$

This also holds if $G^t - G^{t+1}$ is replaced by $\min\{1 + K_1/s, G^t - G^{t+1}\}$.

Proof. Combining Lemma 13 and Claim 22, one obtains that the drift of G_1 is at least

$$\mathbf{E}\left[G_1^t - G_1^{t+1} \mid x, \lambda\right] \geq \Pr[B]\,(\alpha_1 - K_1) + \mathbf{1}_{\lambda<n} \Pr[\bar{B}]K_1/s - \alpha_2 e^{-\beta\lambda}, \tag{11}$$

for some constants $\alpha_1, \alpha_2, \beta > 0$.

We choose $K_1 = \alpha_1/2$ so that the drift for any t with $Z^t > 0$ is at least

$$\mathbf{E}\left[G_1^t - G_1^{t+1} \mid x, \lambda\right] \geq \Pr[B]K_1 + \mathbf{1}_{\lambda<n} \Pr[\bar{B}]K_1/s - \alpha_2 e^{-\beta\lambda}.$$

- If $\lambda < n$: then as we want s small enough, we may assume that $s < 1$. In this setting the drift is lower bounded by

$$\begin{aligned}\mathbf{E}[G_1^t - G_1^{t+1} \mid x, \lambda] &\geq \Pr[B]K_1 + \Pr[\bar{B}]K_1/s - \alpha_2 e^{-\beta\lambda} \\ &= K_1 + \Pr[\bar{B}]K_1(1-s)/s - \alpha_2 e^{-\beta\lambda}.\end{aligned}$$

Note that there is $\lambda_0 = \lambda_0(\alpha_2, \beta, K_1)$ such that for $\lambda \geq \lambda_0$ the last term can be bounded as $\alpha_2 e^{-\beta\lambda} \leq K_1/2$, in which case the drift is at least $K_1/2$. For the remaining case, recall Lemma 12 guarantees that $\Pr[\bar{B}] \geq e^{-b_2\lambda}$ for some constant $b_2 > 0$ depending only on c. Hence, for a choice of s small enough we can achieve $\Pr[\bar{B}]K_1(1-s)/s \geq \alpha_2 e^{-\beta\lambda}$, so the drift stays above $K_1 > K_1/2$.

- If $\lambda \geq n$: the drift is

$$\mathbf{E}\left[G_1^t - G_1^{t+1} \mid x, \lambda\right] \geq \Pr[B]K_1 - \alpha_2 e^{-\beta n}.$$

The first term is at least $\Pr[B]K_1 \geq (1 - (1 - c/n)^{\lfloor\lambda\rceil Z^t})K_1 \geq (1 - e^{-c})K_1$ while the second is $e^{-\Omega(n)} = o(1)$; this implies that the drift is at least $\frac{(1-e^{-c})}{2}K_1$ for sufficiently large n.

To see why the statement also holds for $\min\{1 + K_1/s\,,\ G^t - G^{t+1}\}$, we recall that $G^t = Z^t + H_1^t$, that Lemma 13 holds for $\min\{1, Z^t - Z^{t+1}\}$ and that H_1 may increase by at most K_1/s in each step. This implies that the first formula (11) also holds if we replace $G^t - G^{t+1}$ by $\min\{1 + K_1/s\,,\ G^t - G^{t+1}\}$ and all following arguments are unchanged. □

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

Proof of Theorem 18. Corollary 23 guarantees that for s sufficiently small there is $\delta > 0$ such that the drift of G_1 is at least $\mathbf{E}[G_1^t - G_1^{t+1} \mid x, \lambda] \geq \delta$ whenever $Z^t > 0$. Let T be the first point in time when either $G_1^T = 0$ or $Z^T = 0$. Then the drift bound for G_1 applies to all $t < T$, and by Theorem 2 we have $\mathbf{E}[T] \leq G_1^0/\delta \leq (n + K_1 \log(\lambda_{\max}))/\delta = O(n)$. By Lemma 21, $G_1^T = 0$ implies $Z^T = 0$, so in particular at time T we have $x^T = (1, \dots, 1)$ and Theorem 18 is proved. □

4.3. Expected number of generations when $c = 1$

We will now prove Theorem 19; that is, we will show that the self-adjusting EA is also efficient when the mutation rate is $1/n$. The reason we need to treat this case differently from the previous one is because of the expected number of bits gained when increasing the fitness. If we set $c = 1$, the drift obtained in Claim 15 is no longer constant but proportional to Z^t/n. In particular, in the last stages of the exploration, the drift is a lot smaller and this results in a looser bound for the number of generations. Still, the proof is similar to the one for $c < 1$, but we need to choose a different potential function.

Definition 24 *(Potential function for $c = 1$).* Let

$$h_2(\lambda) := K_2 \cdot \max\{0, \tfrac{1}{\lambda} - \tfrac{1}{\lambda_{\max}}\},$$

where K_2 is a constant to be chosen later, and $\lambda_{\max} = F^{1/s}n$. Then for $x \in \{0,1\}^n$ and $\lambda \in [1, \infty)$ we define

$$g_2(x, \lambda) := \textsc{Zm}(x) + h_2(\lambda),$$

and we set $H_2^t := h_2(x^t, \lambda^t)$ and $G_2^t := g_2(x^t, \lambda^t)$, and as before $Z^t := \textsc{Zm}(x^t)$.

As before, we have a lemma stating that the deviation between $\textsc{Zm}(x)$ and $g_2(x,\lambda)$ is small.

Lemma 25 *("Sandwich" inequalities). For all x, λ we have*

$$g_2(x,\lambda) - K_2\left(1 - \frac{1}{\lambda_{\max}}\right) \leq \textsc{Zm}(x) \leq g_2(x,\lambda).$$

In particular $g_2(x,\lambda) = 0$ implies that $\textsc{Zm}(X) = 0$.

Proof. Similar to the proof of Lemma 21, the proof follows from the fact that

$$h_2(\lambda) = K_2 \max\left\{0, \frac{1}{\lambda} - \frac{1}{\lambda_{\max}}\right\} \in \left[0, K_2\left(1 - \frac{1}{\lambda_{\max}}\right)\right],$$

when $\lambda \in [1,\infty)$. □

The drift of Z is known from Lemma 13, so to compute the drift of G it suffices to compute that of H_2. As before, we abbreviate $x = x^t$ and $\lambda = \lambda^t$ where the index is clear from the context.

Claim 26. *At all times $t \geq 0$ with $Z^t > 0$ we have*

$$\mathbf{E}\left[H_2^t - H_2^{t+1} \mid x, \lambda\right] \geq -\frac{K_2}{\lambda}(F-1)\Pr[B] + \frac{K_2}{\lambda}\left(1 - F^{-1/s}\right)\Pr\left[\bar{B}\right]\mathbf{1}_{\lambda<n}.$$

Proof of Claim 26. Similar to the proof of Claim 22, we analyse the drift of H_2^t. We first give a general bound that we will use for the case that the fitness increases. We have $\lambda^{t+1} \geq \lambda^t/F$ and thus $(\lambda^t)^{-1} - (\lambda^{t+1})^{-1} \geq (1-F)/\lambda^t$. Hence, $H_2^t - H_2^{t+1} \geq K_2(1-F)/\lambda^t$ for all t. In particular,

$$\mathbf{E}[H_2^t - H_2^{t+1} \mid x^t, \lambda^t, f(x^{t+1}) > f(x^t)] \geq K_2\frac{(1-F)}{\lambda^t}.$$

For $f(x^{t+1}) \leq f(x^t)$, we have $\lambda^{t+1} \geq \lambda^t$, and thus $H_2^t - H_2^{t+1} \geq 0$. If additionally $\lambda^t < \lambda_{\max}F^{-1/s} = n$, then $\lambda^t < F^{1/s}\lambda^t = \lambda^{t+1} \leq \lambda_{\max}$, and hence $H_2^t - H_2^{t+1} = K_2(1-F^{-1/s})/\lambda^t$. Summarizing,

$$\mathbf{E}[H_2^t - H_2^{t+1} \mid x^t, \lambda^t, f(x^{t+1}) \leq f(x^t)] \geq \begin{cases} K_2(1-F^{-1/s})/\lambda^t & \text{if } \lambda_t < n; \\ 0 & \text{if } n \leq \lambda_t. \end{cases}$$

This gives

$$\begin{aligned}\mathbf{E}[H_2^t - H_2^{t+1}] \geq{}& K_2(1-F)/\lambda^t \cdot \Pr[f(x^{t+1}) > f(x^t)] \\ &+ K_2(1-F^{-1/s})/\lambda^t \cdot \Pr[f(x^{t+1}) \leq f(x^t)] \cdot \mathbf{1}_{\lambda^t<n},\end{aligned}$$

where the conditioning on x, λ is implicit. Recall that B is the event that at least one of the offspring flips a 0-bit of X_t, which is a necessary condition for $f(x^{t+1}) > f(x^t)$. Also, we have $1-F<0$ and $1-F^{-1/s}>0$ due to $F>1$, so replacing $\Pr[f(x^{t+1}) > f(x^t)]$ by its upper bound $\Pr[B]$ and replacing $\Pr[f(x^{t+1}) \leq f(x^t)]$ by its lower bound $\Pr[\bar{B}]$, we conclude the proof. □

We can bound the drift of G_2^t from below as follows.

Corollary 27. *There exist constants $0 < s_0$ and $1 < F_0$ such that the following holds. For all $0 < s \leq s_0$ and all $1 < F \leq F_0$ there exists a choice of K_2 and a constant $\delta > 0$ such that for all times t with $Z^t > 0$,*

$$\mathbf{E}[G_2^t - G_2^{t+1} \mid x, \lambda] \geq \delta G_2^t/n.$$

Proof. We will first show

$$\mathbf{E}[G_2^t - G_2^{t+1} \mid x, \lambda] \geq \delta' Z^t/n \tag{12}$$

for some $\delta' > 0$. Combining Lemma 13 together with Claim 26 we obtain that the drift of G_2 is at least

$$\begin{aligned}\mathbf{E}[G_2^t - G_2^{t+1} \mid x, \lambda] \geq{}& \Pr[B]\Big(\alpha_1\frac{Z^t}{n} - K_2(F-1)\frac{1}{\lambda}\Big) - \alpha_2 e^{-\beta\lambda} \\ &+ \Pr[\bar{B}]K_2(1-F^{-1/s}) \cdot \frac{1}{\lambda}\mathbf{1}_{\lambda\leq n},\end{aligned} \tag{13}$$

for some constants $\alpha_1, \alpha_2, \beta > 0$.

We will argue that if $F > 1$ and $s > 0$ are both small enough and if K_2 is chosen appropriately, then the drift of G_2^t is of order Z^t/n. We will choose F, s later but we may already choose $K_2 = \alpha_1/(2(F-1))$ so that the drift is at least

$$\begin{aligned}\mathbf{E}[G_2^t - G_2^{t+1} \mid x, \lambda] \geq \Pr[B]\alpha_1 \left(\frac{Z^t}{n} - \frac{1}{2\lambda}\right) - \alpha_2 e^{-\beta\lambda} \\ + \Pr[\bar{B}]\alpha_1 \frac{1 - F^{-1/s}}{2(F-1)} \cdot \frac{1}{\lambda}\mathbf{1}_{\lambda \leq n}.\end{aligned}$$

Our proof is based on a case distinction. Let b_3 be the constant of Lemma 12, $\gamma := 1 - e^{-b_3}$ and let $\tilde{\lambda} > 0$ be such that $\gamma\alpha_1/(4\lambda) - \alpha_2 e^{-\beta\lambda} \geq 0$ holds for all $\lambda \geq \tilde{\lambda}$.

- If $\lambda \leq \max\{\tilde{\lambda}, n/Z^t\}$: then by ignoring the first positive contribution in (13), the drift is at least

$$\mathbf{E}[G_2^t - G_2^{t+1} \mid x, \lambda] \geq -\Pr[B]\alpha_1 \frac{1}{2\lambda} - \alpha_2 e^{-\beta\lambda} + \Pr[\bar{B}]\alpha_1 \frac{1 - F^{-1/s}}{2(F-1)} \cdot \frac{1}{\lambda}.$$

Splitting the positive contribution into three equal parts gives

$$\begin{aligned}\mathbf{E}[G_2^t - G_2^{t+1} \mid x, \lambda] \geq \Pr[\bar{B}]\frac{1}{3}\alpha_1 \frac{1 - F^{-1/s}}{2(F-1)} \cdot \frac{1}{\lambda} \\ + \left(\Pr[\bar{B}]\frac{1}{3} \cdot \frac{1 - F^{-1/s}}{(F-1)} - \Pr[B]\right)\frac{\alpha_1}{2\lambda} \\ + \Pr[\bar{B}]\frac{1}{3}\alpha_1 \frac{1 - F^{-1/s}}{2(F-1)} \cdot \frac{1}{\lambda} - \alpha_2 e^{-\beta\lambda}.\end{aligned} \tag{14}$$

Recall that Lemma 12 guarantees that $\Pr[\bar{B}] = e^{-\Theta(\lambda Z^t/n)}$. For the currently considered range of λ, we have $\Pr[\bar{B}] = \Omega(1)$, while $\Pr[B] = O(1)$. Also, $(1 - F^{-1/s})/(F-1) \overset{s \to 0}{\to} 1/(F-1) \overset{F \to 1}{\to} +\infty$, so a choice of F, s small enough (but constant) guarantees that the second and third line in (14) are both non-negative. This means that in the range of λ considered, the drift is at least some multiple of $1/\lambda$, which is at least $\delta' Z^t/n$ for a small enough constant δ'.

- If $\lambda > \max\{\tilde{\lambda}, n/Z^t\}$: then by ignoring the last positive contribution in (13) we see that the drift is at least

$$\begin{aligned}\mathbf{E}[G_2^t - G_2^{t+1} \mid x, \lambda] &\geq \Pr[B]\alpha_1 \left(\frac{Z^t}{n} - \frac{1}{2\lambda}\right) - \alpha_2 e^{-\beta\lambda} \\ &\geq \Pr[B]\alpha_1 \frac{Z^t}{2n} - \alpha_2 e^{-\beta\lambda} \\ &\geq \Pr[B]\alpha_1 \frac{Z^t}{4n} + \Pr[B]\alpha_1 \frac{1}{4\lambda} - \alpha_2 e^{-\beta\lambda}.\end{aligned}$$

Lemma 12 states that $\Pr[B] \geq 1 - e^{-b_3\lambda Z^t/n} \geq 1 - e^{-b_3} = \gamma$ since $\lambda > Z^t/n$ and by definition of γ. Since $\lambda > \tilde{\lambda}$, the last two contributions sum up to a non-negative constant so that the drift is at least $\gamma\alpha_1 \frac{Z^t}{4n} \geq \delta Z^t/n$.

At any time when $Z^t \geq 1$ and $\lambda^t \geq 1$ we have

$$G_2^t = Z^t + K_2 \max\{0, 1/\lambda - 1/\lambda_{\max}\} \leq Z^t(1 + K_2),$$

so $Z^t \geq G_2^t/(1 + K_2)$. Letting $\delta' = \delta/(K_2 + 1)$ we have for all $t \geq 0$ with $Z^t > 0$

$$\mathbf{E}[G_2^t - G_2^{t+1} \mid x, \lambda] \geq \delta' G_2^t.$$

This proves (12). To relate Z^t to G_2^t, recall that by Lemma 25, we have $Z^t \geq G_2^t - K_2$. To obtain a multiplicative drift, we distinguish two cases. If $Z^t \geq K_2$, then we have $Z^t \geq Z^t/2 + K_2/2 \geq (G_2^t - K_2)/2 + K_2/2 = G_2^t/2$. On the other hand, if $0 < Z^t < K_2$, then we have $Z^t \geq 1 = 2K_2/(2K_2) \geq G_2^t/(2K_2)$. Hence, for all times $t < T$ we have

$$\mathbf{E}\left[G_2^t - G_2^{t+1} \mid x, \lambda\right] \geq \delta' Z^t/n \geq \frac{\delta' G_2^t}{n \max\{2, 2K_2\}},$$

which concludes the proof. □

Now we are ready to prove the main result of this subsection.

Proof of Theorem 19. Corollary 27 guarantees that at all times before the time T when the optimum is found,

$$\mathbf{E}\left[G_2^t - G_2^{t+1} \mid x, \lambda\right] \geq \delta G_2^t/n,$$

for a constant $\delta > 0$. Moreover, $G_2^t \leq Z^t \leq n + K_2$ holds for all $t \geq 0$ by Lemma 25. Let $g_{\min} := \inf\{g_2(x,\lambda) \mid x \in \{0,1\}^n, \textsc{Zm}(x) > 0, \lambda \geq 1\} = K_2$. By the Multiplicative Drift Theorem 3,

$$\mathbf{E}[T] \leq \frac{n\left(1+\log\left(\frac{g_2(x^0,\lambda^0)}{g_{\min}}\right)\right)}{\delta} \leq \frac{n\left(1+\log\left(\frac{n+K_2}{K_2}\right)\right)}{\delta} = O(n\log n),$$

which concludes the proof. □

4.4. Bound on the number of evaluations

We have proved that the number of generations is respectively $O(n)$ or $O(n\log n)$ if $c < 1$ or $c = 1$. We will now turn our attention to the total number of function evaluations. For $c = 1$, we again need to assume that F is sufficiently close to one. More precisely, we will show the following theorems.

Theorem 28. *Let $0 < c < 1 < F$ be constants. Then there exist constants $C, s_0 > 0$ such that for all $s \leq s_0$ and every dynamic monotone function, the expected number of function evaluations of the SA-$(1,\lambda)$-EA with success rate s, update strength F and mutation probability c/n is at most $Cn\log n$.*

Theorem 29. *There exist constants $C, s_0 > 0$ and $F_0 > 1$ such that for all $s \leq s_0$, all $1 < F < F_0$ and every dynamic monotone function, the expected number of function evaluations of the SA-$(1,\lambda)$-EA with success rate s, update strength F and mutation probability $1/n$ is at most $Cn^2\log\log n$.*

Remark 30. Theorem 28 is tight since any unary unbiased algorithm needs at least $\Omega(n\log n)$ function evaluations to optimize OneMax [39]. On the other hand, Theorem 29 is not tight. Calculating a bit more precisely would allow to replace the $\log\log n$ factor by an even smaller factor. However, we suspect that even the main order n^2 is not tight, since the $(1+1)$-EA with $c = 1$ is known to need time $O(n^{3/2})$ even in the pessimistic PO-EA model [48], which includes every dynamic monotone function. The order $n^{3/2}$ is tight for the PO-EA, but a stronger bound of $O(n\log^2 n)$ is known for all static monotone functions [45], and an $O(n\log n)$ bound is known for all dynamic linear functions [53].

We conjecture that the number of function evaluations required to optimise *static* monotone functions is linear up to some logarithmic factors (in fact, we conjecture $O(n\log n)$) even for $c = 1$. However, the methods used in [45] are rather different from the ones in this paper, so it remains unclear whether they can be transferred.

We also conjecture that *dynamic* monotone functions are harder to optimise, i.e., that $O(n\log n)$ generations and $O(n^{3/2})$ evaluations are tight. More precisely, we conjecture that the 'adversarial' Dynamic BinVal described in our conclusion is the hardest dynamic monotone function for the SA-$(1,\lambda)$-EA, and requires $\Omega(n\log n)$ generations and $\Omega(n^{3/2})$ evaluations.

Remark 31. Our approach uses the best-so-far ZeroMax value Z_*^t defined below as in [32,38]. However, apart from that our proof is rather different. In fact, we believe that the proof in these papers is not fully correct. In the proof of Theorem 3.5 in [32], the authors bound the number of evaluations per generation by identically distributed random variables, and use Wald's equation to bound the total number of evaluations. However, Wald's equation is only true for the sum of *independent* random variables (or for similar conditions, e.g. [59]), a condition that is not satisfied in this situation. (The random variables are identically distributed, but not *independently* identically distributed.) Thus we need to use a different approach.

To avoid the issue mentioned above, we will decompose the interval $[n]$ into smaller 'sub-intervals' and we will show that with very high probability, the time needed for $\textsc{Zm}(x)$ to 'traverse' such an interval is of the expected order. We will compute the expected number of children at each of those steps, and will conclude using linearity of expectation.

To prove concentration of the time needed to traverse 'sub-intervals' we will use Theorem 7 in the case $c < 1$ and Theorem 8 in the case $c = 1$. The key ideas are summarised in the following lemmas. The first one is an adaptation of one proved by Hevia Fajardo and Sudholt. Below, we let $Z_*^t := \min_{t' \leq t}(Z^{t'})$ be the smallest value of Z^t observed until time t. Naturally, the process is unaware of this value but it will turn out useful for the analysis. We will apply the following lemmas to intermediate stages of a run, so we will consider an arbitrary starting population size λ^{init} in them.

Lemma 32 *(Fajardo, Sudholt [32,38]). Consider the SA-$(1,\lambda)$-EA as in Theorems 28 or 29, with an arbitrary initial search point and an initial value of $\lambda = \lambda^{\text{init}}$. There exists a constant $C > 0$ such that at all times $t \geq 0$ and for all $z > 0$ we have*

$$\mathbf{E}\left[\lambda^t \cdot \mathbf{1}_{Z_*^t \geq z}\right] \leq \lambda^{\text{init}}/F^t + Cn/z.$$

Lemma 33. *Consider the self-adjusting $(1, \lambda)$-EA as in Theorems 28 or 29, with an arbitrary initial search point and an initial value of $\lambda = \lambda^{\text{init}}$. Let T denote the first time t at which $\lambda^t \leq 8en\log n/Z^t$. There exists an absolute constant $C > 0$ such that*

$$\mathbf{E}\left[\sum_{t=1}^{T} \lambda^t\right] \leq C\lambda^{\text{init}}.$$

Lemma 34. *Let (a, b) be an interval of length $b - a = \log n$. Consider the self-adjusting $(1, \lambda)$-EA with $c < 1$ as in Theorem 28, with an initial search point $x = x^{\text{init}}$ such that $\textsc{Zm}(x^{\text{init}}) \leq b$, and an arbitrary initial value of λ.*

Let T be first time t at which $Z^t \leq a$. Then there exists an absolute constant $D > 0$ such that $T \leq D\log n$ with probability at least $1 - n^{-4}$.

Proof of Lemma 32. We will compute the expectation using the following formula

$$\mathbf{E}[\lambda^t \cdot \mathbf{1}_{Z^t_* \geq z}] \leq 1 + \sum_{\ell=1}^{\infty} \Pr\left[\lambda^t \geq \ell, Z^t_* \geq z\right]. \tag{15}$$

Let ℓ be an integer; if $\ell \leq \max\{\lambda^{\text{init}}/F^t, n/z\}$, then $\Pr[\lambda^t \geq \ell] = 1$. Otherwise, observe that for $\lambda^t \geq \ell$ there has to exist a time before t when λ increases, i.e. when the fitness does not improve. We may then write

$$\mathbf{1}_{\lambda^t \geq \ell,\ Z^t_* \geq z} = \sum_{k=1}^{t} \mathbf{1}_{\lambda^t \geq \ell,\ Z^t_* \geq z,\ t-k \text{ is the last time when } \lambda \text{ increases}}. \tag{16}$$

Note that if $(t - k)$ is the last time when λ increases, then the number of children at this time must be $\lfloor \lambda^t \cdot F^{k-1/s} \rceil \geq -1 + \ell \cdot F^{k-1/s}$. Naturally, if $Z^t_* \geq z$, we must also have $Z^{t-k}_* \geq z$. In particular, we find that if the following event holds

$$\left\{\lambda_t \geq \ell, \quad Z^t_* \geq z, \quad t - k \text{ is the last time when } \lambda \text{ increases}\right\},$$

then so does

$$\{\lambda_{t-k} \geq \ell F^{k-1/s},\ Z^{t-k}_* \geq z,\ \text{no improvement made at time } t-k\}. \tag{17}$$

If $Z^{t-k}_* \geq z$, the probability of a single child improving the fitness is at least

$$(1 - c/n)^{n - Z^{t-k}} \cdot \left(1 - (1 - c/n)^{Z^{t-k}}\right) \geq e^{-c} \cdot \left(1 - e^{-cz/n}\right) \geq \frac{cz}{2e^c n},$$

where the last step holds by Lemma 10 since $cz/n \leq 1$. In particular, this implies that the probability of the event in (17) is at most

$$\begin{aligned}
&\Pr\left[\lambda_{t-k} \geq \ell F^{k-1/s}, \quad Z^{t-k}_* \geq z, \quad \text{no improvement is made at time } t-k\right] \\
&\qquad \leq \left(1 - \frac{cz}{2e^c n}\right)^{\lfloor \ell F^{k-1/s} \rfloor}.
\end{aligned}$$

Replacing in (16) and using $F^k = e^{k\log F} \geq 1 + k\log F$ gives

$$\begin{aligned}
\Pr\left[\lambda_t = \ell, \textsc{Zm}^*_t \geq z\right] &\leq \sum_{k=1}^{t} \left(1 - \frac{cz}{2e^c n}\right)^{\lfloor \ell \cdot F^{k-1/s} \rfloor} \\
&\leq \sum_{k=1}^{\infty} \left(1 - \frac{cz}{2e^c n}\right)^{-1+\ell\cdot(1+k\log F)F^{-1/s}} \\
&= \left(1 - \frac{cz}{2e^c n}\right)^{-1+\ell F^{-1/s}} \cdot \frac{\left(1 - \frac{cz}{2e^c n}\right)^{\ell\log F F^{-1/s}}}{1 - \left(1 - \frac{cz}{2e^c n}\right)^{\ell\log F F^{-1/s}}} \\
&\leq C\left(1 - \frac{cz}{2e^c n}\right)^{\ell F^{-1/s}},
\end{aligned}$$

for a sufficiently large constant C since $\ell \geq n/z$. Now using Equation (15) and taking the trivial upper bound of 1 for the probability of the first $\max\{n/z, \lambda^{\text{init}}/F^t\}$ terms, we obtain

$$\begin{aligned}
\mathbf{E}\left[\lambda^t \cdot \mathbf{1}_{Z_*^t \geq z}\right] &\leq 1 + \frac{\lambda^{\mathrm{init}}}{F^t} + n/z + C \sum_{\ell=n/z}^{\infty} \left(1 - \frac{cz}{2e^c n}\right)^{\ell F^{-1/s}} \\
&\leq 1 + \frac{\lambda^{\mathrm{init}}}{F^t} + n/z + C \sum_{\ell=0}^{\infty} \left(1 - \frac{cz}{2e^c n}\right)^{\ell F^{-1/s}} \\
&\leq 1 + \frac{\lambda^{\mathrm{init}}}{F^t} + n/z + C \frac{1}{1 - \left(1 - \frac{cz}{2e^c n}\right)^{F^{-1/s}}} \\
&\leq 1 + \frac{\lambda^{\mathrm{init}}}{F^t} + n/z + C \frac{2e^c F^{1/s} n}{cz} \left(1 + \frac{cz}{2e^c n F^{1/s}}\right) \\
&\leq \frac{\lambda^{\mathrm{init}}}{F^t} + C' \frac{n}{z},
\end{aligned}$$

for a large constant $C' > 0$. □

Proof of Lemma 33. Consider a time $t < T$ such that $\lambda^t \geq 8en\log n/Z^t$. The probability that a child improves the fitness is at least

$$(1-c/n)^{n-Z^t}\left(1-(1-c/n)^{Z^t}\right) \geq e^{-c}\left(1-e^{-cZ^t/n}\right) \geq cZ^t/(2e^c n),$$

where the last step holds by Lemma 10 since $cZ^t/n \leq 1$. Hence the probability that all the $\lfloor \lambda^t \rceil$ children fail to improve the fitness is at most $\left(1 - cZ^t/(2e^c n)\right)^{\lfloor \lambda^t \rceil} \leq \exp\left(-cZ^t \lfloor \lambda^t \rceil/(2e^c n)\right) = o(1)$. From this, we easily compute

$$\mathbf{E}\left[\lambda^{t+1} \cdot \mathbf{1}_{t+1<T} \mid \lambda_t\right] = (1-o(1))\lambda^t/F + o(1)\lambda^t F^{1/s} \leq \lambda^t/F^{1/2}.$$

This recursively implies that $\mathbf{E}\left[\lambda^t \cdot \mathbf{1}_{t<T}\right] \leq \lambda^{\mathrm{init}} \cdot F^{-t/2}$. Using $\lambda^t \leq \lambda^{t-1}F^{1/s}$, we can now conclude,

$$\begin{aligned}
\mathbf{E}\left[\sum_{t=1}^{T} \lambda^t\right] &= \mathbf{E}\left[\sum_{t=1}^{\infty} \lambda^t \cdot \mathbf{1}_{t \leq T}\right] \leq \mathbf{E}\left[F^{1/s} \sum_{t=1}^{\infty} \lambda^{t-1} \cdot \mathbf{1}_{t \leq T}\right] \\
&= F^{1/s} \sum_{t=0}^{\infty} \mathbf{E}\left[\lambda^t \cdot \mathbf{1}_{t<T}\right] \leq F^{1/s} \sum_{t=0}^{\infty} \lambda^{\mathrm{init}} F^{-t/2} \leq C\lambda^{\mathrm{init}},
\end{aligned}$$

for some constant C. □

Proof of Lemma 34. Recall the definition

$$g_1(x,\lambda) = \textsc{Zm}(x) + h_1(\lambda) = \textsc{Zm}(x) + K_1 \max\{0, \log_F(\lambda_{\max}/\lambda)\},$$

and $G_1^t := g_1(x^t, \lambda^t)$. We define $\Gamma^t := \min\{1 + K_1/s, G_1^t - G_1^{t+1}\}$. We observe that $\sum_{t=0}^{\tau} \Gamma^t \leq G_1^0 - G_1^\tau \leq K_1 \log_F \lambda_{\max} + b - Z^\tau$, so if we define T' as the first time τ when $\sum_{t=1}^{\tau} \Gamma^t \geq K_1 \log_F \lambda_{\max} + b - a$, we clearly have $Z^\tau \leq a$. In other words, $T \leq T'$ and we will show the desired tail bound for T'.

To obtain the tail bound for T', we observe that h_1 is decreasing and that $h(\lambda) - h(\lambda F) \leq K_1$ for all λ. By Corollary 23, the drift of Γ_t is at least a constant $\varepsilon > 0$ and Lemma 17 applied with $C_1 = K_1/s$, $C_2 = K_1$ guarantee that $\varepsilon - \Gamma^t$ is sub-Gaussian. Thus Theorem 7 is applicable. Let D be the constant from Theorem 7. Then we choose $\tau = \max\{4/D, 2/\varepsilon\} \cdot \log n$ and Theorem 7 immediately implies that $\Pr[T > \tau] \leq \Pr[T' > \tau] \leq n^{-4}$. □

We are now ready to prove Theorem 28. We look at a slight alteration of the SA-$(1,\lambda)$-EA, working in exactly the same way as the ‘normal’ process except that we introduce some *idle steps* in which the algorithm does not do anything. Moreover, we divide a run of the algorithm into *blocks* and *phases* as follows. For simplicity, we will assume in the following that $n/\log n$ is an integer. A block starts with an initialisation phase which lasts until the condition $\lambda^t \leq F^{1/s} 8en\log n/Z^t$ is met. Once this phase is over, the block runs for $n/\log n$ phases of length $D\log n$, with D the constant of Lemma 34. During the i-th such phase the process attempts to improve Z^t from $n - i\log n$ to $n - (i+1)\log n$. If such an improvement is made before the $D\log n$ steps are over, then the process remains idle during the remaining steps of that phase. We call the non-idle steps *active*.

If a phase fails to make the correct improvement in $D\log n$ generations, or if $\lambda^t \geq F^{1/s} 8en\log n/Z_*^t$ at any point after the initialisation phase is over, then the whole block is considered a failure, and the next block starts. Obviously, the entire process stops (and succeeds) if the optimum is found. With this partitioning of a run, we will prove Theorem 28.

Proof of Theorem 28. The proof relies on the following two facts:

(i) every block finds the optimum whp;
(ii) consider a block starting with $\lambda = \lambda^{\text{init}}$, then the expected number of function evaluations during this block is at most $K(\lambda^{\text{init}} + n\log n)$ for a constant $K = K(c, F, s)$.

It is rather easy to see how those two items imply the theorem. The algorithm starts with an initial value of $\lambda = 1$, so the expected number of function evaluations in the first block is at most $O(n\log n)$. Recall that a block terminates as soon as λ goes above $F^{1/s}8en\log n/Z_*^t$ after the initialisation phase. This means that any block-run after the first one will start with $\lambda^{\text{init}} \le F^{2/s}8en\log n/Z_*^t$ and by ii its expected total number of evaluations is also $O(n\log n)$. The success of each block is at least $1-o(1)$ for all possible $x^{\text{init}}, \lambda^{\text{init}}$ it starts with, so by i we have an expected $(1+o(1))$ blocks, each requiring an expected $O(n\log n)$ evaluations, hence the result.

To conclude, we will now prove the two items above. Let us start with i which is simpler: the initialisation never fails since it runs until it succeeds, i.e., until λ gets small enough or the optimum is found. By Lemma 34, 'crossing' any interval of size $\log n$ fails with probability at most n^{-4}. By union bound over all $n/\log n$ phases of the block, the probability that one of them fails for this reason is at most $n^{-3}\log^{-1} n = o(n^{-3})$. Finally, the block might also fail because we have $\lambda^t \ge F^{1/s}8en\log n/Z_*^t$ at some point, which means that the $(t-1)$-th step was not successful despite $\lambda^{t-1} \ge 8en\log n/Z^{t-1}$, since $Z_*^t \le Z^{t-1}$. This happens with probability at most n^{-2} by Lemma 12, so by union bound over the Dn generations in the different phases, the probability that this happens at some time during a given block is at most $o(1)$.

We now prove ii. Consider any $1 \le i \le n/\log n$; we will show that the number of function evaluations in the i-th phase of the block is at most of order $\frac{n}{n-i\log n}$. In a slight abuse of notation, for $t \in [D\log n]$ we will let λ_i^t denote the value of λ in the t-th step *of the i-th phase*, and we set λ_i^t to be 0 if step t is idle, e.g. if the improvement to $Z^t \le n-(i+1)\log n$ has already been found, or if the i-th phase does not happen because the block has failed before that.

Since the previous phase is successful, the value of λ at the start of this phase must be $\lambda_i^0 \le C\frac{n\log n}{n-i\log n}$. By definition, t is only active if Z^t has never been at or below $n-(i+1)\log n$, so by Lemma 32 we have

$$\mathbf{E}\left[\lambda_i^t\right] \le C'\left(\lambda_i^0/F^t + \frac{n}{n-(i+1)\log n}\right).$$

Note that in each round, the number of function evaluations is $\lfloor\lambda_i^t\rceil \le 1+\lambda_i^t$. Summing over the $D\log n$ steps of the i-th phase, we see that the total number of function evaluations during this phase satisfies

$$\mathbf{E}\left[\sum_{t=1}^{D\log n}(1+\lambda_i^t)\right] \le C''\frac{n\log n}{n-(i+1)\log n},$$

for a constant $C'' > 0$. Hence, excluding the initialisation phase, the total number of function evaluations is in expectation at most

$$C''\sum_{i=1}^{n/\log n}\frac{n\log n}{n-i\log n} = C''\sum_{j=1}^{n/\log n}\frac{n\log n}{j\log n} \le C'''n\log n$$

for a new constant $C''' > 0$. Lemma 33 immediately gives that the expected number of function evaluations of the initialisation phase is at most $C'\lambda^{\text{init}}$, and ii is proved. □

The proof of Theorem 29 is extremely close to that above, so we will only give a sketch of it. The main difference is the way the phases of a block are defined: during the i-th phase, the algorithm attempts to improve Z^t from $n/\log^{i-1} n$ to $n/\log^i n$ in $Dn\log\log n$ steps for a large constant D. One checks that $n/\log^i n = n/e^{i\log\log n}$ is less than 1 for $i > \log n/\log\log n$, so we have $1+\log n/\log\log n$ phases in a block. Since G_2^t has multiplicative drift, Theorem 8 immediately gives that the probability that a phase fails to improve Z^t is at most $\log^{-2} n$, so the probability that any phase in a fixed block fails is $O(\log n/\log\log n \cdot \log^{-2} n) = o(1)$. In the i-th phase, the expected value of λ at each step is at most $O(\log^i n)$. The total number of function evaluations per block after the initialization phase is then

$$O(1)\sum_{i=0}^{\log n/\log\log n} n\log\log n\cdot\log^i n = O(n^2\log\log n),$$

which implies Theorem 29.

5. Close to the optimum success rates become asymptotically irrelevant

In this section, we will show that one can still have efficient search even when s is large, provided one starts close enough to the optimum. More precisely, we will prove the following theorem. For simplicity, we only treat the case $c < 1$.

Theorem 35. *Let $0 < c < 1 < F$ be constants. For every $s > 0$, there exists an $\varepsilon > 0$ such that for any initial search point x^0 satisfying $\textsc{Zm}(x^0)/n \le \varepsilon$ and for any initial population size $\lambda^{\text{init}} \ge 1$ the following holds. For every dynamic monotone function, with high probability the number of generations of the SA-$(1,\lambda)$-EA with success rate s, update strength F, and mutation probability c/n and initial state $(x^0, \lambda^{\text{init}})$ is $O(n)$. Additionally, the number of function evaluations is $\omega(n \log n)$ with probability $o(1)$.*

The approach will be essentially the same as in Section 4.2. The main difference lies in the potential function: we will need to introduce a second penalty term into the part $h(\lambda)$ that depends on λ. Moreover, when s is large, no potential function can have strong positive drift towards the optimum for all values of Z^t, as it would otherwise contradict the negative results from Section 6. Hence, we will only show that the potential has positive drift when $Z^t/n \le 2\varepsilon$. Then by the Negative Drift Theorem, starting from $Z^t/n \le \varepsilon$ it is unlikely that the exploration reaches a search point for which $Z^t/n > 2\varepsilon$ in polynomial time. Hence, the algorithm stays in a range where the drift is positive, and by the Additive Drift Theorem the optimum is found efficiently.

Definition 36 *(Potential function for positive result near optimum).* For all $\lambda \ge 1$, we set

$$h_3(\lambda) = K_1 \max\left\{0, \log_F \lambda_{\max}/\lambda\right\} + K_2 e^{-K_3\lambda},$$

where $\lambda_{\max} = F^{1/s}n$ and the constants $K_1, K_2, K_3 > 0$ will be fixed later. Then for $x \in \{0,1\}^n$ and $\lambda \in [1,\infty)$ we define

$$g_3(x,\lambda) := \textsc{Zm}(x) + h_3(\lambda),$$

and we set $H_3^t := h_3(x^t, \lambda^t)$ and $G_3^t := g_3(x^t, \lambda^t)$, and as usual $Z^t := \textsc{Zm}(x^t)$.

As before, we get a sandwich lemma.

Lemma 37 *("Sandwich" inequalities). For all x, λ we have*

$$g_3(x,\lambda) - K_1 \log_F \lambda_{\max} - K_2 \le \textsc{Zm}(x) \le g_3(x,\lambda).$$

Proof. The proof runs as for Lemma 21, except that we also use $K_2 e^{-K_3\lambda} \in [0, K_2]$. □

As in Section 4, $A = A^t$ denotes the event that some child of x^t flips no one-bit and $B = B^t$ the event that some child of x^t flips (at least) one zero-bit. Also recall that we abbreviate $x = x^t$ and $\lambda = \lambda^t$ when t is clear from the context.

Claim 38. *At all times $t \ge 1$ with $Z^t > 0$ we have*

$$\begin{aligned}\mathbf{E}[H_3^t - H_3^{t+1} \mid x, \lambda] \ge \mathbf{1}_{\lambda<n} \Pr[\bar{B}]\big(K_1/s + K_2(1 - e^{-K_3(F^{1/s}-1)})e^{-K_3\lambda}\big) \\ - \Pr[B] \cdot \big(K_1 + K_2 e^{-K_3\lambda/F}\big).\end{aligned}$$

Proof. The proof is extremely similar to that of Claim 22. In particular, the contribution of the first term $K_1 \max\left\{0, \log_F \lambda_{\max}/\lambda\right\}$ is exactly the same, so we will only compute the contribution of the second term. Since that term is non-negative at time t and is at most $K_2 e^{-K_3\lambda/F}$ at time $t+1$ due to $\lambda^{t+1} \ge \lambda/F$, it contributes at least $-K_2 e^{-K_3\lambda/F}$ to the difference $H_3^t - H_3^{t+1}$.

In the case $f(x^{t+1}) \le f(x^t)$ of non-success we have $\lambda^{t+1} = \lambda F^{1/s}$, so we may use the exact contribution to $H_3^t - H_3^{t+1}$, which is

$$K_2(1 - e^{-K_3\lambda(F^{1/s}-1)})e^{-K_3\lambda} \ge K_2(1 - e^{-K_3(F^{1/s}-1)})e^{-K_3\lambda},$$

since $\lambda \ge 1$. As in Section 4.1, observing that $\Pr[f(x^{t+1}) > f(x^t)] \le \Pr[B]$ and using the law of total expectation gives the result. □

Corollary 39. *There exist constants $\varepsilon, \delta, K_1, K_2, K_3 > 0$ (which may only depend on c, F, s) such that for all times t when $0 < Z^t/n \le 2\varepsilon$ we have*

$$\mathbf{E}\left[G_3^t - G_3^{t+1} \mid x, \lambda : \textsc{Zm}(x)/n \le 2\varepsilon\right] \ge \delta.$$

This also holds if we replace $G_3^t - G_3^{t+1}$ by $\min\{1 + K_1/s + K_2\,,\ G_3^t - G_3^{t+1}\}$.

Proof. Combining Claims 15, 16 and 38, there exist constants $\alpha_1, \alpha_2, \alpha_3, \beta > 0$ (which may only depend on c, F, s) such that at all times t with $Z^t > 0$ we have

$$\begin{aligned}\mathbf{E}[G_3^t - G_3^{t+1} \mid x, \lambda] \geq \Pr[B]\left(\alpha_1 - K_1 - \alpha_2 K_2 e^{-K_3\lambda/F}\right) - \alpha_3 e^{-\beta\lambda} \\ + \Pr[\bar{B}]\mathbf{1}_{\lambda<n}\left(\alpha_4 K_2 K_3 e^{-K_3\lambda} + K_1/s\right). \end{aligned} \tag{18}$$

We choose $K_1 = \alpha_1/2$, $K_3 = \beta/2$, $K_2 \geq 2\alpha_3/(\alpha_4 K_3)$. We will prove a constant lower bound in two different cases, depending on whether λ is small or not.

- If $\lambda \leq \frac{F}{K_3}\log\frac{2\alpha_2 K_2}{K_1}$: then, ignoring the (positive) contribution of $\Pr[B](\alpha_1 - K_1)$ and $\Pr[\bar{B}]K_1/s$, we see that the drift of g is at least

$$\begin{aligned}\mathbf{E}[G_3^t - G_3^{t+1} \mid x, \lambda] &\geq -\Pr[B]\alpha_2 K_2 e^{-K_3\lambda/F} - \alpha_3 e^{-\beta\lambda} + \Pr[\bar{B}]\alpha_4 K_2 K_3 e^{-K_3\lambda} \\ &\geq -\Pr[B]\alpha_2 K_2 + \alpha_3 e^{-K_3\lambda}(2\Pr[\bar{B}] - 1), \\ &\geq -\Pr[B]\alpha_2 K_2 + \alpha_3\left(\frac{K_1}{2\alpha_2 K_2}\right)^F (2\Pr[\bar{B}] - 1) \end{aligned} \tag{19}$$

where in the second step we bounded $e^{-K_3\lambda/F} \leq 1$ and used $\beta \geq K_3$ and $K_2 \geq 2\alpha_3/(\alpha_4 K_3)$, and in the third step we used $\lambda \leq F/K_3 \cdot \log(2\alpha_2 K_2/K_1)$. Recall that $\Pr[\bar{B}] \geq e^{-b_2\lambda Z^t/n}$ for some $b_2 > 0$ by Lemma 12. Since λ is bounded by a constant in the current case, if we choose ε small enough (w.r.t. all previous constants) then we can achieve $2\Pr[\bar{B}] - 1 \geq 1/2$ and $\Pr[B]\alpha_2 K_2 \leq \frac{\alpha_3}{4}\left(\frac{K_1}{2\alpha_2 K_2}\right)^F$. Then the drift must be at least

$$\mathbf{E}[G_3^t - G_3^{t+1} \mid x, \lambda] \geq \frac{\alpha_3 K_1^F}{4(2\alpha_2 K_2)^F} \geq \delta,$$

for a $\delta > 0$ chosen small enough.

- If $\frac{F}{K_3}\log\frac{2\alpha_2 K_2}{K_1} \leq \lambda < n$: then the first contribution in (18) is at least $\Pr[B]K_1/2$. Hence, the drift is at least

$$\begin{aligned}\mathbf{E}[G_3^t - G_3^{t+1} \mid x, \lambda] &\geq \Pr[B]\frac{K_1}{2} - \alpha_3 e^{-\beta\lambda} + \Pr[\bar{B}]\left(\alpha_4 K_2 K_3 e^{-K_3\lambda} + \frac{K_1}{s}\right) \\ &= \Pr[B]\frac{K_1}{2} + \Pr[\bar{B}]\frac{K_1}{s} + \left(2\Pr[\bar{B}] - e^{-K_3\lambda}\right)\cdot\alpha_3 e^{-K_3\lambda} \\ &\geq \frac{K_1}{\max\{2, s\}} + \left(2\Pr[\bar{B}] - e^{-K_3\lambda}\right)\cdot\alpha_3 e^{-K_3\lambda}, \end{aligned}$$

where the second line holds since $K_3 = \beta/2$ and $K_2 = 2\alpha_3/(\alpha_4 K_3)$. Recall that $\Pr[\bar{B}] = (1 - c/n)^{\lfloor\lambda\rceil Z^t} \geq e^{-4c\lambda\varepsilon}$ whenever $Z^t/n \leq 2\varepsilon$ and n is sufficiently large. Choosing ε small guarantees that this is larger than $e^{-K_3\lambda}$ for all λ, meaning that the drift of G_3^t is at least $K_1/\max\{2, s\} \geq \delta$ for a suitable $\delta > 0$.

- If $\lambda \geq n$: then in (18) every negative contribution is of order $e^{-\Omega(n)}$ while the term $\Pr[B](\alpha_1 - K_1) = \Pr[B]K_1$ is $\Theta(1)$, so the drift is at least δ for some $\delta > 0$.

For the last part of the statement, we use the same argument as for Corollary 23, using that (18) also holds for $\min\{1, G^t - G^{t+1}\}$ since Claims 15 and 16 do, and h may only increase by $K_1/s + K_2$ at each step. □

We may now prove Theorem 35.

Proof of Theorem 35. Let $\varepsilon, \delta, K_1, K_2, K_3 > 0$ be the constants from Corollary 39 and assume the initial search point x^0 is such that $\textsc{Zm}(x^0)/n \leq \varepsilon$. Analogously to the proof of Theorem 28, we define $\Gamma^t = \min\{1 + K_1/s + K_2, G_3^t - G_3^{t+1}\}$, let T be the first time t when $Z^t = 0$ or $G_3^t = 0$, and observe that this actually implies $Z^T = 0$. Moreover, we define T' as the first time when $Z^t/n > 2\varepsilon$.

By Corollary 39 the drift at any time $t < \min\{T, T'\}$ is at least

$$\mathbf{E}\left[\mathbf{1}_{t<\min\{T,T'\}}\Gamma^t \mid x, \lambda\right] \geq \delta.$$

As in the proof of Theorem 28, we use Lemma 17 to see that $\delta - \Gamma^t$ is sub-Gaussian since h_3 is decreasing may not increase too much at each step.

In particular, Theorem 7 gives that, for a suitable constant $D > 0$, the event $E := \{T > Dn$ and $\sum_{\tau=0}^{Dn}\Gamma^\tau < \varepsilon n + K_1\log\lambda_{\max} + K_2\}$ has probability $\Pr[E] = e^{-\Omega(n)}$. If the second event does not happen, $\sum_{\tau=0}^{Dn}\Gamma^\tau \geq \varepsilon n + K_1\log\lambda_{\max} + K_2$, then by the Sandwich Lemma 37 this implies $Z^t \leq 0$ for $t = Dn$ and thus $T \leq Dn$. Hence, $\Pr[T \leq Dn] \geq \Pr[\bar{E}] = 1 - e^{-\Omega(n)}$, and the statement about the number of generations is proven. For the number of function evaluations, in the proof in Theorem 28 we use the potential function as a black box (except for the Sandwich Lemma), so the proof carries over. □

6. Small success rates yield exponential runtimes

The aim of this section is to show that for large s, that is, for a small enough success rate, the SA-$(1, \lambda)$-EA needs superpolynomial time to find the optimum of any dynamic monotone function. The reason is that the algorithm has negative drift in a region that is still far away from the optimum, in linear distance. In fact, as we have shown in Section 5, the drift is *positive* close to the optimum. Thus the hardest region for the SA-$(1, \lambda)$-EA is not around the optimum. This surprising phenomenon was discovered for OneMax in [38]. We show that it is not caused by any specific property of OneMax, but that it occurs for *every* dynamic monotone function. Even in the OneMax case, our result is slightly stronger than [32], since they show their result only for $1 < F < 1.5$, while ours holds for all $F > 1$. On the other hand, they give an explicit constant $s_1 = 18$ for OneMax.

Theorem 40. *Let $0 < c \leq 1 < F$. For every $\varepsilon > 0$, there exists $s_1 > 0$ such that for all $s \geq s_1$ the following holds. For every dynamic monotone function and every initial search point x^{init} satisfying $\text{Zm}(x^{\text{init}}) \geq \varepsilon n$ the number of generations of the SA-$(1, \lambda)$-EA with success rate s, update strength F, and mutation probability c/n is $e^{\Omega(n/\log^2 n)}$ with high probability.*

Definition 41 *(Potential function for negative result).* Given F, we define

$$h_4(\lambda) := -K_4 \log_F^2(\lambda F) = -K_4 \cdot (\log_F(\lambda) + 1)^2$$

with K_4 a positive constant to be chosen later. As before, we define the potential function to be the sum of $\text{Zm}(x)$ and $h_4(\lambda)$:

$$g_4(x, \lambda) = \text{Zm}(x) + h_4(\lambda).$$

As usual, we set $G_4^t := g_4(x^t, \lambda^t)$, $H_4^t := h_4(\lambda^t)$ and $Z^t := \text{Zm}(x^t)$. Contrary to the previous sections, we now are now aiming to show that the difference $G_4^{t+1} - G_4^t$ is positive in expectation. (Note the switched order of $t + 1$ and t.) This will require approaches slightly different from the ones we used so far.

The theorem will be proved using the following lemmas. Recall from Section 3.1 the event B that at least one child flips a zero-bit.

Lemma 42. *There exists a constant $\alpha_1 > 0$ depending only on c such that at all times t we have*

$$\mathbf{E}[Z^{t+1} - Z^t \mid x, \lambda] \geq -\Pr[B]\alpha_1(1 + \log \lambda).$$

Lemma 43. *There exist constants $\varepsilon, \alpha_2 > 0$ depending only on c, F such that if $Z^t \leq \varepsilon n$ and $\lambda \leq F$, then*

$$\mathbf{E}[Z^{t+1} - Z^t \mid x, \lambda] \geq \alpha_2.$$

Lemma 44. *Assume that $s \geq 1 \geq c$. At all times t with $Z^t > 0$ we have*

$$\mathbf{E}[H^{t+1} - H^t \mid x, \lambda] \geq \frac{1}{3} \Pr[B] K_4 (1 + \log_F \lambda) \mathbf{1}_{\lambda \geq F} - \frac{3}{s} K_4 (1 + \log_F \lambda)$$

Proof of Lemma 42. The event $\bar{B}$ implies $\text{supp}(x^{t+1}) \subseteq \text{supp}(x^t)$ and thus $Z^{t+1} - Z^t \geq 0$. Hence, $\mathbf{E}[Z^{t+1} - Z^t \mid \bar{B}] \geq 0$. By the law of total probability, we may thus bound

$$\begin{aligned} \mathbf{E}[Z^{t+1} - Z^t] &= \Pr[B] \cdot \mathbf{E}[Z^{t+1} - Z^t \mid B] + \Pr[\bar{B}] \cdot \mathbf{E}[Z^{t+1} - Z^t \mid \bar{B}] \\ &\geq \Pr[B] \cdot \mathbf{E}[Z^{t+1} - Z^t \mid B]. \end{aligned} \tag{20}$$

To bound the conditional expectation, let N^j be the number of zero-bits flipped by the j-th individual, and let $N := \max_j\{N^j\}$. We have $Z^{t+1} - Z^t \geq -N$, so we would like to bound $\mathbf{E}[-N \mid B]$. The events B^j are positively correlated with the event $N \geq z$, for every $z \geq 1$. Therefore,

$$\begin{aligned} \Pr[N \geq z \mid B] &\leq \Pr[N \geq z \mid B, B^1, \ldots, B^{\lfloor\lambda\rceil}] = \Pr[N \geq z \mid B^1, \ldots, B^{\lfloor\lambda\rceil}] \\ &= 1 - \prod_{j=1}^{\lfloor\lambda\rceil} (1 - \Pr[N^j \geq z \mid B^j]). \end{aligned}$$

As in the proof of Claim 16, we can couple the one-bit flips in y^j given B^j by first sampling the position l of the leftmost one-bit flip, and then flipping all bits to the right of l independently with probability c/n. Since there are less than

n positions to the right of l, this shows that N^j is dominated by $1+N'$, where N' follows a $\mathrm{Bin}(n, c/n)$ distribution. In particular, by the Chernoff bound, Theorem 11, $\Pr[N^j \geq z \mid B^j] \leq \Pr[N' \geq z-1] \leq e^{-\alpha_0(z-1)}$ for a constant α_0 that only depends on c. Hence,

$$\begin{aligned}\Pr[N \geq z \mid B] &\leq 1 - \prod_{j=1}^{\lfloor\lambda\rceil}(1-\Pr[N^j \geq z \mid B^j]) \leq 1-(1-e^{-\alpha_0(z-1)})^{\lfloor\lambda\rceil}\\ &\leq \min\{1, \lfloor\lambda\rceil e^{-\alpha_0(z-1)}\},\end{aligned}$$

and

$$\begin{aligned}\mathbf{E}[N \mid B] &= \sum_{z=1}^{\infty}\Pr[N \geq z \mid B] \leq \sum_{z=1}^{\infty}\min\{1, \lfloor\lambda\rceil e^{-\alpha_0(z-1)}\}\\ &\leq 1+\log\lfloor\lambda\rceil + \sum_{z=\log\lfloor\lambda\rceil+1}^{\infty}\lfloor\lambda\rceil e^{-\alpha_0(z-1)} \leq \alpha_1(1+\log\lambda)\end{aligned}$$

for a suitable constant $\alpha_1 > 0$. Combining this with (20), we obtain

$$\begin{aligned}\mathbf{E}[Z^{t+1}-Z^t] &\geq \Pr[B]\cdot\mathbf{E}[Z^{t+1}-Z^t \mid B]\\ &\geq \Pr[B]\cdot\mathbf{E}[-N \mid B] \geq -\Pr[B]\alpha_1(1+\log\lambda),\end{aligned}$$

as desired. □

Proof of Lemma 43. For all $j \in [\lfloor\lambda\rceil]$, let us denote by M^j the number of one-bits flipped by the j-th offspring and $M = \min_j M^j$. We also define N^j as the number of zero-bits flipped by the j-th child and let $N = \max_j N^j$.

Clearly, $Z^{t+1} - Z^t \geq M - N$, so it suffices to prove that $\mathbf{E}[M-N] \geq \alpha_2$ for some constant α_2.

Observe that M is the minimum of $\lfloor\lambda\rceil \leq \lfloor F\rceil$ i.i.d. random variables following a binomial distribution $\mathrm{Bin}(n-Z^t, c/n)$. In particular,

$$\begin{aligned}\Pr[M \geq 1] &= \left(1-(1-c/n)^{(n-Z^t)}\right)^{\lfloor\lambda\rceil}\\ &\geq \left(1-e^{-(1-\varepsilon)c}\right)^{\lfloor F\rceil},\end{aligned}$$

since $Z^t \leq \varepsilon n$. From this we deduce that $\mathbf{E}[M] \geq (1-e^{-(1-\varepsilon)c})^{\lfloor F\rceil} = \Omega(1)$.

Observe now that $N \leq \sum_j N^j$. Since each N^j follows a binomial distribution $\mathrm{Bin}(Z^t, c/n)$ the expected value of N is at most

$$\mathbf{E}[N] \leq \lfloor\lambda\rceil\, Z^t c/n \leq \varepsilon c\,\lfloor F\rceil\,.$$

Choosing ε small enough and $\alpha_2 = \left(1-e^{-(1-\varepsilon)c}\right)^{\lfloor F\rceil} - \varepsilon c\,\lfloor F\rceil$ gives the result. □

Proof of Lemma 44. Conditioned on $f(x^{t+1}) \leq f(x^t)$ we have

$$\begin{aligned}H_4^{t+1}-H_4^t &= -K_4\log_F^2(\lambda F^{1+1/s}) + K_4\log_F^2(\lambda F)\\ &= -K_4\left(\log_F^2(\lambda F) + 2\log_F(\lambda F)/s + 1/s^2 - \log_F^2(\lambda F)\right)\\ &= -\frac{1}{s}K_4\,(2\log_F\lambda + 2 + 1/s)\\ &\geq -\frac{3}{s}K_4\,(1+\log_F\lambda)\,,\end{aligned}$$

since $s \geq 1$.

If we now condition on $f(x^{t+1}) > f(x^t)$ and assume $\lambda \geq F$, we have

$$\begin{aligned}H_4^{t+1}-H_4^t &= -K_4\log_F^2(\lambda) + K_4\log_F^2(\lambda F)\\ &= K_4\,(2\log_F\lambda + 1)\\ &\geq K_4(1+\log_F\lambda).\end{aligned}$$

We observe that h_4 is decreasing with λ, so when $\lambda < F$ we may simply lowerbound the drift by 0.

The law of total probability then gives

$$\mathbf{E}\left[H_4^{t+1}-H_4^t\right]\geq \Pr[f(x^{t+1})>f(x^t)]K_4(1+\log_F\lambda)\mathbf{1}_{\lambda\geq F}$$
$$-\Pr[f(x^{t+1})\leq f(x^t)]\frac{3}{s}K_4(1+\log_F\lambda).$$

Clearly $\Pr[f(x^{t+1})\leq f(x^t)]\leq 1$, so to obtain the result is suffices to prove that $\Pr[f(x^{t+1})>f(x^t)]\geq \Pr[B]/3$.

Recall that B is the event that some offspring flips a zero-bit of the parent into a one-bit at time t. Assume that B holds, and let $j\in[\lfloor\lambda\rceil]$ be the index of a child which flips a zero-bit of the parent. Clearly, if y^j flips no one-bit of x^t into a zero-bit, then $f(y^j)>f(x)$ and the fitness increases at step t. The event that y^j flips no one-bit is independent of B and has probability $(1-c/n)^{n-Z^t}\geq e^{-c}$. In particular, this implies that $\Pr[f(x^{t+1})>f(x^t)]\geq e^{-c}\Pr[B]\geq\Pr[B]/3$ since $c\leq 1$. □

Corollary 45. *For all $0<c\leq 1<F$ and every sufficiently small $\varepsilon>0$ there exists $s_1>0$ such that for all $s\geq s_1$ the following holds. There exists a constant $\delta>0$ such that if $\varepsilon n/2\leq Z^t\leq\varepsilon n$ then*

$$\mathbf{E}\left[G_4^{t+1}-G_4^t\mid x,\lambda\right]\geq\delta.$$

Proof. We take $\varepsilon>0$ so small that Lemma 43 is applicable, and let α_1,α_2 be the other constants from Lemmas 42 and 43.

We will show that for a sufficiently large (but constant) s, the drift of G_4^t is at least a constant δ when $Z^t\in[\varepsilon n/2,\varepsilon n]$. We distinguish on whether λ is small or not.

If $\lambda\geq F$, then Lemmas 42 and 44 combine into

$$\mathbf{E}[G_4^{t+1}-G_4^t\mid x,\lambda]\geq\Pr[B]\left(\tfrac{1}{3}K_4(1+\log_F\lambda)-\alpha_1(1+\log\lambda)\right)$$
$$-\tfrac{3}{s}K_4(1+\log_F\lambda).$$

We choose K_4 large enough so that $\alpha_1(1+\log\lambda)\leq K_4(1+\log_F\lambda)/12$ for all λ. The drift is then

$$\begin{aligned}\mathbf{E}[G_4^{t+1}-G_4^t\mid x^t,\lambda^t]&\geq\tfrac{1}{4}\Pr[B]K_4(1+\log_F\lambda)-\tfrac{3}{s}K_4(1+\log_F\lambda)\\&=K_4(1+\log_F\lambda)\left(\tfrac{1}{4}\Pr[B]-\tfrac{3}{s}\right).\end{aligned}\tag{21}$$

Recall that Lemma 12 guarantees that $\Pr[B]\geq 1-e^{-b_3Z^t\lambda/n}$ for some positive constant b_3. In particular, since $Z^t\geq\varepsilon n/2$,

$$\Pr[B]\geq 1-e^{-b_3\varepsilon\lambda/2}\geq 1-e^{-b_3\varepsilon/2}$$

is at least a constant. For a choice of s larger than $24(1-e^{-b_3\varepsilon/2})$, the bound (21) is at least $\mathbf{E}[G_4^{t+1}-G_4^t\mid x,\lambda]\geq\delta$ for some constant $\delta>0$.

If $\lambda<F$, then Lemmas 43 and 44 guarantee that

$$\begin{aligned}\mathbf{E}[G_4^{t+1}-G_4^t\mid x,\lambda]&\geq\alpha_2-\frac{3}{s}K_4(1+\log_F\lambda)\\&\geq\alpha_2-\frac{6}{s}K_4.\end{aligned}$$

For a choice of s large enough this is at least δ, for some constant $\delta>0$. □

We are now ready to prove the main theorem of this section. Essentially, it follows from Corollary 45 and the Negative Drift Theorem 4. However, compared to the other sections, there is a slight complication since the difference $|G_4^t-Z^t|=K_4\log_F^2(\lambda F)$ is not bounded. However, we will prove that with overwhelming probability the difference does not grow larger than $K_4\sqrt{n}$.

Proof of Theorem 40. Let $\Lambda:=n^2/F$ and let T be the first point in time when $Z^t\leq\varepsilon n/2$. We first show that with overwhelming probability, we have $\lambda^t\leq\Lambda$ for all $1\leq t\leq\min\{T,e^n\}$. Indeed, to obtain some $\lambda>\Lambda$, it would be necessary to have a step with $\lambda>\Lambda F^{-1/s}$ that does not improve the fitness. If this were to happen before time T, it must happen in a step with $Z^t\geq\varepsilon n/2$. By Lemma 12, the probability to have a non-improving step is $e^{-\Omega(\lambda)}$. By a union bound, the probability that such a step happens before time e^n is at most $e^{n-\Omega(\lambda)}=o(1)$. Hence, w.h.p. $\lambda^t\leq\Lambda$ for all $1\leq t\leq\min\{T,e^n\}$. Note that in this case we have $|G_4^t-Z^t|\leq 4K_4\log_F^2 n$, so in particular, $G_4^t>4K_4\log_F^2 n$ implies that $Z^t>0$ for $\lambda\leq\Lambda$.

In the following, we will apply the Negative Drift Theorem 4 to G_4^t. The drift condition is satisfied by Corollary 45 whenever $Z^t\in[\varepsilon n/2,\varepsilon n]$, which is implied whenever $G_4^t\in[\varepsilon n/2+4K_4\log_F^2 n,\varepsilon n]$ and $\lambda\leq\Lambda$.

For the step size condition, let L^j denote the total number of bits flipped in y^j, and $L:=\max\{L^j\}_j$. Since L^j follows a $\mathrm{Bin}(n,c/n)$ distribution, by the Chernoff bound, Theorem 11, there is a constant $\beta>0$ such that $\Pr[L^j\geq z]\leq e^{-\beta z}$ for all $z \geq 0$. Let $r := 4K_4 \log_F n/\beta$, and note that we can achieve $|H_4^{t+1} - H_4^t| \leq r/2$ when $\lambda \leq \Lambda$, by making $\beta > 0$ smaller if necessary. Then for all $j \geq 1$,

$$\begin{aligned}\Pr[|G_4^{t+1} - G_4^t| \geq jr] &\leq \Pr[|Z^{t+1} - Z^t| \geq (j-1/2)r] \leq \Pr[L \geq (j-1/2)r] \\ &= 1 - \big(1 - \Pr[L^1 \geq (j-1/2)r]\big)^{\lfloor \lambda \rceil} \\ &\leq 1 - (1 - e^{-\beta(j-1/2)r})^{\lfloor \lambda \rceil} \leq 1 - (1 - n^{-4(j-1/2)})^{\Lambda+1} \\ &\leq 1 - e^{-\frac{1}{2}(\Lambda+1)n^{-4(j-1/2)}} = 1 - e^{-n^{-\Omega(j)}} \\ &= n^{-\Omega(j)} \leq e^{-j},\end{aligned}$$

where the last inequality holds for n sufficiently large. Thus the step size condition of Theorem 4 is satisfied, and we obtain that w.h.p. $G_4^t \geq \varepsilon n/2 + 4K_4 \log_F^2 n$ for $e^{\Omega(n/\log^2 n)}$ steps if $\lambda^t \leq \Lambda$ during this time. Since the latter also holds w.h.p., this implies $T = e^{\Omega(n/\log^2 n)}$ w.h.p., which concludes the proof. □

7. Simulations

In this section, we provide simulations that complement our theoretical analysis. The functions optimized in our simulations include OneMax, Binary, HotTopic [46], BinaryValue, and Dynamic BinVal [41], where Binary is defined as $f(x) = \sum_{i=1}^{\lfloor n/2 \rfloor} x_i n + \sum_{i=\lfloor n/2 \rfloor+1}^{n} x_i$, and BinaryValue is defined as $f(x) = \sum_{i=1}^{n} 2^{i-1} x_i$. The definition of HotTopic can be found in [46], and we set the parameters to $L = 100$, $\alpha = 0.25$, $\beta = 0.05$, and $\varepsilon = 0.05$. Dynamic BinVal is the dynamic environment which applies the BinaryValue function to a random permutation of the n bit positions, see [41] for its formal definition. In all experiments, we start the SA-$(1,\lambda)$-EA with a randomly sampled search point and an initial offspring size of $\lambda^{\text{init}} = 1$. The algorithm terminates when the optimum is found or after $500n$ generations. The code for the simulations can be found at https://github.com/zuxu/OneLambdaEA.

7.1. Threshold of s

In Fig. 1, we follow the same setup as in [32], but for a larger set of functions. We observe exactly the same threshold $s = 3.4$ for OneMax. For the other monotone functions of our choice, the threshold effect happens before $s = 3.4$, which suggests that some hard monotone functions might have a lower allowance for the value of s than OneMax, other than conjectured by Hevia Fajardo and Sudholt in [32].

7.2. Effect of F

We have shown that the SA-$(1,\lambda)$-EA with $c < 1$ optimizes every dynamic monotone function efficiently when s is sufficiently small and is inefficient when s is too large. Both results hold for arbitrary F. It is natural to assume that there is a threshold s_0 between the efficient and inefficient regime. However, Fig. 2 below shows that the situation might be more complicated. For this plot, we have first empirically determined an efficiency threshold for s on Dynamic BinVal (see Fig. 1), then fixed s slightly below this threshold and systematically varied the value of F. For this intermediate value of s, we see that there is a phase transition in terms of F.

Hence, we conjecture that there is no threshold s_0 such that the SA-$(1,\lambda)$-EA is efficient for all $s < s_0$ and all $F > 1$, and inefficient for all $s > s_0$ and all $F > 1$. Rather, we conjecture that there is 'middle range' of values of s for which it depends on the value of F whether the SA-$(1,\lambda)$-EA is efficient. Note that we know from this paper that this phenomenon can *only* occur for a 'middle range': both for sufficiently small s (Theorems 18, 28), and for sufficiently large s (Theorem 40), the value of F does not play a role.

In general, smaller values of F seem to be beneficial. However, the correlation is not perfect, see for example the dip for $c = 0.98$ and $F = 5.5$ in the left subplot of Fig. 2. These dips also happen for some other combinations of s, F and c (not shown), and they seem to be consistent, i.e., they do not disappear with a larger number of runs or larger values of n up to $n = 5000$. To test whether this is due to the rounding scheme, we checked whether the effect disappears if we round λ in each generation stochastically to the next integer; e.g., $\lambda^t = 2.6$ means that in generation t we create two offspring with probability 40% and three offspring with probability 60%. The effect remains, and the runtime still seems to depend on F in a non-monotone fashion, see the right subplot of Fig. 2.

The impact of F is visible for all ranges $c < 1$, $c = 1$ and $c > 1$. For $c = 1$ we have only proven efficiency for sufficiently small F. However, we conjecture that there is no real phase transition at $c = 1$, and the 'only' difference is that our proof methods break down at this point. For the fixed s, with increasing c the range of F becomes narrower and restricts to smaller values while larger values of c admit a larger range of values for F.

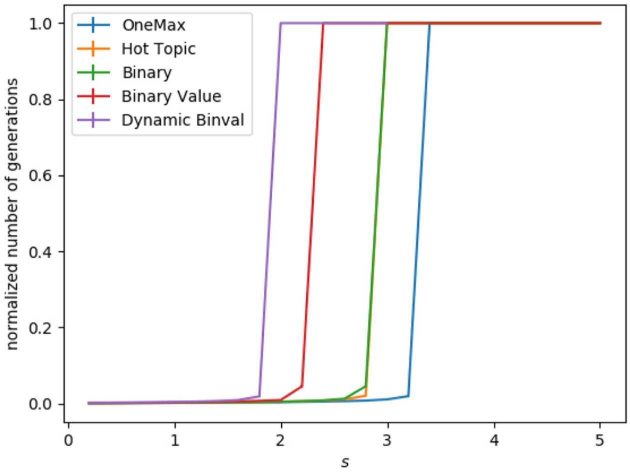

Fig. 1. Average number of generations of the self-adjusting $(1, \lambda)$-EA with $F = 1.5$ and $c = 1$ in 10 runs when optimizing monotone functions with $n = 10000$, normalised and capped at $500n$ generations. HotTopic and Binary behave similarly, so the corresponding curves have a large overlap. The evaluated values of s range from 0.2 to 5 with a step size of 0.2 for all functions except that Dynamic BinVal was not evaluated for $3.2 \leq s \leq 5$ due to performance issues.

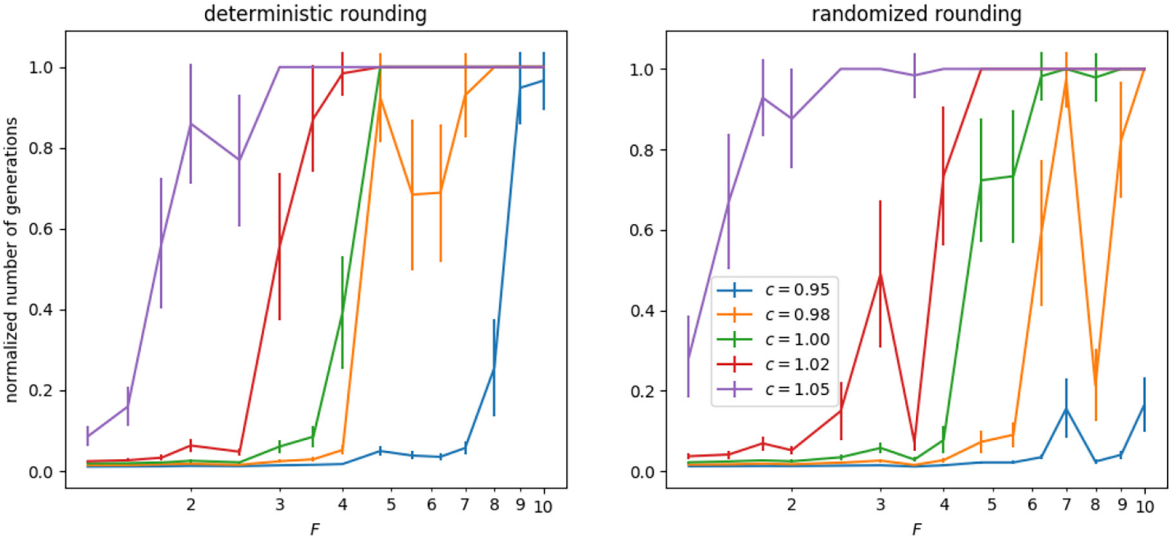

Fig. 2. Average number of generations of the self-adjusting $(1, \lambda)$-EA with $s = 1.8$ and in 50 runs when optimizing Dynamic BinVal with $n = 1000$, normalised and capped at $500n$ generations. The left and right subplots correspond to the deterministic and randomized rounding schemes respectively. The length of each vertical bar is one standard deviation at its corresponding data point.

8. Conclusion

In this paper, we have studied the SA-$(1, \lambda)$-EA on dynamic monotone functions. Hevia Fajardo and Sudholt had shown an extremely strong dependency of the performance on the success rate s for the OneMax benchmark. We have shown that there is nothing specific to OneMax about the situation. The same effect happens for any (static or dynamic) monotone fitness function: for small values of s, the SA-$(1, \lambda)$-EA is efficient on all dynamic monotone functions, while for large values of s, the SA-$(1, \lambda)$-EA is inefficient on every dynamic monotone function. In the latter case, the bottleneck is not around the optimum, but rather in some area of linear distance from the optimum. Thus the SA-$(1, \lambda)$-EA is one of the surprising examples showing that some algorithms may fail in easy fitness landscapes, but succeed in hard fitness landscapes.

Hevia Fajardo and Sudholt have conjectured that the problem becomes worse the easier the fitness landscape is. Concretely, they conjectured that any parameter choice that works for OneMax should also give good result for any other landscape [38]. In a companion paper [60], we disprove this conjecture, but for an unexpected reason: there are different ways to measure 'easiness' of a fitness landscape. While it is theoretically proven that OneMax is the easiest fitness function with respect to decreasing the distance from the optimum [54], this is not the aspect that matters for the SA-$(1, \lambda)$-EA. Here, the important aspect is how easy it is to find a fitness improvement, since this may induce too small target population sizes in the SA-$(1,\lambda)$-EA. For finding fitness improvements, there are easier functions than OneMax, for example the dynamic BinVal function [41] or HotTopic functions [46], see [60] for details. It remains open to determine the easiest dynamic monotone function f_{easiest} with respect to fitness improvements. A candidate for f_{easiest} might be the 'adversarial' Dynamic BinVal, which we define as Dynamic BinVal (see Section 7) with the exception that the permutation is not random but chosen so that any 0-bit is heavier than any 1-bit. With this fitness function, any 0-bit flip gives a fitter child, regardless of the number of 1-bit flips, so it is intuitively convincing that it should be the easiest function with respect to fitness improvement.

Moreover, the conjecture of Hevia Fajardo and Sudholt might still hold if we replace OneMax by f_{easiest}. I.e., is it true that any parameter choice that works for f_{easiest} also works for any other dynamic monotone function, and perhaps even in yet more general settings?

Apart from that, the most puzzling part of the picture is the experimental finding that in a 'middle regime' of success rates, the update strength F seems to play a role in a non-monotone way (for fixed success rate s). It is open to prove theoretically that there is indeed such a 'middle regime' where F plays a role at all. For why this effect is non-monotone in F, we do not even have a good hypothesis. As outlined in Section 7, it does not seem to be a rounding effect. This shows that we are still missing important parts of the overall picture.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

We thank Dirk Sudholt for helpful discussions during the Dagstuhl seminar 22081 "Theory of Randomized Optimization Heuristics", and we thank Mario Hevia Fajardo for sharing his code for comparison.

We also thank the two anonymous reviewers for their valuable input and suggestions. Their feedback contributed to improving the quality of this paper, and we are grateful for their time and effort.

References

[1] M. Kaufmann, M. Larcher, J. Lengler, X. Zou, Self-adjusting population sizes for the $(1,\lambda)$-EA on monotone functions, in: Parallel Problem Solving from Nature, PPSN, Springer, 2022.

[2] A.E. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms, IEEE Trans. Evol. Comput. 3 (1999) 124–141.

[3] J. Jägersküpper, T. Storch, When the plus strategy outperforms the comma strategy and when not, in: Foundations of Computational Intelligence, FOCI, IEEE, 2007, pp. 25–32.

[4] J.E. Rowe, D. Sudholt, The choice of the offspring population size in the $(1,\lambda)$ evolutionary algorithm, Theor. Comput. Sci. 545 (2014) 20–38.

[5] D. Antipov, B. Doerr, Q. Yang, The efficiency threshold for the offspring population size of the (μ,λ) EA, in: Genetic and Evolutionary Computation Conference, GECCO, 2019, pp. 1461–1469.

[6] B. Doerr, T. Jansen, D. Sudholt, C. Winzen, C. Zarges, Optimizing monotone functions can be difficult, in: International Conference on Parallel Problem Solving from Nature, Springer, 2010, pp. 42–51.

[7] B. Doerr, T. Jansen, D. Sudholt, C. Winzen, C. Zarges, Mutation rate matters even when optimizing monotonic functions, Evol. Comput. 21 (1) (2013) 1–27.

[8] J. Lengler, A. Steger, Drift analysis and evolutionary algorithms revisited, Comb. Probab. Comput. 27 (4) (2018) 643–666.

[9] G. Badkobeh, P.K. Lehre, D. Sudholt, Unbiased black-box complexity of parallel search, in: Parallel Problem Solving from Nature, PPSN, Springer, 2014, pp. 892–901.

[10] S. Böttcher, B. Doerr, F. Neumann, Optimal fixed and adaptive mutation rates for the LeadingOnes problem, in: Parallel Problem Solving from Nature, PPSN, Springer, 2010, pp. 1–10.

[11] B. Doerr, C. Doerr, F. Ebel, From black-box complexity to designing new genetic algorithms, Theor. Comput. Sci. 567 (2015) 87–104.

[12] B. Doerr, C. Doerr, J. Yang, Optimal parameter choices via precise black-box analysis, Theor. Comput. Sci. 801 (2020) 1–34.

[13] B. Doerr, C. Witt, J. Yang, Runtime analysis for self-adaptive mutation rates, Algorithmica 83 (4) (2021) 1012–1053.

[14] G. Karafotias, M. Hoogendoorn, Á.E. Eiben, Parameter control in evolutionary algorithms: trends and challenges, IEEE Trans. Evol. Comput. 19 (2) (2014) 167–187.

[15] A. Aleti, I. Moser, A systematic literature review of adaptive parameter control methods for evolutionary algorithms, ACM Comput. Surv. 49 (3) (2016) 1–35.

[16] B. Doerr, C. Doerr, Optimal parameter choices through self-adjustment: applying the 1/5-th rule in discrete settings, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015, pp. 1335–1342.

[17] J. Lässig, D. Sudholt, Adaptive population models for offspring populations and parallel evolutionary algorithms, in: Foundations of Genetic Algorithms, FOGA, 2011, pp. 181–192.

[18] B. Doerr, C. Gießen, C. Witt, J. Yang, The $(1+\lambda)$ evolutionary algorithm with self-adjusting mutation rate, Algorithmica 81 (2) (2019) 593–631.

[19] A. Rajabi, C. Witt, Self-adjusting evolutionary algorithms for multimodal optimization, in: Genetic and Evolutionary Computation Conference, GECCO, 2020, pp. 1314–1322.

[20] B. Doerr, C. Doerr, J. Lengler, Self-adjusting mutation rates with provably optimal success rules, Algorithmica 83 (10) (2021) 3108–3147.

[21] A. Rodionova, K. Antonov, A. Buzdalova, C. Doerr, Offspring population size matters when comparing evolutionary algorithms with self-adjusting mutation rates, in: Genetic and Evolutionary Computation Conference, GECCO, 2019, pp. 855–863.

[22] A. Lissovoi, P.S. Oliveto, J.A. Warwicker, On the time complexity of algorithm selection hyper-heuristics for multimodal optimisation, Proc. AAAI Conf. Artif. Intell. 33 (1) (2019) 2322–2329.

[23] A. Lissovoi, P. Oliveto, J.A. Warwicker, How the duration of the learning period affects the performance of random gradient selection hyper-heuristics, Proc. AAAI Conf. Artif. Intell. 34 (3) (2020) 2376–2383.

[24] A. Lissovoi, P.S. Oliveto, J.A. Warwicker, Simple hyper-heuristics control the neighbourhood size of randomised local search optimally for LeadingOnes, Evol. Comput. 28 (3) (2020) 437–461.

[25] B. Doerr, A. Lissovoi, P.S. Oliveto, J.A. Warwicker, On the runtime analysis of selection hyper-heuristics with adaptive learning periods, in: Genetic and Evolutionary Computation Conference, GECCO, 2018, pp. 1015–1022.

[26] B. Doerr, C. Doerr, T. Kötzing, Static and self-adjusting mutation strengths for multi-valued decision variables, Algorithmica 80 (5) (2018) 1732–1768.

[27] M.A. Hevia Fajardo, D. Sudholt, On the choice of the parameter control mechanism in the (1+(λ, λ)) genetic algorithm, in: Genetic and Evolutionary Computation Conference, GECCO, 2020, pp. 832–840.

[28] A. Mambrini, D. Sudholt, Design and analysis of schemes for adapting migration intervals in parallel evolutionary algorithms, Evol. Comput. 23 (4) (2015) 559–582.

[29] B. Case, P.K. Lehre, Self-adaptation in nonelitist evolutionary algorithms on discrete problems with unknown structure, IEEE Trans. Evol. Comput. 24 (4) (2020) 650–663.

[30] A. Rajabi, C. Witt, Evolutionary algorithms with self-adjusting asymmetric mutation, in: Parallel Problem Solving from Nature, PPSN, Springer, 2020, pp. 664–677.

[31] B. Doerr, C. Doerr, Theory of parameter control for discrete black-box optimization: provable performance gains through dynamic parameter choices, in: Theory of Evolutionary Computation, 2020, pp. 271–321.

[32] M.A. Hevia Fajardo, D. Sudholt, Self-adjusting population sizes for non-elitist evolutionary algorithms: why success rates matter, arXiv preprint, arXiv: 2104.05624, 2021.

[33] S. Kern, S.D. Müller, N. Hansen, D. Büche, J. Ocenasek, P. Koumoutsakos, Learning probability distributions in continuous evolutionary algorithms–a comparative review, Nat. Comput. 3 (1) (2004) 77–112.

[34] I. Rechenberg, Evolutionsstrategien, in: Simulationsmethoden in der Medizin und Biologie, Springer, 1978, pp. 83–114.

[35] L. Devroye, The compound random search, Ph.D. dissertation, Purdue Univ., West Lafayette, IN, 1972.

[36] M. Schumer, K. Steiglitz, Adaptive step size random search, IEEE Trans. Autom. Control 13 (3) (1968) 270–276.

[37] A. Auger, Benchmarking the (1+ 1) evolution strategy with one-fifth success rule on the BBOB-2009 function testbed, in: Genetic and Evolutionary Computation Conference, GECCO, 2009, pp. 2447–2452.

[38] M.A. Hevia Fajardo, D. Sudholt, Self-adjusting population sizes for non-elitist evolutionary algorithms: why success rates matter, in: Genetic and Evolutionary Computation Conference, GECCO, 2021, pp. 1151–1159.

[39] P.K. Lehre, C. Witt, Black-box search by unbiased variation, Algorithmica 64 (2012) 623–642.

[40] J. Lengler, X. Zou, Exponential slowdown for larger populations: the $(\mu+1)$-EA on monotone functions, Theor. Comput. Sci. 875 (2021) 28–51.

[41] J. Lengler, J. Meier, Large population sizes and crossover help in dynamic environments, in: Parallel Problem Solving from Nature, PPSN, Springer, 2020, pp. 610–622.

[42] J. Lengler, S. Riedi, Runtime analysis of the $(\mu+1)$-EA on the dynamic BinVal function, in: Evolutionary Computation in Combinatorial Optimization, EvoCom, Springer, 2021, pp. 84–99.

[43] M.A. Hevia Fajardo, D. Sudholt, Hard problems are easier for success-based parameter control, in: Genetic and Evolutionary Computation Conference, GECCO, 2022, pp. 796–804.

[44] J. Jorritsma, J. Lengler, D. Sudholt, Comma selection outperforms plus selection on onemax with randomly planted optima, in: Genetic and Evolutionary Computation Conference, GECCO, 2023.

[45] J. Lengler, A. Martinsson, A. Steger, When does hillclimbing fail on monotone functions: an entropy compression argument, in: Analytic Algorithmics and Combinatorics, ANALCO, SIAM, 2019, pp. 94–102.

[46] J. Lengler, A general dichotomy of evolutionary algorithms on monotone functions, IEEE Trans. Evol. Comput. 24 (6) (2019) 995–1009.

[47] T. Jansen, On the brittleness of evolutionary algorithms, in: Foundations of Genetic Algorithms, FOGA, Springer, 2007, pp. 54–69.

[48] S. Colin, B. Doerr, G. Férey, Monotonic functions in EC: anything but monotone!, in: Genetic and Evolutionary Computation Conference, GECCO, 2014, pp. 753–760.

[49] B. Doerr, C. Doerr, Optimal static and self-adjusting parameter choices for the $(1+(\lambda,\lambda))$ genetic algorithm, Algorithmica 80 (5) (2018) 1658–1709.

[50] B. Doerr, C. Doerr, T. Kötzing, Provably optimal self-adjusting step sizes for multi-valued decision variables, in: International Conference on Parallel Problem Solving from Nature, Springer, 2016, pp. 782–791.

[51] T. Kötzing, Concentration of first hitting times under additive drift, Algorithmica 75 (3) (2016) 490–506.

[52] J. Lengler, Drift analysis, in: Theory of Evolutionary Computation, Springer, 2020, pp. 89–131.

[53] J. Lengler, U. Schaller, The $(1+1)$-EA on noisy linear functions with random positive weights, in: Symposium Series on Computational Intelligence, SSCI, IEEE, 2018, pp. 712–719.

[54] B. Doerr, D. Johannsen, C. Winzen, Multiplicative drift analysis, Algorithmica 64 (2012) 673–697.

[55] P.S. Oliveto, C. Witt, Improved time complexity analysis of the simple genetic algorithm, Theor. Comput. Sci. 605 (2015) 21–41.

[56] B. Doerr, L.A. Goldberg, Drift analysis with tail bounds, in: Parallel Problem Solving from Nature, PPSN, Springer, 2010, pp. 174–183.

[57] G.R. Grimmett, et al., Percolation, vol. 321, Springer Science & Business Media, 1999.

[58] B. Doerr, Probabilistic tools for the analysis of randomized optimization heuristics, in: Theory of Evolutionary Computation, Springer, 2020, pp. 1–87.

[59] B. Doerr, M. Künnemann, Optimizing linear functions with the (1+ λ) evolutionary algorithm—different asymptotic runtimes for different instances, Theor. Comput. Sci. 561 (2015) 3–23.

[60] M. Kaufmann, M. Larcher, J. Lengler, X. Zou, OneMax is not the easiest function for fitness improvements, https://arxiv.org/abs/2204.07017, 2022.