Algorithmica (2025) 87:1804–1863
https://doi.org/10.1007/s00453-025-01330-y

Comma Selection Outperforms Plus Selection on OneMax with Randomly Planted Optima

Joost Jorritsma[1] · **Johannes Lengler**[2] · **Dirk Sudholt**[3]

Received: 12 June 2023 / Accepted: 9 June 2025 / Published online: 18 August 2025

Abstract

Evolutionary algorithms (EAs) are general-purpose optimisation algorithms that maintain a population (multiset) of candidate solutions and apply variation operators to create new solutions called offspring. A new population is typically formed using one of two strategies: a $(\mu + \lambda)$ EA (plus selection) keeps the best μ search points out of the union of μ parents in the old population and λ offspring, whereas a (μ, λ) EA (comma selection) discards all parents and only keeps the best μ out of λ offspring. Comma selection may help to escape from local optima, however when and how it is beneficial is subject to an ongoing debate. We propose a new benchmark function to investigate the benefits of comma selection: the well known benchmark function OneMaxwith randomly planted local optima, generated by frozen noise. We show that comma selection (the $(1, \lambda)$ EA) is faster than plus selection (the $(1 + \lambda)$ EA) on this benchmark, in a fixed-target scenario, and for offspring population sizes λ for which both algorithms behave differently. For certain parameters, the $(1, \lambda)$ EAfinds the target in $\Theta(n \ln n)$ evaluations, with high probability (w.h.p.), while the $(1 + \lambda)$ EAw.h.p. requires $\omega(n^2)$ evaluations. We further show that the advantage of comma selection is not arbitrarily large: w.h.p. comma selection outperforms plus selection at most by a factor of $O(n \ln n)$ for most reasonable parameter choices. We develop novel methods for analysing frozen noise and give powerful and general fixed-target results with tail bounds that are of independent interest.

Keywords Runtime analysis · Non-elitism · Comma strategies · Fixed-target running times · Drift analysis · Multimodal optimisation

Joost Jorritsma, Johannes Lengler and Dirk Sudholt contributed equally to this work.

An extended abstract of this work was published in [33]. It announced the results presented here, but mostly without proofs. This comprehensive and improved manuscript contains all proofs, including lemmas not stated in [33] that may be of independent interest.

Extended author information available on the last page of the article

1 Introduction

Evolutionary Algorithms (EAs) are general-purpose optimisation heuristics that borrow ideas from the natural evolution of species. They maintain a *population* (multiset) of candidate solutions and apply variation operators such as mutation and/or crossover to generate new solutions called *offspring*. Then a process of selection is used to generate a new population from search points in the current population, referred to as *parents*, and the newly created offspring. Selection is based on the objective values of candidate solutions, given by a *fitness function* and the objective value of a search point is referred to as *fitness value*. Selection strategies include $(\mu + \lambda)$-selection (*plus selection*) in which the population contains μ search points, λ offspring are being created and then the new population is formed from picking the μ best individuals from the union of μ parents and λ offspring. Plus selection ensures that the best fitness value seen so far is being maintained in the population, a concept known as *elitism*.[1] In contrast, (μ, λ)-selection (*comma selection*) discards the parents and selects the μ best search points exclusively from the set of λ offspring.

Choosing an appropriate selection strategy is an important aspect that can have a drastic impact on performance. While elitism is helpful to exploit the best-so-far solution, a common concern is that the algorithm might get stuck in local optima. The *escape hypothesis* posits that non-elitism might help in such cases [2]. Indeed, the disadvantage of elitism can be measured by the *elitist black-box complexity* of a problem, and some (artificial) problems show an exponential penalty of elitism [2–3].

Comma selection is a popular non-elitist selection strategy as the parent(s) are not allowed to compete for survival and the best offspring may be worse than the best parent. Despite its popularity, it is still unclear how good comma selection is at helping with escaping local optima. Some deceptive landscapes, in which other non-elitist mechanisms can help, can still deceive comma selection [3]. A working group at a 2022 Dagstuhl seminar "came to the conclusion that there is a gap between theory and practice as we are lacking convincing examples (apart from Cliff) where comma selection provably helps, whereas in practice comma strategies seem to be quite popular to escape from local optima" [4].

The main theoretical results for comma strategies at local optima are for the benchmarks Jump and Cliff. It is known that comma selection in the (μ, λ) EA does not work better than plus selection on the Jump function, for arbitrary population sizes μ and λ [5]. For the Cliff function, there is a choice of λ such that the $(1, \lambda)$ EA "only" takes time $O(n^{3.97\cdots})$, while the plus strategy needs exponential time for any λ [7–8]. However, the optimisation time of the $(1, \lambda)$ EA is still rather high (albeit polynomial), and the dependence on λ is rather tricky. There are some promising efforts to develop self-adjusting mechanisms that can adapt λ during the run of the algorithm, but these come with their own pitfalls [9–11]. In particular, the optimisation time for Cliff can be reduced to $O(n \ln n)$ by a self-adjusting mechanism which resets λ periodically, but this mechanism needs to be well-aligned with the problem [8].

[1] Some authors define elitism in a stronger way, so that the whole population must consist of the best-so-far search points [1].

Arguably, Cliff (and also Jump) captures a rather specific situation that might be atypical for local optima. We will not give the definition of Cliff, but only describe the atypical situation. When the algorithm is in a local optimum x, and accepts a Hamming neighbour y that is closer to the optimum ("down the cliff"), then *most neighbours of y have the same fitness as x* and thus are "back up on the cliff". In particular, a random walk without selective pressure would likely lead back to the local optimum (or an equivalent search point). It is rather hard to imagine this situation in practice, where at or near an optimum (global or local), random walks typically increase the distance from the optimum and decrease fitness. This is not just a coincidental feature of the Cliff function; the analysis of this phenomenon is at the heart of all runtime analyses of Cliff. Thus, Cliff(and Jump) might not be the best test function for understanding local optima.

One issue with the $(1, \lambda)$ EAis that one needs to find a compromise between two trends: if λ is too large, then every generation contains a clone of the parent, so the $(1, \lambda)$ EAjust imitates an elitist algorithm. But if λ is too small then the algorithm can not cope with situations in which improvements are hard to find. If the goal is to find a global optimum, then it is hard to balance those two aspects. However, it is less problematic in fixed-target optimisation, which is why we study this setting. We will discuss this point in more detail in Sect. 1.3.

1.1 Distorted OneMax

We introduce an alternative model of local optima, which we call distorted OneMax or $\text{DISOM} = \text{DISOM}_{d,p}$ for short. It could also be called "OneMaxwith planted local optima" or "OneMaxwith frozen Bernoulli noise". It is based on the popular benchmark function $\text{ONEMAX}(x) := \sum_{i=1}^{n} x_i$ that simply counts the number of ones in a bit string $x = (x_1, x_2, \ldots, x_n)$ of length n. We start with all search points being assigned their OneMaxfunction value and introduce two real-valued parameters $p \in [0, 1]$ and $d > 0$. Then for each search point x, with probability p we increase its fitness by d, independently of the other search points. Hence, we artificially "plant" a potential local optimum in x. (It does not always need to be a local optimum since a fitter neighbour of x could also be distorted.) Note that the distortion is part of the fitness function, and thus the fitness function is a static function. Hence, if an algorithm evaluates the same search point x several times, it will always detect the same fitness. This is different from models with noisy fitness evaluations, in which several queries on the same search point can give different fitness values.

The runtime analysis of evolutionary algorithms on stochastic problems is a rapidly emerging research area, with results on prior noise (altering the genotype before the fitness evaluation) [12], posterior noise (returning a fitness value obscured by noise) [13] and partial evaluation of search points [14]. Analyses were performed for simple (1+1) EAs and (1+λ) EAs [12, 16–18], population-based algorithms [13, 19], including non-elitist algorithms [20], ACO [22–23], estimation-of-distribution algorithms [24] and, most recently, the famous NSGA-II algorithm [25, 26].[2] All the cited papers consider *dynamic* noise in which each query gives an independent noisy

[2] The paper [25] used the noise model presented in our preliminary work [33].

fitness. Before our work, the only other theoretical analysis involving *frozen* noise we are aware of is [27]. In this paper, a frozen noise model was studied for the compact Genetic Algorithm cGA on OneMaxand Gaussian noise was added to all search points. Our model is different as it adds Bernoulli noise, that is, the added noise is either 0 or d. During the reviewing process of this manuscript, the disOMbenchmark has been picked up in [28], where it was embedded into a hierarchy of benchmark functions, and in [29], where it was shown that plus strategies are slowed down much more dramatically if the amount of distortion d is drawn randomly for each distorted point.

1.2 Main Results

We study the fixed-target performance, that is, the random time needed to evolve a solution of at least some given target fitness [30], with fitness target $n - k^*$ of the $(1, \lambda)$ EAand the $(1 + \lambda)$ EAon distorted OneMaxwith parameters $p \in [0, 1]$ and $d > 1$. We will explain those choices in more detail in Sect. 1.3 below. We denote by $T^{\text{comma}} = T^{\text{comma}}_{k^*,\lambda,d,p}$ and $T^{\text{plus}} = T^{\text{plus}}_{k^*,\lambda,d,p}$ the number of function evaluations until the $(1, \lambda)$ EAand the $(1 + \lambda)$ EAfind a search point of fitness at least $n - k^*$ on $\text{DISOM}_{d,p}$, respectively.

We give matching upper and lower bounds on T^{comma} and also on T^{plus} in Theorem 1.1 below for a wide range of parameters k^*, λ, d, p, which hold with high probability.[3] For those parameters T^{plus} is by a factor $1/p$ larger than T^{comma}. In particular, we show that there are parameters for which comma selection reduces the runtime from nearly quadratic to quasi-linear. Since the assumptions on the parameters are a bit technical, we state them together with a discussion in Assumption 1.4 in Sect. 1.3 below. Intuitively, the parameters λ and k^* must be chosen such that the $(1, \lambda)$ EAefficiently reaches fitness target $n - k^*$, but it does not create a clone of the parent in each generation. We state our main theorem.

Theorem 1.1 *Under Assumption 1.4 on the parameters below, with high probability*

$$T^{\text{comma}}_{k^*,\lambda,d,p} = \Theta\big(n \ln n\big), \tag{1}$$

$$T^{\text{plus}}_{k^*,\lambda,d,p} = \Theta\big(\tfrac{n \ln n}{p}\big). \tag{2}$$

For any p such that $p = \omega(1/(n \ln n))$ and $p = n^{-\Omega(1)}$ there are k^*, λ and d such that Assumption 1.4 is satisfied. Thus, (1) and (2) may differ by a factor of almost $n \ln n$.

It may seem like an unimportant quirk that we have used a w.h.p. statement instead of expectations, but this is not so. Indeed, the distorted OneMax function easily leads to regimes in which expectations are completely meaningless, since they are dominated by events of tiny probability which contribute gigantic terms to the

[3] *With high probability* (w.h.p.) means with probability $1 - o(1)$ as $n \to \infty$.

expectation. This is a known phenomenon, see [1] for an in-depth discussion in the context of elitist black-box complexity. In our situation, we give the following proposition as example. We remark that those parameters are outside of the regimes of Assumption 1.4.

Proposition 1.2 *For $p = 2^{-n}$, $d = n - 0.5$, $k^* = 0$ and $\lambda = 3 \ln n$,*

$$\mathbb{E}\big[T^{\text{comma}}_{k^*,\lambda,d,p}\big] = O(n \ln n) \quad \textit{and} \quad \mathbb{E}\big[T^{\text{plus}}_{k^*,\lambda,d,p}\big] = n^{\Omega(n)}, \tag{3}$$

but with high probability

$$T^{\text{comma}}_{k^*,\lambda,d,p} = O(n \ln n) \quad \textit{and} \quad T^{\text{plus}}_{k^*,\lambda,d,p} = O(n \ln n). \tag{4}$$

Following the terminology of [1] for black-box complexity, we also call the expected times in (3) the *Las Vegas runtimes*, and for $r \in [0, 1]$ we call the *r-Monte Carlo runtime* the time until the algorithm finds the target with probability at least $1 - r$. Proposition 1.2 states that the Las Vegas runtime and the *r*-Monte Carlo runtime (for any constant *r*) differ dramatically for the $(1 + \lambda)$ EA. In general, Monte Carlo runtimes are more informative since they make a statement about *typical* outcomes.[4]

Theorem 1.1 gives a factor $1/p$ between the comma and the plus strategy that is arbitrarily close to $n \ln n$. The next theorem shows that this is the largest possible factor for Monte Carlo runtimes if the $(1, \lambda)$ EA is efficient on DISOM. Note that the factor for Las Vegas runtimes can be huge by Proposition 1.2.

Theorem 1.3 *Let $C, \varepsilon > 0$ be constants. Assume that the parameters $\lambda \geq 1$, $k^* \in [n^\varepsilon, n/6]$, $p \in [0, 1]$, and $T \in [1, n^C]$ (all possibly depending on n) are such that with high probability*

$$T^{\text{comma}}_{k^*,\lambda,d,p} \leq T. \tag{5}$$

Then there exists a constant $C' > 0$ such that with high probability

$$T^{\text{plus}}_{k^*,\lambda,d,p} \leq \begin{cases} T, & \textit{if } p \cdot T = o(1) \textit{or} \lambda > C' \ln(n), \\ O\big(n(\lambda + \ln(\frac{n}{k^*}))\big), & \textit{if } p \cdot n(\lambda + \ln(\frac{n}{k^*})) = o(1), \\ O\big(T/p\big), & \textit{otherwise, provided that} \lambda = O(\ln(\frac{n}{k^*})). \end{cases}$$

Moreover, in all three cases $T^{\text{plus}} = O(T \cdot n \ln n)$ w.h.p.

[4] Usually, there is another reason to prefer Monte Carlo runtimes, since with good Monte-Carlo runtimes we can restart the algorithm if a run gets stuck. However, the situation here is a bit more subtle, since DISOMis a randomized function. Thus, there are two forms of randomness: one from the random choice of the fitness function, and one from the random decision of the algorithms. If the long expected runtime comes from an atypical fitness function, the problem is not solved by restarting the algorithm. However, it is easy to find a *fixed* function for which Proposition 1.2 still holds, for example the ONEMAXfunction with a single planted local optimum of value $n - 1$ at the all-zero string. This is implicitly shown in the proof of Proposition 1.2.

We believe that the condition on λ in the third case is not needed, but keep it to avoid even more technical complexity. We emphasize that we do not make Assumption 1.4 here. In fact, the conditions in Theorem 1.3 are much more general than Assumption 1.4 and also cover many degenerate parameter settings. For example, for small λ the $(1,\lambda)$ EAmay be inefficient and not have runtime $\Theta(n \ln n)$. We caution that Theorem 1.3 is far from trivial. We do make heavy use of the condition $k^* \geq n^\varepsilon$ in the proof. Moreover, we believe that the statement would be wrong if the $(1+\lambda)$ EAwould break fitness ties in favour of the parent. We discuss those matters in more detail in Sect. 6.

1.3 Parameter Setup

We will now explain which regimes are reasonable to consider for the parameters k^*, λ, d, p. Note that all of $d = d(n)$, $k^* = k^*(n)$, $p = p(n)$ and $\lambda = \lambda(n)$ may depend on n.

We will use the abbreviation $\eta := e/(e-1)$. This is helpful since the probability that a mutation is not identical to the parent (is not a *clone*) is $\approx 1 - 1/e = \eta^{-1}$. We write $q{:=}\eta^{-\lambda}$ for the (approximate) probability to have no clone in the offspring population. This is a central quantity, since it is the probability of escaping from a local optimum in the $(1,\lambda)$ EA. It has been known for decades that if $\lambda \geq C \ln n$ for a large C, then the $(1,\lambda)$ EAmimics an elitist algorithm because then $q = n^{-C\cdot\ln(\eta)} \approx n^{-0.46C}$ [6, 31]. For example, if $C \geq 5$ then $q = o(n^{-2})$ and w.h.p. the parent will be cloned in all of the first n^2 generations. Hence, the $(1,\lambda)$ EAjust behaves as the $(1+\lambda)$ EAin this regime. Since we are interested in potential differences between those two algorithms, we will thus consider regimes where $q \geq n^{-1+\varepsilon}$, or equivalently $\lambda \leq (1-\varepsilon)\log_\eta n$, since this is the regime where the two algorithms behave differently [31]. Note that this choice is not wise when the algorithm is supposed to find the optimum, since the $(1,\lambda)$ EAwith $\lambda \leq (1-\varepsilon)\log_\eta n$ is inefficient in finding the optimum of any function with unique optimum [7]. However, such a λ is fine for fixed-target optimisation, i.e., if we are interested in finding a search point of fitness at least $n - k^*$, as long as $\lambda \geq (1+\varepsilon)\log_\eta(n/k^*)$ [32], or equivalently $q \leq (k^*/n)^{1+\varepsilon}$. Thus, we require $(1+\varepsilon)\log_\eta(n/k^*) \leq \lambda \leq (1-\varepsilon)\log_\eta n$ for some constant $\varepsilon > 0$. This range is non-empty if $k^* = n^{\Omega(1)}$, so we will make this restriction. This also avoids some complications at the optimum, since the optimum of DISOMmay be not at $\vec{1} = (1,\dots,1)$, but at a distorted point whose fitness exceeds n. For this reason we will assume $d \leq k^*$, so that the target fitness $n - k^*$ cannot be achieved by search points at distance larger than $2k^*$ from $\vec{1}$. Finally, for simplicity we will also assume $k^* = n^{1-\Omega(1)}$, which implies $\lambda = \Theta(\ln n)$.

We argue that considering a fixed-target scenario is of interest since it reveals search dynamics and benefits of non-elitism that were previously hidden from view. Previous work on the $(1,\lambda)$ EA only considered global optimisation and required offspring population sizes λ that only permitted very few fitness-decreasing steps. The range of λ values identified in this work admits new insights on the $(1,\lambda)$ EA that demonstrate the benefits of non-elitism in a regime where fitness-decreasing steps occur frequently.

The $(1+\lambda)$ EAwith $\lambda = O(\ln n)$ needs $O(n \ln n)$ fitness evaluations to optimize OneMax. If $p = o\left(\frac{1}{n \ln n}\right)$, then w.h.p. the $(1+\lambda)$ EAwill not encounter any distorted search points before finding the optimum. Thus, we may ignore the case $p = o\left(\frac{1}{n \ln n}\right)$. Finally, we will make two more assumptions for technical simplicity. Firstly, we require $p = o(k^*/n)$. This ensures that close to the target fitness $n - k^*$, the probability $\Theta(k^*/n)$ that an offspring is closer to $(1, \dots, 1)$ dominates the probability p that the offspring is distorted. The regime $p = \omega(k^*/n)$ is rather different because even if the comma strategy escapes from a local maximum into a non-distorted point, it will likely return to a distorted point before having the chance to make an improvement. We leave the study of this regime for future work.

Secondly, we assume that $q = \omega(p\lambda)$, or equivalently $p = o\left(\frac{1}{\lambda \eta^\lambda}\right)$. Note that $p\lambda$ is roughly the probability of sampling a distorted offspring in one generation, while q is the probability of escaping a local optimum in the $(1, \lambda)$ EA. Thus, we assume that escaping is to be more likely than sampling another local optimum. This condition simplifies the analysis in some places, but we don't believe it is actually needed, and we hope that we can remove it in future work. In fact, we will require the slightly stronger condition $q \geq p^{1-\varepsilon}$, or equivalently $\lambda \leq (1-\varepsilon)\log_\eta(1/p)$. This is stronger than $q = \omega(p\lambda)$ since the other conditions already imply $p = n^{-\Omega(1)}$ and $\lambda = \Theta(\ln n)$. Summarizing, we will make the following assumption for our main theorem, Theorem 1.1 above.

Assumption 1.4 Let $q := \eta^{-\lambda}$ for $\eta := e/(e-1)$, and let $\varepsilon > 0$ be any constant. We assume $k^* = n^{\Omega(1)}$ and $k^* = n^{1-\Omega(1)}$, $p = \omega(\frac{1}{n \ln n})$, and

$$p^{1-\varepsilon} \leq q \leq (k^*/n)^{1+\varepsilon}. \tag{6}$$

Finally, we assume that $d \in [(1+\varepsilon)\frac{\ln(n/p)}{\ln(n/k^*)}, k^*]$.

We have already motivated the assumptions on p, k^* and λ, and the upper bound on d. The lower bound on d will come out of the proof of the lower bound on T^{plus}, see Equation (35) in that proof. By the assumptions on k^* and p we have $\ln(n/p) = \Theta(\ln n)$ and $\ln(n/k^*) = \Theta(\ln n)$, so the lower bound on d is just a constant.

Note that we can write (6) equivalently as a condition on λ:

$$(1+\varepsilon)\log_\eta(n/k^*) \leq \lambda \leq (1-\varepsilon)\log_\eta(1/p), \qquad \text{so} \qquad \lambda = \Theta(\ln n). \tag{7}$$

Together with (6), Assumption 1.4 implies $q = n^{-\Theta(1)}$, and thus for some $\delta > 0$,

$$p \leq q^{1+\varepsilon/(1-\varepsilon)} \leq qn^{-\delta} \leq n^{-2\delta}, \quad \frac{q}{\lambda p} \geq \frac{p^{-\varepsilon}}{\lambda} \geq n^{2\delta\varepsilon}, \quad \text{and} \quad p \leq q \leq k^*/n^{1+\delta}. \tag{8}$$

We discuss briefly the possible ranges of the parameters. The values of p, q, and k^*/n are coupled by (6), but p can be arbitrarily close to $\frac{1}{n \ln n}$ and k^* can take a value n^c for a constant $c < 1$ that is arbitrarily close to 1. Hence, for any constant $0 < c < 1$ we may set any one of the three values p, q, or k^*/n to n^{-c} and still satisfy Assump-

tion 1.4 by choosing the other two values appropriately. As discussed above, values of p or q which are much smaller than $\frac{1}{n \ln n}$ do not lead to interesting regimes, because respectively the algorithm does not encounter distorted points or it mimics a plus strategy. The restrictions on q always determine λ up to constant factors, where the interval may be more or less narrow depending on p and k^*. This discussion also implies the second statement of Theorem 1.1, that we may choose parameters yielding a factor of more than n between T^{comma} and T^{plus}.

Proof (*Parameters yielding quasi-linear factor in Theorem* 1.1) Let $0 < \delta \leq 1$ and $p = p(n)$ such that $p \leq n^{-\delta}$ and $p = \omega\left(\frac{1}{n \ln n}\right)$. Then Assumption 1.4 is satisfied by choosing $\lambda := \left\lceil \frac{\delta}{2} \log_\eta n \right\rceil$ such that $q = \eta^{-\lambda} = \Theta(n^{-\delta/2})$ and setting $k^* := n^{1-\delta/4}$ and d as a sufficiently large constant. □

1.4 Structure of the Paper

The structure of the paper is as follows. In Sect. 2 we give a formal description of the setup. Section 3 contains tools: after some basic probabilistic bounds we give domination results comparing plus with comma strategies, different population sizes, and OneMaxwith other functions. Then we derive high-probability fixed-target runtime bounds, which hold for arbitrary fitness functions and which are tight on OneMax. Both the domination results and the fixed-target runtime bounds are very general and may be of independent interest.

In Sect. 4 we show that the comma strategy is not substantially slowed down by the distorted points. We do this by defining a suitable potential function and carefully investigating its drift. A main obstacle is that due to the frozen noise, we do not get fresh randomness in each generation, i.e., we can not just assume that an offspring is distorted with probability p. We solve this problem by showing that w.h.p. no distorted search point is evaluated twice. The beginning of Sect. 4 contains a more detailed overview.

In Sect. 5 we prove the lower runtime bound for the plus strategy. In Sect. 6 we give an upper runtime bound, which in particular implies Theorem 1.3. The upper bound is surprisingly complicated, and again this is due to the frozen noise and the dependencies that come with it. While this was a mere complication for the comma strategy, here it could change the process substantially, and it is only mitigated because the algorithm moves quickly on fitness plateaus of distorted points of equal fitness. We conjecture that the runtime would be much higher if fitness ties were broken in favour of the parent, see the beginning of Sect. 6 for a more thorough discussion.

Finally, in Sect. 7 we show how our analysis implies Theorems 1.1 and Proposition 1.2, before we conclude with some open problems in Sect. 8. Note that an extended abstract of this work was published in [33].

2 Notation and Preliminaries

2.1 General Notation

We write $[n] := \{1, \dots, n\}$. We denote search points by $x = (x_1, \dots, x_n) \in \{0,1\}^n$, and the OneMaxvalue of x is $\mathrm{OM}(x) := \sum_{i \in [n]} x_i$. We denote by $\vec{1} = (1, \dots, 1)$ and $\vec{0} = (0, \dots, 0)$ the unique search points with $\mathrm{OM}(\vec{1}) = n$ and $\mathrm{OM}(\vec{0}) = 0$. For $x, y \in \{0,1\}^n$, the *Hamming distance* $H(x, y)$ of x and y is the number of positions $i \in [n]$ such that $x_i \neq y_i$. We call y a (Hamming) *neighbour* of x if $H(x, y) = 1$. We set $\eta := e/(e-1)$, and we denote by $\log_2 n$ the binary logarithm of n, by $\ln n$ the natural logarithm of n, and by $\log_\eta n := \ln n / \ln \eta$ the logarithm with base η.

For an event $\mathcal{E}$ we denote by $\mathbb{1}\{\mathcal{E}\}$ the indicator variable of $\mathcal{E}$, i.e., $\mathbb{1}\{\mathcal{E}\} = 1$ if $\mathcal{E}$ occurs and $\mathbb{1}\{\mathcal{E}\} = 0$ otherwise.

2.2 Distorted OneMax

We start with a formal definition of the DisOneMax function $\mathrm{DisOM} : \{0,1\}^n \to \mathbb{R}_{\geq 0}$. It comes with two parameters: $p \in [0,1]$ is the probability of a distortion and $d \in \mathbb{R}_{>0}$ indicates the magnitude of the distortion. We partition the search space $\{0,1\}^n$ into two sets $\mathcal{C}$ and $\mathcal{D}$ of "clean" and "distorted" points, respectively, where for each $x \in \{0,1\}^n$ we have

$$x \in \mathcal{D} \text{ with probability } p, \qquad x \in \mathcal{C} \text{ otherwise,} \tag{9}$$

independently of the other points. We define the DisOneMax function $\mathrm{DisOM} = \mathrm{DisOM}_{d,p}$ as

$$\mathrm{DisOM}(x) := \mathrm{OM}(x) + d \cdot \mathbb{1}\{x \in \mathcal{D}\}. \tag{10}$$

2.3 Algorithms

In Algorithms 1 and 2 we give the pseudocode for fixed-target optimisation of a fitness function f with the $(1, \lambda)$ EAand the $(1 + \lambda)$ EArespectively. Both algorithms create offspring using *standard bit mutation:* each bit is flipped independently with probability $1/n$. Unless mentioned otherwise, we assume that the mutation rate is $1/n$. The *running time* is the number of function evaluations until the target is met. Note that in our context of DisOM, the target will always be to reach fitness at least $n - k^*$.

Algorithm 1 $(1, \lambda)$ EA for maximizing f to target $n - k^*$.

Initialization: $t = 0$; pick $x^{(0)}$ uniformly at random from $\{0,1\}^n$.
Optimization: **while** $f(x^{(t)}) \leq n - k^*$ **do**
  Mutation: **for** $j \in \{1, \ldots, \lambda\}$ **do**
    $y^{(t,j)} \leftarrow \text{mutate}(x^{(t)})$ by flipping each bit of $x^{(t)}$ independently with probability $1/n$;
  Selection: Let $y^{(t)} = \arg\max\{f(y^{(t,1)}), \ldots, f(y^{(t,\lambda)})\}$, breaking ties uniformly at random;
  Update: $x^{(t+1)} = y^{(t)}$; $t \leftarrow t + 1$;

Algorithm 2 $(1 + \lambda)$ EA for maximizing f to target $n - k^*$.

Initialization: $t = 0$; pick $x^{(0)}$ uniformly at random from $\{0,1\}^n$.
Optimization: **while** $f(x^{(t)}) \leq n - k^*$ **do**
  Mutation: **for** $j \in \{1, \ldots, \lambda\}$ **do**
    $y^{(t,j)} \leftarrow \text{mutate}(x^{(t)})$ by flipping each bit of $x^{(t)}$ independently with probability $1/n$;
  Selection: Let $y^{(t)} = \arg\max\{f(y^{(t,1)}), \ldots, f(y^{(t,\lambda)})\}$, breaking ties uniformly at random;
  Update: **if** $f(y^{(t)}) \geq f(x^{(t)})$ **then** $x^{(t+1)} = y^{(t)}$; **else** $x^{(t+1)} = x^{(t)}$;
    $t \leftarrow t + 1$;

We recall that $T^{\text{comma}} = T^{\text{comma}}_{k^*,\lambda,d,p}$ and $T^{\text{plus}} = T^{\text{plus}}_{k^*,\lambda,d,p}$ are the number of function evaluations until the $(1, \lambda)$ EAand the $(1 + \lambda)$ EArespectively find a search point of fitness at least $n - k^*$ on $\text{DISOM}_{d,p}$. By the *time* that an algorithm spends, we refer to the number of function evaluations in all generations up to the first generation in which the target fitness is reached for the first time. So when the algorithm runs for time T, then it runs for $\lceil T/\lambda \rceil$ generations.

2.4 Drift Theorems

Drift analysis turns bounds on the expected progress towards a target state into bounds on the expected time for hitting said target [34]. We will use the variable drift theorem [35] as stated in [34, Theorem 2.3.3] and the multiplicative drift theorem with tail bounds [36] as stated in [34, Theorem 2.4.5].

Theorem 2.1 *(Variable Drift Theorem) Let $(X_t)_{t \geq 0}$ be a sequence of non-negative random variables with a finite state space $\mathcal{S} \subseteq \mathbb{R}^+_0$ such that $0 \in \mathcal{S}$. Let $s_{\min} := \min(\mathcal{S} \backslash \{0\})$, let $T := \inf\{t \geq 0 \mid X_t = 0\}$, and for $t \geq 0$ and $s \in \mathcal{S}$ let $\Delta_t(s) := \mathbb{E}[X_t - X_{t+1} \mid X_t = s]$. If there is an increasing function $h : \mathbb{R}^+ \to \mathbb{R}^+$ such that for all $s \in \mathcal{S} \backslash \{0\}$ and all $t \geq 0$*

$$\Delta_t(s) \geq h(s),$$

then

$$\mathbb{E}[T] \leq \frac{s_{\min}}{h(s_{\min})} + E\left[\int_{s_{\min}}^{X_0} \frac{1}{h(\sigma)}\,\mathrm{d}\sigma\right],$$

where the expectation in the latter term is over the random choice of X_0.

Theorem 2.2 *(Multiplicative drift theorem with tail bounds) Let $(X_t)_{t\geq 0}$ be a sequence of non-negative random variables with a finite state space $\mathcal{S} \subseteq \mathbb{R}_0^+$ such that $0 \in \mathcal{S}$. Let $s_{\min} := \min(\mathcal{S}\backslash\{0\})$, and let $T := \inf\{t \geq 0 \mid X_t = 0\}$. Suppose that $X_0 = s_0$, and that there exists $\delta > 0$ such that, for all $s \in \mathcal{S}\backslash\{0\}$ and all $t \geq 0$,*

$$\mathbb{E}[X_t - X_{t+1} \mid X_t = s] \geq \delta s.$$

Then, for all $r \geq 0$,

$$\mathbb{P}\left(T > \left\lceil \frac{r + \ln(s_0/s_{\min})}{\delta} \right\rceil\right) \leq e^{-r}.$$

We also provide the following lower bound on the drift of the $(1,\lambda)$ EAon OneMax. It is extracted from the proof of [37, Theorem 4.9], which is a refinement of [7, Theorem 9].

Theorem 2.3 *([37]) On OneMax, the drift of the $(1,\lambda)$ EAat Hamming distance k from the optimum, if $k+1 \geq 9n\left(\frac{e}{e-1}\right)^{-\lambda}/\lambda$ and $\lambda \geq 9$, is at least*

$$\begin{cases} \frac{3-e}{6}, & if\ \lambda\ k\ \geq (3-e)n, \\ \frac{\lambda k}{n} \cdot \frac{3-e}{9}, & otherwise. \end{cases} \tag{11}$$

2.5 Concentration Bounds

We will use the following concentration bounds. First we give the Chernoff bound for the sum of independent Bernoulli random variables [38, Theorem 1.10.21 and 1.10.5].

Theorem 2.4 *(Chernoff bound) Let $X_1, \ldots, X_n$ be independent Bernoulli random variables and $X := \sum_{i=1}^n X_i$. If $\mu \geq \mathbb{E}[X]$, then for all $\delta \geq 0$,*

$$\mathbb{P}(X \geq (1+\delta)\mu) \leq \exp\left(-\frac{\min\{\delta, \delta^2\}}{3}\mu\right).$$

If $\mu \leq \mathbb{E}[X]$, then for all $\delta \geq 0$

$$\mathbb{P}\big(X \leq (1-\delta)\mu\big) \leq \exp\Big(-\frac{\delta^2}{2}\mu\Big).$$

Next we give a version for the sum of independent *geometric* random variables [39, Theorem 2.3].

Theorem 2.5 *Let $X_1, \dots, X_n$ be independent geometric random variables with success probabilities $p_1, \dots, p_n > 0$. Let $p_{\min} := \min\{p_i \mid i \in [1..n]\}$. Let $X := \sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[X] = \sum_{i=1}^n 1/p_i$. For all $\delta \geq 0$,*

$$\mathbb{P}(X \geq (1+\delta)\mu) \leq \frac{1}{1+\delta}(1-p_{\min})^{\mu(\delta-\ln(1+\delta))}.$$

3 General Tools

In this section we collect some lemmas which we need for our results, but which may be useful in other contexts as well. Section 3.1 collects basics about the $(1,\lambda)$ EA, in particular bounds on the OneMax-drift and on the probability of producing a clone. In Sect. 3.2, we prove that the $(1+\lambda)$ EAis the most efficient algorithm on OneMax(in the sense of stochastic domination) among all algorithms which create their offspring in batches of size λ (which will be formalised in Theorem 3.5) via standard bit mutation with the same fixed mutation rate. This also holds in fixed-target settings. As a corollary, we obtain that the $(1+\lambda)$ EAon OneMaxis the fastest algorithm to reach a fixed Hamming distance from the optimum among all $(1+\lambda)$ algorithms and all fitness functions with a unique global optimum. In Sect. 3.3 we show high-probability upper and lower runtime bounds for the $(1+\lambda)$ EAand $(1,\lambda)$ EAon OneMaxwith prescribed start and target fitness. The results in this section will not come as a big surprise for experts since similar, but more specific, statements were known before. However, this is the first time they are proven in such generality.

3.1 Properties of the $(1,\lambda)$ EA

The following lemma summarizes known results on transition probabilities and expectations from the literature [7, 37, 40].

Lemma 3.1 *Let $X_t := n - OM(x^{(t)})$ be the current distance to the optimum of the $(1,\lambda)$ EAon OneMaxand let $\Delta_k := (X_t - X_{t+1} \mid X_t = k)$ be the progress in one iteration. Recall $\eta = e/(e-1)$ and $q = \eta^{-\lambda}$. Then*

$$\mathbb{P}(\Delta_k \geq 1) \geq 1 - \left(1 - \frac{k}{en}\right)^{\lambda} \geq \frac{\lambda k}{en + \lambda k} \geq \frac{1}{2}\min\left\{1, \frac{\lambda k}{en}\right\},$$
$$\mathbb{P}(\Delta_k < 0) \leq q.$$

If $\lambda \geq 8nq/(k+1)$ and $\lambda = \omega(1)$,

$$\mathbb{E}(\Delta_k) \geq \mathbb{E}(\min\{\Delta_k, 1\}) \geq \min\left\{1, \frac{\lambda k}{n}\right\} \cdot \frac{3-e}{9}.$$

All statements also hold when replacing the $(1, \lambda)$ EAwith the $(1 + \lambda)$ EA, without any restrictions on λ.

Proof We start with the $(1, \lambda)$ EA. The first two statements are shown in [40, Lemma 2.2], except for the last step on $\mathbb{P}(\Delta_k \geq 1)$, which holds since either $en + \lambda k \leq 2en$ or $en + \lambda k \leq 2\lambda k$.

We turn to a lower bound on $\mathbb{E}(\Delta_k)$. The condition on λ implies $k + 1 \geq \frac{8nq}{\lambda}$ and $\lambda \geq 9$ as required by Theorem 2.3. Then the drift of the $(1, \lambda)$ EAin terms of the Hamming distance to the optimum, if the current Hamming distance is k, is at least

$$\begin{cases} \frac{3-e}{6}, & \text{if } \lambda\ k \geq (3-e)n, \\ \frac{\lambda k}{n} \cdot \frac{3-e}{9}, & \text{otherwise.} \end{cases} \tag{12}$$

Thus, the drift is at least $\min\{1, \frac{\lambda k}{n}\} \cdot \frac{3-e}{9}$ for all $k \geq 1$. We note that the cited lower bounds on the drift, and the above arguments, only consider steps decreasing the distance by exactly 1. Hence the drift bounds remain valid for the drift of $\min\{\Delta_k, 1\}$.

For the $(1 + \lambda)$ EA, the first two statements also apply since steps increasing the distance to the optimum are rejected. Then, for any $\lambda \in \mathbb{N}$, the lower bound on $\mathbb{E}(\Delta_k)$ follows directly from the first statement: $\mathbb{E}(\Delta_k) \geq \mathbb{P}(\Delta_k \geq 1) \geq \frac{\lambda k}{en+\lambda k}$. Following [37], if $\lambda k < (3-e)n$,

$$\frac{\lambda k}{en + \lambda k} \geq \frac{\lambda k}{en + (3-e)n} = \frac{\lambda k}{3n} > \frac{\lambda k}{n} \cdot \frac{3-e}{9}$$

and if $\lambda k \geq (3-e)n$,

$$\frac{\lambda k}{en + \lambda k} \geq \frac{(3-e)n}{en + (3-e)n} = \frac{3-e}{3} > \frac{3-e}{6}.$$

This shows (12) for the $(1 + \lambda)$ EAand continuing as above completes the proof. □

The next lemma confirms that the probability of not creating a clone is approximately q.

Lemma 3.2 *Let $\lambda = o(n)$ and $q = \eta^{-\lambda}$. Then the probability of an iteration of the $(1, \lambda)$ EAand the $(1 + \lambda)$ EAof not creating a clone of the current search point is at least q and at most $q(1 + o(1))$.*

Proof The probability of not creating a clone during λ offspring creations is

$$\left(1 - \left(1 - \tfrac{1}{n}\right)^n\right)^\lambda,$$

as all λ offspring creations must flip at least one bit, and the latter happens with probability $1-(1-\frac{1}{n})^n$. Using $(1-\frac{1}{n})^n \leq 1/e \leq (1-\frac{1}{n})^{n-1}$, we get

$$\left(1-\left(1-\tfrac{1}{n}\right)^n\right)^\lambda \geq \left(1-\tfrac{1}{e}\right)^\lambda = q,$$

and

$$\begin{aligned}\left(1-\left(1-\tfrac{1}{n}\right)^n\right)^\lambda &= \left(1-\left(1-\tfrac{1}{n}\right)^{n-1}\left(1-\tfrac{1}{n}\right)\right)^\lambda \\ &\leq \left(1-\tfrac{1}{e}\left(1-\tfrac{1}{n}\right)\right)^\lambda \\ &= \left(1-\tfrac{1}{e}\right)^\lambda \left(1+\tfrac{1}{(e-1)n}\right)^\lambda \\ &\leq q\cdot\exp\left(\tfrac{\lambda}{(e-1)n}\right) = q\cdot(1+o(1)),\end{aligned}$$

since $\frac{\lambda}{n} = o(1)$. □

Next we show that it is unlikely to have any mutation which flips more than $c\ln n$ bits within the first $n^2\ln(n)$ function evaluations. As a result, if $\lambda = O(\ln n)$, w.h.p. no such mutation occurs in the first n^2 generations. We remark that the n^2 is arbitrary and could be replaced by any other polynomial term in *n*.

Lemma 3.3 *For every constant $c > 0$, the probability of an offspring having a Hamming distance of at least $c\ln(n)$ to its parent is $n^{-\Omega(\ln\ln n)}$. Hence, w.h.p. each offspring generated during the first $n^{O(1)}$ evaluations in $(1,\lambda)$ EAor $(1+\lambda)$ EAhas Hamming distance at most $c\ln(n)$ from its parent.*

Proof The probability that a standard bit mutation flips at least $c\ln(n)$ bits is bounded by the probability of flipping any $c\ln(n)$ bits chosen from *n* bits, which is

$$\binom{n}{c\ln(n)} n^{-c\ln(n)} \leq \frac{1}{(c\ln(n))!} = (\ln(n))^{-\Omega(\ln(n))} = n^{-\Omega(\ln\ln(n))},$$

where we used $n! \geq (n/e)^n = n^{\Omega(n)}$. The second part of the statement follows by a union bound over polynomially many offspring. □

3.2 Domination Results

We continue with domination results. It was first shown in [41] that OneMaxis the easiest function for the $(1+1)$EA, and these results were extended later in [42–45]. Here we show that OneMaxis also the easiest function for the $(1+\lambda)$ EA, and that conversely the $(1+\lambda)$ EAis the fastest mutation-based algorithm on OneMaxwhich creates solutions in batches, see Theorem 3.5. Both statements also hold in fixed-target settings. This result has very powerful implications, and we first give three immediate consequences in Theorem 3.4. We say that a random variable *Y stochasti-*

cally dominates a random variable X, written as $Y \succeq X$, if $\mathbb{P}(Y \geq s) \geq \mathbb{P}(X \geq s)$ for all $s \in \mathbb{R}$, or, equivalently, $\mathbb{P}(Y \leq s) \leq \mathbb{P}(X \leq s)$ for all $s \in \mathbb{R}$.

Theorem 3.4 *Let $a, b \in [0, n]$, $\lambda \in \mathbb{N}$, and consider the $(1+1)$EA, the $(1, \lambda)$ EAand the $(1+\lambda)$ EAwith the same mutation rate $r \leq 1/2$ and with starting points x^{one}, x^{comma} and x^{plus} respectively. Let $f : \{0,1\}^n \to \mathbb{R}$ be any fitness function.*

(a) *Assume $\text{OM}(x^{\text{comma}}) \leq a$ and $\text{OM}(x^{\text{plus}}) \geq a$. Let $T^{\text{plus}}(\text{OM})$ be the number of generations until the $(1+\lambda)$ EAon OneMax creates a search point x with $\text{OM}(x) \geq b$, and let $T^{\text{comma}}(f)$ be the number of generations until the $(1, \lambda)$ EAon f creates a search point x* with $\text{OM}(x) \geq b$. *Then $T^{\text{comma}}(f)$ stochastically dominates $T^{\text{plus}}(\text{OM})$.*
(b) *Assume $\text{OM}(x^{\text{one}}) \geq a$ and $\text{OM}(x^{\text{plus}}) \leq a$. On OneMax, let T^{one} and T^{plus} be the number of function evaluations until the respective algorithm finds a search point x* with $\text{OM}(x) \geq b$. *Then T^{plus} stochastically dominates T^{one}.*
(c) *Let $x^* \in \{0,1\}^n$ and let $x^f, x^{\text{OM}} \in \{0,1\}^n$ be such that $H(x^f, x^*) \geq a$ and $H(x^{\text{OM}}, \vec{1}) \leq a$. Let T^f be the hitting time of the set $\{x : H(x, x^*) \leq b\}$ for the $(1+\lambda)$ EAwith $x^{\text{plus}} = x^f$ on f and let T^{OM} be the hitting time of the set $\{x : H(x, \vec{1}) \leq b\}$ for the $(1+\lambda)$ EAwith $x^{\text{plus}} = x^{\text{OM}}$ on OneMax.* Then *T^f stochastically dominates T^{OM}.*

Part (b) was shown in [46]. For (c), the most natural case is that x^* is the unique global optimum of f, but this is not required.

In fact, all three parts of Theorem 3.4 are just special cases of the following, more general theorem. It says that the $(1+\lambda)$ EAon OneMaxis faster than any other evolutionary algorithm using standard bit mutation (in the framework defined below) with the same mutation rate if it creates offspring in batches of size λ. Crucially, this holds for any selection strategy for the parents, which may be chosen from all previously generated search points. Thus, it is also independent of the fitness function, since this "only" decides which individuals may reproduce.

Theorem 3.5 *Let $\lambda \in \mathbb{N}$, $a, b \in [0, n]$, mutation rate $r \leq 1/2$ and let $x^{\mathcal{A}}, x^{\text{plus}} \in \{0,1\}^n$ with $OM(x^{\mathcal{A}}) \leq a$ and $OM(x^{plus}) \geq a$. Consider any algorithm $\mathcal{A}$ with the following scheme. The algorithm starts by creating $x^{\mathcal{A}}$. In each generation, it uses an arbitrary mechanism to select λ (not necessarily distinct) parents among all previously created search points, and creates λ offspring by applying standard bit mutation with mutation probability r to them.*

Let S_t be the set of search points that $\mathcal{A}$ creates in the first t generations, and let $X^{t,\mathcal{A}} := \max\{OM(x) \mid x \in S_t\}$. Let $X^{t,\text{plus}}$ be the OM-value of the $(1+\lambda)$ EAwith standard bit mutation and mutation probability p on OneMaxafter t generations if started in x^{plus}. Then $X^{t,\text{plus}}$ stochastically dominates $X^{t,\mathcal{A}}$.

Moreover, let $T^{\mathcal{A}} := \min\{t : X^{t,\mathcal{A}} \geq b\}$ and $T^{\text{plus}} := \min\{t : X^{t,\text{plus}} \geq b\}$. Then $T^{\mathcal{A}}$ stochastically dominates T^{plus}.

Proof The proof is based on Lemma 6.1 in [43], which states that if y and y' are obtained by standard bit mutation of x and x' respectively, with mutation probability $r \leq 1/2$, and if $\mathrm{OM}(x) \leq \mathrm{OM}(x')$ then $\mathrm{OM}(y')$ stochastically dominates $\mathrm{OM}(y)$.

We show that for all $b \in [0, n]$,

$$\mathbb{P}(X^{t,\mathcal{A}} \geq b) \leq \mathbb{P}(X^{t,\mathrm{plus}} \geq b). \tag{13}$$

We use induction over t. For $t = 0$, (13) is satisfied because the $(1+\lambda)$ EAstarts with $\mathrm{OM}(x^{\mathrm{plus}}) \geq a$, while $\mathcal{A}$ starts with $\mathrm{OM}(x^{\mathcal{A}}) \leq a$. So we assume that (13) holds for some $t \geq 0$ and show the same statement for $t+1$. Since $X^{t,\mathrm{plus}}$ stochastically dominates $X^{t,\mathcal{A}}$, we can couple them such that $X^{t,\mathrm{plus}} \geq X^{t,\mathcal{A}}$. We will show that for all $s \in [1, n]$,

$$\mathbb{P}\big(X^{t+1,\mathcal{A}} \geq b \mid X^{t,\mathrm{plus}} = s\big) \leq \mathbb{P}\big(X^{t+1,\mathrm{plus}} \geq b \mid X^{t,\mathrm{plus}} = s\big). \tag{14}$$

Note that we condition on the same event on both sides, so (13) follows from (14) by the law of total probability, that is,

$$\begin{aligned}
\mathbb{P}\big(X^{t+1,\mathcal{A}} \geq b\big) &= \sum_s \mathbb{P}\big(X^{t,\mathrm{plus}} = s\big)\mathbb{P}\big(X^{t+1,\mathcal{A}} \geq b \mid X^{t,\mathrm{plus}} = s\big) \\
&\leq \sum_s \mathbb{P}\big(X^{t,\mathrm{plus}} = s\big)\mathbb{P}\big(X^{t+1,\mathrm{plus}} \geq b \mid X^{t,\mathrm{plus}} = s\big) \\
&= \mathbb{P}\big(X^{t+1,\mathrm{plus}} \geq b\big).
\end{aligned}$$

When $s \geq b$, there is nothing to show since the $(1+\lambda)$ EAis elitist and thus the right-hand side of (14) is one. So, we fix some value $s < b$ and condition on $X^{t,\mathrm{plus}} = s$. Denote by $p_{\mathrm{imp}} = p_{\mathrm{imp}}(s)$ the probability that a particular offspring of the $(1+\lambda)$ EAin generation $t+1$ creates an offspring of OM-value at least b. Then $\mathbb{P}\big(X^{t+1,\mathrm{plus}} \geq b \mid X^{t,\mathrm{plus}} = s\big) = 1 - (1 - p_{\mathrm{imp}})^{\lambda}$. On the other hand, since $X^{t,\mathcal{A}} \leq X^{t,\mathrm{plus}} = s$, the algorithm $\mathcal{A}$ has only created potential parents of OM-value at most s until generation t. Hence, by [43, Lemma 6.1], any offspring y of $\mathcal{A}$ in generation $t+1$ satisfies $\mathbb{P}\big(\mathrm{OM}(y) \geq b \mid X^{t,\mathrm{plus}} = s\big) \leq p_{\mathrm{imp}}$, and this bound holds independently for all λ offspring of $\mathcal{A}$. Hence,

$$\begin{aligned}
\mathbb{P}\big(X^{t+1,\mathcal{A}} \geq b \mid X^{t,\mathrm{plus}} = s\big) &\leq 1 - (1 - p_{\mathrm{imp}})^{\lambda} \\
&= \mathbb{P}\big(X^{t,\mathrm{plus}} \geq b \mid X^{t+1,\mathrm{plus}} = s\big).
\end{aligned}$$

This concludes the induction and proves the first domination statement. The second domination statement is just a reformulation of the first one: the event "$T^{\mathcal{A}} \leq t$" is identical to the event "$X^{t,\mathcal{A}} \geq b$", since both express that within the first t generations $\mathcal{A}$ creates a search point of OM-value at least b. Hence,

$$\mathbb{P}\big(T^{\mathcal{A}} \leq t\big) = \mathbb{P}\big(X^{t,\mathcal{A}} \geq b\big) \leq \mathbb{P}\big(X^{t,\mathrm{plus}} \geq b\big) = \mathbb{P}\big(T^{\mathrm{plus}} \leq t\big).$$

□

Proof of Theorem 3.4 For part (a), we just need to observe that the $(1, \lambda)$ EAon any fitness function f falls into the category of $\mathcal{A}$ in Theorem 3.5. So we may choose the $(1, \lambda)$ EAfor $\mathcal{A}$.

Part (b) follows by setting $\lambda_{3.5} := 1$ in Theorem 3.5, where we use the subscript 3.5 to distinguish it from the λ in the statement of this theorem. Then the $(1 + \lambda_{3.5})$ EA from that lemma is just the $(1 + 1)$EA, and for algorithm $\mathcal{A}$ we can take the $(1 + \lambda)$ EA. Then Theorem 3.5 implies that T^{plus} stochastically dominates T^{one}.

For (c), by symmetry it suffices to show the statement for the case $x^* = \vec{1}$. We again apply Theorem 3.5, where we choose for $\mathcal{A}$ the $(1 + \lambda)$ EAon f. (Note that f decides which of the previously generated solutions is allowed to survive and produce offspring.)

Then Theorem 3.5 implies that T^f stochastically dominates T^{OM}. □

As the proof shows, we could have replaced the $(1, \lambda)$ EAin Theorem 3.4(a) by any other algorithm creating offspring in batches of size λ via standard bit mutation as in Theorem 3.5.

3.3 High-Probability Fixed Target Results

Now we give upper and lower bounds for the time that the $(1 + \lambda)$ EAand the $(1, \lambda)$ EAneed to reach some target fitness on OneMax, in the regime where the $(1, \lambda)$ EAis efficient. We show that it is exponentially unlikely to deviate from the expectation by more than a constant factor. We will use those results for proving the lower bound in Theorem 1.1 in Sect. 5 and at several places in the proof of Theorem 1.3 in Sect. 6.

The time bounds match known ones for the $(1 + \lambda)$ EA[47, 48] and the $(1, \lambda)$ EA[6, 7, 37, 40], albeit that the dependency on λ can be improved slightly, see [49–51] and we do not have tight leading constants. The strength of our result lies in its generality and exponentially small tail bounds. Related previous work includes upper tail bounds [36] and lower tail bounds for OneMax[52], a review of fixed-target results in [30] and black-box complexity lower bounds with tail bounds for unary unbiased black-box algorithms [51] in a framework similar to ours. The latter work includes a fixed-target scenario of getting close (in Hamming distance) to global or local optima. Our tail bounds are stronger than the previous ones. Notably, this is achieved by using simple, direct arguments and applying Chernoff-type bounds, instead of using drift analysis.

Theorem 3.6 *Consider an algorithm $\mathcal{A}$ as in Theorem 3.5 with the standard mutation rate of $r := 1/n$ and a fitness function $f : \{0, 1\}^n \to \mathbb{R}$. Let $a, b \in [0, n]$ with $a > b$ and fix a search point $x^* \in \{0, 1\}^n$. Let $T^{\mathcal{A},f}$ denote the number of evaluations made by $\mathcal{A}$ on f to reach a search point within Hamming distance at most b of x^*.*

There are positive constants c_1, c_2 such that the following holds.

1. *For every algorithm $\mathcal{A}$, every fitness function f and every target search point x^*, if $\mathcal{A}$ starts with a population of search points that all have Hamming distance at least a to x^*,*

$$\mathbb{P}\left(T^{\mathcal{A},f} \leq c_1 n \ln(a/b)\right) \leq e^{-\Omega(\min\{a-b,b\})}.$$

2. *For $f = \textsc{OneMax}$, $x^* = \vec{1}$ and either $\mathcal{A} = (1+\lambda)$ EAwith arbitrary λ (including the (1+1) EA) or $\mathcal{A} = (1,\lambda)$ EAwith $\lambda \geq \max\{\log_\eta(n/b), 16(e+1)/3\}$, if $\mathcal{A}$ starts with a search point at Hamming distance at most a to x^*,*

$$\mathbb{P}\left(T^{\mathcal{A},f} \geq c_2\left((a-b)\lambda + \ln(a/b)n\right)\right) \leq e^{-\Omega(\min\{a-b,b\})}.$$

We start with the first statement, which is an upper bound on the lower tail. Note that it suffices to prove the tail bound for the $(1+1)$EAon OneMax, since by Theorem 3.4 the same tail bound on the number of function evaluations also holds for the $(1+\lambda)$ EAon OneMaxby (b) and for any other $\mathcal{A}$ and f by Theorem 3.5.

The next technical lemma will be used to show the following. Once the distance to $\vec{1}$ in the $(1+1)$EAdecreases to a value at most k, the probability of overshooting the target by some value d is exponentially small in d. The statement is phrased more generally, describing the outcome of one standard bit mutation, as it may be of independent interest (e. g. for estimating probabilities of overshooting local optima on functions of unitation, cf. Lemma 9 in [53]).

Lemma 3.7 *Consider a standard bit mutation with mutation rate 1/n of a search point at Hamming distance i from some arbitrary but fixed search point x^* and let X denote the random Hamming distance from x^* in the offspring. Then for all $n \in \mathbb{N}$, all $k < i$ and all $m \in \{0,\dots,k\}$ we have*

$$\mathbb{P}(X \leq k - m \mid X \leq k) \leq 2^{-m}.$$

Proof Without loss of generality we assume $x^* = \vec{1}$, that is, X is the number of zeros in the offspring. The claim is trivial for $m = 0$ and $n = 1$ (as $n = 1$ implies $i = 1, k = 0$ and $m = 0$), hence we assume $m \geq 1$ and $n \geq 2$ in the following. Let L denote the random number of flipped ones, that is, bit flips that *increase* the distance to $\vec{1}$. We show that

$$\mathbb{P}(X \leq k - m \mid X \leq k, L = \ell) \leq 2^{-m}$$

for all feasible values $\ell \in \{0,\dots,n-i\}$. This implies the claim by the law of total probability. Note that it suffices to consider $\ell \leq k - m$ as otherwise the left-hand side is 0. Let F denote the random number of flipped zeros (i. e. bits that *decrease* the distance to $\vec{1}$), then we have

$$\mathbb{P}(X \leq k-m \mid X \leq k, L=\ell) = \frac{\mathbb{P}(X \leq k-m \mid L=\ell)}{\mathbb{P}(X \leq k \mid L=\ell)} = \frac{\mathbb{P}(F \geq i-k+m+\ell)}{\mathbb{P}(F \geq i-k+\ell)}$$

as flipping at least $i-k+\ell$ zeros and flipping ℓ ones yields a search point with at most k zeros, and flipping at least $i-k+\ell+m$ zeros and flipping ℓ ones yields a search point with at most $k-m$ zeros. Note that F is a binomial random variable with parameters i and $1/n$.

We claim that for all $x \in \{1,\dots,i\}$ and all $m \in \{1,\dots,k\}$ we have

$$\frac{\mathbb{P}(F=x+m)}{\mathbb{P}(F=x)} \leq 2^{-m}.$$

This implies the statement as then

$$\begin{aligned}\frac{\mathbb{P}(F \geq i-k+m+\ell)}{\mathbb{P}(F \geq i-k+\ell)} &= \frac{\sum_{j=0}^{k-m-\ell} \mathbb{P}(F=i-k+m+\ell+j)}{\sum_{j=0}^{k-\ell} \mathbb{P}(F=i-k+\ell+j)} \\ &\leq \frac{\sum_{j=0}^{k-m-\ell} 2^{-m} \cdot \mathbb{P}(F=i-k+\ell+j)}{\sum_{j=0}^{k-\ell} \mathbb{P}(F=i-k+\ell+j)} \leq 2^{-m}.\end{aligned}$$

It remains to prove the claim. The claim is trivial for $i < x+m$ as then there are less than $x+m$ zeros and $\mathbb{P}(F=x+m)=0$, hence we assume $x+m \leq i$. Then we get

$$\begin{aligned}\frac{\mathbb{P}(F=x+m)}{\mathbb{P}(F=x)} &= \frac{\binom{i}{x+m}(1/n)^{x+m}(1-1/n)^{i-x-m}}{\binom{i}{x}(1/n)^{x}(1-1/n)^{i-x}} \\ &= \frac{x!(i-x)!}{(x+m)!(i-x-m)!} \cdot \left(\frac{1}{n-1}\right)^m \\ &= \prod_{j=0}^{m-1} \frac{i-x-j}{x+m-j} \cdot \left(\frac{1}{n-1}\right)^m.\end{aligned}$$

Now, if $i-x < x+m$ then all factors in the above $\prod$-term are at most $\frac{n-1}{n}$ (as $x+m-j \geq i-x-j+1$, $i-x-j \leq n-1$ and the function $\frac{z-1}{z}$ is increasing in z for $z \in \mathbb{N}$) and we obtain $\left(\frac{n-1}{n}\right)^m \cdot \left(\frac{1}{n-1}\right)^m = n^{-m} \leq 2^{-m}$ as an upper bound, using $n \geq 2$. Otherwise, if $i-x \geq x+m$, the factors are non-decreasing with j and so we may bound all factors by that for $j=m-1$. This yields the upper bound

$$\left(\frac{i-x-m+1}{x+1}\right)^m \cdot \left(\frac{1}{n-1}\right)^m \leq \left(\frac{n-1}{x+1}\right)^m \cdot \left(\frac{1}{n-1}\right)^m = \left(\frac{1}{x+1}\right)^m \leq 2^{-m}$$

where we used $x \geq 1$, $m \geq 1$ and $i \leq n$. □

Lemma 3.8 *Consider the (1+1) EA on an arbitrary fitness function and fix an arbitrary search point x^*. Let $a, b \in [n]$ with $a > b$ and let T denote the number of iterations for the (1+1) EA to reach a Hamming distance of at most b from x^* when starting at Hamming distance a from x^*. Then*

$$\mathbb{P}\left(T \leq \frac{(a-b)n}{2a}\right) \leq e^{-(a-b)/6}.$$

Moreover, if $a \geq 2b$ then

$$\mathbb{P}\left(T \leq \frac{\log_2(a/b)n}{16}\right) \leq e^{-\Omega(b)}.$$

Proof Without loss of generality, we assume $x^* = \vec{1}$. In order to go from distance a to b in at most $(a-b)n/2a$ generations, there must be at least $a-b$ bit flips among the initial a positions of 0-bits during those generations. This necessary condition is irrespective of the fitness function. In each generation, each of the a positions has probability $1/n$ to be flipped. In $(a-b) \cdot \frac{n}{2a}$ generations, there are thus $a \cdot (a-b) \cdot \frac{n}{2a} = (a-b)n/2$ chances for these bits to flip, each with probability $1/n$. Thus, the total number of bit flips among the a positions is given by a Binomial distribution $\mathrm{Bin}((a-b)n/2, 1/n)$, which has expectation $(a-b)/2$. By Chernoff bounds (Theorem 2.4), the probability of having at least $a-b$ bit flips among the a positions is at most

$$\exp\left(-\frac{(a-b)}{6}\right).$$

For the second statement, we apply the first statement repeatedly. Let ℓ be the smallest integer such that $a/2^\ell < b$. Since $a \geq 2b$, we have $\ell > 1$.

We first describe the main idea before going into detail. We split the interval $[b, a]$ into $\ell - 1 \geq 1$ intervals $[b_i, a_i]$ that are mutually non-overlapping (except possibly for the extremes b_i and a_i). For each interval $[b_i, a_i]$ we consider the random time T_i for reaching a distance of at most b_i, when starting with a distance of a_i. More specifically, we define $b_i := a/2^i$ and $a_1 := a$ deterministically (ignoring rounding issues for the sake of readability). The values a_i for $2 \leq i \leq \ell - 1$ are defined as random variables: a_i is the random distance to the optimum reached when the distance decreases to a value at most b_{i-1} for the first time. Note that we may have $a_i = b_{i-1}$ in case the algorithm stops at distance b_{i-1}, but we may also have $a_i < b_{i-1}$ in case the target distance of b_{i-1} is overshot. In particular, the next interval $[b_i, a_i]$ can be much smaller than $[b_i, b_{i-1}]$ and it may, in theory, even be empty if $a_i < b_i$. However, we shall show that, typically, all intervals are still large.

For $1 \leq i \leq \ell - 1$ we define E_i as the event that $(a_i - b_i) \geq (b_{i-1} - b_i)/2 = (a/2^{i-1} - a/2^i)/2 = a/2^{i+1}$. This means that, even

when the previous target distance b_{i-1} was overshot, at least half the distance between b_{i-1} and b_i still needs to be covered. Note that E_1 is always true as $a_1 = a$ was chosen deterministically. Conditional on $E_1 \cap \ldots \cap E_{\ell-2}$ we have $T \geq \sum_{i=1}^{\ell-1} T_i$. For all non-negative numbers $U_1, \ldots, U_{\ell-1}$ (which will be made precise later) and $U := \sum_{i=1}^{\ell-1} U_i$, we have that $T \leq U$ implies that $T_i \leq U_i$ for at least one i. Consequently, $\mathbb{P}(T \leq U) \leq \mathbb{P}(\exists i : T_i \leq U_i) \leq \sum_{i=1}^{\ell-1} \mathbb{P}(T_i \leq U_i)$ by a union bound. Incorporating the aforementioned condition more formally, by the law of total probability,

$$\begin{aligned}
\mathbb{P}(T \leq U) &= \mathbb{P}(T \leq U \mid E_1 \cap \ldots \cap E_{\ell-2})\, \mathbb{P}(E_1 \cap \ldots \cap E_{\ell-2}) \\
&\quad + \mathbb{P}\left(T \leq U \mid \overline{E_1 \cap \ldots \cap E_{\ell-2}}\right) \mathbb{P}(\overline{E_1 \cap \ldots \cap E_{\ell-2}}) \\
&\leq \mathbb{P}(T \leq U \mid E_1 \cap \ldots \cap E_{\ell-2}) + \mathbb{P}(\overline{E_1 \cap \ldots \cap E_{\ell-2}}) \\
&\leq \mathbb{P}(T \leq U \mid E_1 \cap \ldots \cap E_{\ell-2}) + \sum_{i=1}^{\ell-2} \mathbb{P}(\overline{E_i}) \\
&\leq \sum_{i=1}^{\ell-1} \mathbb{P}(T_i \leq U_i \mid E_i) + \sum_{i=1}^{\ell-2} \mathbb{P}(\overline{E_i}).
\end{aligned}$$

The probabilities $\mathbb{P}(\overline{E_i})$ for $2 \leq i \leq \ell - 1$ are bounded as follows. Applying Lemma 3.7 with $k := b_{i-1}$ and $m := (b_{i-1} - b_i)/2 = a/2^{i+1}$, in the first generation in which the distance decreases to a value at most b_{i-1}, it is at least

$$b_{i-1} - m = b_i + (b_{i-1} - b_i) - \frac{b_{i-1} - b_i}{2} = b_i + \frac{b_{i-1} - b_i}{2}$$

with probability $\mathbb{P}(E_i) \geq 1 - 2^{-a/2^{i+1}}$. Consequently, using $a \geq 2^{\ell-1} b$ (which follows from the definition of ℓ),

$$\begin{aligned}
\sum_{i=1}^{\ell-2} \mathbb{P}(\overline{E_i}) &\leq \sum_{i=1}^{\ell-2} 2^{-a/2^{i+1}} \leq \sum_{i=1}^{\ell-2} 2^{-b \cdot 2^{\ell-1}/2^{i+1}} = \sum_{j=0}^{\ell-3} 2^{-b \cdot 2^j} \leq \sum_{j=0}^{\ell-3} 2^{-b \cdot (j+1)} \\
&\leq 2^{-b} \cdot \sum_{j=0}^{\infty} 2^{-bj} = 2^{-b} \cdot \frac{1}{1 - 2^{-b}} \leq 2 \cdot 2^{-b},
\end{aligned}$$

as $b \geq 1$.

We proceed with bounding $\mathbb{P}(T_i \leq U_i \mid E_i)$ where we define $U_i := \frac{(a_i - b_i)n}{2a_i}$. We apply the first statement with parameters b_i and a_i and obtain

$$\mathbb{P}(T_i \leq U_i \mid E_i) \leq e^{-(a_i - b_i)/6} \leq e^{-a/(6 \cdot 2^{i+1})} = e^{-b \cdot 2^{\ell-i-2}/6}.$$

Summing up these probabilities for all i yields

$$\sum_{i=1}^{\ell-1} \mathbb{P}\left(T_i \le U_i \mid E_i\right) \le \sum_{i=1}^{\ell-1} e^{-b\cdot 2^{\ell-i-2}/6} = \sum_{j=0}^{\ell-2} e^{-b\cdot 2^j/12}$$
$$\le \sum_{j=0}^{\ell-2} e^{-b\cdot(j+1)/12} = e^{-b/12} \sum_{j=0}^{\ell-2} e^{-b\cdot j/12}$$
$$\le e^{-b/12} \sum_{j=0}^{\infty} e^{-b\cdot j/12} = e^{-b/12} \cdot \frac{1}{1-e^{-b/12}}.$$

Since $(a_i - b_i) \ge a/2^{i+1}$ and $a_i \le b_{i-1} = a/2^{i-1}$, we get

$$U := \sum_{i=1}^{\ell-1} U_i = \sum_{i=1}^{\ell-1} \frac{(a_i - b_i)n}{2a_i} \ge \sum_{i=1}^{\ell-1} \frac{an/2^{i+1}}{2a/2^{i-1}} = \frac{n}{8} \sum_{i=1}^{\ell-1} 1 \ge \frac{\ell n}{16}$$

using $\ell \ge 2$ in the last step. Since $a/2^\ell < b$ and $\ell > \log_2(a/b)$, we have

$$\mathbb{P}\left(T \le \frac{\log_2(a/b)n}{16}\right) \le \mathbb{P}(T \le U) \le 2 \cdot 2^{-b} + O(e^{-b/12}) = e^{-\Omega(b)}.$$

□

Now we prepare the proof of the second statement of Theorem 3.6, the bound on the upper tail. To this end, we couple the progress to the following set of independent random variables.

Definition 3.9 For $t \in \mathbb{N}$ and $i \in \{1, \dots, t\}$ define independent random variables

$$Z_i := \begin{cases} +1 & \text{with probability } \frac{3}{4} \\ -j & \text{with probability } \frac{3}{4} \cdot 4^{-j}, for j \in \mathbb{N}. \end{cases}$$

Note that the Z_i are i. i. d. with expectation

$$\mathbb{E}(Z_1) = \frac{3}{4} - \sum_{j=1}^{\infty} \frac{3}{4} \cdot j 4^{-j} = \frac{3}{4}\left(1 - \frac{4}{9}\right) = \frac{5}{12},$$

where we used the equality $\sum_{j=1}^{\infty} j x^j = \frac{x}{(1-x)^2}$ for $0 < x < 1$.

Lemma 3.10 *Consider the $(1, \lambda)$ EAon OneMaxwith a target distance of k^* and $\lambda \ge \max\{\log_\eta(n/k^*), 16(e+1)/3\}$. Let X_t denote the distance to the optimum at time t. Then for all $k \ge k^*$ the progress $(X_t - X_{t+1} \mid X_t = k, X_{t+1} \ne X_t)$, that is, the change in distance conditional on $X_{t+1} \ne X_t$, stochastically dominates Z_i.*

The same statement also holds when replacing the $(1, \lambda)$ EAwith the $(1 + \lambda)$ EAand dropping the condition on λ.

Proof For the $(1+\lambda)$ EAthe statement is trivial. Since on OneMaxthe distance cannot increase, $X_{t+1} \neq X_t$ implies $X_{t+1} < X_t$ and hence $(X_t - X_{t+1} \mid X_t = k, X_{t+1} \neq X_t) \geq 1$. Since $Z_i \leq 1$, the conditional progress stochastically dominates Z_i. So now we consider the $(1,\lambda)$ EA.

Let $\Delta_k := (X_t - X_{t+1} \mid X_t = k)$ and $C := \frac{16}{3}$. By Lemma 3.1,

$$\mathbb{P}(\Delta_k \geq 1) \geq \frac{\lambda k}{en + \lambda k},$$

and

$$\mathbb{P}(\Delta_k < 0) \leq q = \eta^{-\lambda}.$$

We claim that $\eta^{-\lambda} \leq \frac{1}{C} \cdot \frac{\lambda k}{en+\lambda k}$, which is equivalent to $\lambda \geq \log_\eta(C(1+\frac{en}{\lambda k}))$. Using $k \geq k^*$ and $\lambda \geq C(e+1)$, we have $\log_\eta(C(1+\frac{en}{\lambda k})) \leq \log_\eta(C + \frac{en}{(e+1)k^*})$. Now, if $C \leq \frac{n}{(e+1)k^*}$ then $\log_\eta(C + \frac{en}{(e+1)k^*}) \leq \log_\eta(n/k^*) \leq \lambda$ by assumption on λ. Otherwise, if $C > \frac{n}{(e+1)k^*}$ then $\log_\eta(C + \frac{en}{(e+1)k^*}) \leq \log_\eta(C(e+1)) < C(e+1) \leq \lambda$ also by assumption on λ. Thus, we get

$$\mathbb{P}(\Delta_k = -1) \leq \mathbb{P}(\Delta_k < 0) \leq \eta^{-\lambda} \leq \frac{1}{C} \cdot \frac{\lambda k}{en + \lambda k} \leq \frac{1}{C} \cdot \mathbb{P}(\Delta_k \geq 1). \quad (15)$$

Now fix $j \in [2, n-k]$. The $(1,\lambda)$ EAonly increases its distance to $\vec{1}$ by j if all λ offspring increase the distance by at least j. A necessary condition is that at least j bits flip in all λ offspring. The probability of this event is at most

$$\mathbb{P}(\Delta_k = -j) \leq \left(\binom{n}{j} n^{-j} \right)^\lambda \leq \left(\frac{1}{j!} \right)^\lambda = \eta^{-\lambda} \left(\frac{\eta}{j!} \right)^\lambda.$$

The term $q = \eta^{-\lambda}$ was already bounded by $\frac{1}{C} \cdot \mathbb{P}(\Delta_k \geq 1)$ in (15). As $j \geq 2$ we have $\eta/(j!) < 1$ and the requirements on λ imply $\lambda \geq 8$. Thus, $(\eta/(j!))^\lambda \leq (\eta/(j!))^8$ and for $j = 2$ we verify numerically that $\eta^8/2^8 = (\eta/(j!))^8 \leq C^{-j+1} = \frac{3}{16}$. The inequality also holds for all $j > 2$ since the function $C^j/(j!)^8$ is decreasing with j: for all $j \geq 2$ we have

$$\frac{C^{j+1}/((j+1)!)^8}{C^j/(j!)} = \frac{C}{(j+1)^8} \leq \frac{16}{3 \cdot (2+1)^8} = \frac{16}{19683} < 1.$$

Thus, we have shown $(\eta/(j!))^\lambda \leq C^{-j+1}$ and, consequently, $\mathbb{P}(\Delta_k = -j) \leq C^{-j} \cdot \mathbb{P}(\Delta_k \geq 1)$. This implies

$$\mathbb{P}(\Delta_k \neq 0) \leq \mathbb{P}(\Delta_k \geq 1) + \sum_{j=1}^{\infty} C^{-j} \cdot \mathbb{P}(\Delta_k \geq 1) = \frac{C}{C-1} \cdot \mathbb{P}(\Delta_k \geq 1).$$

Now the claim follows from

$$\mathbb{P}(\Delta_k \geq 1 \mid \Delta_k \neq 0) = \frac{\mathbb{P}(\Delta_k \geq 1)}{\mathbb{P}(\Delta_k \neq 0)} \geq \frac{\mathbb{P}(\Delta_k \geq 1)}{\frac{C}{C-1} \cdot \mathbb{P}(\Delta_k \geq 1)} = \frac{C-1}{C} \geq \frac{3}{4} = \mathbb{P}(Z_i = 1)$$

and, for all $j \in \mathbb{N}$,

$$\mathbb{P}(\Delta_k = -j \mid \Delta_k \neq 0) = \frac{\mathbb{P}(\Delta_k = -j)}{\mathbb{P}(\Delta_k \neq 0)} \leq \frac{C^{-j} \cdot \mathbb{P}(\Delta_k \geq 1)}{\mathbb{P}(\Delta_k \geq 1)} \leq \frac{3}{4} \cdot 4^{-j} = \mathbb{P}(Z_i = -j).$$

□

Now we give a Chernoff-type deviation bound for the sum of Z_i variables.

Lemma 3.11 *Consider random variables $Z_1, \ldots, Z_t$ and $Z := \sum_{i=1}^{t} Z_i$ as in Definition 3.9. Then*

$$\mathbb{P}\left(Z \leq \frac{\mathbb{E}(Z)}{2}\right) \leq \left(\frac{57}{58}\right)^t.$$

Proof We follow the proof of Chernoff bounds. Let $\gamma := \ln(7/6)$ and note that $\gamma > 0$. Using Markov's inequality and $\mathbb{E}(Z) = t \cdot \frac{5}{12}$,

$$\mathbb{P}\left(Z \leq \frac{\mathbb{E}(Z)}{2}\right) = \mathbb{P}(e^{-\gamma Z} \geq e^{-\gamma t \cdot 5/24}) \leq \frac{\mathbb{E}(e^{-\gamma Z})}{e^{-\gamma t \cdot 5/24}}.$$

We simplify the numerator as follows, exploiting the independence of the Z_i's:

$$\mathbb{E}(e^{-\gamma Z}) = \mathbb{E}\left(\prod_{i=1}^{t} e^{-\gamma Z_i}\right) = \prod_{i=1}^{t} \mathbb{E}\left(e^{-\gamma Z_i}\right).$$

By the density of the random variables Z_i from Definition 3.9, along with $e^{\gamma}/4 < 1$,

$$\mathbb{E}\left(e^{-\gamma Z_i}\right) = \frac{3}{4} \cdot e^{-\gamma} + \sum_{j=1}^{\infty} e^{\gamma j} \cdot \frac{3}{4} \cdot 4^{-j} = \frac{3}{4}\left(e^{-\gamma} + \frac{e^{\gamma}/4}{1 - e^{\gamma}/4}\right).$$

Recalling $\gamma = \ln(7/6)$, this simplifies to

$$\mathbb{E}\left(e^{-\gamma Z_i}\right) = \frac{3}{4}\left(\frac{6}{7} + \frac{7/24}{1-7/24}\right) = \frac{453}{476}.$$

Plugging this back in yields

$$\mathbb{E}(e^{-\gamma Z}) = \left(\frac{453}{476}\right)^t,$$

so

$$\frac{\mathbb{E}(e^{-\gamma Z})}{e^{-\gamma t \cdot 5/24}} = \left(\frac{453}{476}\right)^t \cdot \left(\frac{7}{6}\right)^{t \cdot 5/24}.$$

It is easily verified numerically that $(453/476) \cdot (7/6)^{5/24} \leq 57/58$, which completes the proof. □

The coupling allows us to derive tail bounds for the $(1, \lambda)$ EAand the $(1 + \lambda)$ EAon OneMax.

Lemma 3.12 *Let $a, b \in [n]$ with $a > b$. Let T denote the random number of iterations for the $(1, \lambda)$ EA with $\lambda \geq \max\{\log_\eta(n/b), 16(e+1)/3\}$ (cf. Lemma 3.10) or the $(1 + \lambda)$ EA with arbitrary $\lambda \in \mathbb{N}$ to reach a distance of at most b from $\tilde{1}$ when starting at distance at most a on OneMax. Then*

$$\mathbb{P}\left(T \geq (a-b) \cdot \frac{48}{5}\left(1 + \frac{en}{\lambda b}\right)\right) \leq 2(57/58)^{a-b}.$$

Moreover, if $a \geq 2b$ then

$$\mathbb{P}\left(T \geq \frac{48}{5} \cdot (a-b) + \frac{48en}{5\lambda} \cdot (\log_2(a/b) + 1)\right) \leq 116(57/58)^b.$$

Proof We first consider the $(1, \lambda)$ EA. Let X_t denote the distance of the current search point to the optimum at time t. Since we are only interested in reaching a distance of at most b, we may assume that the process remains at distance b as soon as a distance of $\leq b$ is reached for the first time. We call a step t *relevant* if $X_{t+1} \neq X_t$ or if $X_{t+1} = X_t \leq b$. As long as $X_t \geq b$, the probability of a relevant step is at least $\mathbb{P}(X_{t+1} > X_t \mid X_t \geq b) \geq \frac{\lambda X_t}{en + \lambda X_t} \geq \frac{\lambda b}{en + \lambda b} =: p_b$ by Lemma 3.1. Let $r := (a-b) \cdot \frac{24}{5}$ and $T := 2r/p_b$. Thus, the number of relevant steps in T iterations stochastically dominates a sum of T i. i. d. Bernoulli random variables $Y_1, \ldots, Y_T$ with parameters p_b.

Let $Y := \sum_{t=1}^{T} Y_t$ and note that $\mathbb{E}(Y) = Tp_b = 2r$. By Chernoff bounds, $\mathbb{P}(Y \leq \mathbb{E}(Y)/2) = e^{-r/4} = e^{-(a-b)\cdot 6/5}$. By stochastic domination, this also constitutes an upper bound on the probability of having fewer than r relevant steps.

Now assume the algorithm makes at least r relevant steps. By Lemma 3.10, the total progress in these steps stochastically dominates the sum of r variables, $Z{:=}Z_1 + Z_2 + \cdots + Z_r$ where the Z_i are defined as in Definition 3.9. The expected progress is $\mathbb{E}(Z) = r \cdot \mathbb{E}(Z_1) = r \cdot \frac{5}{12}$. By Lemma 3.11, the probability of the progress being at most $\mathbb{E}(Z)/2 = r \cdot \frac{5}{24} = a - b$ is at most $(57/58)^r = (57/58)^{a-b}$. Since by assumption we start at a distance at most a, a progress of at least $a - b$ implies that a distance of at most b has been reached. Taking a union bound over the two failure events proves the first claim.

For the second statement, we apply the first statement repeatedly by splitting the interval $[b, a]$ into several intervals (similarly to the proof of Lemma 3.8). Let ℓ be the smallest integer such that $a/2^\ell < b$, that is, $\ell = \lfloor \log_2(a/b) \rfloor + 1$. Since $a \geq 2b$, we have $\ell > 1$. Now we apply the first statement $\ell - 1$ times, with parameters $(a_1, b_1), (a_2, b_2), \ldots, (a_{\ell-1}, b_{\ell-1})$ chosen as $a_1{:=}a$, $b_{\ell-1}{:=}b$, $a_i{:=}a/2^{i-1}$ for $i \in [2, \ldots, \ell - 1]$ and $b_i{:=}a/2^i$ for $i \in [1, \ldots, \ell - 2]$. Note that $a_i - b_i = a/2^i$ for all $i \in [1, \ell - 2]$ and $a_{\ell-1} - b_{\ell-1} \in [a/2^{\ell-1}, a/2^{\ell-2}]$. This means that all interval lengths $a_i - b_i$ are decreasing exponentially with a base of two, except for the last interval, which can be slightly larger in order to make up for a possible remainder in the distance between a and b.

Let T_i denote the random number of generations in which the distance is in $[b_i, a_i]$. Since all distances in $[b, a]$ are counted towards some T_i, we have $T \leq \sum_{i=0}^{\ell-1} T_i$ (the inequality holds since distances of b_i are counted towards T_i and T_{i+1}). Given any non-negative numbers $L_1, \ldots, L_{\ell-1}$ and $L{:=}\sum_{i=1}^{\ell-1} L_i$, the event $T \geq L$ implies that $T_i \geq L_i$ for at least one i. Hence, $\mathbb{P}(T \geq L) \leq \sum_{i=1}^{\ell-1} \mathbb{P}(T_i \geq L_i)$. We choose $L_i{:=}(a_i - b_i) \cdot \frac{48}{5}\left(1 + \frac{en}{\lambda b_i}\right)$ according to the first statement applied to a_i and b_i. Then we have

$$L = \sum_{i=1}^{\ell-1} (a_i - b_i) \cdot \frac{48}{5}\left(1 + \frac{en}{\lambda b_i}\right) = \frac{48}{5} \sum_{i=1}^{\ell-1} (a_i - b_i) + \frac{48en}{5\lambda} \sum_{i=1}^{\ell-1} (a_i - b_i) \cdot \frac{1}{b_i}.$$

The first sum is telescopic and simplifies to $a - b$. In the second sum, summands for $i \leq \ell - 2$ simplify to 1 since $a_i - b_i = a/2^i = b_i$. The last summand is at most twice as large as the previous ones, hence the whole sum is at most ℓ. Together, since $\ell \leq \log_2(a/b) + 1$,

$$L \leq \frac{48}{5} \cdot (a - b) + \frac{48en}{5\lambda} \cdot (\log_2(a/b) + 1).$$

Thus,

$$\mathbb{P}\left(T \geq \frac{48}{5} \cdot (a - b) + \frac{48en}{5\lambda} \cdot (\log_2(a/b) + 1)\right) \leq \mathbb{P}(T \geq L) \leq \sum_{i=1}^{\ell-1} \mathbb{P}(T_i \geq L_i)$$

and summing up all failure probabilities from applications of the first statement yields a probability bound of

$$\begin{aligned}\sum_{i=1}^{\ell-1} 2(57/58)^{a/2^i} &\geq \sum_{i=1}^{\ell-1} 2(57/58)^{b\cdot 2^{\ell-1-i}} = \sum_{j=0}^{\ell-2} 2(57/58)^{b\cdot 2^j} \\ &\leq \sum_{j=0}^{\ell-2} 2(57/58)^{b\cdot(j+1)} = 2(57/58)^b \sum_{j=0}^{\ell-2} (57/58)^{bj} \\ &\leq 2(57/58)^b \cdot \frac{1}{1-(57/58)^b} \leq 116(57/58)^b\end{aligned}$$

using $b \geq 1$ in the last step. □

Now we can finally prove Theorem 3.6.

Proof of Theorem 3.6 For the first statement, by symmetry we may assume $x^* = \vec{1}$. By Theorem 3.5, the time $T^{\mathcal{A},f}$ stochastically dominates $T^{\text{plus}}(\text{OM})$ and by Theorem 3.4 (b), the latter time stochastically dominates T^{one} using the notation from Theorem 3.4. Hence it suffices to prove the statement for $\mathcal{A}$ =(1+1) EA and $f = \textsc{OneMax}$.

We choose $c_1{:=}1/(16\ln 2)$. If $b \geq a/2$ then we argue as follows. Using $x \geq \ln(1+x)$ for all $x \in \mathbb{R}$, we get

$$\frac{a-b}{2a} \geq \frac{1}{4}\cdot\frac{a-b}{b} \geq \frac{1}{4}\cdot\ln\left(1+\frac{a-b}{b}\right) = \frac{1}{4}\cdot\ln(a/b) \geq c_1\ln(a/b).$$

Along with Lemma 3.8,

$$\mathbb{P}\left(T^{\text{one}} \leq c_1\ln(a/b)n\right) \leq \mathbb{P}\left(T^{\text{one}} \leq \frac{(a-b)n}{2a}\right) \leq e^{-(a-b)/6}.$$

As the exponent is $-\Omega(\min\{a-b,b\})$, this implies the claim.

If $b < a/2$ then Lemma 3.8 yields

$$\mathbb{P}\left(T^{\text{one}} \leq \frac{\log_2(a/b)n}{16}\right) \leq e^{-\Omega(b)} = e^{-\Omega(\min\{a-b,b\})}$$

and this implies the claim since $\log_2(a/b)/16 = c_1\ln(a/b)$.

For the second statement we will apply Lemma 3.12 with $b{:=}k^*$. We choose $c_2{:=}\frac{96e}{5\ln 2}$. If $b \geq a/2$ then we apply $\frac{x}{x+1} \leq \ln(1+x)$ for all $x > -1$ to $x{:=}(a-b)/b$ and get

$$\frac{a-b}{b} \leq 2\cdot\frac{a-b}{a} = 2\cdot\frac{(a-b)/b}{a/b} \leq 2\ln(1+(a-b)/b) = 2\ln(a/b).$$

Note that Lemma 3.12 bounds the number of generations, hence we need to multiply with λ to obtain a bound on the number of function evaluations. This yields an upper bound on the number of evaluations of

$$\frac{48}{5} \cdot (a-b)\lambda + \frac{48en}{5} \cdot \frac{a-b}{b} \leq \frac{48}{5} \cdot (a-b)\lambda + \frac{96en}{5} \cdot \ln(a/b)$$

that holds with probability $1 - 2(57/58)^{a-b} \geq 1 - e^{-\Omega(a-b)}$. Since $48/5 < 96e/5 \leq c_2$, this proves the claim.

If $b < a/2$ then we note that

$$\log_2(a/b) + 1 \leq 2\log_2(a/b) = \frac{2}{\ln 2} \cdot \ln(a/b).$$

Invoking Lemma 3.12 and multiplying by λ yields a bound of

$$\frac{48}{5} \cdot (a-b)\lambda + \frac{48en}{5} \cdot (\log_2(a/b) + 1) \leq \frac{48}{5} \cdot (a-b)\lambda + \frac{96en}{5\ln 2} \cdot \ln(a/b).$$

Noting that both leading constants are at most c_2 and that the failure probability from Lemma 3.12 is at most $116(57/58)^b = e^{-\Omega(\min\{a-b,b\})}$ completes the proof. □

4 Comma Strategy Escapes Traps

In this section we prove the upper bound on T^{comma} in (1) in Theorem 1.1. The proof consists of the following steps: first we introduce dyDisOM, a "dynamic" version of disOM, and study the drift of the $(1,\lambda)$ EAon dyDisOMusing drift analysis for a suitable potential function. Studying dyDisOMinstead of disOMsimplifies the drift analysis, since the "frozen" noise in disOMintroduces dependencies with respect to distorted points that are hard to control. Using the bounds on the drift, we compute the expected running time of the $(1,\lambda)$ EAon dyDisOM. Unfortunately, our potential does not satisfy the conditions to employ a standard additive drift theorem with tail bounds. Still, we obtain an "almost-matching" upper bound (a factor $\ln n$ larger than Theorem 1.1) on the running time which holds w.h.p, see Lemma 4.3.

Using the non-sharp upper bound, we show in Lemma 4.4 that the algorithm never moves more than $o(\ln^4(n))$ away from the target. This argument is inspired by the 'ratchet argument' introduced in [54]: when viewed from a distance, the current fitness can only increase (like a ratchet tool that only allows movement in one direction), except for a limited amount of slack. This event we use to sharpen our upper bound to obtain concentration around the expected running time to prove the upper bound in Theorem 1.1. This bootstrapping argument requires a fine control on the drift. In particular, we need to consider steps towards the optimum by more than 1 (contrary to the process in Definition 3.9).

Moreover, the ratchet argument allows to argue that no distorted search point is evaluated twice. This will imply that w.h.p. $T^{\text{comma,dy}} = T^{\text{comma}}$, i.e., the number of

function evaluations until the $(1,\lambda)$ EAfinds a search point of fitness at least $n-k^*$ on DYDISOM, is the same as the number of function evaluations on DISOM.

4.1 Dynamic Distorted OneMax

We introduce a dynamic version of DISOMin which we reveal the sets of distorted and clean points gradually, and in which previously sampled distorted points can become clean later (but not the other way around). Let $s := t\lambda + j$ for $j \in [\lambda]$, and let $y^{(s)}$ be the s-th sampled search point after initialization, and $x^{(t)}$ for the current search point. Note that $y^{(s)}$ is not necessarily accepted. We write for $\mathcal{C}_s$ for the set of clean points discovered before time $s+1$. With $\mathcal{C}_0 = \{x^{(0)}\}$, we iteratively define for $s \geq 1$

$$\mathcal{C}_s = \begin{cases} \mathcal{C}_{s-1} \cup \{y^{(s)}\}, & \text{with probability } 1-p, \text{if } y^{(s)} \neq x^{(t)}, \\ \mathcal{C}_{s-1}, & \text{otherwise.} \end{cases}$$

Note that $\mathcal{C}_{s-1} \subseteq \mathcal{C}_s$ for all s, reflecting that clean points remain clean forever. Given $\mathcal{C}_s$, we define

$$\text{DYDISOM}(y^{(((s)))}) := \begin{cases} \text{OM}(x), & \text{if } x \in \mathcal{C}_s, \\ \text{OM}(x)+d, & \text{otherwise.} \end{cases}$$

We use drift analysis [34] to analyse the $(1,\lambda)$ EAon DYDISOM. For convenience, we use a potential that approximates the distance to the target, so we use the ZEROMAX function $\text{ZM}(x) := n - \text{OM}(x)$. We frequently abbreviate $\text{ZM}(x) = k$. We introduce some extra notation to define the potential function. Let $Y_1, \ldots, Y_\lambda \sim \text{Bin}(k, 1/n)$ be i. i. d. binomial random variables, and define $Y^*(k) := \max(Y_1, \ldots, Y_\lambda)$. The random variable Y_i represents the number of 0-bits flipped into a 1 by the i-th offspring when the parent has $\text{ZM}(x) = k$. Next we define a potential function which penalises being in a distorted point, since this makes it harder to find improvements. Finding the right trade-off is the heart of our analysis of DYDISOM. For $x \in \{0,1\}^n$ and some suitably chosen constant $\delta > 0$ we define the potential $P(x)$ of a search point x as

$$P(x) := \text{ZM}(x) + \mathbb{1}\{x \notin \mathcal{C}_s\} \frac{\delta}{\lambda p} \mathbb{E}\big[Y^*(\text{ZM}(x))\big]. \tag{16}$$

We will compute bounds on the drift

$$\Delta(x) := \mathbb{E}\big[P(x^{(t)}) - P(x^{(t+1)}) \mid x^{(t)} = x\big]. \tag{17}$$

Note that by our sign convention, a positive drift corresponds to progress towards smaller potentials, and thus we want to compute a positive lower bound on $\Delta(x)$ since we aim to establish an upper bound on the running time. We comment briefly on the second term in the potential $P(x)$, which balances two effects: on the one hand it is sufficiently small so that the event that a distorted offspring is found from a clean point (which happens with probability at most λp) yields a small negative

contribution to the drift; on the other hand it is sufficiently large so that the drift from a distorted point is of the same order as the drift from a clean point, even though the probability of making a jump is much smaller.

We state Lemmas 4.1–4.4, then show that an upper bound on T^{comma} follows from them, and eventually prove the stated lemmas. As always, all Landau notation is with respect to $n \to \infty$, and k^*, k, λ, and p may be functions of n. The first three items in the following lemma are due to [48].

Lemma 4.1 *[48, Lemma 4] Let, for some $k \geq 1$, $Y_1, \dots, Y_\lambda \sim \mathrm{Bin}(k, 1/n)$ be i. i. d. binomial random variables, and define $Y^*(k) := \max(Y_1, \dots, Y_\lambda)$. It follows that*

(1) *if there exists $\alpha > 0$ with $\alpha = O(1)$ such that $k = n/(\ln^\alpha \lambda)$, and $\lambda = \omega(1)$, then*

$$\mathbb{E}[Y^*(k)] = (1 \pm o(1)) \frac{\ln(\lambda)}{(\alpha+1)\ln\ln(\lambda)},$$

(2) *for all k, we have $\mathbb{E}[Y^*(k)] \geq \frac{\lambda k}{\lambda k + n}$, and thus $\mathbb{E}[Y^*(k)] = \Omega\big(\frac{\lambda k}{n}\big)$ when $k \leq n/\lambda$.*

(3) *if $k = o(n/\lambda)$, then $\mathbb{E}[Y^*(k)] = (1 - o(1))\frac{\lambda k}{n}$.*

(4) $\mathbb{E}[Y^*(n)] = O(\ln \lambda)$.

Proof We only prove the fourth item. We use a truncation argument and bound the maximum of random variables by its sum to obtain for $y \in \mathbb{N}$

$$\mathbb{E}[Y^*(n)] \leq y + \mathbb{E}\left[\sum_{i \in [\lambda]} \mathbb{1}\{Y_i \geq y\} Y_i\right] \leq y + \lambda \sum_{k=y}^{n} k \binom{n}{k} (1/n)^k \leq y + \lambda \sum_{k=y}^{n} \frac{k}{k!}.$$

Using Stirling's approximation, it follows that the sum on the right-hand side is of order $o(1/\lambda)$ when $y = \lceil 2 \ln \lambda \rceil$. □

The following lemma provides useful bounds for the drift analysis. We postpone the proof.

Lemma 4.2 *Consider the setting of Lemma 4.1 under Assumption 1.4. For $k \geq k^*$ it holds that*

(1) $\mathbb{E}[Y^*(k + \ln n)] = (1 + o(1))\mathbb{E}[Y^*(k)]$.
(2) $\lambda p \ln(n) = o(\mathbb{E}[Y^*(k)])$.
(3) $q \ln(n) = o(\mathbb{E}[Y^*(k)])$.
(4) $1/n = o(\mathbb{E}[Y^*(k)])$.

Having stated the above two general lemmas, we next state a lemma that obtains bounds on the drifts and a non-sharp upper bound on T^{comma} on DYDISOM, i.e., it contains an additional ln-factor compared to Theorem 1.1. Our lower bound on the

drift matches the drift of the $(1,\lambda)$ EAand OneMaxwith the same parameters (using the ZeroMaxpotential for those cases). Let $T^{\text{com, dy}} = T^{\text{com, dy}}_{k^*,\lambda,d,p}$ denote the number of evaluations for the $(1,\lambda)$ EAto reach the target k^* on dyDisOM.

Lemma 4.3 *Under Assumption 1.4, we have, with the potential P(x) defined in (16),*

$$\begin{array}{lll} (1) & \Delta(x) = \Omega(\mathbb{E}[Y^*(ZM(x))]) & \text{for all } x \text{ such that } ZM(x) \in [k^*, n], \\ (2) & \Delta(x) = \Omega(\mathbb{E}[P(x)\frac{\lambda}{n}]), & \text{for all } x \text{ such that } ZM(x) \in [k^*, n/\lambda]. \end{array}$$

Moreover, $T^{\text{comma,dy}}_{k^*,\lambda,d,p} \leq n \ln^2 n$ w.h.p.

Note that the two cases are not mutually exclusive. The following lemma shows that even though the noise in dyDisOMis dynamic, these dynamics are not seen by the algorithm since each sampled point always returns the same function value w.h.p. Moreover, it shows that w.h.p., for all $t \leq T^{\text{comma}}$ simultaneously, the algorithm never jumps to a search point with much smaller OM-value than the current search point $x^{(t)}$.

Lemma 4.4 *Under Assumption 1.4 the following two statements hold w.h.p.*

(1) *For dyDisOM, if there exists* $s < t \leq T^{\text{comma,dy}}$ *such that* $y^{(s)} = y^{(t)}$, *then* $\text{dyDisOM}(y^{(s)}) = \text{dyDisOM}(y^{(t)})$.
(2) $\forall r < r' \leq \lceil T^{\text{comma}}/\lambda \rceil : \text{OM}(x^{(r')}) \geq \text{OM}(x^{(r)}) - \varepsilon \ln^4(n)$.

We postpone the proof and first prove the upper bound on T^{comma}, assuming Lemmas 4.2–4.4.

Proof of Theorem 1.1, upper bound on T^{comma} We consider a run of the algorithm on dyDisOM, and make the connection to disOMat the end of the proof. Since the drift of the potential is similar to the drift of the $(1,\lambda)$ EAand the $(1+\lambda)$ EAon OneMax, this part is similar to previous analyses of those situations [48, 49]. We split the run of the $(1,\lambda)$ EAon dyDisOMinto two phases. Let T_1 denote the number of function evaluations until the first time that the $(1,\lambda)$ EAmoves to a point X_{T_1} that has ZM -value at most $n/\lambda - \varepsilon \ln^4(n)$, where ε is as in Lemma 4.4(2). Let T_2 denote the number of function evaluations after moving to X_{T_1} until moving to a point with ZM-value at most k^*. Then $T^{\text{comma,dy}} \leq T_1 + T_2$, since a distorted point with ZM -value at most $k^* + d$ (which has fitness at least $n - k^*$) may have been found before $T_1 + T_2$. We will argue that w.h.p. $T_1, T_2 = O(n \ln n)$.

We first consider T_1, for which we apply a variable drift theorem and Markov's inequality (the presence of the indicator in the potential in (16) prevents direct use of drift theorems with tail bounds, since the decay on the jump size distribution does not decay exponentially in the jump size). By Lemmas 4.1(1-2) and 4.3, we obtain that

$$\Delta(x) \geq \begin{cases} \Omega\left(\frac{\ln \lambda}{2 \ln \ln \lambda}\right), & \text{if } \text{ZM}(x) = k \geq n/\ln \lambda, \\ \Omega(1), & \text{if } \text{ZM}(x) = k \geq n/\lambda - \varepsilon \ln^4 n. \end{cases}$$

The expected potential of the initial solution, $P(x^{(0)})$, is bounded from above as follows. Since $x^{(0)}$ is distorted with probability p, $\mathbb{E}[\mathbb{1}\{x \notin \mathcal{C}_s\}] = p$. Using definitions from Lemma 4.1, $Y^*(k) := \max(Y_1, \dots, Y_\lambda) \leq \sum_{i=1}^{\lambda} Y_i$ and $\mathbb{E}[Y^*(k)] \leq \lambda \mathbb{E}[Y_1] = \lambda k/n \leq \lambda$. Plugging this, along with the trivial bound $\mathrm{ZM}(x^{(0)}) \leq n$, into (16) yields

$$\mathbb{E}[P(x^{(0)})] \leq \mathbb{E}[\mathrm{ZM}(x^{(0)})] + \mathbb{E}\left[\mathbb{1}\{x \notin \mathcal{C}_s\} \frac{\delta}{\lambda p} \mathbb{E}\big[Y^*(\mathrm{ZM}(x^{(0)}))\big]\right] \leq n + \delta.$$

Applying the variable drift theorem (Theorem 2.1) to the potential, the expected number of generations $\mathbb{E}[\lceil T_1/\lambda \rceil]$ is of order at most

$$\begin{aligned}
\mathbb{E}[\lceil T_1/\lambda \rceil] &\leq O\left(1 + \mathbb{E}\left[\int_{n/\ln\lambda}^{P(x^{(0)})} \frac{\ln\ln\lambda + \ln(n/x)}{\ln\lambda} \mathrm{d}x\right] + \int_{n/\lambda - \varepsilon \ln^4(n)}^{n/\ln\lambda} 1 \mathrm{d}x\right) \\
&\leq O\left(1 + \mathbb{E}\left[\int_{n/\ln\lambda}^{P(x^{(0)})} \frac{\ln\ln\lambda + \ln\ln\lambda}{\ln\lambda} \mathrm{d}x\right] + \frac{n}{\ln\lambda}\right) \\
&= O\left(1 + (\mathbb{E}[P(x^{(0)})] - n/\ln\lambda) \cdot \frac{2\ln\ln\lambda}{\ln\lambda} + \frac{n}{\ln\lambda}\right) \\
&\leq O\left(n \frac{\ln\ln\lambda}{\ln\lambda} + \frac{n}{\ln\lambda}\right) = O\Big(n \frac{\ln\ln\lambda}{\ln\lambda}\Big).
\end{aligned} \tag{18}$$

Since the number of function evaluations is by a factor λ larger than the number of generations, it follows by Markov's inequality that

$$\mathbb{P}\big(T_1 \geq n \ln n\big) \leq \frac{O\big(n \frac{\ln\ln\lambda}{\ln\lambda} \lambda\big)}{n \ln n} = O\Big(\frac{\ln\ln\ln n}{\ln\ln n}\Big) = o(1), \tag{19}$$

using that $\lambda = \Theta(\ln(n))$ by Assumption 1.4.

Recall that T_1 is the first time that the current search point x satisfies $\mathrm{ZM}(x) \leq n/\lambda - \varepsilon \ln^4 n$. By Lemma 4.4(2), w.h.p. the $(1, \lambda)$ EAnever visits a search point x' with $\mathrm{ZM}(x') > n/\lambda$ after time T_1. By Lemma 4.3, we have for all x with $\mathrm{ZM}(x) \leq n/\lambda$ that

$$\Delta(x) = \Omega\Big(P(x) \cdot \frac{\lambda}{n}\Big),$$

which corresponds to a multiplicative drift for all the visited search points after T_1 w.h.p. Working under the condition of the above multiplicative drift, we set up to apply the multiplicative drift theorem with tail bounds (Theorem 2.2). Since our aim is to reach a point with $\mathrm{ZM}(x) \leq k^*$, we define a distance function $\Phi(x) := P(x)$ if $\mathrm{ZM}(x) > k^*$ and $\Phi(x) := 0$ otherwise. Thus, reaching a Φ-value of 0 corresponds to our target, and since we only decreased the values for target search points, the expected change in $\Phi(x)$ for all x with $\Phi(x) > 0$ is bounded from below by the drift of the potential:

$$\mathbb{E}[\Phi(x^{(t)}) - \Phi(x^{(t+1)}) \mid \Phi(x^{(t)}) > 0] \geq \Delta(x^{(t)}) = \Omega\left(\Phi\left(x^{(t)}\right) \cdot \tfrac{\lambda}{n}\right).$$

Hence we also have a multiplicative drift with respect to Φ. The maximum Φ -value is bounded by the maximum potential in the regime $\mathrm{ZM}(x) \leq n/\lambda$, $\Phi_{\max} := \frac{n}{\lambda} + \frac{\delta}{\lambda p} \cdot \max_k \mathbb{E}[Y^*(k)] \leq n + \frac{\delta}{p}$ using $\mathbb{E}[Y^*(k)] \leq \lambda$ as derived above. Since $p = \omega(1/(n \ln n))$ and δ is constant, $\Phi_{\max} = o(n \ln n)$. Recall that T_2 is the random number of evaluations until reaching a point with $\mathrm{ZM}(x) \leq k^*$ from X_{T_1} and that the corresponding number of generations is $\lceil T_2/\lambda \rceil$. Applying the multiplicative drift theorem with tail bounds (Theorem 2.2) to Φ yields that, for any $r > 0$,

$$\mathbb{P}\Big(\lceil T_2/\lambda \rceil > \Theta\big(\tfrac{n}{\lambda}(\ln(\Phi_{\max}) + r)\big)\Big) \leq \exp(-r).$$

Choosing $r := \ln n$ and using $\ln(\Phi_{\max}) = O(\ln n)$ yields that w.h.p. the number of generations to reach a point with $\mathrm{ZM}(x) \leq k^*$ from X_{T_1} is at most $O(n \ln(n)/\lambda)$. The number of function evaluations in this phase is by a factor λ larger, namely $O(n \ln n)$, as required. Combined with (18) this implies that $T^{\text{comma,dy}} = O(n \ln n)$ w.h.p.

We will now translate this bound on DYDISOM(with dynamic noise) to a bound on DISOM(with frozen noise). For DYDISOMand DISOM, each point is distorted at the first time it is sampled with probability p independently of other points. Consequently, $x^{(0)}$ is clean w.h.p. Moreover, by Lemma 4.4(1) w.h.p. the function values of points sampled in DYDISOMnever change, so the runs on the dynamic and the 'frozen' model are identical w.h.p. Hence, the upper bound in (1) on $T^{\text{comma}}_{k^*,\lambda,d,p}$ follows from $T^{\text{comma,dy}}_{k^*,\lambda,d,p} = O(n \ln n)$ with high probability.

It is left to prove Lemmas 4.2–4.4.

Proof of Lemma 4.2 *Part 1.* The lower bound $\mathbb{E}[Y^*(k + \ln(n))] \geq \mathbb{E}[Y^*(k)]$ is trivial, since $\mathbb{E}[Y^*(k)]$ is increasing in k: $Y^*(k)$ is the maximum of λ iid binomial random variables with parameters k and $1/n$. We now show a helping statement for the upper bound. Let $0 \leq k_1 < k_2$, we show that for two independent random variables $Y^*(k_1)$ and $Y^*(k_2)$ it holds that $\mathbb{E}[Y^*(k_1 + k_2)] \leq \mathbb{E}[Y^*(k_1) + Y^*(k_2)]$. Let $Z_{i,j} \sim \mathrm{Ber}(1/n)$ be i. i. d. Bernoulli random variables for $i, j \geq 1$, so we may couple the random variables as follows

$$\begin{aligned} Y^*(k_1 + k_2) &= \max_{i \leq \lambda}(Z_{i,1} + \cdots + Z_{i,k_1+k_2}), \\ Y^*(k_1) &= \max_{i \leq \lambda}(Z_{i,1} + \cdots + Z_{i,k_1}), \\ Y^*(k_2) &= \max_{i \leq \lambda}(Z_{i,k_1+1} + \cdots + Z_{i,k_2}). \end{aligned}$$

For any s,

$$\begin{aligned} \mathbb{P}\big(Y^*(k_1 + k_2) - Y^*(k_1) \geq s\big) &\leq \mathbb{P}\big(\exists i \leq \lambda : Z_{i,k_1+1} + \cdots + Z_{i,k_1+k_2} \geq s\big) \\ &= \mathbb{P}\big(Y^*(k_2) \geq s\big). \end{aligned}$$

Hence, $Y^*(k_1+k_2)-Y^*(k_1)$ is stochastically dominated by $Y^*(k_2)$. Consequently, $\mathbb{E}[Y^*(k+\ln n)] \leq \mathbb{E}[Y^*(k)] + \mathbb{E}[Y^*(\ln n)]$ by stochastic domination and linearity of expectation (cf. Lemma 1.8.2 in [38]). Part 1 follows if we show that $\mathbb{E}[Y^*(\ln n)] = o\big(\mathbb{E}[Y^*(k)]\big)$ for all $k \geq k^*$. We apply Lemma 4.1(3) for $k_{4.1} = \ln n$, using that $k^* = n^{\Omega(1)}$ and $k^* = n^{1-\Omega(1)}$ by Assumption 1.4, to obtain

$$\mathbb{E}[Y^*(\ln n)] = (1-o(1))\frac{\lambda \ln n}{n} \ll (1-o(1))\frac{\lambda k^*}{n} = \mathbb{E}[Y^*(k)].$$

Parts 2–4. $\mathbb{E}[Y^*(k)]$ is minimized at $k = k^*$, and thus by Lemma 4.1(3) $\mathbb{E}[Y^*(k)] = \Omega(\lambda k/n)$. So for part 2 we have to verify that $p \ln n = o(k/n)$, which follows from (8). Analogously, for part 3 we have to verify $q \ln n = o(\lambda k/n)$, which also follows from (7) and (8). For part 4, we need $1 = o(\lambda k)$, which holds as both $\lambda, k = \omega(1)$.

Proof of Lemma 4.3 We will establish a lower bound on the drift $\Delta(x)$ defined in (17). Define

$$\Delta^+(x) := \mathbb{E}[\max\{P(x^{(t)}) - P(x^{(t+1)}), 0\} \mid x^{(t)} = x],$$
$$\Delta^-(x) := \mathbb{E}[\max\{P(x^{(t+1)}) - P(x^{(t)}), 0\} \mid x^{(t)} = x],$$

so that

$$\Delta(x) = \Delta^+(x) - \Delta^-(x). \tag{20}$$

We will first obtain lower bounds on $\Delta^+(x)$, then obtain upper bounds on $\Delta^-(x)$, distinguishing in both cases between clean and distorted points x. Then we combine the bounds and find $\Delta(x) = \Omega(\mathbb{E}[Y^*(k)])$ under Assumption 1.4. Eventually we obtain an upper bound on $\mathbb{P}(T^{\text{comma}} \geq n \ln^2 n)$. □

4.2 Forward Progress from Clean Points

First, assume that x is a clean point with $\mathrm{ZM}(x) = k$. Let X_i denote the number of bits flipped from 1 to 0 for the i-th offspring (with $i \leq \lambda$), similarly let Y_i be the number of bits flipped from 0 to 1 for the i-th offspring. Moreover, we write $Y^*(k) := \max_i(Y_i)$ and i^* for the index such that $Y_{i^*} = Y^*(k)$ (with arbitrary tie-breaking rule). Let $\mathcal{E}_1$ be the event that all offspring are clean. Then,

$$\begin{aligned}\Delta^+(x) &\geq \mathbb{E}[\max_{i\leq\lambda}(Y_i - X_i, 0)\mathbb{1}\{\mathcal{E}_1\}] + \mathbb{E}[\max\{P(x^{(t)}) - P(x^{(t+1)}), 0\}\mathbb{1}\{\neg\mathcal{E}_1\} \mid x^{(t)} = x] \\ &\geq \mathbb{E}[Y^*(k)\mathbb{1}\{X_{i^*} = 0\}\mathbb{1}\{\mathcal{E}_1\}],\end{aligned} \tag{21}$$

considering only the case where all offspring are clean, and in which the i^*-th offspring has no 1-bits flipped into a 0-bit, so $X_{i^*} = 0$. We first condition on the set of all mutated bit positions where x is 0 (which uniquely determines $Y^*(k)$), and then on the set of all mutated bit positions. In the formula we abbreviate those sets by "mutated 0-bits" and "mutated bits" respectively. We obtain

$$\begin{aligned}\Delta^+(x) &\geq \mathbb{E}\Big[\mathbb{E}\big[Y^*(k)\mathbb{1}\{X_{i^*}=0\}\mathbb{E}[\mathbb{1}\{\mathcal{E}_1\} \mid \text{ mutated bits}] \mid \text{mutated 0-bits}\big]\Big] \\ &= \mathbb{E}\Big[Y^*(k)\mathbb{E}\big[\mathbb{1}\{X_{i^*}=0\}\mathbb{P}\big(\mathcal{E}_1 \mid \text{ mutated bits}\big) \mid \text{mutated 0-bits}\big]\Big],\end{aligned}$$

where the outer expectation is taken over the possible sets of mutated bits. Since already visited clean points remain clean, and newly visited points are distorted with probability p, by a union bound $\mathbb{P}\big(\mathcal{E}_1 \mid \text{ mutated bits}\big) \geq 1-\lambda p = 1-o(1)$. By independence, the probability that no 1-bits are mutated is at least $(1-1/n)^{n-k} \geq 1/e$. So we obtain for clean x that

$$\Delta^+(x) \geq (1-o(1))\mathbb{E}[Y^*(k)]/e. \tag{22}$$

4.3 Forward Progress from Distorted Points

If x is a distorted point, we consider the additional event $\mathcal{E}_2$ that there is no clone of the current individual among the offspring, and the event $\mathcal{E}_3$ that there exists i such that $X_i \leq 1$. As a result, conditional on the event $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ there is no clone of the parent, all offspring are clean, and the number of 0-bits increases by at most 1, i.e., $\mathrm{ZM}(x^{(t+1)}) - \mathrm{ZM}(x^{(t)}) \leq 1$. Hence, for sufficiently large n,

$$\begin{aligned}\Delta^+(x) &\geq \mathbb{E}[\big(P(x^{(t)}) - P(x^{(t+1)})\big)\mathbb{1}\{\mathcal{E}_1\}\mathbb{1}\{\mathcal{E}_2\}\mathbb{1}\{\mathcal{E}_3\} \mid x^{(t)}=x] \\ &\geq \big(\delta/(\lambda p)\cdot\mathbb{E}[Y^*(k)]-1\big)\cdot\mathbb{P}\big(\mathcal{E}_1\cap\mathcal{E}_2\cap\mathcal{E}_3\big) \\ &\geq \delta/(2\lambda p)\cdot\mathbb{E}[Y^*(k)]\cdot\mathbb{P}\big(\mathcal{E}_1\cap\mathcal{E}_2\cap\mathcal{E}_3\big),\end{aligned}$$

since $\frac{\mathbb{E}[Y^*(k)]}{\lambda p} = \omega(1)$ by Lemma 4.2(2). We focus on $\mathbb{P}\big(\mathcal{E}_1\cap\mathcal{E}_2\cap\mathcal{E}_3\big)$. We first condition on the set of all mutated bits, and then take an expectation over all such set. This set determines whether the events $\mathcal{E}_2$ and $\mathcal{E}_3$ hold, but does not contain the information to determine if one of the offspring is distorted, i.e., the event $\mathcal{E}_1$. We obtain by the law of total probability and a union bound over the possibly distorted offspring that

$$\begin{aligned}\mathbb{P}\big(\mathcal{E}_1\cap\mathcal{E}_2\cap\mathcal{E}_3\big) &= \mathbb{E}\big[\mathbb{1}\{\mathcal{E}_2\cap\mathcal{E}_3\}\cdot\mathbb{P}\big(\mathcal{E}_1 \mid \text{ mutated bits}\big)\big] \\ &\geq (1-\lambda p)\cdot\mathbb{P}\big(\mathcal{E}_2\cap\mathcal{E}_3\big) \\ &\geq (1-o(1))\cdot\big(\mathbb{P}(\text{ no clone}) - \mathbb{P}(\forall i\leq\lambda : X_i\geq 2)\big),\end{aligned}$$

applying a union bound and substituting the definition of $\mathcal{E}_2$ and the complement of $\mathcal{E}_3$ in the second inequality. By Lemma 3.2 and using that all $X_i + Y_i \sim \mathrm{Bin}(n,1/n)$ are iid, it follows that for n sufficiently large (using $\lambda=\omega(1)$ and $1/\eta = 0.63.. > 1/2$)

$$\begin{aligned}\mathbb{P}\big(\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3\big) &\geq (1-o(1))\left((1/\eta)^\lambda - \left(\binom{n}{2}\frac{1}{n^2}\right)^\lambda\right)\\ &\geq (1-o(1))\big((1/\eta)^\lambda - (1/2)^\lambda\big)\\ &= (1-o(1))\big(q - 2^{-\lambda}\big) \geq q/2,\end{aligned}$$

since $q = \eta^{-\lambda}$ by definition. We obtain for n large that

$$\Delta^+(x) \geq \delta\mathbb{E}[Y^*(k)]\frac{q}{4\lambda p} \geq \mathbb{E}[Y^*(k)], \tag{23}$$

since $p\lambda = O(p\ln(n)) = o(q)$ by Assumption 1.4.

Combined with (22), this implies $\Delta^+(x) \geq \mathbb{E}[Y^*(k)]/(2e)$ for both clean and distorted x when n is sufficiently large.

We will now establish upper bounds on $\Delta^-(x)$.

4.4 Backwards Progress from Clean Points

Let $\mathcal{E}_4$ be the event that there exists a distorted offspring, $\mathcal{E}_5$ be the event that all offspring flip at most $\ln(n)$ bits, and recall that $\mathcal{E}_2$ denotes the event that there is no clone of the parent. We distinguish several cases depending on the occurrence of the events $\mathcal{E}_2$, $\mathcal{E}_4$, and $\mathcal{E}_5$. For any expression r, we define $r^+ := \max\{r, 0\}$. We abbreviate $P_t = P(x^{(t)})$ and obtain

$$\begin{aligned}\Delta^-(x) = \mathbb{E}\big[(P_{t+1}-P_t)^+\mathbb{1}\{\mathcal{E}_4\}\mathbb{1}\{\mathcal{E}_5\}\big]\\ + \mathbb{E}\big[(P_{t+1}-P_t)^+\mathbb{1}\{\neg\mathcal{E}_2\}\mathbb{1}\{\neg\mathcal{E}_4\}\mathbb{1}\{\mathcal{E}_5\}\big]\\ + \mathbb{E}\big[(P_{t+1}-P_t)^+\mathbb{1}\{\mathcal{E}_2\}\mathbb{1}\{\neg\mathcal{E}_4\}\mathbb{1}\{\mathcal{E}_5\}\big]\\ + \mathbb{E}\big[(P_{t+1}-P_t)^+\mathbb{1}\{\neg\mathcal{E}_5\}\big].\end{aligned} \tag{24}$$

We consider the four terms in the sum separately. For the first term, the indicator $\mathbb{1}\{x \notin \mathcal{C}_s\}$ in (16) is 0 for $x^{(t)}$ and is potentially 1 for $x^{(t+1)}$, in which case the additional term is at most $\frac{\delta}{\lambda p}\cdot\mathbb{E}[Y^*(k+\ln n)]$, using that the expectations $\mathbb{E}[Y^*(k)]$ are increasing in k, and that the maximal jump size is bounded from above by $\ln(n)$ due to the intersection with $\mathcal{E}_5$. Note that the factor $\mathbb{1}\{\mathcal{E}_5\}$ allows us to assume that the event $\mathcal{E}_5$ occurs without needing to condition on it. Since $\mathbb{E}[Y^*(k+\ln(n))] = (1+o(1))\mathbb{E}[Y^*(k)]$ by Lemma 4.2, we have

$$\begin{aligned}\mathbb{E}\big[(P_{t+1}-P_t)^+\mathbb{1}\{\mathcal{E}_4\}\mathbb{1}\{\mathcal{E}_5\}\big] &\leq \big(\ln n + \tfrac{\delta}{\lambda p}\cdot\mathbb{E}[Y^*(k+\ln(n))]\big)\cdot\mathbb{P}\big(\mathcal{E}_4\big)\\ &\leq (1+o(1))\big(\lambda p\ln(n) + 2\delta\cdot\mathbb{E}[Y^*(k)]\big),\end{aligned} \tag{25}$$

using also $\mathbb{P}\big(\mathcal{E}_4\big) = \mathbb{P}\big(\exists\text{distorted offspring}\big) \leq \lambda p$ by a union bound.

The second term in (24) equals 0: there is no distorted offspring conditionally on $\neg\mathcal{E}_4$, and there exists a clone of the parent conditionally on $\neg\mathcal{E}_2$, so the ZM value does not increase and the indicators in (16) are both 0.

For the third term in (24) we observe that the potential can increase by at most $\ln(n)$ since there is no distorted offspring. Conditional on the event $\mathcal{E}_2$ there is no clone of the parent, so by Lemma 3.2

$$\mathbb{E}\big[(P_{t+1} - P_t)^+ \mathbb{1}\{\mathcal{E}_2\}\mathbb{1}\{\neg\mathcal{E}_4\}\mathbb{1}\{\mathcal{E}_5\}\big] \leq \ln(n) \cdot \mathbb{P}\big(\nexists \text{ clone}\big) = (1+o(1))\ln(n)q. \quad (26)$$

We turn to the fourth term in (24). We use that the maximal difference in potential between two search points is $n + \delta\mathbb{E}[Y^*(n)]/(\lambda p)$. Since $Y^*(n)$ is the maximum of $\lambda = \Theta(\log n)$ iid $\mathrm{Bin}(n, 1/n)$ random variables, $\mathbb{E}[Y^*(n)] = O(\log\lambda) = O(\ln\ln n)$ by Lemma 4.1(4). Thus, the maximal difference in potential is $O(n\ln\ln n)$, using also that $p = \omega(1/(n\ln n))$ by Assumption 1.4. By Lemma 3.3 and a union bound over the λ offspring

$$\mathbb{E}\big[(P_{t+1} - P_t)^+ \mathbb{1}\{\neg\mathcal{E}_5\}\big] \leq o(n\ln^2 n) \cdot \lambda \cdot \mathbb{P}\big(\text{ offspring flips } \geq \ln(n) \text{ bits }\big) = o(1/n). \quad (27)$$

Uniting the bounds (24–27) and recalling that the second term in (24) is 0, we obtain for clean x that

$$\Delta^-(x) \leq (1+o(1))2\delta\mathbb{E}[Y^*(k)] + O(\lambda p\ln(n)) + O(q\ln(n)) + o(1/n). \quad (28)$$

The last three terms are of smaller order than $\mathbb{E}[Y^*(k)]$ by Lemma 4.2(2-4). Combining this bound with the forward drift $\Delta^+(x)$ from (22), and substituting the bounds from Lemma 4.2(2–4), it follows that for a constant $\delta \in (0, 1/(4e))$

$$\Delta(x) \geq \Omega\big(\mathbb{E}[Y^*(k)]\big). \quad (29)$$

We turn to the proof of Lemma 4.3(2) starting from a clean point. By the definition of the potential in (16) for clean points, it remains to show that $\mathbb{E}[Y^*(k)] = \Omega\big(\frac{\lambda k}{n}\big)$ when $k \leq n/\lambda$. This follows from Lemma 4.1(2).

4.5 Backwards Progress from Distorted Points

Recall that $\mathcal{E}_2$ denotes the event that there is no clone of the current search point among the offspring. We aim to bound

$$\Delta^-(x) = \mathbb{E}[(P_{t+1} - P_t)^+ \mathbb{1}\{\mathcal{E}_2\}] + \mathbb{E}[(P_{t+1} - P_t)^+ \mathbb{1}\{\neg\mathcal{E}_2\}].$$

We first consider the case when there is a clone, i.e., the event $\mathcal{E}_2$ does not hold. If the selected offspring is again distorted, then it must have the same or a smaller ZM -value than x to be accepted, and the indicator in (16) does not increase since the expectations $\mathbb{E}[Y^*(k)]$ are increasing in k. If the selected offspring is *not* distorted, then it must also have smaller ZM-value, and the indicator decreases. Overall, the potential cannot increase when there is a clone of x. Hence,

$$\Delta^-(x) = \mathbb{E}[(P_{t+1} - P_t)^+ \mathbb{1}\{\mathcal{E}_2\}]. \quad (30)$$

Distinguishing whether there is an offspring that flips more than $\ln(n)$ bits (event $\mathcal{E}_5$), we obtain by the same reasoning as in (27)

$$\begin{aligned}\Delta^-(x) &\le \mathbb{E}[(P_{t+1}-P_t)^+\mathbb{1}\{\mathcal{E}_2\}\mathbb{1}\{\mathcal{E}_5\}] + \mathbb{E}[(P_{t+1}-P_t)^+\mathbb{1}\{\mathcal{E}_2\}\mathbb{1}\{\neg\mathcal{E}_5\}] \\ &\le \mathbb{E}[(P_{t+1}-P_t)^+\mathbb{1}\{\mathcal{E}_2\}\mathbb{1}\{\mathcal{E}_5\}] + \mathbb{E}[(P_{t+1}-P_t)^+\mathbb{1}\{\neg\mathcal{E}_5\}] \\ &\le \mathbb{E}[(P_{t+1}-P_t)^+\mathbb{1}\{\mathcal{E}_2\}\mathbb{1}\{\mathcal{E}_5\}] + o(1/n).\end{aligned} \tag{31}$$

On the event that there are at most $\ln(n)$ bit flips, considering the worst case that the selected offspring is distorted, we obtain by Lemma 4.2(1)

$$\begin{aligned}P_{t+1}-P_t &\le \mathrm{ZM}(x^{(t+1)}) - \mathrm{ZM}(x^{(t)}) + \tfrac{\delta}{\lambda p}\cdot\mathbb{E}[Y^*(\mathrm{ZM}(x^{(t+1)})) - Y^*(\mathrm{ZM}(x^{(t)}))] \\ &\le \ln(n) + \tfrac{\delta}{\lambda p}\cdot o\big(\mathbb{E}[Y^*(\mathrm{ZM}(x^{(t)}))]\big).\end{aligned}$$

Substituting this back into (31) and using that by Lemma 3.2 $\mathbb{P}\big(\mathcal{E}_2\big) = \mathbb{P}\big(\text{ no clone }\big) \le 2q$, we obtain by Lemma 4.2(1)

$$\Delta^-(x) \le 2q\cdot\big(\ln(n) + \tfrac{\delta}{\lambda p}\cdot o\big(\mathbb{E}[Y^*(k)]\big)\big) + o(1/n).$$

By Lemma 4.2, we have $q\ln(n), 1/n = o\big(\mathbb{E}[Y^*(k)]\big)$. Recalling the lower bound on $\Delta^+(x)$ from (23), we obtain

$$\begin{aligned}\Delta(x) &\ge \frac{q\delta}{\lambda p}\Big(\tfrac{1}{4}\mathbb{E}[Y^*(k)] - o\big(\mathbb{E}[Y^*(k)]\big)\Big) - o\big(\mathbb{E}[Y^*(k)]\big) = \Omega\big(\tfrac{q}{\lambda p}\mathbb{E}[Y^*(k)]\big) \\ &= \Omega(\mathbb{E}[Y^*(k)]),\end{aligned}$$

using that $q/(\lambda p) = \omega(1)$ by Assumption 1.4.

To finish the proof of the first part of Lemma 4.3, we are left to show that $\Delta(x) = \Omega(P(x)\frac{\lambda}{n})$ when $\mathrm{ZM}(x) \le n/\lambda$ and x is distorted. Let $k = \mathrm{ZM}(x)$. By definition of the potential in (16), since x is distorted, $P(x) = k + \frac{\delta}{\lambda p}\mathbb{E}[Y^*(k)]$. Along with the bound $\Delta(x) = \Omega\big(\frac{q}{\lambda p}\mathbb{E}[Y^*(k)]\big)$ shown above, it suffices to prove the second equality in

$$\Delta(x) = \Omega\Big(\frac{q}{\lambda p}\mathbb{E}[Y^*(k)]\Big) = \Omega\Big(\frac{\lambda k}{n} + \frac{1}{pn}\mathbb{E}[Y^*(k)]\Big) = \Omega\Big(P(x)\frac{\lambda}{n}\Big),$$

which is implied if both

$$\frac{q}{\lambda p}\mathbb{E}[Y^*(k)] = \Omega\Big(\frac{\lambda k}{n}\Big) \qquad \text{and} \qquad \frac{q}{\lambda p}\mathbb{E}[Y^*(k)] = \Omega\Big(\frac{1}{pn}\mathbb{E}[Y^*(k)]\Big) \Leftrightarrow \frac{q}{\lambda} = \Omega\Big(\frac{1}{n}\Big).$$

The bound on the left-hand side holds since $\mathbb{E}[Y^*(k)] = \Omega(\lambda k/n)$ by Lemma 4.1(2) when $k \le n/\lambda$, and $q/(\lambda p) = n^{\Omega(1)} = \Omega(1)$ by (8). We prove the bound on the right-hand side using Assumption 1.4 and $\lambda = \Theta(\log n)$ by (7).

$$q/\lambda \geq p^{1-\varepsilon}/\lambda = \omega\big((n\log n)^{\varepsilon-1}/\log n\big) = \omega\big(n^{\varepsilon}(\log n)^{\varepsilon-2}/n\big) = \Omega(1/n).$$

4.6 Upper Bound $T^{\text{comma,dyn}}$

We will now obtain an upper bound on $T^{\text{comma,dy}}$ that holds w.h.p. We can now use that $\Delta(x) = \Omega(\mathbb{E}[Y^*(k)])$. Moreover, by Lemmas 4.3(1) and 4.1(2), we have $\Delta(x) = \Omega(\min\{1, \lambda k/n\})$. By the variable drift theorem (Theorem 2.1), we obtain for the number of generations $\lceil T^{\text{comma,dy}}/\lambda \rceil$ that

$$\mathbb{E}\big[\lceil T^{\text{comma,dy}}/\lambda \rceil\big] \leq O\left(\frac{n}{\lambda} + \int_1^{n/\lambda} \frac{1}{\lambda x/n}\mathrm{d}x + \int_{n/\lambda}^{n} 1\mathrm{d}x\right) = O(n)$$

for $\lambda = \Theta(\ln(n))$.

The number of function evaluations is by a factor λ larger. Hence, by Markov's inequality,

$$\mathbb{P}\big(T^{\text{comma,dy}} \geq n\ln^2(n)\big) \leq \frac{O(n\ln(n))}{n\ln^2(n)} = o(1).$$

□

Proof of Lemma 4.4 We first verify the first part of Lemma 4.4, and verify the second statement at the end. The only way there could exist $t' < t$ such that $y^{(t')} = y^{(t)}$, but $\textsc{dyDisOM}(y^{(t')}) \neq \textsc{dyDisOM}(y^{(t)})$, is when $y^{(t')}$ is distorted and resampled at a later time t at which it is clean (clean points remain clean). Hence, it suffices to argue that w.h.p. the algorithm never resamples a point that was distorted at the first time it was sampled. To do so, we will analyse the positive and negative jumps on the OneMaxscale that the $(1,\lambda)$ EAmay perform on dyDisOMin polynomial time intervals.

Let $s := \lceil C(n/k^*)\ln^4(n)\rceil$ for some large constant $C > 0$, and define for the t-th function evaluation the random times $t_1(t), \ldots, t_s(t)$, which are the first s unique evaluations after the $(t-1)$-st evaluation from a clean parent. If for some $j \in [s]$ there are no j unique evaluations after the $(t-1)$-st evaluation from a clean point because the algorithm terminated, we set $t_j(t) := T^{\text{comma,dy}}$. Formally, we define for $j \in [s]$

$$\mathcal{T}_c := \{t : x^{(\lceil t/\lambda\rceil - 1)} \text{ is clean}\}, \quad t_s(t) := \min\left\{\tau \in [t, T^{\text{comma,dy}}] : \left|\bigcup_{r\in\mathcal{T}_c, r\geq t} \{y^{(r)}\}\right| = j\right\},$$

and set $t_j(t) := T^{\text{comma,dy}}$ if there is no such τ.

We will show that there exists $C', \varepsilon > 0$ such that w.h.p. the following four events hold:

(i) $\forall t \leq n^2$, the total time in the interval $[t, t_s(t)]$ at which the parent is distorted is at most $O(\ln^3(n)/q)$.
(ii) $\forall t \leq n^2, r \in [t_s(t), T^{\text{comma,dy}}]$, each sampled point $y^{(r)}$ satisfies $\text{OM}(y^{(r)}) > \text{OM}(x^{(\lceil t/\lambda \rceil - 1)}) + C' \ln(n)$.
(iii) $\forall t \leq n^2$, each sampled point $y^{(r)}$ satisfies $|\text{OM}(y^{(t)}) - \text{OM}(x^{(\lceil t/\lambda \rceil - 1)})| < C' \ln(n)$.
(iv) the number of sampled distorted points until the fixed target is reached is $O(pn \ln(n))$.

We stress that events (i–iii) are defined for all $t \leq n^2$ *simultaneously*. Below we invoke Lemma 4.3 to argue that we do not need to consider the run of the algorithm for $t > n^2$ as $T^{\text{comma,dy}} \leq n^2$ w.h.p. We first show that the statement follows under the assumption that these events hold w.h.p. Recall that it is sufficient to bound the probability that a distorted point is resampled on the intersection of the events (i-iv). If an offspring $y^{(t)}$ with $\text{OM}(y^{(t)}) = \ell$ is distorted, then by event (iii) its parent (which is $x^{(\lceil t/\lambda \rceil - 1)}$) has OM-value at least $\ell - C' \ln(n)$. By event (ii-iii), the only times at which the distorted point $y^{(t)}$ can be resampled is during the interval $[t, t_s(t)]$. At each time $r \in [t, t_s(t)]$ the probability that $y^{(t)}$ is resampled is at most $1/n$ since at least one bit needs to be flipped from the clean parent $x^{(\lceil t/\lambda \rceil - 1)}$. By a union bound over the precisely $s = \Theta((n/k^*) \ln^4(n))$ times in $[t, t_s(t)]$ at which the parent is clean (definition of $t_1(t), \ldots, t_s(t)$), and the $O(\ln^3(n)/q)$ times at which the parent is distorted (event (i)), the probability that the point $y^{(r)}$ is resampled in this interval is at most

$$O(\ln^4(n)/k^* + \ln^3(n)/(nq)).$$

By a union bound over the at most $O(pn \ln(n))$ distorted points visited (event (iv)), no distorted point is resampled with probability at least

$$\begin{aligned} &1 - O(pn \ln(n) \cdot (\ln^4(n)/k^* + \ln^3(n)/(nq))) \\ &\quad = 1 - O(\max\{\ln^5(n) \cdot p \cdot n/k^*, \ln^4(n)p/q\}) = 1 - o(1), \end{aligned}$$

since $p \leq k^*/n^{1+\delta} = o(k^*/(n \ln^5(n)))$ and $p \leq qn^{-\delta} = o(q/\ln^4(n))$ by Assumption 1.4, see (8). Thus, the first lemma statement follows if we show that the four events hold with high probability.

4.7 Preparations Events (i–ii)

We will analyse the $(1, \lambda)$ EAon DyDisOMduring intervals $[t, t_s(t)]$ to obtain bounds for events (i-ii). We start with the number of OM-improving steps in $[t, t_s(t)]$, for which we only consider steps at which the parent is clean. Similar to Lemma 3.1, the probability of an OM-improving step (for $r \in [t, t_s(t)]$ such that $r \leq T^{\text{comma,dy}}$) is at least $c\lambda k^*/n$ for some small constant $c > 0$, independently of the history. So the expected number of OM-improving steps is at least $\lfloor s/\lambda \rfloor \cdot c\lambda k^*/n = \Omega(s \cdot k^*/n) = \Omega(\ln^4(n))$. By a Chernoff bound it follows that

w.h.p., for all $t \leq n^2$ the number of OM-improving generations in $[t, t_s(t)]$ is at least $\hat{C} \ln^4(n)$ for some constant $\hat{C}$ that depends on the sufficiently large constant C in the definition of s.

We move on to the drift away from the target during the interval $[t, t_s(t)]$. We will bound the total number of times that the algorithm decreases the OM value from above, distinguishing between jumps from/to either clean or distorted points. We first show that the number of offspring (including non-unique evaluations) created from clean parents in the interval $[t, t_s(t)]$ is at most $3\,s$ for all $t \leq n^2$ simultaneously. To show this, we argue that each offspring $r \in [t, t_s(t)]$ has not been sampled in $[t, r-1]$ with probability at least 1/2 for all n sufficiently large, independently of the past. With probability $1/e + o(1)$, $x^{(r)}$ is a clone of the parent. In time $[t, t_s(t) - 1]$, at most $s - 1$ unique evaluations from a clean parent have been made by definition. Therefore, with probability at most $(s-1)/n = o(1)$, $x^{(r)}$ coincides with one of the previous unique evaluations other than the parent, and $x^{(r)}$ is a unique evaluation with probability at least $1 - 1/e - o(1) \geq 1/2$. As a result, by a Chernoff bound,

$$\begin{aligned} \mathbb{P}(\exists t \leq \min(n^2, T^{comma,dy}) : \quad & \#\text{off spring sampled } in[t, t_s(t)]\text{from clean parents}) \\ & \text{is } at \text{ least } 3s \\ & \leq n^2 \mathbb{P}(Bin(3s, 1/2) \geq s) \\ & \leq n^2 \exp(-\Omega(s)) = n^2 \exp(-\Omega(\ln^4(n))) = o(1). \end{aligned} \tag{32}$$

4.8 Clean to Clean

We start with backward jumps from clean points to other clean points, whose total size is w.h.p. bounded from above by $O(\ln(n))$ times the number of generations in which there is no clone of the parent (the upper bound on the jump size comes from Lemma 3.3). Since there is no clone of the parent with probability at most $2q$ by Lemma 3.2, the number of generations without clone is stochastically dominated by a Binomial random variable $\mathrm{Bin}(\lceil 3s/\lambda \rceil, 2q) \preccurlyeq \mathrm{Bin}(\max\{\lceil 3s/\lambda \rceil, \ln(n)/(2q)\}, 2q)$, which has expectation $\max(2q\lceil 3s/\lambda \rceil, \ln(n))$. Here, we used that the number of generations in $[t, t_s(t)]$ is at most $3s/\lambda$ w.h.p. by (32); the maximum is useful in the next steps to guarantee polynomial decay of error probabilities via Chernoff bounds in the following steps.

By the Chernoff bound in Theorem 2.4, the probability that for a fixed t the number of non-clone generations in $[t, t_s(t)]$ exceeds $(12+1)\max\{2q\lceil 3\,s/\lambda \rceil, \ln(n)\}$ is at most

$$\exp\big(-\tfrac{12}{3}\max(2q\lceil 3s/\lambda \rceil, \ln(n))\big) \leq \exp\big(-4\ln(n)\big) = n^{-4}. \tag{33}$$

By a union bound, with probability $1 - O(n^{-2})$ there is no $t \leq n^2$ such that the number of non-clone generations in $[t, t_s(t)]$ exceeds $(12+1)\max\{2q\lceil 3\,s/\lambda \rceil, \ln(n)\}$. By (8) which follows from Assumption 1.4, and $s = \Theta((\ln^4 n)n/k^*)$, it follows that $qs/\lambda = \Theta((\ln^3 n)n^{-\delta}) = o(\ln n)$, and thus w.h.p. there is no $t \leq n^2$ such that the number of non-clone generations in $[t, t_s(t)]$ exceeds $13\ln(n)$. The jump sizes away from the optimum from all such jumps sum up to at most $O(\ln^2(n))$.

4.9 Clean to Distorted, Distorted to Clean

The number of distorted offspring sampled in $[t, t_s(t)]$ from clean parents is stochastically dominated by $\mathrm{Bin}(3\,s, p) \preccurlyeq \mathrm{Bin}(\max\{3\,s, \ln(n)/p\}, p)$: the number of offspring sampled from a clean parent is at most $3s$ w.h.p. by (32), and in DYDISOM-previously sampled clean points remain clean, while other points are distorted independently with probability p. Similar to the application of the Chernoff bound in (33),

$$\begin{aligned}
&\mathbb{P}\left(\exists t \le \min(n^2, T^{\text{comma,dy}}) : \begin{array}{l}\#\text{distorted offspring sampled from clean parents} \\ \text{in}[t, t_s(t)]\text{is at least } 13\max(3sp, \ln n)\end{array}\right) \\
&\quad \le \mathbb{P}\Big(\mathrm{Bin}(\max\{3s, \ln(n)/p\}, p) \ge 13\max(3sp, \ln n)\Big) \\
&\quad \le n^2 \exp\big(-\tfrac{12}{3}\max(3sp, \ln n)\big) \le n^{-2}
\end{aligned}$$

By (8), $sp = \Theta((\ln^4 n)pn/k^*) = O((\ln^4 n)n^{-\delta}) = o(\ln n)$, so the number of distorted offspring sampled from clean points is at most $13\ln n$ w.h.p. When jumping from a clean point to a distorted point, by Lemma 3.3, the OM-value decreases by at most $O(\ln(n))$. Hence, for all t the total decrease of OM-value during the interval $[t, t_s(t)]$ is $O(\ln^2(n))$. The number of jumps from a distorted point to a clean point is also $O(\ln(n))$, since it is bounded from above by the number of sampled distorted points from a clean parent, and at each step the decrease is at most $O(\ln(n))$.

4.10 Distorted to Distorted

To analyse the number of jumps between distorted points, we establish an upper bound on the total number of jumps between distorted points before jumping to a clean point. The probability of moving between two generations is at least q, considering only the case in which there is no clone of the parent. The probability of moving from a distorted point to a distorted point is at most λp. Hence, the probability of jumping from a distorted point to another distorted point at its next move is bounded from above by $\lambda p/q \le 1/2$ (using that $p\ln(n)/q = o(1)$ by Assumption 1.4), independently of the past. Thus, the number of consecutive moves between distorted points is stochastically dominated by a geometric random variable $\mathrm{Geo}(1/2)$. Hence, the probability that there are $6\ln(n)$ consecutive moves between distorted points starting from a distorted point is at most

$$(1-1/2)^{3\ln(n)/(1/2)} \le \exp(-3\ln(n)). \tag{34}$$

By a union bound over the at most n^2 distorted points, this event does not occur for any of the first n^2 distorted points w.h.p. Combined with the at most $O(\ln(n))$ sampled distorted offspring from clean points, for all t the total number of visited distorted points in the time interval $[t, t_s(t)]$ is at most $O(\ln^2(n))$, and at each jump between distorted points away from the optimum is at most $O(\ln(n))$ by Lemma 3.3. Hence, the OM-value decreases at most by $O(\ln^3 n)$.

4.10.1 Event (i)

W.h.p., as noted in the preparatory reasoning, for all t the number of visited distorted points during $[t, t_s(t)]$ is $O(\ln^2(n))$. For large enough n, the number of generations to leave a distorted point is stochastically dominated by a $\mathrm{Geo}(q)$ random variable with $q = \eta^{-\lambda}$, since q is a lower bound on the probability that there is no clone of the parent in one generation by Lemma 3.2. Similar to (34), w.h.p. none of the first n^2 times that we visit a distorted point, the algorithm stays longer in the distorted point than $O(\ln(n)/q)$ generations. Since w.h.p., for all t the total number of visited distorted points during $[t, t_s(t)]$ is $O(\ln^2(n))$, event (i) holds w.h.p.

4.10.2 Event (ii)

W.h.p., the total negative progress from clean parents is at most $O(\ln^2 n)$ by the preparatory reasoning, and the total negative progress from distorted parents is $O(\ln^3 n)$ by Lemma 3.3. Since the number of OM-improving steps in each interval $[t, t_s(t)]$ is $\Omega(\ln^4 n)$ w.h.p, it follows that there exists $\varepsilon > 0$ such that w.h.p.

(iia) $\forall t \leq n^2$, the progress in time $[t, t_s(t)]$ is at least $2\varepsilon \ln^4 n$.

Let $t' = t_s(t)$ for some t such that $y^{(t)}$ is distorted. W.h.p., by the preparatory reasoning, during the interval $[t', t_s(t')]$ the total number of backward jumps is bounded by $O(\ln^2 n)$, and each of them has size at most $O(\ln n)$. Thus, w.h.p. all parents x' in the interval $[t', t_s(t')]$ satisy $\mathrm{OM}(x') \geq \mathrm{OM}(x^{(\lceil t'/\lambda \rceil - 1)}) - O(\ln^3 n)$, and by Lemma 3.3 the distance from any parent to its offspring is at most $O(\ln n)$. Hence, all offspring $y^{(r)}$ with $r \in [t', t_s(t')]$ satisfy for any constant $\varepsilon > 0$ and for sufficiently large n,

$$\begin{aligned}
\mathrm{OM}(y^{(r)}) &\geq \mathrm{OM}(x^{(\lceil t'/\lambda \rceil - 1)}) - O(\ln^3 n) \\
&\geq \mathrm{OM}(x^{(\lceil t'/\lambda \rceil - 1)}) - \varepsilon \ln^4 n \\
&\overset{\text{(ii}_\text{a}\text{)}}{\geq} \mathrm{OM}(x^{(\lceil t/\lambda \rceil - 1)}) + \varepsilon \ln^4 n.
\end{aligned}$$

Again by Lemma 3.3 it follows that $\mathrm{OM}(y^{(t)}) > \mathrm{OM}(x^{(\lceil t/\lambda \rceil - 1)}) - O(\ln n)$, so $\mathrm{OM}(y^{(r)}) > \mathrm{OM}(y^{(t)})$ for all t and all $r \in [t_s(t), t_s(t_s(t))]$. Since this holds w.h.p. for all t simultaneously, combining this inequality for all t yields that event (ii) holds w.h.p.

4.10.3 Event (iii)

By Lemma 3.3, no offspring flips more than $c \ln n$ bits w.h.p.

4.10.4 Event (iv)

The first time that each sampled point is sampled, it is distorted with probability p, independently of the rest. Since a clean point remains clean and $T^{\text{comma,dy}} = O(n \ln n)$

w.h.p. by Lemma 4.3, the total number of distorted points is w.h.p. stochastically dominated by a binomial random variable $\mathrm{Bin}(\hat{C}n\ln n, p)$ for some $\hat{C} > 0$. The first statement of the lemma follows by Chernoff's bound, see Theorem 2.4, and the reasoning below the definition of the three events.

The second part follows by reasoning analogously to the reasoning in the proof that event (ii) holds w.h.p.

5 Traps Slow Down The Plus Strategy

We now prove the lower bounds on T^{plus} in Theorem 1.1. For this direction, we still have dependency with the history when we sample an offspring, but the dependency is in a direction which helps us. While it may happen that we sample a search point that we have already examined before, this can never give a strict fitness improvement because the $(1+\lambda)$ EAis elitist. As long as the $(1+\lambda)$ EAdoes not leave the distorted set, this allows us to dominate the progress on DISOMby the progress of an algorithm on ONEMAXwhich automatically rejects every offspring with probability $1-p$ before evaluating its fitness.

Proof of Theorem 1.1, lower bound on T^{plus} Let us consider distances from $\vec{1}$ in the two intervals $I := (2k^*, 2k^*n^{2\delta}]$ and $I' := [k^*n^{\delta}, k^*n^{2\delta}] \subset I$ for some sufficiently small $\delta > 0$ so that $k^*n^{2\delta}\lambda = o(n)$ (which exists by Assumption 1.4). Recall that by our assumption that $d \le k^*$, all points in I have fitness less than $n - 2k^* + d \le n - k^*$, so it is necessary to traverse I. Let $\mathcal{D}$ denote the set of distorted points. We will show that w.h.p.

(i) we visit a distorted point with ZM-value in I',
(ii) afterwards the algorithm does not leave $\mathcal{D}$ for time $n\ln(n)/p$, and
(iii) it takes $\Omega(n\ln(n)/p)$ generations to reach fitness at least $n-k^*$ within $\mathcal{D}$ when starting in I'.

The three items imply the lower bound (2). We prove the items one by one, starting with (i). Observe that w.h.p. the algorithm does not jump over the interval I', since in the first $O(n\ln(n)/p) = o(n^2\ln^2 n)$ number of generations, each offspring is within Hamming distance $O(\ln n)$ from its parent by Lemma 3.3, and $|I'| = \Omega(k^*n^{2\delta}) = \omega(\ln n)$. Assume that the algorithm enters I' at a clean point $x^{(0)}$, since otherwise there is nothing to show. Note that the distance from $x^{(0)}$ to the all-ones bit string is at least $k^*n^{2\delta} - O(\ln(n)) = \Omega(k^*n^{2\delta})$ w.h.p. (also by Lemma 3.3). As long as it does not move to a distorted point, it mimics perfectly the behaviour on ONEMAX. In particular, by Theorem 3.6(1) with $a = \Omega(k^*n^{2\delta})$ and $b = k^*n^{\delta}$, w.h.p. the $(1+\lambda)$ EAneeds to produce $\Omega(n\ln\frac{a}{b}) = \Omega(n\ln(n^{\delta})) = \Omega(n\ln n)$ offspring to cross the interval I' on ONEMAX. The offspring produced in this time do not have to be all different from each other. Let us investigate such a run on ONEMAXfurther, starting in $x^{(0)}$. Since each mutation has probability $\Omega(1)$ to produce a (Hamming) neighbour of the parent, w.h.p. $\Omega(n\ln n)$ offspring are neighbours of their respective parents. We denote the set of these offspring by S.

We will argue next that this set S contains $\Omega(n \ln n)$ *different* individuals. For brevity, we say that a search point is in I' if its ZM-value is in I'. For any parent x in I', the probability that an offspring of x is strictly fitter than x is at least $1/(ek^*n^\delta)$, since this is a lower bound for the probability of flipping a zero-bit and no other bit. Hence, for any sequence of $k^*n^{2\delta}$ mutations of parents in I', where the same parent may occur multiple times, the probability that none of them is a fitness improvement is at most

$$\mathbb{P}(\text{no improvement among} k^*n^{2\delta} \text{offspring}) \leq \left(1 - \frac{1}{ek^*n^\delta}\right)^{k^*n^{2\delta}} \leq e^{-n^\delta/e}.$$

This is stretched exponentially small, so by a union bound, w.h.p. this does not occur in $O(n \ln(n)/p)$ generations, i.e., whenever the algorithm produces $k^*n^{2\delta}$ offspring, it increases the fitness at least once. This means that on any fitness level $i \in I'$, the algorithm produces at most $k^*n^{2\delta} + \lambda$ offspring from parents with ZM -value i, where the $+\lambda$ takes into account that after finding an improvement the algorithm still evaluates the rest of this generation. Now consider the total number of offspring on fitness level i that are produced. Since w.h.p. every offspring has Hamming distance $O(\ln n)$ from its parent, such offspring can only be produced from fitness levels $i \pm O(\ln n)$, and on each level at most $k^*n^{2\delta} + \lambda$ offspring are produced. Hence, using $\lambda = \Theta(\ln n)$ by Assumption 1.4, w.h.p. S contains at most $O(\ln n \cdot (k^*n^{2\delta} + \lambda)) = O(k^*n^{2\delta}\lambda) = o(n)$ search points per fitness level in I'.

Assuming that this event holds, we reconsider the run of the algorithm. Whenever the algorithm is in a parent x, it has visited $o(n)$ Hamming neighbours of x up to this point. Therefore, each offspring has probability $\Omega(1)$ to be a neighbour not visited before. By the Chernoff bound, w.h.p. this process produces $\Omega(n \ln n)$ *different* search points in distance one from their parent while producing $\Omega(n \ln n)$ offspring. So w.h.p. $|S| = \Omega(n \ln n)$.

Finally, due to the length of the interval, at most $O(k^*n^{2\delta})$ generations improve the fitness, so the number of offspring in those generations is at most $O(\lambda k^*n^{2\delta}) = o(n \ln n)$. We remove those offspring from S, yielding a set S' of size $\Omega(n \ln n)$. All the $\Omega(n \ln n)$ individuals in S' are clean with probability $(1-p)^{\Omega(n \ln n)} = o(1)$ as $p = \omega(1/(n \ln n))$ by Assumption 1.4. Hence, w.h.p. at least one point in S' is distorted.

Recall that the runs on DISOMand OMare identical up to the point when the first distorted search point is accepted. So unless the $(1+\lambda)$ EAaccepts some distorted point, it queries the same points in S'. However, let $y \in S'$ be distorted with parent x which is within Hamming distance 1 from y. The fitness of y is $\text{DISOM}(y) = \text{OM}(y) + d \geq \text{OM}(x)$. The probability that the same generation contains a fitter non-distorted offspring not in S is at most $O(\lambda k^*n^{2\delta-1}) = o(1)$, as it has to flip at least one of its $O(k^*n^{2\delta})$ zero bits. So, no other search point in the ONEMAXrun in that generation has a higher fitness than x w.h.p., the offspring y will be accepted (or another distorted offspring in the same generation). This proves (i).

For (ii), assume that the algorithm is in a distorted point $x \in I$, and consider a clean offspring y of x. The algorithm can only prefer y over

x if $\mathrm{OM}(y) = f(y) \geq f(x) = \mathrm{OM}(x) + d$. Hence, in order to accept a clean offspring, we need to decrease the Hamming distance to the origin by at least d. By [38, Lemma 1.10.37], the probability that an offspring decreases its Hamming distance by at least d from its parent with at most $k^* n^{2\delta}$ zeros is at most $O\left(\left(k^* n^{2\delta}/n\right)^d\right)$. By the union bound, the probability that this happens in time $n\ln(n)/p$ is $O\left(\frac{n\ln(n)}{p} \cdot n^{2\delta d}\left(\frac{k^*}{n}\right)^d\right)$. This expression is decreasing in d, and hence it is at most $O\left(\frac{n\ln(n)}{p} \cdot n^{2\delta d_{\min}}\left(\frac{k^*}{n}\right)^{d_{\min}}\right)$, where $d_{\min} := (1+\varepsilon)\ln(n/p)/\ln(n/k^*)$ is the (not necessarily integral) lower bound on d in Assumption 1.4. Plugging in, the second factor simplifies to

$$(k^*/n)^{d_{\min}} = e^{-d_{\min}\ln(n/k^*)} = e^{-(1+\varepsilon)\ln(n/p)} = (p/n)^{1+\varepsilon}. \tag{35}$$

Therefore, the probability that at least one of $n\ln(n)/p$ offspring decreases the Hamming distance by at least d is at most

$$O\left(\frac{n\ln(n)}{p} \cdot n^{2\delta d_{\min}}\left(\frac{p}{n}\right)^{1+\varepsilon}\right) = O\left(n^{2\delta d_{\min}}\left(\frac{p}{n}\right)^{\varepsilon}\ln n\right) = o(1)$$

for $\delta = \delta(\varepsilon) > 0$ sufficiently small. In other words, w.h.p. no clean offspring is accepted for time $n\ln(n)/p$. If $d = \omega(1)$, we can bound d in the above from below by a large constant, and upper bounds on the probability of the bad event remains valid. So the conclusion also holds without the assumption that d is constant.

It remains to show (iii). Consider a modified $(1+\lambda)$ EA, which starts in a distorted point in I', but which automatically discards all clean points, regardless of their fitness. Then, to make an improving step within $\mathcal{D}$, the algorithm needs to query a search point that *(a)* improves the Hamming distance from $\vec{1}$ and *(b)* that is distorted. We will show that we can couple the performance with a run on OneMaxin which each offspring is discarded with probability at least $1-p$ before considering it for selection. Let us call this a $(1-p)$-rejection run. Indeed, assume the modified $(1+\lambda)$ EAruns on disOMand its current search point is $x \in I$. Assume further that it samples an offspring y which is by an additive term $r > 0$ closer to the optimum. There are two cases. Either y has not been sampled before. In this case, y is clean with probability $1-p$, and thus rejected with this probability. Or y has been sampled before. In this case, y was clean (otherwise we would have moved there earlier), and thus it is rejected with probability 1. Therefore, for any $r > 0$, the probability of moving r closer to the optimum within $\mathcal{D}$ is at most the probability of moving r closer to the optimum in a $(1-p)$-rejection run on OneMax. Since even the $(1+1)$EAon OneMaxtakes time $\Omega(n\ln(n))$ to cross I from any point in I' by Theorem 3.6(1), a $(1-p)$-rejection run of the $(1+1)$EAon OneMaxtakes time $\Omega(n\ln(n)/p)$. (We need to wait expected time $1/p$ before considering an offspring.) By Theorem 3.4, the same is true for the $(1+\lambda)$ EA. This proves (iii) and concludes the proof of the lower bound on T^{plus} in Theorem 1.1.

6 Plus Strategy Still Escapes

The main goal of this section is to prove Theorem 1.3, which gives an upper bound on the runtime of $(1+\lambda)$ EAon DISOMin terms of an upper bound on the runtime of $(1,\lambda)$ EAon DISOM. We remark that the results obtained here immediately imply the upper bound on T^{plus} in Theorem 1.1. We split the proof of Theorem 1.3 into two cases. We start with a rather simple lemma for the case which covers the first two settings in Theorem 1.3.

Lemma 6.1 *Consider the setting of Theorem 1.3. There exists a constant $C' > 0$ such that w.h.p.*

$$T^{\text{plus}} \leq \begin{cases} T, & if p \cdot T = o(1), or \lambda \geq C' \ln(n), \\ O\big(n \cdot (\lambda + \ln(\frac{n}{k^*}))\big), & if p \cdot n \cdot \big(\lambda + \ln(\frac{n}{k^*})\big) = o(1). \end{cases}$$

Proof To prove the upper bound on the first line, we distinguish two caes. For the first case with $pT = o(1)$, consider a run of the $(1+\lambda)$ EAon OMfor time T. Then w.h.p. fitness target $n - k^*$ is reached before T by Theorem 3.4(a). We can couple this run with a run on DISOMuntil the first time that a distorted search point is queried. By a union bound, the probability that this happens before generation T is $p \cdot T = o(1)$. Hence, with high probability the two runs on OMand on DISOMare identical until time T, and w.h.p. fitness target k^* is reached before then. Hence, w.h.p. $T^{\text{plus}} \leq T$.

We turn to the second case $\lambda \geq C' \ln(n)$, which may overlap with the first case. Since we assume that $T \leq n^C$ in Theorem 1.3, there exists $C' > 0$ such that for $\lambda \geq C' \ln(n)$ in all first n^C/λ generations there is a clone of the parent in a run of the $(1,\lambda)$ EA w.h.p. Hence, in those cases the $(1,\lambda)$ EAmimics the $(1+\lambda)$ EAand the statement follows since the $(1,\lambda)$ EAfinds the target before time T w.h.p.

For the third case, we observe that a run of the $(1+\lambda)$ EAon OMfinds the target in time $O(n \cdot (\lambda + \ln(\frac{n}{k^*})))$ by Theorem 3.6(2), and the third case follows similar to the first case by coupling the runs of the algorithm with its run on ONEMAX and noting that w.h.p. none of the points sampled during this time is distorted.****

The next lemma is the most complicated step. We will show that the local optima in DISOMcan increase the runtime of the $(1,\lambda)$ EAat most by a factor of $O(1/p)$. Note that this statement seems very intuitive: the algorithm can always make progress by staying within the distorted points. If the probability of making an improving step on ONEMAXis p_{imp}, then the probability of making an improving step on DISOMshould be $p_{\text{imp}} \cdot p$ since we need to find an OM-improving step, and the offspring needs to be distorted.

However, this intuition can be misleading. The problem is that the distortions are fixed, and that we do not get fresh randomness each time. Assume we are at distance k from the optimum, and let us focus on single-bit flips for illustration. Since k of the n neighbours are improving, each single-bit flip has a chance of k/n to be improving, so on ONEMAXwe need to wait for n/k single-bit flips in expectation.

For DISOM, if we are in a distorted point x at distance k from the optimum, then the probability that a single-bit flip is distorted *and* OM-improving (i.e., closer to $\vec{1}$ than the parent) is naively pk/n, so it is tempting to assume that one simply needs to wait for $n/(pk)$ generations in expectation. However, this is not true: it could happen that all k OM-improving neighbours of x are clean. In fact, this is a *likely* scenario since the expected number of OM-improving distorted neighbours is pk, which may be of order $o(1)$.[5] In this case, we can never escape from x by a single-bit flip. (In other words: DISOMhas a local optimum, which was the main goal of the introduction of DISOM.) So in this case, we need at least *two-bit flips* to escape.

This could potentially be very costly, but fortunately we can profit from two-bit flips to *the same* OM-*level*, i.e., OM-neutral mutations which lead to search points in the same Hamming distance from the optimum. Those two-bit flips are much cheaper than OM-*improving* two-bit flips and provide fresh randomness. Mind that this is a real and important issue, and the following lemma would be wrong if the $(1+\lambda)$ EAbroke ties in favour of the parent. This is also why we are uncertain whether Theorem 1.3 transfers similarly to other functions. In fact, we conjecture that it is false for other linear functions.

Lemma 6.2 *Consider the setting of Theorem 1.3, and assume additionally that* $p \cdot n \cdot (\lambda + \ln(n/k^*)) = \Omega(1)$, *and* $p \cdot T = \Omega(1)$ *and* $\lambda < C' \ln n$ *with* C' *from Lemma 6.1. Let* T^{DISOM} *be the fixed-target hitting time of the* $(1+\lambda)$ *EAon* DISOM*for target fitness* $n - k^*$. *Then w.h.p.,*

$$T^{\text{DISOM}} = O\big(\lambda n + \tfrac{n}{p} \ln(\tfrac{n}{k^*})\big).$$

Proof W.h.p. the initial point of the $(1+\lambda)$ EAhas distance at least $n/3$ from $\vec{1}$. Hence, by Theorem 3.6(1), we can assume without loss of generality that $T = \Omega(n \ln(n/k^*))$, so $p = \Omega(1/(n \ln n))$ as $pT = \Omega(1)$, where T is from the statement of Theorem 1.3 which serves as an arbitrary upper bound for the runtime of $(1,\lambda)$ EAon DISOMthat holds with high probability. □

We first make some observations that allow us to simplify the problem. On DISOM, every search point in Hamming distance at most k^* from $\vec{1}$ has fitness at least $n - k^*$. (The converse is not true in general.) Therefore, it suffices to bound the time until the $(1+\lambda)$ EAreaches Hamming distance k^* from $\vec{1}$, since this time is at least as large as T^{DISOM}.

Next, consider a run of the $(1+\lambda)$ EAon DISOM. Let $T_{\mathcal{C}}^{\text{DISOM}}$ and $T_{\mathcal{D}}^{\text{DISOM}}$ denote the number of function evaluations that are performed while the parent is in a clean and in a distorted state, respectively. We split

$$T^{\text{DISOM}} = T_{\mathcal{C}}^{\text{DISOM}} + T_{\mathcal{D}}^{\text{DISOM}}. \tag{36}$$

[5] Recall that p can be arbitrarily close to $1/(n \ln n)$. In fact, this is a particularly interesting case since it gives the largest factor between T^{comma} and T^{plus} in Theorem 1.1.

6.1 Evaluations from A Clean Parent

We will first show that $T_{\mathcal{C}}^{\text{DISOM}} = O(\lambda n + n \ln(n/k^*))$ w.h.p. To see this, we introduce some terminology. We call *level k* the set of all search points at distance k from $\vec{1}$. We denote the set of clean points on level k by $\mathcal{C}_k$, and the set of distorted points on level k by $\mathcal{D}_k$. We call an offspring OM-*improving* if the offspring has strictly smaller distance from $\vec{1}$ than the parent. For a parent in fitness level k, let $p_{\text{imp}} = p_{\text{imp}}(k)$ be the probability that a mutation creates an OM-improving offspring, and $p_{\text{imp},1} = p_{\text{imp},1}(k)$ be the probability that a mutation is a single-bit flip that creates an OM-improving offspring. Then both $p_{\text{imp}} = \Theta(k/n)$ and $p_{\text{imp},1} = \Theta(k/n)$.

Let T_k be the number of offspring that the $(1+\lambda)$ EAon DISOMgenerates with parents in $\mathcal{C}_k$. Note that the algorithm may leave and re-enter $\mathcal{C}_k$ if there are distorted points of the same fitness. But as soon as it generates an OM-improving offspring y from a parent $x \in \mathcal{C}_k$ on level k, y is strictly fitter than x (regardless of whether y is distorted or not) and the algorithm never returns to $\mathcal{C}_k$ afterwards. Hence, every offspring of a parent in $\mathcal{C}_k$ has probability at least p_{imp} to leave $\mathcal{C}_k$ for good. Note that the other offspring in the same generation are still generated, which adds at most $\lambda - 1$ additional offspring to T_k. Therefore, T_k is stochastically dominated by $T'_k + \lambda - 1$, where T'_k follows a geometric distribution $\text{Geo}(p_{\text{imp}}(k))$ with $p_{\text{imp}}(k) = \Theta(k/n)$. Summing over all $k \in [k^*, n]$, $T_{\mathcal{C}}^{\text{DISOM}}$ is dominated by $(\lambda-1)(n-k^*) + \sum_{k=k^*+1}^{n} T'_k \le \lambda n + \sum_{k=k^*+1}^{n} T'_k$ for independent geometric random variables T'_k. Since $k^* < n/6$ by assumption in Theorem 1.3, the sum has expectation $\mu = \Theta(\sum_{k=k^*+1}^{n} n/k) = \Theta(n \ln(n/k^*))$, where we used that the hidden constant is uniform over all k. Moreover, the probability that the sum exceeds its expectation by a constant factor is $e^{-\Omega(p_{\min}\mu)}$ by Theorem 2.5, where $p_{\min} = \min\{p_{\text{imp}}(k) \mid k \in [k^*, n]\} = \Theta(k^*/n)$ and thus $p_{\min}\mu = \Theta(k^* \ln(n/k^*))) = \omega(1)$. In either case, w.h.p. $T_{\mathcal{C}}^{\text{DISOM}} = O(\lambda n + n \ln(n/k^*))$.

6.1.1 Evaluations from a Distorted Parent

Hence, it remains to bound $T_{\mathcal{D}}^{\text{DISOM}}$. Note that, if d is an integer, the algorithm might leave and return to $\mathcal{D}_k$ by visiting clean points of the same fitness (at level $k+d$). We will ignore this complication for now, and only return to it in the very end of this proof.

We partition $\mathcal{D}_k$ into two sets of *good* and *bad* points. The set $\mathcal{D}_k^{\text{good}}$ contains all $x \in \mathcal{D}_k$ for which the algorithm has so far queried at most $k/2$ of the k OM-improving neighbours of x, and we define $\mathcal{D}_k^{\text{bad}} := \mathcal{D}_k \setminus \mathcal{D}_k^{\text{good}}$.

Assume that the algorithm is in a search point $x \in \mathcal{D}_k^{\text{bad}}$. We define for $i \ge 2$

$$N^{(1)}(x) := \{x' \in \mathcal{D}_k \mid H(x', x) = 2, H(x', \vec{1}) = H(x, \vec{1})\}, \tag{37}$$

$$N_x(x') := \{x'' \in N^{(1)}(x') \cap \mathcal{D}_k : H(x, x'') = H(x, x') + 2\}. \tag{38}$$

$$N^{(i)}(x) := \bigcup_{x' \in N^{(i-1)}(x)} N_x(x') \qquad \text{for all} i \geq 2. \tag{39}$$

As we will show, the sequence of $N^{(i)}(x)$ constitutes a network on which the algorithm may move relatively quickly. Recall the parameter ε from the condition $k^* \geq n^\varepsilon$ in Theorem 1.3. In the following, we will show for the constant $i_0 := \max\{6, \lceil 1 + 1/\varepsilon \rceil\}$ that w.h.p.

(i) for each level k, it does not happen that the algorithm is in $\mathcal{D}_k$ after having queried more than $2\ln(n)/p$ different search points from level $k-1$; and
(ii) for all parents x in the first n^3 steps of the algorithm, all $x' \in N^{(i)}(x)$ for any $i \leq i_0$, and all $k^* \geq n^\varepsilon$, $N_x(x')$ have size $\Theta(pkn)$; moreover $|N^{(i)}(x)| = \Theta((pkn)^i)$ for all $i \leq i_0$; and
(iii) for all parents x, at least half of the search points in $N^{(i_0)}(x)$ are good.

We bound the probabilities of the three events separately, and afterwards show that on the intersection of these events the algorithm leaves $\mathcal{D}_k$ sufficiently fast.

6.1.2 Event (i)

Every offspring from level $k-1$ has probability p to be distorted, in which case the algorithm will never return to level k. If the algorithm queries $2\ln(n)/p$ different offspring from level $k-1$, the probability that none of them is distorted is $(1-p)^{2\ln(n)/p} \leq e^{-2\ln n} = n^{-2}$. By a union bound over the at most n levels, the probability that this happens for any level is $o(1)$. Thus, event (i) holds w.h.p. Note that event (i) implies that the algorithm queries at most $2\ln(n)/p$ OM-improving offspring with parents from level k before leaving $\mathcal{D}_k$.

6.1.3 Event (ii)

We first make an observation for a distorted vertex x' at level k, for which at most one of the points at the same level at Hamming-distance 2 from x' has been visited before. Then the expected number of distorted points at distance 2 from x' stochastically dominates a binomially distributed random variable with parameters $(k-1)(n-k-1)$ and p, and is stochastically dominated by 1 plus a binomial random variable with the same parameters. As noted at the beginning of the proof, we assume w.l.o.g. that $p = \Omega(1/(n\ln n))$, so $pkn \geq \Omega(k/\ln n) \geq \Omega(n^\varepsilon/\ln n) = \Omega(n^{\varepsilon/2})$. By Chernoff's bound, see Theorem 2.4, for any given point x',

$$\begin{aligned}\mathbb{P}\big((n-k-1)(k-1)/2 \leq N_x(x') \leq 1 + 2(n-k-1)(k-1)\big) &\geq 1 - \exp(-\Theta(nkp)\big) \\ &\geq 1 - \exp(-\Omega(n^{\varepsilon/2})).\end{aligned}$$

We next argue that we can take a union bound over all $x' \in N^{(i_0)}(x)$ and all parents x. Indeed, the first time that a vertex x' is within Hamming distance $2i_0$ of a sampled

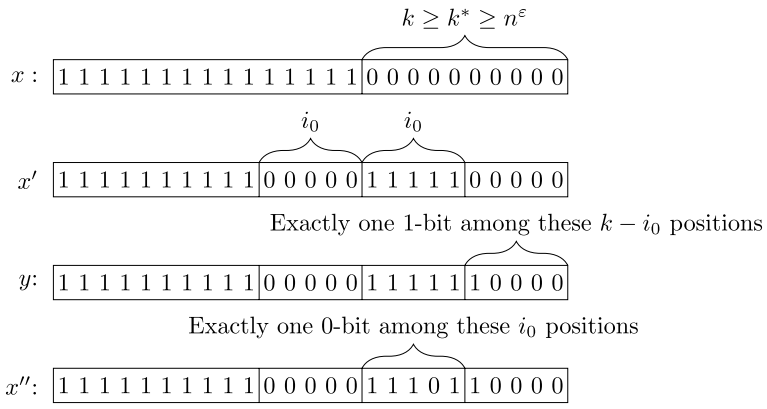

Fig. 1 Illustration of the counting argument in (iii). Consider a bad point $x' \in N^{(i_0)}(x)$, a OM-improving neighbour y of x' such that $H(x, y) = H(x, x') + 1$ and another (different from x') bad point $x'' \in N^{(i_0)}(x)$ that also has y as Hamming-improving neighbour. Then y must be obtained from x' by flipping one of $k - i_0$ positions to from zero to one, and x'' is obtained from y by flipping one of i_0 one-bits to zero. Hence, y counts as neighbour towards at most $i_0 + 1$ bad points

offspring x, we can apply this bound, because at this time the algorithm has visited at most one point in Hamming distance 2 from x' at level k, namely possibly the offspring x. (There is exactly one such point if the algorithm decides to visit x and if $H(x, x') = 2$, otherwise there are no such points). During the first n^3 queries, there are most $n^3 \sum_{i \leq i_0} (kn)^i \leq i_0 n^{3+2i_0} = o(\exp(n^{\varepsilon/2}))$ points within Hamming distance i_0 of a sampled offspring, allowing to take a union bound over all these points. Therefore, w.h.p. for all parents in the first n^3 steps, all $x' \in N^{(i)}(x)$ for any $i \leq i_0$, each $N_x(x')$ has size $\Theta(pkn)$. Each time we increment i, each point $x' \in N^{(i)}(x)$ can add up to $O(pkn)$ points from its $N_x(x')$. Hence, $N^{(i+1)}(x)$ is by at most $O(pkn)$ larger than $N^{(i)}(x)$. By induction over i, this implies that $|N^{i_0}(x)| = O((pkn)^{i_0})$ with high probability for all parents x, since i_0 is constant. For a lower bound, we use the same idea, but we must take into account that we double-count some search points, since the same search point may be contained in $N_x(x')$ for several x'. We now quantify this overlap. If $x'' \in N^{(i)}(x)$, then there are at most i^2 points x' for which $x'' \in N_x(x')$ (compared to x'', each such x' coincides with x in one of i flipped 1-bits in x'', and in one of i flipped 0-bits). Thus, x'' is counted at most i^2 times, or in formula

$$|N^{(i_0)}(x)| \geq \frac{1}{i_0^2}\Theta(pkn)|N^{(i_0-1)}(x)| \geq \cdots \geq \prod_{i \in [i_0]} \frac{1}{i^2}\Theta(pkn) = \Theta((pkn)^{i_0}).$$

6.1.4 Event (iii)

To prove that (iii) holds with high probability, let us assume for the sake of contradiction that at least half of $N^{(i_0)}(x)$ is bad for some x. Then each of the bad points $x' \in N^{(i_0)}(x)$ has at least $k/2$ OM-improving neighbours that are already queried. Moreover, at least $k/2 - i_0$ of these neighbours y have distance $2i_0 + 1$ from x, see Fig. 1 for an illus-

tration. On the other hand, each such y (in distance $k-1$ from the optimum and in distance $2i_0+1$ from x) differs from x in exactly i_0+1 one-bits and i_0 zero-bits, and is therefore neighbour of at most i_0+1 points $x' \in N^{(i_0)}(x)$. Hence, the algorithm has queried at least $|N^{(i_0)}(x)|/2 \cdot (k/2-i_0)/(i_0+1) = \Omega(nk) = \Omega(n^{1+\varepsilon})$ different OM-improving neighbours from parents of level k. This is a contradiction, since the number of queried OM-improving neighbours is at most $2\ln(n)/p \le 2n\ln^3 n$ w.h.p. by event (i).

We will now bound the runtime of the algorithm conditional on the intersection of the three events (i-iii). Let $\mathcal{F}_t$ denote all information of the algorithm up to time $t \le n^3$, i.e., all sampled search points, their fitnesses and the selection (which is non-deterministic in case of ties), and assume that it satisfies the events (i-iii). Let M_k denote the number of moves the algorithm makes within $\mathcal{D}_k$ before leaving it, i.e., the number of times that $x^{(t)} \in \mathcal{D}_k$ and $x^{(t+1)} \neq x^{(t)}$. We will show that there exists a constant $\sigma > 0$ such that given $\mathcal{F}_t$ satisfying (i-iii)

$$M_k \preccurlyeq (i_0+1)\mathrm{Geo}(\sigma) \preccurlyeq (i_0+1) + \mathrm{Geo}\Big(\frac{\sigma}{i_0+1}\Big), \tag{40}$$

and after each jump to a vertex x in $\mathcal{D}_k$, the number of function evaluations before moving $J_k(x)$ (either within $\mathcal{D}_k$ or to a vertex outside $\mathcal{D}_k$) satisfies

$$J_k(x) \preccurlyeq \lambda + \mathrm{Geo}\Big(\sigma\frac{pk}{n}\Big), \tag{41}$$

regardless of x being good or bad. Since the sum of a geometric number of independent identically distributed geometric random variables is again geometrically distributed,[6], we may bound the number $T_k^{\mathcal{D}}$ of offspring generated from parents in $\mathcal{D}_k$ by

$$\begin{aligned} T_{k,\mathcal{D}} &\preccurlyeq \sum_{i=1}^{M_k} \Big(\lambda + \mathrm{Geo}_i\Big(\sigma\frac{pk}{n}\Big)\Big) \\ &\preccurlyeq (i_0+1)\lambda + \sum_{i=1}^{i_0+1} \mathrm{Geo}_i\Big(\sigma\frac{pk}{n}\Big) + \lambda\mathrm{Geo}\Big(\frac{\sigma}{i_0+1}\Big) + \mathrm{Geo}\Big(\frac{\sigma^2 pk}{(i_0+1)n}\Big), \end{aligned} \tag{42}$$

where all geometric random variables within the sums are independent. We remark that the second, third, and fourth term on the second line are not independent; this is not important for the analysis below that uses a union bound.

[6] This statement can be derived either via probability generating functions or from the theory of infinitely divisible distributions [55, Chapter XVII] but can also be seen elementarily: consider a pair of coins with probabilities p and q, and count the number X of rounds until both simultaneously come up heads. Let $X_1, X_2, \ldots$ be the number of rounds until the first coin comes up heads. Then X is distributed as $X_1 + X_2 + \ldots + X_k$, where $k \sim \mathrm{Geo}(q)$ and each X_i is distributed as $\mathrm{Geo}(p)$, but X is also distributed as $\mathrm{Geo}(pq)$. Hence, both distributions must be identical.

Recall that we assumed $d \notin \mathbb{N}$. Hence, each level $\mathcal{D}_k$ is entered at most once. Therefore, it follows that (using the convention $0 \cdot \infty = 0$)

$$T_{\mathcal{D}}^{\text{DISOM}} \preccurlyeq T_{\mathcal{D}}^{\text{DISOM}} \mathbb{1}\{T_{\mathcal{D}}^{\text{DISOM}} \leq n^3\} + \infty \cdot \mathbb{1}\{T_{\mathcal{D}}^{\text{DISOM}} > n^3\}, \tag{43}$$

where $T_{\mathcal{D}}^{\text{DISOM}} = \sum_{k=k^*}^{n} T_{k,\mathcal{D}}$ can be bounded by (42).

To prove (40), we will show that (a) every move from a good parent leaves $\mathcal{D}_k$ with constant probability; (b) starting from a parent in $\mathcal{D}_k$ (which may be either good or bad) the algorithm lands after i_0 moves either at a point $x' \in \mathcal{D}_k$ which is good with at least constant probability, or it leaves $\mathcal{D}_k$. Given (a) and (b), the factor $\text{Geo}(\rho)$ in the first bound in (40) represents the number of iterations of multiples of $i_0 + 1$ moves until $\mathcal{D}_k$ is left (after each iteration $\mathcal{D}_k$ is left with constant probability). So (a) and (b) will imply $M_k \preccurlyeq (i_0 + 1)\text{Geo}(\sigma)$. The second bound in (40) is an elementary fact about the geometric distribution which can be seen as follows. Consider a process where a biased coin with probability $\sigma/(i_0 + 1)$ of heads is flipped until the first appearance of heads, and then is continued to be flipped until the number X of flips reaches the next multiple of $i_0 + 1$. Then X is dominated by $\text{Geo}(\sigma/(i_0 + 1)) + (i_0 + 1)$. On the other hand, if we split the sequence of coin flips into batches of size $(i_0 + 1)$ then the probability of heads in each batch is at most σ by a union bound. Hence, the number of batches until the first appearance of heads dominates $\text{Geo}(\sigma)$, and thus X dominates $(i_0 + 1)\text{Geo}(\sigma)$. This proves the second bound in (40). It remains to show (a) and (b).

(a) Moving from a good parent. Consider an offspring from a parent in $\mathcal{D}_k^{\text{good}}$. With probability $\Theta(p_{\text{imp}})$ the offspring is OM-improving, with probability $\Theta(1)$ it hasn't been visited before (by definition of a good point, half of the OM-improving points have not been visited, and each offspring at Hamming distance one is equally likely), and with probability p it is distorted. Thus, the probability that this offspring makes the algorithm leave $\mathcal{D}_k$ is $\Omega(p_{\text{imp}}p) = \Omega(pk/n)$. To show that the algorithm leaves $\mathcal{D}_k$ with constant probability, we have to argue that the probability that an offspring leads to a move within $\mathcal{D}_k$ is $O(pk/n)$.

In order to create a search point in Hamming distance $2i$ on the same fitness level, it is necessary to flip exactly i zero-bits. There are $\binom{k}{i} \leq k^i$ possibilities for the zero-bits, and the probability of flipping them is n^{-i}. Hence, the probability to create such an offspring is at most $(k/n)^i$, and by summing over all $i \geq 1$ this probability is $O(k/n)$ because the sum is dominated by the term for $i = 1$. The algorithm only moves to such an offspring if it is in $\mathcal{D}_k$, which happens with probability p. Hence, the probability that an offspring leads to a move within $\mathcal{D}_k$ is also $O(pk/n)$, proving (a).

(b) Moving to a good parent.

Let us fix some $x' \in N^{(i_0)}(x)$, and let $x = x^{(0)}, x^{(1)}, \ldots, x^{(i_0)} = x'$ be a chain of search points such that $x^{(i)} \in N_x(x^{(i-1)})$. Let $\mathcal{E}$ be the event that $(x^{(i)})_{i=1}^{i_0}$ are the next i_0 search points that the algorithm visits, where we do not count idle generations in which the algorithm stays in the same search point. We claim that

$$\mathbb{P}(\mathcal{E} \mid \text{ not leaving } \mathcal{D}_k \text{ for } i_0 \text{ moves, event (i-iii) }) = \Omega(1/|N^{(i_0)}(x)|).$$

First note that the probability that the algorithm moves within $\mathcal{D}_k \subseteq \mathcal{D}$ in one generation is $O(\lambda p p_{\text{imp}})$, since for moving it is necessary to flip at least one zero-bit (which has probability $\Theta(p_{\text{imp}})$), and the offspring needs to be distorted (probability $O(p)$). Finally, the factor λ comes from a union bound over the λ offspring per generation. On the other hand, the probability of moving from $x^{(i)}$ to $x^{(i+1)}$ is $\Theta(\lambda/n^2) = \Theta(\lambda p_{\text{imp}}/(kn)) = \Theta(\lambda p p_{\text{imp}}/(pkn))$. Hence, the conditional probability of moving to $x^{(i)}$, on moving at all, is $\Omega(1/(pkn))$. Iterating this over the i_0 steps (where i_0 is a constant), we obtain

$$\begin{aligned}&\mathbb{P}(\mathcal{E} \mid \text{ not leaving } \mathcal{D}_k \text{ for } i_0 \text{ moves, event (i-iii) }) \\ &\quad = \prod_{i=0}^{i_0-1} \Omega(1/(pkn)) = \Omega((pkn)^{-i_0}) \overset{\text{event (iii)}}{=} \Omega(1/|N^{(i_0)}(x)|).\end{aligned}$$

Summing over paths to the at least $|N^{(i_0)}|/2$ many points in $N^{(i_0)}$ that are good at any time of the algorithm, see event (ii), it follows that with constant probability the algorithm is either in a good parent after i_0 moves, or has left $\mathcal{D}_k$. By the reasoning below (43), this proves the stochastic domination (40) for δ sufficiently small, as by (a) the algorithm leaves $\mathcal{D}_k$ in the next move with constant probability.

To prove (41), we bound the time for the next move from above by the number of function evaluations until a point within $\mathcal{D}_k$ is sampled (the additive λ corrects for the evaluations of the remaining offspring in the same generation in which we move). Each sampled offspring is distorted with probability $\Omega(p)$, and is within $\mathcal{D}_k$ with probability $\Omega(k/n)$. This proves (41). Thus also (42) follows.

From (42) and $T_{\mathcal{D}}^{\text{DISOM}} = \sum_{k=k^*}^{n} T_{k,\mathcal{D}}$ it follows that

$$\begin{aligned}T_{\mathcal{D}}^{\text{DISOM}} &\preccurlyeq \sum_{k=k^*}^{n} \left((i_0+1)\lambda + \sum_{i=1}^{i_0+1} \text{Geo}_i\Big(\delta\frac{pk}{n}\Big) + \lambda \text{Geo}\Big(\frac{\delta}{i_0+1}\Big) + \text{Geo}\Big(\frac{\delta^2 pk}{(i_0+1)n}\Big)\right) \\ &\le O(\lambda n) + \sum_{i=1}^{i_0+1} \sum_{k=k^*}^{n} \text{Geo}_i\Big(\frac{\delta pk}{n}\Big) + \lambda \sum_{k=k^*}^{n} \text{Geo}\Big(\frac{\delta}{i_0+1}\Big) + \sum_{k=k^*}^{n} \text{Geo}\Big(\frac{\delta^2 pk}{(i_0+1)n}\Big),\end{aligned}$$

where the geometric random variables *within* each sum are independent. The expectation of each sum is $O(\lambda n + \frac{n}{p}\ln(n/k^*))$, with minimal parameter $p_{\min} = \Omega(pk^*/n)$. Concentration for each of the sums follows from Theorem 2.5, and by a union bound the lemma follows if $d \notin \mathbb{N}$.

Finally, for $d \in \mathbb{N}$ it can happen that the algorithm leaves and re-enters $\mathcal{D}_k$ via $\mathcal{C}_{k-d}$. However, any offspring of a parent in $\mathcal{C}_{k-d}$ has a chance of $\Omega(\frac{k-d}{n}) = \Omega(\frac{k}{n})$ to strictly improve fitness, where we used $k \ge d + k^* \ge 2d$. This is larger than the lower bound of $\Omega(kp/n)$ for finding an improving offspring that we used above for distorted points. Moreover, it might happen that a trajectory to a good distorted point is interrupted because the algorithm jumps to a clean point. But since reaching a clean point is *better* than reaching a good distorted point (has a higher chance of $\Omega(k/n)$

for finding a strictly fitter offspring than our estimated $\Omega(kp/n)$ for good distorted points), this can only accelerate the algorithm. Hence, the same argument remains valid if the algorithm visits points in $\mathcal{C}_{k-d}$.

Now we have all ingredients to prove Theorem 1.3.

Proof of Theorem 1.3 Since we switch between the fitness functions $f \in \{\text{OneMax}, \text{disOM}\}$ and several target fitnesses k^*, we include the indices f and k^* in the notation.

The first two cases of the bound are implied by Lemma 6.1.

A necessary condition for reaching fitness level $n - k^*$ is to reach Hamming distance $k^* + d$ from $\vec{1}$. By Theorem 3.5 the $(1, \lambda)$ EAon disOMis at most as fast as the $(1 + \lambda)$ EAon OneMax, so we also have $T^{\text{OM,plus}}_{k^*+d} \leq T$ w.h.p. By Theorem 3.6 we have $T^{\text{OM,plus}}_{k^*} = \Theta(n \log(n/k^*)) = \Theta(n \log(n/(k^* + d))) = \Theta(T^{\text{OM,plus}}_{k^*+d})$ w.h.p., where we use $\lambda = O(\ln(n/k^*))$ and where the middle step follows because $k^* \leq k^* + d \leq 2k^* \leq n/3$. Hence, $T = \Omega(n\lambda + \frac{n}{p} \ln(n/k^*))$ and thus $T^{\text{plus}} = O(T/p)$ by Lemma 6.2.

7 Combining All Results

We verify that the previous sections combined prove Theorem 1.1.

Proof of Theorem 1.1 The upper bound on T^{comma} is proven in Sect. 4. We verify now the lower bound on T^{comma}. Observe that T^{comma} stochastically dominates the time T' until the $(1, \lambda)$ EAfinds an offspring y with $\text{ZM}(y) \geq k^* + d$. By Theorem 3.6(1) it follows that $T' = \Omega(n \ln n)$ (setting $\mathcal{A} = (1, \lambda)$ EA, $a = \Theta(n)$ and $b = k^* + d$).

We turn to T^{plus}. The upper bound follows immediately from the upper bound on T^{comma}, since by Theorem 1.3 (which holds for a wider range of parameters than assumed in Assumption 1.4 in Theorem 1.1) the runtime of the $(1 + \lambda)$ EAon disOMis at most a factor $O(1/p)$ slower than on OneMax when $\lambda = \Theta(\ln n)$ and $k^* = n^{1-\Omega(1)}$. The lower bound on T^{plus} is given in Sect. 5.

Lastly, we give the proof of Proposition 1.2.

Proof of Proposition 1.2 For the w.h.p. statement in Proposition 1.2, we just observe that for OneMax, w.h.p. both algorithms find the optimum with $O(n \ln n)$ fitness evaluations [7]. Since p is so tiny, w.h.p. none of the $O(n \ln n)$ visited search points is distorted, and thus w.h.p. the runtime on OneMaxand on disOMis the same.

Let us now consider $\mathbb{E}[T^{\text{plus}}]$. With probability $p = 2^{-n}$, the all-zero string $\vec{0}$ is distorted. With probability $(1 - p)^{2^n - 1} \approx 1/e$, the other $2^n - 1$ search points are not distorted. So with probability $\Theta(p)$, we have $\vec{0}$ as the unique distorted search point. If this happens, then the $(1 + \lambda)$ EAstarts in $\vec{0}$ with probability $1/2^n$. If this happens, $\vec{0}$ has a fitness of $n - 0.5$ and the $(1 + \lambda)$ EAcan only escape by sampling the global

optimum. To do that, the algorithm needs to flip all n bits at the same time. The probability to do this is n^{-n}, so the algorithm needs expected time n^n to escape. In total, this scenario contributes $\Theta(p \cdot 2^{-n} \cdot n^n) = n^{\Omega(n)}$ to $\mathbb{E}[T^{\text{plus}}]$. This proves the lower bound on $\mathbb{E}[T^{\text{plus}}]$.

For the $(1, \lambda)$ EA, first note that if the algorithm visits any distorted search point except $\vec{0}$, then this search point has fitness at least n, so that the fitness target is achieved. Hence, the algorithm terminates when it reaches any distorted search point except $\vec{0}$. Therefore, any distorted search point except for $\vec{0}$ makes the runtime smaller. We may thus pessimistically assume that all search points except for $\vec{0}$ are clean.

This leaves us with two cases. If $\vec{0}$ is also clean, then DISOM = OM, and the expected runtime of the $(1, \lambda)$ EAwith $\lambda = 3 \ln n$ is $O(n \ln n)$ in this case [7]. In the other case, $\vec{0}$ is distorted. Note that this case occurs only with probability $p = 2^{-n}$. If the algorithm does not hit $\vec{0}$, then we can argue as before, so let us pessimistically assume that the algorithm starts in $\vec{0}$ or hits it during its run. Then after an expected $\Theta(q^{-1}) = n^{O(1)}$ generations, it does not duplicate $\vec{0}$ and thus proceeds to a search point $x \neq \vec{0}$. Thus, there is an i^* such that $x_{i^*} \neq 0$. We claim that the $(1, \lambda)$ EAhas a probability of at least $\rho = n^{-O(1)}$ to reach the optimum in $n^{O(1)}$ steps from x without visiting $\vec{0}$ again. Note that this implies the bound on $\mathbb{E}[T^{\text{comma}}]$, since whenever the $(1, \lambda)$ EAis in $\vec{0}$, it has a probability of $\Omega(\rho)$ of leaving $\vec{0}$ and reaching the optimum in the next $n^{O(1)}$ steps. Hence, this case contributes at most $p \cdot O(1/\rho) \cdot n^{O(1)} = 2^{-n} n^{O(1)} \leq 1$ to $\mathbb{E}[T^{\text{comma}}]$.

So it remains to prove the estimate for ρ. Let $C > 0$ be a large constant to be fixed later. With probability $(1 - 1/n)^{Cn \ln n} \geq e^{-2C \ln n} = n^{-2C}$, the $(1, \lambda)$ EAdoes not flip the i^*-th bit in any of the next $2Cn \ln n$ mutations. If this happens, then the algorithm in particular does not return to $\vec{0}$ during this time. Moreover, if the i^*-th bit remains unchanged then the algorithm simply optimizes an $(n-1)$-dimensional ONEMAXinstance during this time.[7] If C is sufficiently large, the expected time to reach the optimum of the $(n-1)$-dimensional problem is at most $Cn \ln n$, and by Markov's inequality the probability of needing more than $2Cn \ln n$ steps is at most 1/2. Hence, the probability of finding the optimum in $Cn \ln n$ generations is at least $\frac{1}{2} n^{-2C}$. This concludes the proof.

8 Conclusion

We have shown that a comma strategy can indeed help for dealing with local optima. To this end, we have introduced the new theoretical benchmark DISOM. We believe that this benchmark is of wider interest for studying local optima. As discussed in the introduction, arguably the popular benchmarks JUMPand CLIFFhave rather atypical local optima, and DISOMis a very simple way of adding local optima to the simple OMfunction. Thus, it would be very interesting to investigate how other non-elitist selection mechanisms like tournament selection [56], linear ranking selection, or fit-

[7] Note that for a ONEMAXproblem on $n' := n - 1$ bits, the mutation rate is $1/n = 1/(n'+1)$ instead of $1/n'$. However, it is clear that this deviation is negligible.

ness-proportionate selection [57] perform (see [31, 58] for overviews on non-elitist selection), and whether this can be phrased more generally in terms of selective pressure [59]. It would also be interesting to see whether this allows for parameter settings that find the optimum on DISOMefficiently, rather than only reaching a fixed target.

Our proof also gives insights into how the $(1, \lambda)$ EAescapes local optima. In particular, in the DISOMlandscape under Assumption 1.4, our proof shows that the $(1, \lambda)$ EAtruly escapes local optima: after escaping, it never hits the same local optimum for the second time.

Interestingly, our proofs suggest that there might be a larger gap between the $(1, \lambda)$ EAand the $(1 + \lambda)$ EAif the distortion d is not fixed for all distorted points, but if it randomly varies a bit by less than ± 0.5. In that case, our proofs for the $(1, \lambda)$ EAwould still go through, but our analysis suggests that the $(1 + \lambda)$ EAmay get stuck in local optima that slow it down even further. Whether this is really the case remains an open question. In follow-up work, larger variations of the distortion (e.g., drawn from a Gaussian distributed with constant variation) were shown to have a devastating impact on the $(1 + \lambda)$ EAand to lead to an exponential slowdown [29].

Of course, ONEMAXis not the only function that can be distorted. The same process can be applied to any other function, for example to any linear function. As discussed before Lemma 6.2, we suspect that for the $(1 + \lambda)$ EAthere is a real difference between ONEMAXand other linear functions, and that the huge fitness plateaus of ONEMAXare important for the $(1 + \lambda)$ EAto be efficient.

Acknowledgements We are thankful for the fruitful discussions at the Dagstuhl seminar 22081 "Theory of Randomized Optimization Heuristics", which triggered this research, as well as the Dagstuhl seminar 22182 "Estimation-of-Distribution Algorithms: Theory and Applications". The work of JJ is supported by the National Science Foundation under Grant No. DMS-1928930, while he was in residence at the Simons Laufer Mathematical Sciences Institute in Berkeley, California, during the spring semester of 2025. We are greatly indebted to the reviewers who made countless constructive suggestions improving the quality of this manuscript, especially one reviewer who went so far as to suggest Fig. 1 and even provided us with the tikz source code for it.

Author Contributions All authors contributed equally to the manuscript.

Funding Open access funding provided by Swiss Federal Institute of Technology Zurich

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. Doerr, C., Lengler, J.: Introducing elitist black-box models: When does elitist behavior weaken the performance of evolutionary algorithms? Evol. Comput. **25**(4), 587–606 (2017)
2. Dang, D.-C., Eremeev, A., Lehre, P.K.: Escaping local optima with non-elitist evolutionary algorithms. In: AAAI Conference on artificial intelligence (AAAI 2021), **35**, 12275–12283 (2021)
3. Dang, D.-C., Eremeev, A., Lehre, P.K.: Non-elitist evolutionary algorithms excel in fitness landscapes with sparse deceptive regions and dense valleys. In: Genetic and evolutionary computation conference (GECCO 2021), pp. 1133–1141 (2021)
4. Auger, A., Fonseca, C.M., Friedrich, T., Lengler, J., Gissler, A.: Theory of Randomized Optimization Heuristics (Dagstuhl Seminar 22081). Dagstuhl Rep. **12**(2), 87–102 (2022)
5. Doerr, B.: Does comma selection help to cope with local optima? Algorithmica **84**, 1659–1693 (2022)
6. Jägersküpper, J., Storch, T.: When the plus strategy outperforms the comma strategy and when not. In: Foundations of computational intelligence (FOCI 2007), pp. 25–32 (2007)
7. Rowe, J.E., Sudholt, D.: The choice of the offspring population size in the $(1, \lambda)$ evolutionary algorithm. Theoret. Comput. Sci. **545**, 20–38 (2014)
8. Hevia Fajardo, M.A., Sudholt, D.: Self-adjusting offspring population sizes outperform fixed parameters on the Cliff function. Artif. Intell. **328**, 104061 (2024). https://doi.org/10.1016/j.artint.2023.104061
9. Kaufmann, M., Larcher, M., Lengler, J., Zou, X.: Self-adjusting population sizes for the $(1, \lambda)$-EA on monotone functions. In: Parallel problem solving from nature (PPSN 2022), pp. 569–585 (2022). Springer
10. Kaufmann, M., Larcher, M., Lengler, J., Zou, X.: OneMax is not the easiest function for fitness improvements. In: Evolutionary computation in combinatorial optimization (EvoCOP 2023), pp. 162–178. Springer, New York (2023)
11. Hevia Fajardo, M.A., Sudholt, D.: Hard problems are easier for success-based parameter control. In: Genetic and evolutionary computation conference (GECCO 2022), pp. 796–804 (2022)
12. Droste, S.: Analysis of the (1+1) EA for a noisy OneMax. In: Proceedings of the genetic and evolutionary computation conference (GECCO 2004), pp. 1088–1099. Springer, New York (2004)
13. Gießen, C., Kötzing, T.: Robustness of populations in stochastic environments. Algorithmica **75**(3), 462–489 (2016)
14. Dang, D.-C., Lehre, P.K.: Runtime analysis of non-elitist populations: From classical optimisation to partial information. Algorithmica **75**(3), 428–461 (2016)
15. Qian, C., Yu, Y., Zhou, Z.-H.: Analyzing evolutionary optimization in noisy environments. Evol. Comput. **26**(1), 1–41 (2018)
16. Qian, C., Bian, C., Jiang, W., Tang, K.: Running time analysis of the $(1 + 1)$-EA for OneMax and LeadingOnes under bit-wise noise. Algorithmica (2018)
17. Sudholt, D.: Analysing the robustness of evolutionary algorithms to noise: refined runtime bounds and an example where noise is beneficial. Algorithmica **83**(4), 976–1011 (2021)
18. Bian, C., Qian, C., Tang, K.: Towards a running time analysis of the (1+1)-EA for OneMax and LeadingOnes under general bit-wise noise. In: Parallel problem solving from nature (PPSN 2018), pp. 165–177. Springer, Cham (2018)
19. Prugel-Bennett, A., Rowe, J., Shapiro, J.: Run-time analysis of population-based evolutionary algorithm in noisy environments. In: Foundations of genetic algorithms (FOGA 2015), pp. 69–75. ACM, New York (2015)
20. Dang, D.-C., Lehre, P.K.: Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms. In: Foundations of genetic algorithms (FOGA 2015), pp. 62–68. ACM, New York (2015)
21. Sudholt, D., Thyssen, C.: A simple ant colony optimizer for stochastic shortest path problems. Algorithmica **64**(4), 643–672 (2012)
22. Doerr, B., Hota, A., Kötzing, T.: Ants easily solve stochastic shortest path problems. In: Genetic and evolutionary computation conference (GECCO 2012), pp. 17–24 (2012)
23. Feldmann, M., Kötzing, T.: Optimizing expected path lengths with ant colony optimization using fitness proportional update. In: Foundations of genetic algorithms (FOGA 2013), pp. 65–74. ACM, New York (2013)

24. Friedrich, T., Kötzing, T., Krejca, M.S., Sutton, A.M.: The compact genetic algorithm is efficient under extreme Gaussian noise. IEEE Trans. Evol. Comput. **21**(3), 477–490 (2017)
25. Dang, D.-C., Opris, A., Salehi, B., Sudholt, D.: Analysing the robustness of NSGA-II under noise. In: Genetic and evolutionary computation conference (GECCO 2023), pp. 642–651 (2023)
26. Dinot, M., Doerr, B., Hennebelle, U., Will, S.: Runtime analyses of multi-objective evolutionary algorithms in the presence of noise. In: International joint conference on artificial intelligence (IJCAI 2023) (2023)
27. Friedrich, T., Kötzing, T., Neumann, F., Radhakrishnan, A.: Theoretical study of optimizing rugged landscapes with the cGA. In: Parallel problem solving from nature (PPSN 2022), pp. 586–599. Springer, New York (2022)
28. Dang, D.-C., Lehre, P.K.: The SLO hierarchy of pseudo-boolean functions and runtime of evolutionary algorithms. In: Genetic and evolutionary computation conference (GECCO 2024), pp. 1551–1559 (2024)
29. Lengler, J., Schiller, L., Sieberling, O.: Plus strategies are exponentially slower for planted optima of random height. In: Genetic and evolutionary computation conference (GECCO 2024), pp. 1587–1595 (2024)
30. Buzdalov, M., Doerr, B., Doerr, C., Vinokurov, D.: Fixed-target runtime analysis. Algorithmica **84**(6), 1762–1793 (2022)
31. Lehre, P.K.: Fitness-levels for non-elitist populations. In: Genetic and evolutionary computation conference (GECCO 2011), pp. 2075–2082 (2011)
32. Antipov, D., Doerr, B., Yang, Q.: The efficiency threshold for the offspring population size of the (μ, λ) EA. In: Genetic and evolutionary computation conference (GECCO 2019), pp. 1461–1469 (2019)
33. Jorritsma, J., Lengler, J., Sudholt, D.: Comma selection outperform plus selection on OneMax with randomly planted optima. In: Genetic and evolutionary computation conference (GECCO 2023). ACM Press, New York (2023). To appear
34. Lengler, J.: Drift analysis. In: Theory of evolutionary computation, pp. 89–131. Springer, New York (2020)
35. Johannsen, D.: Random combinatorial structures and randomized search heuristics. PhD thesis, Universität des Saarlandes (2010)
36. Doerr, B., Goldberg, L.A.: Adaptive drift analysis. Algorithmica **65**(1), 224–250 (2013)
37. Bossek, J., Sudholt, D.: Do additional target points speed up evolutionary algorithms? Theor. Comput. Sci. (2023). https://doi.org/10.1016/j.tcs.2023.113757
38. Doerr, B.: Probabilistic tools for the analysis of randomized optimization heuristics. In: Theory of evolutionary computation, pp. 1–87. Springer, New York (2020)
39. Janson, S.: Tail bounds for sums of geometric and exponential variables. Stat. Probab. Lett. **135**, 1–6 (2018)
40. Hevia Fajardo, M.A., Sudholt, D.: Self-adjusting population sizes for non-elitist evolutionary algorithms: why success rates matter. Algorithmica **86**(2), 526–565 (2024). https://doi.org/10.1007/S00453-023-01153-9
41. Doerr, B., Johannsen, D., Winzen, C.: Multiplicative drift analysis. Algorithmica **4**(64), 673–697 (2012)
42. Sudholt, D.: A new method for lower bounds on the running time of evolutionary algorithms. IEEE Trans. Evol. Comput. **17**(3), 418–435 (2013)
43. Witt, C.: Tight bounds on the optimization time of a randomized search heuristic on linear functions. Comb. Probab. Comput. **22**(2), 294–318 (2013)
44. Doerr, B.: Analyzing randomized search heuristics via stochastic domination. Theoret. Comput. Sci. **773**, 115–137 (2019)
45. Doerr, B.: The runtime of the compact genetic algorithm on jump functions. Algorithmica **83**, 3059–3107 (2021)
46. Jansen, T., Jong, K., Wegener, I.: On the choice of the offspring population size in evolutionary algorithms. Evol. Comput. **13**(4), 413–440 (2005)
47. Lässig, J., Sudholt, D.: Adaptive population models for offspring populations and parallel evolutionary algorithms. In: Foundations of genetic algorithms (FOGA 2011), pp. 181–192. ACM, New York (2011)
48. Gießen, C., Witt, C.: The interplay of population size and mutation probability in the (1+λ) EA on onemax. Algorithmica **78**(2), 587–609 (2017)

49. Doerr, B., Künnemann, M.: How the (1+λ) evolutionary algorithm optimizes linear functions. In: Genetic and evolutionary computation conference (GECCO 2013), pp. 1589–1596. ACM, New York (2013)
50. Badkobeh, G., Lehre, P.K., Sudholt, D.: Unbiased black-box complexity of parallel search. In: Parallel problem solving from nature (PPSN 2014), pp. 892–901. Springer, New York (2014)
51. Lehre, P.K., Sudholt, D.: Parallel black-box complexity with tail bounds. IEEE Trans. Evol. Comput. **24**(6), 1010–1024 (2020)
52. Lehre, P.K., Witt, C.: Tail bounds on hitting times of randomized search heuristics using variable drift analysis. Comb. Probab. Comput. **30**(4), 550–569 (2021)
53. Paixão, T., Pérez Heredia, J., Sudholt, D., Trubenová, B.: Towards a runtime comparison of natural and artificial evolution. Algorithmica **78**(2), 681–713 (2017)
54. Hevia Fajardo, M.A., Sudholt, D.: Self-adjusting population sizes for non-elitist Evolutionary Algorithms: Why success rates matter. In: Genetic and evolutionary computation conference (GECCO 2021), pp. 1151–1159 (2021)
55. Feller, W., et al.: An introduction to probability theory and its applications (1971)
56. Lehre, P.K., Qin, X.: More precise runtime analyses of non-elitist evolutionary algorithms in uncertain environments. Algorithmica, 1–46 (2022)
57. Happ, E., Johannsen, D., Klein, C., Neumann, F.: Rigorous analyses of fitness-proportional selection for optimizing linear functions. In: Genetic and evolutionary computation conference (GECCO 2008), pp. 953–960 (2008)
58. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. In: Foundations of genetic algorithms (FOGA 1991) vol. 1, pp. 69–93 (1991)
59. Lehre, P.K.: Negative drift in populations. In: Parallel problem solving from nature (PPSN 2010), pp. 244–253 (2010). Springer

Authors and Affiliations

Joost Jorritsma[1] · Johannes Lengler[2] · Dirk Sudholt[3]

✉ Joost Jorritsma
joost.jorritsma@stats.ox.ac.uk

✉ Johannes Lengler
johannes.lengler@inf.ethz.ch

✉ Dirk Sudholt
Dirk.Sudholt@uni-passau.de

[1] University of Oxford, Oxford, UK

[2] ETH Zürich, Zürich, Switzerland

[3] University of Passau, Passau, Germany