



Solving Problems with Unknown Solution Length at Almost No Extra Cost

Benjamin Doerr¹ · Carola Doerr² · Timo Kötzing³

Received: 8 June 2017 / Accepted: 16 June 2018 / Published online: 3 July 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Following up on previous work of Cathabard et al. (in: Proceedings of foundations of genetic algorithms (FOGA'11), ACM, 2011) we analyze variants of the $(1+1)$ evolutionary algorithm (EA) for problems with unknown solution length. For their setting, in which the solution length is sampled from a geometric distribution, we provide mutation rates that yield for both benchmark functions ONEMAX and LEADINGONES an expected optimization time that is of the same order as that of the $(1+1)$ EA knowing the solution length. More than this, we show that almost the same run times can be achieved even if *no* a priori information on the solution length is available. We also regard the situation in which neither the number nor the positions of the bits with an influence on the fitness function are known. Solving an open problem from Cathabard et al. we show that, for arbitrary $s \in \mathbb{N}$, such ONEMAX and LEADINGONES instances can be solved, simultaneously for all $n \in \mathbb{N}$, in expected time $O(n(\log(n))^2 \log \log(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$ and $O(n^2 \log(n) \log \log(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$, respectively; that is, in almost the same time as if n and the relevant bit positions were known. For the LEADINGONES case, we prove lower bounds of same asymptotic order of magnitude apart from the $(\log^{(s)}(n))^\varepsilon$ factor. Aiming at closing this arbitrarily small remaining gap, we realize that there is no asymptotically best performance for this problem. For any algorithm solving, for all n , all instances of size n in expected time at most $T(n)$, there is an algorithm doing the same in time $T'(n)$ with $T' = o(T)$. For ONEMAX we show results of similar flavor.

Results presented in this work are based on [11,13].

✉ Carola Doerr
Carola.Doerr@lip6.fr

¹ École Polytechnique, CNRS, LIX - UMR 7161, 91120 Palaiseau, France

² Sorbonne Université, CNRS, LIP6, 75005 Paris, France

³ Hasso-Plattner-Institut, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany

Keywords Black-box optimization · Evolutionary computation · Runtime analysis · Uncertainty · Unknown solution length

1 Introduction

Evolutionary algorithms (EAs) form a class of black-box optimizers, that—thanks to their high flexibility and their ability to produce high-quality solutions for a broad range of problems—are today well-established problem solvers in industrial and academic applications, where they are often used as subroutines for particularly difficult parts of the optimization problem, as well as for several pre- or post-processing steps of state-of-the-art optimization methods. The predominant part of research on EAs focuses on engineering and empirical work. This research is complemented by the *theory of evolutionary computation* (EC), which aims at providing mathematically founded statements about the performance of general-purpose problem solvers like EAs and similar randomized search heuristics, with one of its main objectives being to provide rigorous insights into the *working principles* of such optimization techniques. This way, theory of EC has helped to debunk common misconceptions about the performance of EAs and has served as a source of inspiration to design more efficient EAs.

One of the key performance measure in EC is expected *run time* (also referred to as *optimization time*), i.e., the expected number of function evaluations that are needed until an optimal solution is evaluated for the first time. Run time analysis is hence one of the core topics in the theory of EC. Thanks to significant methodological advances in the last 20 years, run time analysis for static problem settings is today quite well developed [3,23,28]. However, a topic that has received much less attention in the theory of EA literature is the performance of EAs in *uncertain environments*. Uncertainty can come in many guises, for example with respect to function evaluations, the variation operators, or the dynamics of the fitness function. Understanding how evolutionary search algorithms can tackle such uncertain environments is an emerging research topic; see [4] for a survey on examples in combinatorial optimization, but also [25] for a survey also discussing different sources of uncertainty.

Following up on a line of research initiated in [6], we study what EAs can achieve in the presence of uncertainty with respect to the solution length. More precisely, we regard how well an EA can optimize small problems that are hidden in larger representations. Our aim is to design algorithms which, on each such small problem, have a performance comparable to the one that could be obtained with similar algorithms which know the length and the location of the small problem in the representation. Quite surprisingly, we are able to show that already some variants of the simplest evolutionary algorithm, the $(1 + 1)$ EA (cf. Algorithm 1 for the pseudocode of this classic EA), can be very efficient for such problems.

To model uncertainty with respect to the solution length, we assume that we have a large search space $\{0, 1\}^N$ consisting of all bit strings of length N . The objective function f that we want to optimize, however, depends only on a small subset of the bit positions; that is, there is a set $I \subseteq [1..N] := \{1, \dots, N\}$ such that $f(x) = f(x|_I)$ for all $x \in \{0, 1\}^N$, where we write $x|_I$ to denote the bit string composed of only those

positions i with $i \in I$. We call $n := |I|$ the *relevant size* of our optimization problem. We regard two models of uncertainty.

1. *Initial segment uncertainty model*: We assume that I is an initial segment of $[1..N]$, that is, that $I = [1..n]$ for some $n \leq N$. This model was proposed and investigated in [6]; it is motivated there by applications in the design of testing sequences for finite state machines [27].
2. *Unrestricted uncertainty model*: In this model, we allow I to be an arbitrary subset of $[1..N]$ having cardinality $n = |I| \leq N$. Since position-dependent mutation rates cannot reasonably cope with such problems, it was posed as open problem in [6] whether there are evolutionary algorithms that can deal with this stronger type of uncertainty.

1.1 Previous Work

Cathabard et al. [6], were the first to consider, from a theoretical point of view, evolutionary algorithms in environments with unknown solution lengths. Cathabard et al. regard the initial segment uncertainty model and assume that the solution length is sampled from a fixed and known distribution D with finite support. More precisely, they assume that the solution length n is sampled from a truncated version of the geometric distribution, in which the probability mass for values greater than some threshold N is shifted to the event that $n = N$. In this situation, the algorithm designer has access to both the upper bound N for the solution length and the success probability q of the distribution.

Cathabard et al. analyze a variant of the $(1 + 1)$ EA in which each bit is flipped with probability $1/N$ and they also study a variant with position-dependent bit-flip probabilities. In the latter, the i -th bit is flipped independently of all other bits with probability $p_i = 1/(i + 1)$. For any fixed choice of vector of bit flip probabilities \vec{p} , we denote the corresponding algorithm with $(1 + 1)$ EA $_{\vec{p}}$.

Cathabard et al. consider the classic benchmark problems ONEMAX defined by $OM : \{0, 1\}^n \rightarrow \mathbb{R}, x \mapsto \sum_{i=1}^n x_i$ and LEADINGONES defined by $LO : \{0, 1\}^n \rightarrow \mathbb{R}, x \mapsto \max \{i \in \{0, 1, \dots, n\} \mid \forall j \leq i : x_j = 1\}$. They show that, for their two choices of \vec{p} , the $(1 + 1)$ EA $_{\vec{p}}$ has an expected run time which is polynomial in N and n , where the expectation is taken with respect to the solution length and the random decisions of the algorithm. An overview of the precise bounds obtained in [6] is given in Table 1.

1.2 Our Results

We extend the work of Cathabard et al. in several ways. In a first step (Sect. 3) we show that the regarded mutation probabilities are sub-optimal. Making use of the light upper tail of the (truncated) geometric distribution, we design bit flip probabilities that yield significantly smaller expected run times (for both the ONEMAX and the LEADINGONES function). For LEADINGONES, we complement this finding by a lower bound that proves that no mutation probabilities can yield a performance that is better

Table 1 Expected run times of the $(1 + 1)$ EA $_{\vec{p}}$ with $\vec{p} = (p_i)_{i \in \mathbb{N}}$ in the initial segment uncertainty model regarded in [6] in which the solution length n is sampled from the truncated geometric distribution $D = \text{TruncGeo}(N, q)$ with $1/N \leq q \leq 1/2$

Problem	Cor. 4	Results from [6]		Thms. 7 (OM), 10 (LO)	
		$p_i = 1/N$	$p_i = 1/(i + 1)$	$p_i = q/2$	$p_i = q$
ONEMAX $_D$		$\Theta\left(N \log \frac{1}{q}\right)$	$O\left(\frac{1}{q^2} \log N\right)$	$\Theta\left(\frac{1}{q} \log \frac{1}{q}\right)$	$\Theta(N \log N)$
LEADINGONES $_D$	$\Omega\left(1/q^2\right)$	$\Theta(N/q)$	$\Theta\left(1/q^3\right)$	$\Theta\left(1/q^2\right)$	$\Theta(N/q)$

The algorithm designer knows N , i.e., an upper bound for the solution length n

by more than a constant factor than our suggested ones. Table 1 summarizes our findings and compares them to the results from [6].

While in the setting of Cathabard et al. we are in the convenient situation that we have full knowledge of the distribution D from which the solution length is sampled, one may wonder if this knowledge is necessary. We therefore study in Sect. 4 what can be done in the initial segment uncertainty model without any prior knowledge about the solution length. In this situation we require that the algorithm designer chooses bit flip probabilities $(p_i)_{i \in \mathbb{N}}$ such that, regardless of the solution length n , the expected performance of the $(1 + 1)$ EA $_{\vec{p}}$ using bit flip probabilities $\vec{p} = (p_1, \dots, p_n)$ is as small as possible. It is not obvious that this can be done in polynomial time. In fact, for both algorithms studied by Cathabard et al. as well as for any uniform choice ($p_i = p$ for all $i \in \mathbb{N}$) of the bit flip probabilities, the expected run time on this problem is exponential in n if N is exponential in n (cf. Theorems 13 and 14).

We show (Theorems 15 and 16) that not only this problem can be tackled with position-dependent bit flip probabilities, but, quite surprisingly, we shall also see that this can be done in a way that yields almost optimal run times. Indeed, for LEADINGONES our results are only a $(\log(n))^{1+\varepsilon}$ factor worse than the best possible $\Theta(n^2)$ run time bound. This factor can be made even smaller. The precise runtime bounds are $O(n(\log(n))^2 \log^{(2)}(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$ and $O(n^2 \log(n) \log^{(2)}(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$, for ONEMAX and LEADINGONES, respectively, where s is an arbitrary positive integer and $\log^{(j)}$ denotes the j -fold iterated logarithm (cf. Equation (6)).

In Sect. 5 we provide a second way to deal with unknown solution lengths. We introduce an alternative variant of the $(1 + 1)$ EA in which the bit flip probability, which is uniformly and independently applied to each bit position, is chosen according to some (fixed) distribution at the beginning of each iteration. For suitably chosen distributions Q , the expected run times of the respective $(1 + 1)$ EA $_Q$ on ONEMAX and LEADINGONES are of the same asymptotic order as those of the previously suggested solution with position-dependent bit flip probabilities $(p_i)_{i \in \mathbb{N}}$. In particular, for LEADINGONES the expected run times are, simultaneously for all possible solution lengths n , almost of the same order as the expected run time of a best possible $(1 + 1)$ EA knowing the solution length in advance.

This second approach has an advantage over the position-dependent bit flip probabilities in that it effectively ignores bits that do not contribute anything to the fitness

function (*irrelevant bits*). Unlike all previously suggested solutions, it can therefore deal with the unrestricted uncertainty model. These EAs therefore provide a positive answer to the above-mentioned question posed by Cathabard et al. [6, Sect. 6] concerning the ability of EAs do deal with unrestricted uncertainty.

Summarizing the above, we see that the EA variants—with either position-dependent or random mutation rates in the initial segment uncertainty model and using random mutation rates in the unrestricted uncertainty model—are only by a factor of $O(\log(n) \log^{(2)}(n) \dots \log^{(s-1)}(n) (\log^{(s)}(n))^{1+\varepsilon})$ slower than the performance of the $(1 + 1)$ EA on LEADINGONES when the relevant bits are known to the algorithms. Since the unrestricted model contains the initial segment model as a very small subcase (one ONEMAX and LEADINGONES instance of size n instead of $\binom{N}{n} \geq (N/n)^n$ ONEMAX and $n! \binom{N}{n}$ LEADINGONES instances)¹, these results raise the question if one can obtain a superior performance from using position-dependent mutation rates in the much more restricted initial segment uncertainty model. We address this question in Sects. 6 and 7, where we investigate how good the above-described results are.

For LEADINGONES, our lower bounds show that our run time results from Sects. 4 and 5 are very tight (e.g., up to an $O((\log^{(s)}(n))^\varepsilon)$ factor for arbitrary $s \in \mathbb{N}$). This in particular implies that for the LEADINGONES problem, not much can be gained by using position-dependent mutation rates in the heavily restricted initial segment model. Much stronger than that, we show that the $(1 + 1)$ EA with any mutation operator that treats zeros and ones symmetrically, that is, that uses an arbitrary distribution on the subsets of $[1..N]$, selects a random subset according to this distribution, and then flips exactly the bits in this set, cannot obtain a better performance in either the initial segment or the unrestricted model than the $(1 + 1)$ EA with position-dependent rates (in the initial segment model) or the $(1 + 1)$ EA with random mutation rates (in either model). This shows in a very strong sense that the two uncertainty models are equally difficult when optimizing LEADINGONES.

We said above that we have very good lower bounds. We would have liked to claim that we have matching upper and lower performance bounds. We cannot do so, as we observe a curiosity of this optimization under uncertainty problem: There is no asymptotically best performance. More precisely, for any function $T : \mathbb{N} \rightarrow \mathbb{N}$ such that there is an algorithm solving LEADINGONES instances with unknown length n in expected time at most $T(n)$, there is a function $T' : \mathbb{N} \rightarrow \mathbb{N}$ such that $T' = o(T)$ and there is an algorithm solving LEADINGONES instances with unknown length n in expected time at most $T'(n)$.

For the case of ONEMAX we get an analogous result for the unrestricted uncertainty model: any choice of distribution Q over bit flip probabilities can be improved with a distribution Q' so that using bit flip probabilities chosen from Q' lead to an asymptotically better run time than using those chosen from Q . For the case of the initial segment uncertainty model it continues to be open whether there is an optimal run time, or whether there are lower bounds similar to those for LEADINGONES.

¹ Note here that in the initial segment model, we assume that the set of LEADINGONES instances is not invariant under permutation, that is, for a fixed n there is just the usual LEADINGONES instance measuring the largest segment $[1..i] \subseteq [1..n]$ that contains only 1-bits. For the unrestricted model, we assume that the LEADINGONES instance is implemented on the set I of relevant bits in an arbitrary ordering of the bit positions. See Sect. 2.1 for precise definitions.

Table 2 Overview of Results, where “Initial” denotes the initial segment uncertainty model and the second row indicates whether bit flips are made with a uniform or a position-dependent bit flip probability (in one case chosen at random in each iteration)

Setting	Bit flips	OneMax		LeadingOnes	
Initial, random length $\sim \text{Geo}(q)$	unif.	$O(q^{-1} \log q^{-1})$	Thm. 7	$O(q^{-2})$	Thm. 10
				$\Omega(q^{-2})$	Thm. 3
Initial, adversarial length	unif.	$2^{\Omega(n)}$	Thm. 13	$2^{\Omega(n)}$	Thm. 14
	pos.-dep.	$O(n(\log n)^{2+\varepsilon})$	Cor. 17	$O(n^2(\log n)^{1+\varepsilon})$	Cor. 17
Unrestricted uncertainty	unif./rand.	$O(n(\log n)^{2+\varepsilon})$	Cor. 21	$O(n^2 \log n)$	Thm. 34
				$O(n^2(\log n)^{1+\varepsilon})$	Cor. 21
		$\omega(n(\log n)^2)$	Cor. 40	$\omega(n^2 \log n)$	Cor. 35

Results regarding the Random Length Model use the geometric distribution with parameter q , where $1/N < q < 1/2$ and $\varepsilon > 0$. The bounds for fixed bit flip probabilities and uniform and randomized bit flip probabilities in the adversarial model are $O(n(\log(n))^2 \log^{(2)}(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$ and $O(n(\log_b(n))^2 \log_b^{(2)}(n) \dots)$ for ONEMAX, and $O(n^2 \log(n) \log^{(2)}(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$ and $O(n^2 \log_b(n) \log_b^{(2)}(n) \dots)$ for LEADINGONES, where $s \in \mathbb{N}$, $\varepsilon > 0$, and $b < e$ are arbitrary

Our run time results are summarized in Tables 1 and 2. This work builds on results that we have previously announced in [11,13].

2 Problem Setting and Algorithms

We use this section to fix notation (Sect. 2.1), to make precise the models of uncertainty (Sect. 2.2), and to define the EA variants capable of coping with uncertainty (Sect. 2.3).

2.1 Basic Notation, Summability, OneMax, and LeadingOnes

We use the following standard notation. We write \mathbb{N} to denote the positive integers. We write $[a..b]$ to denote the set of integers no less than a and no larger than b . We abbreviate $[n] := [1..n]$.

A sequence \vec{p} is a mapping $\vec{p} : \mathbb{N} \rightarrow \mathbb{R}$, which equivalently can be written as $\vec{p} = (p_n)_{n \in \mathbb{N}}$. We alternate between these two notations. A sequence $\vec{p} = (p_n)_{n \in \mathbb{N}}$ is said to be *monotonically decreasing* if, for all $n \in \mathbb{N}$, it holds that $p_n \geq p_{n+1}$. It is *summable* if the sequence $(S_n)_{n \in \mathbb{N}}$ of partial sums of absolute values $S_n := \sum_{i=1}^n |p_i|$ converges.

We are mostly interested in the asymptotic behavior of our algorithms. We therefore use a lot Landau’s big- and small-Oh notation. For convenience, we write $\vec{p} = o(\vec{q})$ instead of $p(n) = o(q(n))$ whenever the variable n is clear from the context.

We regard in this work the two classic benchmark problems ONEMAX and LEADINGONES, which, for given n , assign to each bit string $x \in \{0, 1\}^n$ the number of ones in x and the number of initial ones before the first zero entry, respectively; that is,

$$\begin{aligned}
 \text{OM}_n &:= \text{ONEMAX}_n(x) = \sum_{i=1}^n x_i, \text{ and} \\
 \text{LO}_n &:= \text{LEADINGONES}_n(x) \\
 &= \max\{i \in [0..n] \mid \forall j \leq i : x_j = 1\}.
 \end{aligned}$$

We omit the subscript n when it is clear from the context.

ONEMAX and LEADINGONES functions are the two best-studied problems in the theory of evolutionary computation literature. How the $(1 + 1)$ EA, $(1 + \lambda)$ EA, $(\mu + 1)$ EA, and $(\mu + \lambda)$ EA optimize these functions is relatively well understood [1, 18,20,24,30] (with the exception of the runtime of the $(\mu + \lambda)$ EA on LEADINGONES). For the performance of the $(1 + 1)$ EA, not only the asymptotic order of magnitude is known [20], namely $\Theta(n \log n)$ for ONEMAX and $\Theta(n^2)$ for LEADINGONES, but even more precise results. Ending a series of works [14,15,20,29] on the expected runtime $E[T]$ of the $(1 + 1)$ EA on ONEMAX, the very precise estimate of $E[T] = en \ln(n) + c_1 n + \frac{1}{2} e \ln(n) + c_2 + O((\log n)/n)$ for concrete constants c_1, c_2 was proven in [22]. For the expected runtime of the $(1 + 1)$ EA on LEADINGONES, the exact value of $E[T] = \frac{1}{2} n^2 ((1 - \frac{1}{n})^{-n+1} - 1 + \frac{1}{n})$ was determined, independently, in [5,29].

We regard in this work ONEMAX and LEADINGONES functions of unknown solution length. If a distribution D is known from which the solution length is sampled we consider the expected run time of our algorithms on ONEMAX_D and LEADINGONES_D , respectively, which are the problems OM_n respectively LO_n with random solution length $n \sim D$. Note here that the expectation is taken both with respect to the random solution length and with respect to the random samples of the algorithm.

2.2 Models of Uncertainty

The focus of our work is on how evolutionary algorithms optimize objective functions that only depend on a small subset of the decision variables. Hence we assume that there is a large search space $\{0, 1\}^N$, but our objective function f can be written as $f(x) = \tilde{f}(x|_I)$ for some subset $I \subset [1..N]$ with $n := |I| \ll N$ and some function $\tilde{f} : \{0, 1\}^I \rightarrow \mathbb{R}$.

It turns out that the dimension N of the large search space is not very relevant for our results. For this reason we assume that the large search space is in fact $\{0, 1\}^{\mathbb{N}}$, that is, the set of all infinite binary sequences. Since the complexity measure we regard is the number of fitness evaluations (and not the number of elementary operations as in classic algorithmics), this expansion of the search space has no influence on our run time statements.

We then regard two *models of uncertainty* about the location of the relevant part of the problem instance. To remain consistent with previous works, we call both a *model of unknown solution length* despite the fact that in the second model also the relevant bit positions are unknown.

Initial segment uncertainty: We assume that the set I of relevant bits is an initial segment $[1..n]$ of the outer space $\{0, 1\}^{\mathbb{N}}$. The number n is not known to the algorithm. When regarding LEADINGONES as fitness function f , we assume that f respects the

usual order of the bit-positions, that is, $f(x) = \text{LO}_n(x) = \max\{i \in [0..n] \mid \forall j \leq i : x_j = 1\}$.

Unrestricted uncertainty: We assume that the set I of relevant bits is any subset of \mathbb{N} of cardinality $|I| = n$. Neither n nor I is known to the algorithm. When regarding the LEADINGONES fitness function, we do not assume that it respects the natural order of the bits. Hence the problem instance is described by I and a bijective mapping $\sigma : [1..n] \rightarrow I$ such that $f(x) = \text{LO}_{n,I,\sigma}(x) = \max\{i \in [0..n] \mid \forall j \leq i : x_{\sigma(j)} = 1\}$.

2.3 Evolutionary Algorithms for Dealing With Unknown Solution Lengths

We now present two ways how evolutionary algorithms can deal with unknown solution length scenarios. Both are based on modifying the mutation operator. Hence, in principle, our ideas can be used in conjunction with any evolutionary algorithm in which mutation is used with significant probability. Nevertheless, to keep things simple and to not obscure the main ideas, we restrict ourselves to the $(1 + 1)$ EA, the simplest of all EAs.

The classic $(1 + 1)$ EA, whose pseudocode is given in Algorithm 1, starts by sampling an initial search point from $\{0, 1\}^n$ uniformly at random. It then proceeds in rounds, each of which consists of a mutation and a selection step. Throughout the whole optimization process the $(1 + 1)$ EA maintains a population size of one, and the individual in this population is always a best-so-far solution. In the *mutation step* of the $(1 + 1)$ EA, the current-best solution x is *mutated* by flipping every bit with probability $1/n$, independently of all other bits. The fitness of the resulting search point y is evaluated and in the *selection step* the parent x is replaced by its offspring y if and only if the fitness of y is at least as good as the one of x . Since we consider maximization problems here, this is the case if $f(y) \geq f(x)$. As usual in the theory of evolutionary algorithms, we are interested in expected *run times*, i.e., the expected number of rounds it takes until the $(1 + 1)$ EA evaluates for the first time a solution of maximal fitness. We therefore do not specify a termination criterion.

Algorithm 1: The $(1 + 1)$ EA with standard mutation probability $1/n$ for the optimization of a pseudo-Boolean function $f : \{0, 1\}^n \rightarrow \mathbb{R}$

```

1 Initialization: Sample  $x \in \{0, 1\}^n$  uniformly at random and compute  $f(x)$ ;
2 Optimization: for  $t = 1, 2, 3, \dots$  do
3    $y \leftarrow x$ ;
4   for  $i = 1, \dots, n$  do independently with probability  $1/n$  set  $y_i \leftarrow 1 - x_i$ ; compute  $f(y)$ ;
5   if  $f(y) \geq f(x)$  then  $x \leftarrow y$ 

```

The first idea to enable the $(1 + 1)$ EA to cope with uncertainty with respect to the solution length is to use *position-dependent* mutation rates [6]. This idea makes sense only for the initial segment uncertainty model. For a given sequence $\vec{p} : \mathbb{N} \rightarrow [0, 1]$, the $(1 + 1)$ EA $_{\vec{p}}$ (cf. Algorithm 2) is the standard $(1 + 1)$ EA, except that in the mutation step the offspring is generated from flipping the bit in position i with position-dependent probability p_i , independently of all other bits.

It is not difficult to see that the $(1 + 1) EA_{\vec{p}}$ indeed generalizes the standard $(1 + 1) EA$. In fact, we obtain the $(1 + 1) EA$ from the $(1 + 1) EA_{\vec{p}}$ if we set $p_i = 1/n$ for all $i \in [n] := \{1, \dots, n\}$. We call such mutation vectors with $p_i = p_j$ for all i, j *uniform* mutation rates.

We note that in [6] position-dependent mutation rates were referred to as *non-uniform*. In the context of our work, however, position-dependent is more precise, since we also consider EA variants that are non-uniform with respect to time.

Algorithm 2: The $(1 + 1) EA_{\vec{p}}$ with position dependent mutation rates $(p_i)_{i \in \mathbb{N}}$ to maximize a pseudo-Boolean function $f : \{0, 1\}^{\mathbb{N}} \rightarrow \mathbb{R}$ with finite number of relevant bits.

```

1 Initialization: Sample  $x \in \{0, 1\}^{\mathbb{N}}$  uniformly at random and compute  $f(x)$ ;
2 Optimization: for  $t = 1, 2, 3, \dots$  do
3    $y \leftarrow x$ ;
4   for  $i \in \mathbb{N}$  do independently with probability  $p_i$  set  $y_i \leftarrow 1 - x_i$  compute  $f(y)$ ;
5   if  $f(y) \geq f(x)$  then  $x \leftarrow y$ 

```

The second idea to overcome the challenges of an unknown solution length scenario, in particular in the unrestricted uncertainty model, is to use a random mutation rate sampled independently in each iteration from a suitable distribution. More precisely, let Q be a probability distribution over $[0, 1]$. For the ease of presentation, we restrict ourselves to discrete distributions. Then the $(1 + 1) EA_Q$ (see Algorithm 3 for the pseudocode) is again the classic $(1 + 1) EA$ with the sole exception that in each iteration t a mutation rate p_t is sampled from Q and then the offspring is generated from the parent individual by flipping each bit independently with probability p_t , that is, by performing standard bit mutation with mutation rate p_t . Again we see that this algorithm generalizes the classic $(1 + 1) EA$, which we obtain for choosing Q to be $1/n$ deterministically.

We will mostly work with distributions Q that have all their probability mass on the values $\{1/i \mid i \in \mathbb{N}\}$. These are most conveniently described via a probability distribution $(p_n)_{n \in \mathbb{N}}$ on \mathbb{N} , that is, via a sequence \vec{p} in $[0, 1]$ such that $\sum_{n=1}^{\infty} p_n = 1$, and then taking $\Pr_{Q(\vec{p})}(1/i) = p_i$. For convenience, we shall allow arbitrary summable sequences \vec{p} in $[0, 1]$ and then set $\Pr_{Q(\vec{p})}(1/i) = p_i / \sum_{n \in \mathbb{N}} p_n$.

Algorithm 3: The $(1 + 1) EA_Q$ with mutation rate distribution Q maximizing a pseudo-Boolean function $f : \{0, 1\}^{\mathbb{N}} \rightarrow \mathbb{R}$ depending only on a finite number of bits.

```

1 Initialization: Sample  $x \in \{0, 1\}^{\mathbb{N}}$  uniformly at random and compute  $f(x)$ ;
2 Optimization: for  $t = 1, 2, 3, \dots$  do
3    $y \leftarrow x$ ;
4   Sample  $p_t$  from  $Q$ ;
5   for  $i \in \mathbb{N}$  do independently with probability  $p_t$  set  $y_i \leftarrow 1 - x_i$  compute  $f(y)$ ;
6   if  $f(y) \geq f(x)$  then  $x \leftarrow y$ 

```

For reasons of completeness, we remark that there are a few other theoretical works using random mutation strengths, however with different targets and with different distributions. For multi-valued representations, more precisely, when the search space is $[0..r-1]^n$, [7, 12] show that a harmonic mutation strength often gives good results. Very roughly speaking, this means that a variable value $k \in [0..r-1]$ is changed to either $k + \delta$ or $k - \delta$ with probability proportional to $1/\delta$, and hence with probability $\Theta(\log(r)/\delta)$. In [19], it was shown that standard-bit mutation with a random mutation rate chosen according to a power-law distribution can greatly speed-up the optimization of non-unimodal functions. We note that the choice of the distribution generally is important. We could not obtain the performance shown in this work with either a harmonic or a power-law mutation strength.

3 Initial Segment Uncertainty Model with Random Solution Length

We first consider the setting introduced by Cathabard et al. [6]. After a short presentation of the model in Sect. 3.1, a general lower bound for LEADINGONES (Sect. 3.2), and the results of [6] in Sect. 3.3, we show that the bounds in [6] can be improved by using different (uniform) mutation rates (Sect. 3.4).

3.1 The Model

Cathabard et al. [6] consider the following model. The algorithm designer knows the distribution D from which the unknown solution length for the initial segment uncertainty model is drawn; only distributions with finite support are considered, so the algorithm designer knows an upper bound N on the actual solution length n . He also knows the class of functions from which the optimization problem is taken (for example ONEMAX or LEADINGONES).

Based on this knowledge, the algorithm designer chooses a vector (p_1, \dots, p_N) of bit flip probabilities indicating with which probability a bit is flipped in each round. As discussed in the previous section, we will also regard a slightly more general model in which the distributions over \mathbb{N} may possibly have infinite support; the algorithm designer then chooses an infinite sequence of bit flip probabilities $\vec{p} = (p_i)_{i \in \mathbb{N}}$. After this choice of bit flip probabilities, the actual solution length n is sampled from the given distribution D . Then the $(1+1)$ EA $_{\vec{p}}$ (Algorithm 2) is run with mutation probabilities $\vec{p} = (p_1, \dots, p_n)$ on the given problem with the given problem length. We use ONEMAX $_D$ and LEADINGONES $_D$, respectively, to denote the problem of optimizing ONEMAX and LEADINGONES, respectively, in this initial segment uncertainty model with random length drawn from D .

Cathabard et al. [6] consider as distribution D the following *truncated geometric distribution*, based on a geometric distribution where the probability mass for values greater than n are moved to n .

Definition 1 ([6]) The *truncated geometric distribution* $\text{TruncGeo}(N, q)$ with truncation parameter N and success probability $q \in (0, 1/N]$ satisfies, for all $n \in \mathbb{N}$, that the probability of $\text{TruncGeo}(N, q) = n$ is

$$\begin{cases} q(1 - q)^{n-1} & \text{if } 1 \leq n \leq N - 1, \\ (1 - q)^{n-1} & \text{if } n = N, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the truncated geometric distribution recovers the geometric distribution $\text{Geo}(q)$ for $N = \infty$.

It is well known, respectively can be found in [6, Proposition 1], that for $X = \text{Geo}(q)$ and $Y = \text{TruncGeo}(N, q)$ with $q \geq 1/N$

$$E[X] = q^{-1} \text{ and } E[Y] = \Theta(q^{-1}). \tag{1}$$

Note that we trivially have $E[Y] \leq E[X]$.

3.2 A General Lower Bound for LeadingOnes

What is a good lower bound for the expected run time of *any* $(1 + 1) \text{EA}_{\vec{p}}$ on ONEMAX or LEADINGONES when the length is sampled from some given distribution D on \mathbb{N} ? If the algorithm designer knew the true length n before he has to decide upon the mutation probabilities (p_1, \dots, p_n) , then the optimal bit flip probability for this solution length could be chosen. We show that for LEADINGONES , if the true length n is known, any setting of the bit-flip probabilities leads to an expected run time of $\Omega(n^2)$ regardless of the choice of \vec{p} .

Lemma 2 *For any fixed solution length n and any vector $\vec{p} = (p_i)_{i \in \mathbb{N}}$ of mutation probabilities, the expected run time of the $(1 + 1) \text{EA}_{\vec{p}}$ on LEADINGONES_n is $\Omega(n^2)$.*

Proof Via arguments analogous to the ones in [5, Sect. 3.3] (see [8, Sect. 2.3] for a formal proof) it is easy to see that the expected run time $E[T]$ of the $(1 + 1) \text{EA}_{\vec{p}}$ on LEADINGONES_n is

$$E[T] = \sum_{i=1}^n \frac{1}{2^{p_i}} \frac{1}{\prod_{j=1}^{i-1} (1 - p_j)}.$$

Using this bound one can easily show that we can assume without loss of generality that the mutation probabilities are monotonically increasing, i.e., $p_i \leq p_{i+1}$ holds for all $i \in [n]$. Indeed, if $p_k > p_{k+1}$ for some $k \in [n]$, the expected run time of the $(1 + 1) \text{EA}_{\vec{p}}$ is larger than that of the $(1 + 1) \text{EA}_{\vec{q}}$ with $\vec{q} = (q_1, \dots, q_n)$, $q_k = p_{k+1}$, $q_{k+1} = p_k$, and $q_i = p_i$ for $i \notin \{k, k + 1\}$.

Let $k := \lfloor n/2 \rfloor$. We have

$$\prod_{j=1}^k (1 - p_j) \leq (1 - p_k)^k \leq e^{-p_k k},$$

which gives

$$E[T] \geq \sum_{i=k+1}^n \frac{1}{2p_i} \frac{1}{\prod_{j=1}^k (1-p_j)} \geq \frac{n}{2} \frac{1}{2p_k} e^{p_k k}.$$

This shows that the overall expected optimization time of the $(1+1)$ EA $_{\vec{p}}$ on LEADINGONES $_n$ is $\Omega(n \exp(p_k k)/p_k)$. For all possible choices of p_k this expression is $\Omega(n^2)$: for $p_k = O(1/n)$ we have $n/p_k = \Omega(n^2)$ and $\exp(p_k k) = \Theta(1)$, while for larger p_k we see that $\exp(p_k k)/p_k$ is a growing function. \square

Using these lower bounds for fixed solution lengths, Jensen's Inequality and the convexity of $n \mapsto n^2$, we get the following general lower bound.

Theorem 3 *Let D be any distribution on \mathbb{N} with a finite expectation of m . Then for any sequence \vec{p} the expected run time of the $(1+1)$ EA $_{\vec{p}}$ on LEADINGONES $_D$ is $\Omega(m^2)$. This bound also applies to the setting in which the algorithm designer can choose the mutation probabilities $\vec{p} = (p_1, \dots, p_n)$ after the solution length $n \sim D$ has been drawn.*

Using Eq. (1), we get the following corollary.

Corollary 4 *Let $N \in \mathbb{N}$ and $q \geq 1/N$. Let $D = \text{TruncGeo}(N, q)$ or $D = \text{Geo}(q)$. For any sequence \vec{p} the expected run time of the $(1+1)$ EA $_{\vec{p}}$ on LEADINGONES $_D$ is $\Omega(q^{-2})$. This bound also applies to the setting in which the algorithm designer can choose the mutation probabilities $\vec{p} = (p_1, \dots, p_n)$ after the solution length $n \sim D$ has been drawn.*

We conjecture that analogous results hold for ONEMAX. More precisely, we conjecture that for any fixed solution length n and any vector \vec{p} of mutation rates the expected optimization time of the $(1+1)$ EA $_{\vec{p}}$ is $\Omega(n \log n)$. With this result, Theorems 3 and Corollary 4 would extend to ONEMAX, with $\Omega(m^2)$ and $\Omega(q^{-2})$ replaced by $\Omega(m \log m)$ and $\Omega(q^{-1} \log q^{-1})$, respectively.

3.3 Known Upper Bounds

Cathabard et al. [6] analyze the run time of the $(1+1)$ EA $_{\vec{p}}$ with uniform mutation probabilities $\vec{p} = (1/N)_{i \in \mathbb{N}}$ and of the $(1+1)$ EA $_i$, which is the $(1+1)$ EA $_{\vec{p}}$ with $\vec{p} = (p_i)_{i \in \mathbb{N}} := (1/(i+1))_{i \in \mathbb{N}}$.

For ONEMAX they obtain the following results.

Theorem 5 (Results for OneMax from [6]) *Let $N \in \mathbb{N}$, $\varepsilon \in (0, 1)$, and $q = N^{-\varepsilon}$. For $D = \text{TruncGeo}(N, q)$ the expected run time of the $(1+1)$ EA $_{\vec{p}}$ on ONEMAX $_D$ is $\Theta(N \log q^{-1})$ when $\vec{p} = (1/N)_{i \in \mathbb{N}}$, while the expected run time of the $(1+1)$ EA $_i$ on ONEMAX $_D$ is $O(q^{-2} \log N)$.*

This result shows that the $(1+1)$ EA $_{\vec{p}}$ with $\vec{p} = (1/N)_{i \in \mathbb{N}}$ outperforms the $(1+1)$ EA $_i$ for $q < 1/\sqrt{N}$, while the latter algorithm is preferable for larger q . As we shall see in

the following section one should not conclude from this result that position-dependent bit flip probabilities are the better choice for this problem.

Remark By using a slightly more careful analysis than presented in [6], the bound for the $(1 + 1) EA_i$ on $ONEMAX_D$ can be improved to $O(q^{-2} \log q^{-1})$. In fact, an analysis similar to the one in Sect. 3.4, that is disregarding outcomes that are much larger than the expectation, will give that result. It can also be shown that the requirement $q = N^{-\varepsilon}$ is not needed as the $O(q^{-2} \log q^{-1})$ bound holds for all $q > 1/N$. It also holds for the (non-truncated) geometric distribution $D = \text{Geo}(q)$.

For $LEADINGONES$, Cathabard et al. show the following results.

Theorem 6 (Results for LeadingOnes from [6]) For N, ε, q , and D as in Theorem 5, the expected run time of the $(1 + 1) EA_{\vec{p}}$ with $\vec{p} = (1/N)_{i \in \mathbb{N}}$ on $LEADINGONES_D$ is $\Theta(Nq^{-1})$, while the expected run time of the $(1 + 1) EA_i$ on $LEADINGONES_D$ is $\Theta(q^{-3})$.

Thus also for $LEADINGONES$ the $(1 + 1) EA_i$ performs better than the $(1 + 1) EA_{\vec{p}}$ with $\vec{p} = (1/N)_{i \in \mathbb{N}}$ when $q > 1/\sqrt{N}$ while the uniform $(1 + 1) EA_{\vec{p}}$ should be preferred for smaller q .

Remark As in the $ONEMAX$ case, the $\Theta(q^{-3})$ bound for the $(1 + 1) EA_i$ holds more generally for all geometric distributions $\text{Geo}(q)$ with parameter $q > 1/N$.

Theorem 6 shows that on $LEADINGONES_D$ the $(1 + 1) EA_i$ loses a factor of $1/q$ with respect to the lower bound given by Corollary 4. This will be improved in the following section.

3.4 Optimal Upper Bounds With Uniform Mutation Probabilities

We show that, for D being the (truncated or non-truncated) geometric distribution, there exist bit flip probabilities $\vec{p} = (p_i)_{i \in \mathbb{N}}$ such that the expected run time of the $(1 + 1) EA_{\vec{p}}$ on $ONEMAX_D$ and $LEADINGONES_D$ is significantly lower than those of the two algorithms studied by Cathabard et al. For $LEADINGONES$ the expected run time of our algorithm matches the lower bound given in Corollary 4 and is thus optimal in asymptotic terms.

In both cases, i.e., both for $ONEMAX_D$ (Theorem 7) and for $LEADINGONES_D$ (Theorem 10), the mutation rates yielding the improvement over the results in [6] are uniform. Our results therefore imply that for $LEADINGONES$, unlike conjectured in [6], one cannot gain more than constant factors from using position-dependent mutation probabilities.

The key observation determining our choice of the mutation probability is the fact that the (truncated) geometric distribution has a light upper tail, so that it is preferable to choose the mutation probability in a way that minimizes the expected optimization time of the $(1 + 1) EA$ when faced with a solution size of average lengths. Hence, if we know the parameters of the distribution, we can choose the mutation probability such that it is (almost) reciprocal in each position to the expected length of the solution

(which we stated in Equation (1) to be proportional to q^{-1}). Thus, in the setting of [6], i.e., for the truncated geometric distribution with parameters N and q , we set $p_i := q/2$ for all $i \in \mathbb{N}$. Our approach naturally also works for the (non-truncated) geometric distribution $\text{Geo}(q)$, which also has a light upper tail.

We remark without proof that similar results hold for distributions that are strongly concentrated around the mean, e.g., binomial distributions, and also strongly concentrated unbounded distributions, such as Poisson distributions.

Theorem 7 *For $N \in \mathbb{N}$ let $1/N \leq q = q(N) < 1/2$. Let $\vec{p} := (q/2)_{i \in \mathbb{N}}$. For $D = \text{Geo}(q)$ and $D = \text{TruncGeo}(N, q)$, the expected run time of the $(1 + 1)$ $\text{EA}_{\vec{p}}$ on ONEMAX_D is $\Theta(q^{-1} \log q^{-1})$.*

We note that the lower bound is immediate. With constant probability, the random instance has length $\Omega(q^{-1})$. Since our $(1 + 1)$ $\text{EA}_{\vec{p}}$ flips bits independently with probability $q/2$, the classic coupon collector type argument (see Lemma 10 in [20]) gives that with probability $\Omega(1)$ after $cq^{-1} \ln(q^{-1})$ iterations, where c is a suitable constant, there is a bit position which was initially zero and which was not flipped in any mutation operation. Hence in this case, which shows up with constant probability, the run time is more than $cq^{-1} \ln(q^{-1})$. This immediately shows that the expected run time is $\Omega(q^{-1} \ln(q^{-1}))$.

For the proof of the upper bound we will use the following upper bound from [29] for the expected run time of the $(1 + 1)$ EA on ONEMAX . A similar upper bound can be found in [31, Theorem 4.1].

Lemma 8 ([29, Theorem 8]) *For a fixed length n and a uniform mutation vector $\vec{p} = (p)_{i \in \mathbb{N}}$ with $0 < p < 1$, the expected run time of the $(1 + 1)$ $\text{EA}_{\vec{p}}$ on ONEMAX_n is at most $(\ln(n) + 1)/(p(1 - p)^n)$.*

Proof (of Theorem 7) We first consider $D = \text{TruncGeo}(N, q)$. We do not worry about constant factors in this analysis and thus bound some expressions generously.

Using Lemma 8 we can bound the expected run time of the $(1 + 1)$ $\text{EA}_{\vec{p}}$ on ONEMAX_D from above by

$$\sum_{n=1}^{N-1} \frac{q(1 - q)^{n-1}(\ln(n) + 1)}{q/2(1 - q/2)^n} + \frac{(1 - q)^{N-1}(\ln(N) + 1)}{q/2(1 - q/2)^N}. \tag{2}$$

To bound the last summand in this expression, we first observe that (using the binomial theorem), for all positive n ,

$$(1 - \frac{q}{2})^n = \left(1 - q + \frac{q^2}{4}\right)^{n/2} > (1 - q)^{n/2}. \tag{3}$$

This shows that the last summand in (2) is at most

$$2(1 - q)^{N/2-1}(\ln(N) + 1)/q,$$

which is $O(q^{-1} \log q^{-1})$. This can be seen as follows. For $q \geq 2 \ln \ln(N)/N$ it holds (using the inequality $1 - q \leq \exp(-q)$) that $(1 - q)^{N/2-1} \leq \exp(-qN/2) \leq 1/\ln(N)$

and thus $2(1 - q)^{N/2-1}(\ln(N) + 1)/q = O(1/q)$, while for $1/N \leq q \leq 2 \ln \ln(N)/N$ we have (for some suitably chosen constant C)

$$\begin{aligned} (1 - q)^{N/2} \ln(N) &\leq (1 - 1/N)^{N/2} \ln(N) \leq C(\ln(N) - \ln(2 \ln \ln N)) \\ &= C \ln(N/(2 \ln \ln N)) \leq C \ln(1/q). \end{aligned}$$

Using again (3) we bound the first part of the sum (2) by

$$\begin{aligned} \frac{2}{1 - q} \sum_{n=1}^{N-1} \frac{(1 - q)^n (\ln(n) + 1)}{(1 - q/2)^n} &\leq \frac{2}{1 - q} \sum_{n=1}^{N-1} (\ln(n) + 1)(1 - q)^{n/2} \\ &= 2 \sum_{n=1}^{N-1} (\ln(n) + 1)(1 - q)^{n/2-1}. \end{aligned}$$

To show that this expression is $O(q^{-1} \log q^{-1})$ we split the sum into blocks of length $k := \lceil 1/q \rceil$ and use again the inequality $1 - q \leq \exp(-q)$. This shows that the last expression is at most

$$\begin{aligned} &2 \sum_{j=0}^{\lceil N/k \rceil - 1} \sum_{\ell=1}^k \exp\left(-q \left(\frac{jk + \ell}{2} - 1\right)\right) (\ln(jk + \ell) + 1) \\ &\leq 2k \sum_{j=0}^{\lceil N/k \rceil - 1} \exp\left(-\frac{1}{k} \left(\frac{jk}{2} - 1\right)\right) (\ln(j + 1) + \ln(k) + 1) \\ &= O(k \ln k), \end{aligned}$$

where the last equality can be best seen by first considering that $\sum_{j=0}^{\lceil N/k \rceil - 1} \exp(-\frac{1}{k} (\frac{jk}{2} - 1))(\ln(k) + 1) = \Theta(\log k)$, while $\sum_{j=0}^{\lceil N/k \rceil - 1} \exp(-\frac{1}{k} (\frac{jk}{2} - 1))(\ln(j + 1)) = O(1)$. Summarizing the computations above we see that (2) is of order at most $q^{-1} \log q^{-1}$.

For $D = \text{Geo}(q)$ the computations are almost identical. By Lemma 8 and (3) the expected run time of the $(1 + 1) \text{EA}_{\vec{p}}$ on ONEMAX_D is at most

$$\begin{aligned} &\frac{2}{1 - q} \sum_{n=1}^{\infty} \frac{(1 - q)^n (\ln(n) + 1)}{(1 - q/2)^n} \\ &\leq 2 \sum_{n=1}^{\infty} (1 - q)^{n/2-1} (\ln(n) + 1) = O(q^{-1} \log q^{-1}), \end{aligned}$$

which can be seen in a similar way as above by splitting the sum into blocks of size $k := \lceil 1/q \rceil$ and using $1 - q \leq \exp(-q)$. □

It is interesting to note that the expected run time increases to between $\Omega(N)$ and $O(N \log N)$ when the mutation probability is chosen to be $\vec{p} = (q)_{i \in \mathbb{N}}$. This can

easily be seen as follows. For the upper bound we use Lemma 8 (ignoring the “+1” terms which are easily seen to play an insignificant role) to obtain that the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ with $\vec{p} = (q)_{i \in \mathbb{N}}$ on ONEMAX $_{\text{TruncGeo}(N, q)}$ is at most

$$\begin{aligned} & \sum_{n=1}^{N-1} \frac{q(1-q)^{n-1} \ln n}{q(1-q)^n} + \frac{(1-q)^{N-1} \ln N}{q(1-q)^N} \\ &= \sum_{n=1}^{N-1} \frac{\ln n}{1-q} + O(\log(N)/q) \\ &= \frac{\ln((N-1)!)}{1-q} + O(N \log N) \\ &= O(N \log N). \end{aligned}$$

We can derive a strong lower bound of $\Omega(N \log N)$ in the case of $2^{-N/3} \leq q \leq 1/N$ from the following one for static solution lengths. See also [31, Sect. 6] for very strong lower bounds.

Lemma 9 ([29, Theorem 9]) *For a fixed length n and a uniform mutation vector $\vec{p} = (p)_{i \in \mathbb{N}}$, the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ on ONEMAX $_n$ is at least $(\ln(n) - \ln \ln n - 3)/(p(1-p)^n)$ for $2^{-n/3} \leq p \leq 1/n$ and at least $(\ln(1/(p^2n)) - \ln \ln n - 3)/(p(1-p)^n)$ for $1/n \leq p \leq 1/(\sqrt{n} \log n)$.*

Thus, the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ with $\vec{p} = (q)_{i \in \mathbb{N}}$ and $2^{-N/3} \leq q \leq 1/N$ on ONEMAX $_{\text{TruncGeo}(N, q)}$ is at least $\sum_{n=1}^{N-1} q(1-q)^{n-1} \frac{(\ln(n) - \ln \ln n - 3)}{q(1-q)^n} \geq \sum_{n=1}^{N-1} \frac{1}{2} \frac{\ln n}{1-q} = \frac{1}{2} \frac{\ln((N-1)!)}{1-q} = \Omega(N \log N)$. Similarly we can get a lower bound of $\Omega(N)$ in case of $1/N \leq q \leq 1/(\sqrt{N} \log N)$ by using the lower bound of $1/(q(1-q)^n)$ for any fixed solution length n .

We now turn our attention to the LEADINGONES problems, where a similar approach as above yields the following result.

Theorem 10 *Let $N \in \mathbb{N}$ and $1/N \leq q \leq 1/2$. Let $\vec{p} := (q/2)_{i \in \mathbb{N}}$. For $D = \text{TruncGeo}(N, q)$ and $D = \text{Geo}(q)$, the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ on LEADINGONES $_D$ is $\Theta(q^{-2})$.*

We will derive this result from the following lemma, which was independently proven in [5, Theorem 3], [29, Corollary 2], and in a slightly weaker form in [26, Theorem 1.2].

Lemma 11 ([5], [29], and [26]) *For a fixed length n and a mutation vector $\vec{p} = (p)_{i \in \mathbb{N}}$ with $0 < p < 1/2$, the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ on LEADINGONES $_n$ is exactly $(1/(2p^2))((1-p)^{-n+1} - (1-p))$.*

Proof (of Theorem 10) We first consider the case that the solution length is sampled from the truncated geometric distribution $\text{TruncGeo}(N, q)$. By Lemma 11 the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ on LEADINGONES $_D$ is

$$\sum_{n=1}^{N-1} q(1-q)^{n-1} \frac{2}{q^2} \left((1-q/2)^{-n+1} - (1-q/2) \right) + A, \tag{4}$$

where A is the summand that accounts for the event that the solution length is N , i.e.,

$$A = (1-q)^{N-1} \frac{2}{q^2} \left((1-q/2)^{-N+1} - (1-q/2) \right) = O\left(q^{-2}\right).$$

To bound expression (4), we use inequality (3), which for $n = 1$ provides $(1-q)^{1/2} < 1 - q/2$. Using this inequality (in the first and the last step in the computations below), and ignoring the “ $-(1 - q/2)$ ” term in (4), we can bound the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ on LEADINGONES $_D$ from above by

$$\begin{aligned} \frac{2}{q} \sum_{n=1}^{N-1} \left(\frac{(1-q)^{n-1}}{(1-q/2)^{n-1}} \right) + O\left(q^{-2}\right) &\leq \frac{2}{q} \sum_{n=0}^{\infty} (1-q)^{n/2} + O\left(q^{-2}\right) \\ &= \frac{2}{q} \frac{1}{1 - (1-q)^{1/2}} + O\left(q^{-2}\right) = O\left(q^{-2}\right). \end{aligned}$$

Similar computations, again using that $(1-q)^{1/2} < 1 - q/2$ show that for $D = \text{Geo}(q)$ the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ on LEADINGONES $_D$ is bounded from above by

$$\frac{2}{q} \sum_{n=1}^{\infty} \frac{(1-q)^{n-1}}{(1-q/2)^{n-1}} \leq \frac{2}{q} \sum_{n=0}^{\infty} (1-q)^{n/2} = \frac{2}{q} \frac{1}{1 - (1-q)^{1/2}} \leq \frac{4}{q^2}.$$

□

Just as for ONEMAX $_D$ (with $D = \text{TruncGeo}(N, q)$) we see that also on LEADINGONES $_D$ the expected run time increases (in this case to $\Theta(N/q)$) when the mutation probability is chosen to be $\vec{p} = (q)_{i \in \mathbb{N}}$. By Lemma 11 this run time equals

$$\begin{aligned} &\sum_{n=1}^{N-1} q(1-q)^{n-1} \frac{1}{2q^2} \left((1-q)^{-n+1} - (1-q) \right) + A \\ &= \frac{1}{2q} \sum_{n=1}^{N-1} (1 - (1-q)^n) + A \\ &= \frac{1}{2q} \left(N - 1 - \frac{1 - (1-q)^N}{q} + 1 \right) + A \\ &= \Theta(N/q) + A, \end{aligned}$$

where A is the summand that accounts for the event that the solution length is N , i.e.,

$$A = (1-q)^{N-1} \frac{1}{2q^2} \left((1-q)^{-N+1} - (1-q) \right) = \Theta\left(q^{-2}\right).$$

4 Initial Segment Uncertainty Model with Arbitrary Solution Length

In the setting described in Sect. 3 it is assumed that the algorithm designer has quite a good knowledge about the solution length. Not only does he know an upper bound of N , but he may also crucially exploit its distribution. Indeed, we make quite heavy use in Theorems 7 and 10 of the fact that the (truncated) geometric distribution has a light upper tail. Since situations may exist in which one cannot rely on so much information, we regard in this section a more general setting in which *no* prior information is given about the possible solution length n . That is, we regard a setting in which the solution length can be an arbitrary positive integer. In this setting neither do we have any upper bounds on n nor any information about its distribution.

As before, our task is to decide upon on a sequence $(p_i)_{i \in \mathbb{N}}$ of mutation probabilities $0 \leq p_i \leq 1$. An adversary may then choose the solution length n and we run the $(1+1)$ EA $_{\vec{p}}$ with $\vec{p} = (p_1, \dots, p_n)$. In practical applications, this can be implemented with a (possibly generous) upper bound on the problem size.

We first show that uniform fixed bit flip probabilities necessarily lead to exponential run times (see Sect. 4.1). We then show two ways out of this problem. In Sect. 4.2 we consider position-dependent bit flip probabilities and in Sect. 5 we show that we can have an efficient algorithm with uniform bit flip probabilities if we choose the bit flip probability randomly in each iteration.

4.1 Non-suitability of Uniform Bit Flip Probabilities

It seems quite intuitive that, if nothing is known about the solution length, there is not much we can achieve with uniform bit flip probabilities. In fact, for any fixed mutation probability $p \in [0, 1]$, we just need to choose a large enough solution length n to see that the $(1+1)$ EA $_{\vec{p}}$ with uniform mutation probability p is very inefficient.

More precisely, using the following statement (which is a simplified version of [31, Theorem 6.5]), we get the lower bound regarding optimizing ONEMAX with uniform bit flip probabilities stated in Theorem 13.

Theorem 12 (from [31]) *Let $0 < \varepsilon < 1$ be a constant. Let $p = O(n^{-2/3-\varepsilon})$ and $\vec{p} := (p)_{i \in \mathbb{N}}$. On any linear function, the expected optimization time of the $(1+1)$ EA $_{\vec{p}}$ is bounded from below by*

$$(1 - o(1)) \frac{1}{p(1-p)^n} \min \left\{ \ln(n), \ln \left(\frac{1}{p^3 n^2} \right) \right\}.$$

Theorem 13 *Let $p \in [0, 1]$ be a constant and $\vec{p} := (p)_{i \in \mathbb{N}}$. Then there exists a positive integer $n_0 \in \mathbb{N}$ such that, for all $n \geq n_0$, the expected run time of the $(1+1)$ EA $_{\vec{p}}$ on ONEMAX $_n$ is $2^{2(n)}$.*

It is quite intuitive that for large p the expected optimization time of the $(1+1)$ EA $_{\vec{p}}$ with $\vec{p} = (p)_{i \in \mathbb{N}}$ is very large also for small problem sizes, as in this case typically too many bits are flipped in each iteration. This has been made precise by Witt, who showed that for p, n with $p = \Omega(n^{\varepsilon-1})$, the expected run time of the $(1+1)$ EA $_{\vec{p}}$ is $2^{\Omega(n^\varepsilon)}$ with probability at least $1 - 2^{-\Omega(n^\varepsilon)}$ [31, Theorem 6.3].

For LEADINGONES we get a similar lower bound from Lemma 11.

Theorem 14 *Let $p \in (0, 1/2)$ be a constant and $\vec{p} := (p)_{i \in \mathbb{N}}$. Then the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ on LEADINGONES $_n$ is $2^{\Omega(n)}$.*

Proof From Lemma 11 we have that the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ is, for n large enough,

$$\frac{1}{2p^2} \left((1 - p)^{-n+1} - (1 - p) \right) \geq \frac{1}{2} (e^{pn-p} - 1) = 2^{\Omega(n)}.$$

□

4.2 Initial Segment Uncertainty Model with Position-Dependent Bit Flip Probabilities

As discussed, one way to achieve efficient optimization with unknown solution length is by using position-dependent mutation rates, that is, different bit positions have different probabilities associated for being flipped during a mutation operation. We shall make use of *summable sequences* \vec{p} to define these bit-flip probabilities. For the convenience of the reader, some basic facts and definitions of summable sequences are summarized in Section A in the “Appendix”.

The advantage of using summable sequences is that the probability of flipping only one single bit is *always* constant, regardless of the total number of bits considered, i.e., regardless of the problem length n . This is in contrast to the sequence $(1/(i + 1))_{i \in \mathbb{N}}$ considered in [6], which is not summable, and which has a chance of $(1/2) \prod_{i=2}^n (1 - 1/(i + 1)) = 1/n$ of flipping only the first bit and a chance of $(1/n) \prod_{i=1}^{n-1} (1 - 1/(i + 1)) = 1/n^2$ of flipping only the n th bit. For this reason the $(1 + 1)$ EA $_i$ is very inefficient for the setting in which the solution length can be arbitrary.

Using the following two theorems for ONEMAX and LEADINGONES, respectively, we will be able to show in Corollary 17 that not knowing the solution length n does not harm the expected run time more than by a factor of order $\log(n) \log^{(2)}(n) \dots \log^{(s-1)}(n) (\log^{(s)}(n))^{1+\varepsilon}$ with respect to the best known (and, in case of LEADINGONES according to Lemma 2 optimal) bound when the problem length is known a priori.

We start with the theorem regarding ONEMAX.

Theorem 15 *Let $\vec{p} = (p_i)_{i \in \mathbb{N}}$ be a monotonically decreasing summable sequence with $\Sigma := \sum_{i=1}^{\infty} p_i < 1$. Then the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ on ONEMAX $_n$ is at most $\log n / (p_n(1 - \Sigma)) = O(\log n / p_n)$.*

Proof We make use of the multiplicative drift theorem [17, Theorem 3] and show that for every n and every search point x with $n - k$ ones, the probability to create in one iteration of the $(1 + 1)$ EA $_{\vec{p}}$ with \vec{p} a search point y with $\text{ONEMAX}_n(y) > \text{ONEMAX}_n(x)$ is at least of order k/p_n . This can in fact be seen quite easily by observing that the probability to increase the ONEMAX-value of x by exactly one is at least

$$\begin{aligned}
 kp_n \prod_{j=1}^n (1 - p_j) &\geq kp_n \left(1 - \sum_{j=1}^n p_j \right) \geq kp_n \left(1 - \sum_{j=1}^{\infty} p_j \right) \\
 &= kp_n(1 - \Sigma),
 \end{aligned}$$

where the first (elementary) estimate is known as Weierstrass product inequality, see, e.g., [9]. From this an upper bound of $\log n/(p_n(1 - \Sigma))$ for the run time of the $(1 + 1) EA_{\vec{p}}$ follows immediately from the multiplicative drift theorem. \square

Next we consider LEADINGONES. The proof follows along similar lines as the one for ONEMAX and uses a fitness level argument instead of multiplicative drift (using additive drift would also be possible). The following statement will be generalized in Lemma 27 but suffices for the considerations made in this section.

Theorem 16 *Let $\vec{p} = (p_i)_{i \in \mathbb{N}}$ be a monotonically decreasing summable sequence with $\Sigma := \sum_{i=1}^{\infty} p_i < 1$. The expected run time of the $(1 + 1) EA_{\vec{p}}$ on $LEADINGONES_n$ is at most $n/(p_n(1 - \Sigma)) = O(n/p_n)$.*

Proof Let $n, k \in \mathbb{N}$ with $k < n$ and let $x \in \{0, 1\}^n$ with $LO(x) = k - 1$. The probability to get in one iteration of the $(1 + 1) EA_{\vec{p}}$ a search point y with $LO(y) > LO(x)$ is at least

$$p_k \prod_{j=1}^{k-1} (1 - p_j) \geq p_k \left(1 - \sum_{j=1}^{k-1} p_j \right) \geq p_k(1 - \Sigma) \geq p_n(1 - \Sigma).$$

By a simple fitness level argument (see, e.g., the work by Sudholt [29] for background and examples of this method), the expected run time of the $(1 + 1) EA_{\vec{p}}$ on $LEADINGONES_n$ is thus at most $n/(p_n(1 - \Sigma))$. \square

It is well known that, for every constant $\varepsilon > 0$, the sequence $(1/(i(\log i)^{1+\varepsilon}))_{i \in \mathbb{N}}$ is summable (this can be proven via Cauchy’s condensation test). It is also monotonically decreasing in i . Theorems 15 and 16 therefore imply that the expected run time of the $(1 + 1) EA_{\vec{p}}$ on $ONEMAX_n$ is $O(n(\log n)^{2+\varepsilon})$ and $O(n^2(\log n)^{1+\varepsilon})$ for $LEADINGONES_n$ when $\vec{p} = (p_i)_{i \in \mathbb{N}} := (1/(2Si(\log i)^{1+\varepsilon}))_{i \in \mathbb{N}}$ with $S := \sum_{i=1}^{\infty} 1/(i(\log i)^{1+\varepsilon})$.

This bound can be improved by regarding the following sequences of iterated logarithms. For every $b > 1, \varepsilon \geq 0, r > 0$, and all positive integers s , set

$$p_b^{s,\varepsilon}(r) := 1/\left(r(\log_b^{(s)}(r))^{1+\varepsilon} \prod_{j=1}^{s-1} \log_b^{(j)}(r) \right), \tag{5}$$

where, for each $b > 1, j \in \mathbb{N}_{\geq 2}$, and $r > 0$, we set

$$\log_b^{(j)}(r) := \begin{cases} \log_b(\log_b^{(j-1)}(r)), & \text{if } \log^{(j-1)}(r) \geq b; \\ 1, & \text{otherwise;} \end{cases} \tag{6}$$

with $\log_b^{(1)}(r) := \log_b(r)$ if $r \geq b$ and $\log_b^{(1)}(r) := 1$ otherwise. When, in the following, the subscript b indicating the base of the logarithm is omitted, it can be assumed to be equal to two.

It is furthermore well known that, for every $\varepsilon > 0$ and every $s \geq 1$, the sequence $(p_2^{s,\varepsilon}(n))_{n \in \mathbb{N}}$ is summable, cf. [21, page 48]. Together with Theorems 15 and 16, this gives the following bounds.

Corollary 17 *Let $s \in \mathbb{N}$ and $\varepsilon > 0$. Set $\vec{p} := (p_i^{s,\varepsilon})_{i \in \mathbb{N}}$. Then the expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ is*

- $O(n(\log(n))^2 \log^{(2)}(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$ on ONEMAX $_n$, and
- $O(n^2 \log(n) \log^{(2)}(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$ for LEADINGONES $_n$.

A natural question arising from the construction in (5) is the summability of those sequences in which the iterated logarithms in the denominator are not interrupted; that is, of the sequences $(p_b^\infty(n))_{n \in \mathbb{N}}$ with elements

$$p_b^\infty(n) := 1 / \left(n \prod_{j=1}^\infty \log_b^{(j)}(n) \right). \tag{7}$$

It is known in the mathematics literature (cf. also Theorem 43 in the ‘‘Appendix’’) that these sequences $(p_b^\infty(n))_{n \in \mathbb{N}}$ are summable if and only if the base b of the logarithm satisfies $b < e := \exp(1)$. This yields the following bounds.

Corollary 18 *Let $b < e$ and $\vec{p} := (p_b^\infty(n))_{n \in \mathbb{N}}$. The expected run time of the $(1 + 1)$ EA $_{\vec{p}}$ is*

- $O(n(\log_b(n))^2 \log_b^{(2)}(n) \dots)$ on ONEMAX $_n$, and
- $O(n^2 \log_b(n) \log_b^{(2)}(n) \dots)$ on LEADINGONES $_n$.

5 Unrestricted Uncertainty Model

In the conclusions of [6] the authors ask the following: how can we optimize efficiently when an upper bound N on the problem length is known, but only n bits at *unknown positions* are relevant for the fitness? It is not difficult to see that our previous solutions with position-dependent bit flip probabilities will not be able to assign appropriate bit flip probabilities to the *relevant* bit positions. However, any uniform choice of bit flip probabilities will effectively ignore irrelevant bit positions. In this section we therefore consider the $(1 + 1)$ EA $_Q$, described as Algorithm 3 in Sect. 2.3, a variation of the $(1 + 1)$ EA where the bit flip probability p is chosen randomly from a distribution Q on $(0, 1)$ in each iteration (the distribution Q does not change over time). This mutation probability is then applied independently to each bit, i.e., each bit of the current best solution is independently flipped with probability p .

To make the problem more explicit, we are asked to find a distribution Q on $[0, 1]$ such that the $(1 + 1)$ EA $_Q$ efficiently optimizes for any $n \in \mathbb{N}$ and any pairwise different $b_1, \dots, b_n \in \mathbb{N}$ the functions

$$\text{ONEMAX}_{b_1, \dots, b_n}(x) := \sum_{i=1}^n x_{b_i}, \text{ respectively}$$

$$\text{LEADINGONES}_{b_1, \dots, b_n}(x) := \max\{i \in [0..n] \mid \forall j \leq i : x_{b_j} = 1\}.$$

In Theorems 19 and 20 we show that such a distribution Q exists. That is, there is a distribution Q such that the corresponding $(1 + 1)$ EA $_Q$ efficiently optimizes any ONEMAX $_{b_1, \dots, b_n}$ and any LEADINGONES $_{b_1, \dots, b_n}$ function, regardless of the number of relevant bits and regardless of their positions.

We start with our main result regarding ONEMAX.

Theorem 19 *Let $(p_i)_{i \in \mathbb{N}} \in (0, 1)^{\mathbb{N}}$ be a monotonically decreasing summable sequence. Set $\Sigma := \sum_{j=1}^{\infty} p_j$. Let Q be the distribution which assigns the mutation probability $1/i$ a probability of p_i/Σ .*

For any pairwise different positive integers b_1, \dots, b_n the expected run time of the $(1 + 1)$ EA $_Q$ on ONEMAX $_{b_1, \dots, b_n}$ is $O(\log(n)/p_{2n})$.

Proof Let $n \in \mathbb{N}$ and b_1, \dots, b_n be pairwise different positive integers.

The probability to sample a mutation probability between $1/(2n)$ and $1/n$ is

$$\sum_{j=n}^{2n} p_j \geq np_{2n}.$$

We disregard all iterations in which we do not sample a mutation probability between $1/(2n)$ and n (they can only be beneficial). Thus, on average, we consider at least one iteration out of $1/(np_{2n})$.

Assuming that x is a search point with $n - \ell$ ones (in the relevant positions) and that the sampled bit flip probability p satisfies $1/(2n) \leq p \leq 1/n$, the probability of gaining one of the remaining ℓ ones in a relevant position and not flipping any other relevant bit is

$$\ell p(1 - p)^{n-1} \geq \ell/(2n)(1 - 1/n)^{n-1} \geq \ell/(2en).$$

Thus, we have an expected progress in each iteration of at least

$$\frac{\ell}{2en} np_{2n} = O(\ell p_{2n}).$$

Therefore, by the multiplicative drift theorem [17, Theorem 3], we need in expectation $O(\log(n)/p_{2n})$ iterations to optimize function ONEMAX $_{b_1, \dots, b_n}$. \square

For LEADINGONES we obtain the following.

Theorem 20 *Let $(p_i)_{i \in \mathbb{N}}$ and Q as in Theorem 19. For any pairwise different $b_1, \dots, b_n \in \mathbb{N}$ the expected run time of the $(1 + 1) EA_Q$ on $LEADINGONES_{b_1, \dots, b_n}$ is $O(n/p_{2n})$.*

Proof Let $n \in \mathbb{N}$ and b_1, \dots, b_n be pairwise different positive integers.

This proof follows along similar lines as the one for ONEMAX. We have again that the probability to have a bit flip probability between $1/(2n)$ and $1/n$ in an iteration is at least np_{2n} .

Let x be a search point with $LEADINGONES_{b_1, \dots, b_n}(x) = \ell$. Given a mutation probability p between $1/(2n)$ and $1/n$, the probability to create in one iteration of the $(1 + 1) EA_Q$ a search point y of fitness greater than ℓ is at least

$$p(1 - p)^{\ell-1} \geq (1/(2n))(1 - 1/n)^{n-1} \geq 1/(2en).$$

Thus, we have an expected progress in each iteration of at least

$$\frac{1}{2en} np_{2n} = O(p_{2n}).$$

Therefore, by the fitness level method (see again [29] for a discussion of this method), we need in expectation $O(n/p_{2n})$ iterations to optimize $LEADINGONES_{b_1, \dots, b_n}$. \square

Similarly to Corollaries 17 and 18 we obtain the following bounds.

Corollary 21 *Let $n \in \mathbb{N}$ and b_1, \dots, b_n be pairwise different positive integers.*

For $s \in \mathbb{N}$, $\varepsilon > 0$, $\vec{p} := (p_i^{s, \varepsilon})_{i \in \mathbb{N}}$, and $Q = Q(\vec{p})$ the expected run time of the $(1 + 1) EA_Q$ is

- $O(n(\log(n))^2 \log^{(2)}(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$ for $ONEMAX_{b_1, \dots, b_n}$, and
- $O(n^2 \log(n) \log^{(2)}(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$ for $LEADINGONES_{b_1, \dots, b_n}$.

For $b < e$, $\vec{p} = (p_b^\infty(n))_{n \in \mathbb{N}}$, and $Q = Q(\vec{p})$ the expected run time of the $(1 + 1) EA_Q$ is

- $O(2n(\log_b(2n))^2 \log_b^{(2)}(2n) \dots)$ for $ONEMAX_{b_1, \dots, b_n}$, and
- $O(2n^2 \log_b(2n) \log_b^{(2)}(2n) \dots)$ for $LEADINGONES_{b_1, \dots, b_n}$.

6 Lower Bounds for LeadingOnes in Both Uncertainty Models

In this section, we prove very sharp lower bounds for the optimization of the $LEADINGONES$ function in the two uncertainty models via the two algorithm classes $(1 + 1) EA_{\vec{p}}$ and $(1 + 1) EA_Q$. Our first result and the central step towards proving the lower bounds is proposing a fairly general class of algorithms that extends both previous algorithm classes (Sect. 6.1). We then show that all three lead to the same run time profiles for the initial segment model (Theorem 26). Since the $(1 + 1) EA_Q$ algorithm class gives the same performance on instances defined on any subset of the bit positions, this general result connects the two uncertainty models and shows that none of our algorithms can have a better performance in the weaker initial segment model.

Given that all algorithm classes are equally powerful, we then concentrate in Sect. 6.2 on the performance of the $(1 + 1)$ EA $_{\vec{p}}$ in the initial segment uncertainty model. In Theorem 34 we show that, for all $s \in \mathbb{N}$, $\Omega(n^2 \log(n) \log^{(2)}(n) \dots \log^{(s)}(n))$ is a lower bound for the expected run time of the LEADINGONES instance of length n . By carefully constructing for each sequence \vec{p} a sequence \vec{q} such that the asymptotic performance of $(1 + 1)$ EA $_{\vec{q}}$ is strictly better than the one of $(1 + 1)$ EA $_{\vec{p}}$, we also show that there is no best asymptotic performance on the initial segment LEADINGONES problem in the $(1 + 1)$ EA $_{\vec{p}}$ algorithms class (and likewise in the $(1 + 1)$ EA $_{\mathcal{Q}}$ class), cf. Sect. 6.2.2.

6.1 Four Algorithm Classes with Equal Power in the Initial Segment Uncertainty Model

To conveniently prove lower bounds for the two algorithm classes $(1 + 1)$ EA $_{\vec{p}}$ and $(1 + 1)$ EA $_{\mathcal{Q}}$, we define the following algorithm class which generalizes both. Consequently, a lower bound for the new algorithm class immediately is a lower bound for both previously regarded ones.

Arbitrary Mutation on Infinite Bit Strings: The $(1 + 1)$ EA $_{\mathbb{P}}$. In simple words, we consider all algorithms that can be defined as follows. Let \mathbb{P} be a probability measure on the set $\Omega = \{0, 1\}^{\mathbb{N}}$ of infinite bit strings. The $(1 + 1)$ EA $_{\mathbb{P}}$, when run on the finite subset $I \subset \mathbb{N}$ of bit positions, is the classic $(1 + 1)$ EA except that the mutation operator (informally speaking) samples a random element X from Ω and then flips exactly those bits $i \in I$ for which $X_i = 1$.

To make this formally correct, we need a mild excursion into probability theory. For all $n \in \mathbb{N}$ and $a_1, \dots, a_n \in \{0, 1\}$, we call the set

$$C(a_1, \dots, a_n) := \{x \in \Omega \mid \forall i \in [n] : x_i = a_i\}$$

a *cylinder* in Ω . Let \mathcal{A} be the σ -algebra generated by all cylinders. The precise structure of \mathcal{A} is not overly important in the remainder except that it contains the following sets $C(y)$.

For a finite set $I \subset \mathbb{N}$ denote by $\{0, 1\}^I$ the set of all finite binary sequences with indices in I , formally, all $y : I \rightarrow \{0, 1\}$. For $y \in \{0, 1\}^I$, let

$$C(y) := \{x \in \Omega \mid \forall i \in I : x_i = y_i\}.$$

Then \mathcal{A} contains all such sets $C(y)$.

Formally speaking, $\{C(y) \mid y \in \{0, 1\}^I\}$ generates a sub- σ -algebra of \mathcal{A} , which is isomorphic to $\{0, 1\}^I$. Consequently, any probability measure \mathbb{P} on the measurable space (Ω, \mathcal{A}) gives rise to a probability measure \mathbb{P}_I on $\{0, 1\}^I$ (with the power set as σ -algebra) defined by

$$\mathbb{P}_I(y) := \mathbb{P}[C(y)]$$

for all $y \in \{0, 1\}^I$.

Building on these considerations, we can define the mutation operator of the $(1 + 1) \text{EA}_{\mathbb{P}}$ when running on a finite set $I \subset \mathbb{N}$ of bit positions as follows. The offspring stemming from a parent $x \in \{0, 1\}^I$ is $x \oplus y$, where $y \in \{0, 1\}^I$ is chosen randomly from the distribution \mathbb{P}_I . Here \oplus denotes the bit-wise exclusive-or. In other words, we flip those bits i of x for which y_i is one.

Next we argue that the $(1 + 1) \text{EA}_{\vec{p}}$ and $(1 + 1) \text{EA}_Q$ classes are subclasses of the class of all algorithms $(1 + 1) \text{EA}_{\mathbb{P}}$.

- Let $\vec{p} = (p_i)_{i \in \mathbb{N}}$ be any sequence in $[0, 1]$. Then the $(1 + 1) \text{EA}_{\vec{p}}$, when run on a finite bit set I , performs mutation by flipping each bit $i \in I$ independently with probability p_i . When defining a probability measure \mathbb{P} on Ω by setting

$$\mathbb{P}[C(a_1, \dots, a_n)] = \prod_{i \in [n]; a_i=1} p_i \prod_{i \in [n]; a_i=0} (1 - p_i)$$

for all cylinders, then the $(1 + 1) \text{EA}_{\vec{p}}$ and the $(1 + 1) \text{EA}_{\mathbb{P}}$ are the same algorithm.

- Let Q be a discrete distribution on $[0, 1]$. Then the $(1 + 1) \text{EA}_Q$, when run on a finite bit set I , performs mutation by first sampling a number q from Q and then flipping each bit $i \in I$ independently with probability q . When defining a probability measure \mathbb{P} on Ω by setting

$$\mathbb{P}[C(a_1, \dots, a_n)] = \sum_q Q(q) \prod_{i \in [n]; a_i=1} q \prod_{i \in [n]; a_i=0} (1 - q),$$

then the $(1 + 1) \text{EA}_Q$ and the $(1 + 1) \text{EA}_{\mathbb{P}}$ are the same algorithm.

For this reason, any lower bound for the best performance achievable with an algorithm from the class $(1 + 1) \text{EA}_{\mathbb{P}}$ immediately carries over to the subclasses $(1 + 1) \text{EA}_{\vec{p}}$ and $(1 + 1) \text{EA}_Q$. Now that we regard the same class $(1 + 1) \text{EA}_{\mathbb{P}}$ of algorithms for the two uncertainty models of an unknown initial segment $[n]$ and an unknown set I of relevant bits, it is clear that a lower bound for the first case carries over to the second. Since our lower bound for the first case will essentially match our upper bounds for both models (as they are equal), it suffices in the following to regard the model of an unknown initial segment $I = [n]$ of bits on which the `LEADINGONES` function to be optimized is defined.

The next step towards the solution of our problem is the following surprising result that for any algorithm $(1 + 1) \text{EA}_{\mathbb{P}}$ there is a randomized local search algorithm (flipping a single randomly chosen bit as mutation operation) that has exactly the same performance on all initial segments (Lemma 22).

Generalized Randomized Local Search on Infinite Bit Strings: $\text{RLS}_{\vec{p}}$. Let p_1, p_2, \dots be non-negative numbers with $\sum_{n \in \mathbb{N}} p_n \leq 1$. Then the algorithm $\text{RLS}_{\vec{p}}$ is a special case of the $(1 + 1) \text{EA}_{\mathbb{P}}$ that

- flips exactly the i th bit with probability p_i
- and flips no bit with probability $1 - \sum_{n \in \mathbb{N}} p_n$.

More formally, $\text{RLS}_{\vec{p}}$ equals $(1 + 1) \text{EA}_{\mathbb{P}}$ for the measure \mathbb{P} defined by $\mathbb{P}[C(a_1, \dots, a_n)] :=$

$$\begin{cases} p_i, & \text{if } a_i = 1 \text{ and } a_j = 0 \text{ for } j \in [n] \setminus \{i\}, \\ 0, & \text{if there are } i, j \in [n] \text{ with } i \neq j \text{ and } a_i = 1 = a_j, \\ 1 - \sum_{n \in \mathbb{N}} p_n, & a_j = 0 \text{ for } j \in [n]. \end{cases}$$

Lemma 22 *Let \mathbb{P} be any probability measure on $\Omega = \{0, 1\}^{\mathbb{N}}$. Then there is a sequence of non-negative numbers $\vec{p} = (p_1, p_2, \dots)$ with $\sum_{n \in \mathbb{N}} p_n \leq 1$ such that, for LEADINGONES , the randomized local search algorithm $\text{RLS}_{\vec{p}}$ and $(1 + 1) \text{EA}_{\mathbb{P}}$ when run on any initial segment of bits have exactly the same run time distribution.*

To prove this lemma, we need the auxiliary result that in any run of an $(1 + 1) \text{EA}_{\mathbb{P}}$ algorithm at any time t the random individual $X^{(t)}$ is identically distributed within each fitness level. This elementary observation has been used previously in [5,16,29] to analyze the optimization of fixed length LEADINGONES instances.

Lemma 23 *Consider a run of an $(1 + 1) \text{EA}_{\mathbb{P}}$ algorithm on the LEADINGONES function defined on the initial segment $[n]$. Denote by $X^{(0)}$ the random initial individual and by $X^{(t)}$, $t \in \mathbb{N}$, the random individual forming the one-element population of the $(1 + 1) \text{EA}_{\mathbb{P}}$ after the t -th iteration. Then, for any two search points $x, y \in \{0, 1\}^n$ with $\text{LO}(x) = \text{LO}(y)$, we have $\Pr[X^{(t)} = x] = \Pr[X^{(t)} = y]$.*

Proof We proceed by induction over time. For $t = 0$, the claim is obvious since $X^{(0)}$ is simply a bit string chosen uniformly at random from $\{0, 1\}^n$, so we even have $\Pr[X^{(t)} = x] = \Pr[X^{(t)} = y]$ for all $x, y \in \{0, 1\}^n$.

Assume the claim to hold for some $t \geq 0$. Since the distribution of $X^{(t+1)}$ is a convex combination of the distributions of $X^{(t+1)}$ conditional on the events $\text{LO}(X^{(t)}) = i$, $i \in [0..n]$, it suffices to show the claim conditioning on $\text{LO}(X^{(t)}) = i$ for some $i \in [0..n]$ (by the law of total probability). So let us assume that $X^{(t)} = i$ for some $i \in [0..n]$. Since the claim is trivial for $i = n$, let us assume $i < n$. Then, with probability one, the first i bits of $X^{(t)}$ are one and $X_{i+1}^{(t)} = 0$. For all $j > i + 1$, the bits $X_j^{(t)}$ are independently and uniformly distributed in $\{0, 1\}$ by our induction hypothesis.

Let $Y \in \{0, 1\}^n$ be the random mutation mask used in the $(t + 1)$ -st iteration, that is, $Y_j = 1$ if and only if the mutation step in this iteration flipped the j -th bit. We again condition on each possible outcome $y \in \{0, 1\}^n$ of Y and prove the claim in this case. Hence let $y \in \{0, 1\}^n$ and condition on $Y^{(t)} = y$. We need to show that $X^{(t+1)}$ in this situation is uniformly distributed within each fitness level. We consider three cases.

Case 1: If y is such that $y_j = 1$ for some $j \leq i$, then the mutation offspring has a lower fitness than $X^{(t)}$ and is not accepted. Hence $X^{(t)}$ and $X^{(t+1)}$ are identically distributed and the claim is obvious.

Case 2: If y is such that $y_j = 0$ for all $j \leq i + 1$, then the mutation offspring has fitness equal to $\text{LO}(X^{(t)})$ and is accepted. The distribution of this $X^{(t+1)}$ is identical to the one of $X^{(t)}$, since the first $i + 1$ bits are not changed and the last $n - (i + 1)$ bits are

obtained from an exclusive-or of the last $n - (i + 1)$ bits of $X^{(t)}$, which are uniformly distributed in $\{0, 1\}^{n-(i+1)}$, and a fixed bit string, namely the last $n - (i + 1)$ bits of y . Since such an exclusive-or is a bijection of $\{0, 1\}^{n-(i+1)}$, this part of $X^{(t+1)}$ remains uniformly distributed in $\{0, 1\}^{n-(i+1)}$.

Case 3: We consider finally the case that $y_{i+1} = 1$ and $y_j = 0$ for all $j \leq i$. In this case, the mutation offspring and hence also $X^{(t+1)}$ have a fitness larger than i . With the same argument as in the previous case, we see that the last $n - (i + 1)$ bits remain independently and uniformly distributed. Consequently, when conditioning on the fitness of $X^{(t+1)}$ being some $j \geq i + 1$, the bits with index larger than $j + 1$ remain independently and uniformly distributed. This again shows the claim. \square

Proof (of Lemma 22) For $n \in \mathbb{N}$, let $C_n := C(a_1, \dots, a_n)$ with $a_1 = \dots = a_{n-1} = 0$ and $a_n = 1$. Let $p_n := \mathbb{P}[C_n]$. Since the sets $(C_n)_{n \in \mathbb{N}}$ are disjoint, we have $\sum_{n \in \mathbb{N}} p_n \leq 1$. We show that the $(1 + 1)$ EA $_{\vec{p}}$ and RLS $_{\vec{p}}$ have an identical optimization behavior.

Let $n \in \mathbb{N}$ and consider the optimization of LEADINGONES on the initial segment $[n]$ both via the $(1 + 1)$ EA $_{\vec{p}}$ and RLS $_{\vec{p}}$. By Lemma 23, both algorithms at any time t have the property that the parent individual $X^{(t)}$ is uniformly distributed within each fitness level. Consequently, the distribution of $X^{(t)}$ is fully described by the fitness level densities $(\Pr[\text{LO}(X^{(t)}) = i])_{i \in [0..n]}$. To show that $X^{(t)}$ has the same distribution for the $(1 + 1)$ EA $_{\vec{p}}$ and RLS $_{\vec{p}}$, it therefore suffices to show that both algorithms have the same fitness level densities.

We do so by induction. There is nothing to show for $t = 0$, since the initial individual is chosen uniformly at random in both algorithms. Assume that at some time $t \geq 0$ both algorithms have the same fitness level densities. We show that this is also true at time $t + 1$. Since $\Pr[\text{LO}(X^{(t+1)}) = i] = \sum_{j=0}^n \Pr[\text{LO}(X^{(t+1)}) = i \mid \text{LO}(X^{(t)}) = j] \Pr[\text{LO}(X^{(t)}) = j]$ and since the $\Pr[\text{LO}(X^{(t)}) = j]$ are identical for both algorithms, it suffices to show that both algorithms have identical $\Pr[\text{LO}(X^{(t+1)}) = i \mid \text{LO}(X^{(t)}) = j]$ for all $i, j \in [0..n]$.

Let $j \in [0..n]$. Since there is nothing to show for $j = n$, let us assume $j < n$. For the $(1 + 1)$ EA $_{\vec{p}}$, the probability to leave the j -th fitness level is the probability to flip the $(j + 1)$ -st bit and to not flip bits $1, \dots, j$. By definition of the $(1 + 1)$ EA $_{\vec{p}}$, this is exactly $\mathbb{P}[C_{j+1}]$. The second algorithm RLS $_{\vec{p}}$, obviously, leaves the j -th fitness level if and only if the $(j + 1)$ -st bit is the single bit that is flipped, which by definition happens with probability $p_{j+1} = \mathbb{P}[C_{j+1}]$. For both algorithms, conditional on that the j -th fitness level being left, the bits with index larger than $j + 1$ are independently and uniformly distributed. Consequently, the number of additional fitness levels gained in addition to the $(j + 1)$ -st one is also equally distributed. \square

We note in the following lemma that, for LEADINGONES, the class of algorithms $(1 + 1)$ EA $_{\vec{p}}$ regarded in [11] is equal in power to the class of algorithms RLS $_{\vec{p}}$. Together with Lemma 22, this shows that all three classes of algorithms are equally powerful, and in particular, that the subclass $(1 + 1)$ EA $_{\vec{p}}$ regarded in [11] was of maximal power.

Lemma 24 *For any sequence of non-negative numbers $\vec{p} = (p_1, p_2, \dots)$ with $\sum_{n \in \mathbb{N}} p_n \leq 1$ there is a sequence $\vec{q} = (q_1, q_2, \dots)$ of numbers in $[0, 1]$ such that*

$RLS_{\vec{p}}$ and $(1 + 1) EA_{\vec{q}}$ for each initial segment of bits have the same run time distribution on LEADINGONES.

Proof Let us assume that we have $p_n < 1$ for all $n \in \mathbb{N}$, as otherwise trivially $\vec{q} := \vec{p}$ suffices. Let $q_1 := p_1$ and recursively $q_n := p_n / \prod_{j=1}^{n-1} (1 - q_j)$. Using induction, it is not difficult to see that $q_n = p_n / (1 - \sum_{j < n} p_j)$. Furthermore, since $1 - \sum_{j < n} p_j \leq p_n$, this also shows that $q_n \leq 1$. As discussed earlier, this $(1 + 1) EA_{\vec{q}}$ is a special case of the $(1 + 1) EA_{\mathbb{P}}$ with $\mathbb{P}[C(a_1, \dots, a_n)] = \prod_{i \in [n]; a_i=1} q_i \prod_{i \in [n]; a_i=0} (1 - q_i)$. In particular, for $a_1 = \dots = a_{n-1} = 0$ and $a_n = 1$, we have $\mathbb{P}[C(a_1, \dots, a_n)] = q_n \prod_{i=1}^{n-1} (1 - q_i)$, which by construction equals p_n . Consequently, if we redo the construction of the proof of Lemma 22 with this $(1 + 1) EA_{\mathbb{P}}$, we obtain our original $RLS_{\vec{p}}$ algorithm. Consequently, the $(1 + 1) EA_{\vec{q}}$ just constructed has an identical performance with $RLS_{\vec{p}}$. \square

An advantage of the class $RLS_{\vec{p}}$ is that the expected run times are easy to compute. The following result will be used in Lemma 27 to bound the run time of the $(1 + 1) EA_{\vec{p}}$. The proof is inspired by the analysis of the expected run time of the $(1 + 1) EA$ on LEADINGONES in [5], however, we are not aware of a previous analysis of the run time of RLS on LEADINGONES. We note that, while this work was under review, a general analysis of $(1 + 1)$ -type algorithms on LEADINGONES via stochastic domination was proposed in [8], which implies our result. Since this proof is very different from ours, and to keep our work self-contained, we still present our original proof.

Lemma 25 *Let $\vec{p} = (p_1, p_2, \dots)$ be a sequence of non-negative numbers with $\sum_{n \in \mathbb{N}} p_n \leq 1$. Then the expected run time of $RLS_{\vec{p}}$ optimizing the LEADINGONES function on the initial segment $[n]$ is $\frac{1}{2} \sum_{i=1}^n \frac{1}{p_i}$, where $\frac{1}{0} := \infty$.*

Proof Let $n \in \mathbb{N}$. Denote by E the expected run time of $RLS_{\vec{p}}$ on LEADINGONES in $[n]$. If there is an $i \in [n]$ with $p_i = 0$, then with positive probability $RLS_{\vec{p}}$ never finds the optimum, hence $E = \infty$ as claimed. So let us assume that all p_i are positive.

Denote by E_i the expected run time of $RLS_{\vec{p}}$ when starting with a random search point of fitness i . Then, trivially, $E_n = 0$. For $i < n$, we observe that when starting with a random search point of fitness i we only make progress when the $(i + 1)$ -st bit is flipped. The expected waiting time for this event is $\frac{1}{p_{i+1}}$. If this event happens, since the bits with index higher than $i + 1$ are independently and uniformly distributed, we move up to fitness level $i + \ell$ with probability $p_{i,i+\ell} := 2^{-\ell}$, $\ell = 1, \dots, n - i - 1$, and move up to fitness level n with probability $p_{i,n} := 2^{-(n-i-1)}$. Since in any case the resulting search point is uniformly distributed on its fitness level, see Lemma 23, we have $E_i = \frac{1}{p_{i+1}} + \sum_{j=i+1}^n p_{i,j} E_j = \frac{1}{p_{i+1}} + \sum_{j=i+1}^{n-1} 2^{-(j-i)} E_j$. Regarding the corresponding expression for E_{i+1} , we see that $\sum_{j=i+2}^{n-1} 2^{-(j-i)} E_j = (E_{i+1} - \frac{1}{p_{i+2}}) / 2$. Substituting this suitably into the expression for E_i , we obtain $E_i = \frac{1}{p_{i+1}} + E_{i+1} - \frac{1}{2p_{i+2}}$. Now an elementary induction for $i = n - 1$ down to 0 shows $E_i = \frac{1}{p_{i+1}} + \frac{1}{2} \sum_{j=i+2}^n \frac{1}{p_j}$.

Since the initial individual is chosen uniformly at random from $\{0, 1\}^n$, we have $E = \sum_{i=0}^{n-1} 2^{-i-1} E_i$, where we exploited $E_n = 0$ already. Using our knowledge about the E_i , we obtain $E = \frac{1}{2} \sum_{i=1}^n \frac{1}{p_i}$ as claimed. \square

The above insight allows us to completely describe the possible performances of $(1 + 1) EA_{\mathbb{P}}$, $(1 + 1) EA_{\vec{p}}$, and $RLS_{\vec{p}}$ algorithms on LEADINGONES defined on an initial segment of unknown length of an infinite sequence of bits. To make things precise, we call the function $E : \mathbb{N} \rightarrow \mathbb{R}$ the *initial segment run time profile* of such an algorithm if its expected run time on the LEADINGONES function defined on the bits with indices in $[n]$ equals $E(n)$.

Theorem 26 (characterization of initial segment run time profiles) *Let $E : \mathbb{N} \rightarrow \mathbb{R}$. The following four properties are equivalent.*

- (i) *There is a sequence $q : \mathbb{N} \rightarrow [0, 1]$ with $\sum_{n=1}^{\infty} q_n \leq 1$ such that for all $n \in \mathbb{N}$, we have $E(n) = \frac{1}{2} \sum_{i=1}^n \frac{1}{q_i}$.*
- (ii) *E is the initial segment run time profile of the $(1 + 1) EA_{\mathbb{P}}$ for some distribution \mathbb{P} .*
- (iii) *E is the initial segment run time profile of the $(1 + 1) EA_{\vec{p}}$ for some $\vec{p} : \mathbb{N} \rightarrow [0, 1]$.*
- (iv) *E is the initial segment run time profile of $RLS_{\vec{p}}$ for some $\vec{p} : \mathbb{N} \rightarrow [0, 1]$ with $\sum_{n=1}^{\infty} p_n \leq 1$.*

If these properties are fulfilled, then q in (i) equals \vec{p} in (iv).

6.2 Computing Concrete Lower Bounds for LeadingOnes

In the remainder of this section, we use Theorem 26 to compute concrete run time bounds for the $(1 + 1) EA_{\vec{p}}$. These bounds then immediately yield performance bounds for all the other settings covered by Theorem 26. We note that some (but not all) of the subsequent run time results can alternatively be obtained by directly analyzing the summable sequences described in Theorem 26.(i). Such an approach would, however, not give any additional insights into the sequences \vec{p} underlying the $(1 + 1) EA_{\vec{p}}$ with initial segment run time profile E , thus motivating us to study this algorithm directly.

6.2.1 Initial Segment Run Time Profile of the $(1 + 1) EA_{\vec{p}}$

Lemma 22 can be used to precisely determine the initial segment run time profile of the $(1 + 1) EA_{\vec{p}}$. From this we can easily compute the following upper and lower bounds. This result extends Theorem 16, which assumes that \vec{p} is monotonically decreasing and summable.

Lemma 27 *Let $\vec{p} = (p_n)_{n \in \mathbb{N}}$ be a sequence of positive numbers. Let $S_0 := 0$ and $S_n := \sum_{i=1}^n p_i$ for all $n \in \mathbb{N}$. The expected optimization time $E[T_{\vec{p}}(n)]$ of the $(1 + 1) EA_{\vec{p}}$ on the LEADINGONES instance of length n in the initial segment uncertainty model can be bounded by*

$$\frac{1}{2} \sum_{i=1}^n \frac{\exp(S_{i-1})}{p_i} \leq E[T_{\vec{p}}(n)] \leq \frac{1}{2} \sum_{i=1}^n \frac{\exp(2S_{i-1})}{p_i}$$

where the upper bound is valid only when $p_n \leq \frac{1}{2}$ for all $n \in \mathbb{N}$.

If \vec{p} is monotonically decreasing, $p_1 \leq 1/2$, and there exists a constant c such that, for all n , $p_n/p_{n/2} \geq c$, then $E[T_{\vec{p}}(n)] = n \exp(\Theta(S_n))/p_n$. For summable sequences satisfying these additional requirements, we thus have $E[T_{\vec{p}}(n)] = \Theta(n/p_n)$.

Proof By Lemma 22 the initial segment run time for the $(1 + 1) EA_{\vec{p}}$ equals that of $RLS_{\vec{q}}$ with $\vec{q} = (p_n \prod_{j < n} (1 - p_j))_{n \in \mathbb{N}}$. To this sequence we apply Lemma 25 to see that the expected run time of the $RLS_{\vec{q}}$ and thus of the $(1 + 1) EA_{\vec{p}}$ equals

$$E[T_{\vec{p}}(n)] = \frac{1}{2} \sum_{i=1}^n \left(\left(\prod_{j < i} (1 - p_j) \right) p_i \right)^{-1}. \tag{8}$$

We use the inequalities $e^{-2x} \leq (1 - x) \leq e^{-x}$, valid for all $x \leq 1/2$, to get the following estimates.

To bound the expected run time from below, we observe that

$$\begin{aligned} E[T_{\vec{p}}(n)] &= \frac{1}{2} \sum_{i=1}^n \left(\left(\prod_{j < i} (1 - p_j) \right) p_i \right)^{-1} \\ &\geq \frac{1}{2} \sum_{i=1}^n \left(\left(\prod_{j < i} \exp(-p_j) \right) p_i \right)^{-1} \\ &= \frac{1}{2} \sum_{i=1}^n \left(\left(\exp \left(- \sum_{j < i} p_j \right) \right) p_i \right)^{-1} \\ &= \frac{1}{2} \sum_{i=1}^n \frac{\exp(S_{i-1})}{p_i}. \end{aligned} \tag{9}$$

If \vec{p} is monotonically decreasing then $S_n \leq 2S_{n/2}$, showing that

$$\begin{aligned} E[T_{\vec{p}}(n)] &\geq \frac{1}{2} \sum_{i=n/2+1}^n \frac{\exp(S_{n/2})}{p_{n/2}} \\ &\geq \frac{1}{2} \frac{n \exp(S_{n/2})}{p_{n/2}}. \end{aligned} \tag{10}$$

If, in addition, for some constant $c > 0$ we have $p_n/p_{n/2} \geq c$, then it can be easily seen from (10) that there is a constant d such that

$$E[T_{\vec{p}}(n)] \geq n \frac{\exp(dS_n)}{p_n}.$$

We now turn to the upper bound. We have

$$\begin{aligned}
 E[T_{\vec{p}}(n)] &= \frac{1}{2} \sum_{i=1}^n \left(\left(\prod_{j<i} (1 - p_j) \right) p_i \right)^{-1} \\
 &\leq \frac{1}{2} \sum_{i=1}^n \left(\left(\prod_{j<i} \exp(-2p_j) \right) p_i \right)^{-1} \\
 &= \frac{1}{2} \sum_{i=1}^n \left(\left(\exp \left(- \sum_{j<i} 2p_j \right) \right) p_i \right)^{-1} \\
 &= \frac{1}{2} \sum_{i=1}^n \frac{\exp(2S_{i-1})}{p_i}. \tag{11}
 \end{aligned}$$

For all positive sequences \vec{p} the series $(S_n)_{n \in \mathbb{N}}$ is monotonically increasing. If \vec{p} is monotonically decreasing, then (11) shows that

$$E[T_{\vec{p}}(n)] \leq \frac{n \exp(2S_n)}{2 p_n}.$$

□

As a side remark, we mention as immediate consequence of Lemma 27 the following bounds on the range in which those sequences leading to an optimal performance of the $(1+1) EA_{\vec{p}}$ on LEADINGONES are placed. Informally, they are somewhere between $(1/n^2)_{n \in \mathbb{N}}$ and $(1/n)_{n \in \mathbb{N}}$.

Corollary 28 *Let \vec{p} be a sequence of positive terms such that, for all $n \in \mathbb{N}$, $p_n \leq 1/n^2$. Then the expected run time of $(1+1) EA_{\vec{p}}$ on the LEADINGONES instance of length n in the initial segment uncertainty model is $\Omega(n^3)$.*

If, for all $n \in \mathbb{N}$, $p_n \geq 2/n$, this expected run time is also $\Omega(n^3)$.

Proof Suppose first that, for all $n \in \mathbb{N}$, $p_n \leq 1/n^2$. Right from Lemma 27, we compute $E[T_{\vec{p}}(n)] \geq \frac{1}{2} \sum_{i=1}^n \frac{1}{i^2} = \Omega(n^3)$.

Suppose now, for all $n \in \mathbb{N}$, $p_n \geq 2/n$. Using Lemma 27 again as well as $\sum_{j=1}^k 1/j \geq \ln(k)$, we compute

$$E[T_{\vec{p}}(n)] \geq \frac{1}{2} \sum_{i=1}^n \exp \left(\sum_{j=1}^{i-1} 2/j \right) \geq \frac{1}{2} \sum_{i=1}^n (i-1)^2 = \Omega(n^3).$$

□

From Lemma 27 we also get, via simple algebraic operations, the following result, which allows us to multiply a summable sequence with constant factors without harming the asymptotic run time profile.

Lemma 29 Let $\vec{p} : \mathbb{N} \rightarrow [0, 1]$ be a summable sequence with $\sum_{i \in \mathbb{N}} p_i =: p, c > 0$, and $\vec{q} = (q_i)_{i \in \mathbb{N}}$ with $q_i := c \cdot p_i$.

For $c \leq 1$ it holds that $E[T_{\vec{q}}(n)] \leq E[T_{\vec{p}}(n)]/c$.

When $c > 1$ and $q_i \in [0, 1/2]$ for all $i \in \mathbb{N}$, then $E[T_{\vec{q}}(n)] \leq (\exp(2cp)/c) E[T_{\vec{p}}(n)]$.

Proof We first regard the case that $c \leq 1$. Note that in this case, for all $j, \frac{1}{1-cp_j} \leq \frac{1}{1-p_j}$. Therefore, by equation (8), for all $n \in \mathbb{N}$,

$$\begin{aligned} E[T_{\vec{q}}(n)] &= \frac{1}{2} \sum_{i=1}^n \frac{1}{q_i} \prod_{j<i} \frac{1}{1-q_j} = \frac{1}{2} \sum_{i=1}^n \frac{1}{cp_i} \prod_{j<i} \frac{1}{1-cp_j} \leq \frac{1}{2} \sum_{i=1}^n \frac{1}{cp_i} \prod_{j<i} \frac{1}{1-p_j} \\ &= \frac{1}{2c} \left(\sum_{i=1}^n \frac{1}{p_i} \prod_{j<i} \frac{1}{1-p_j} \right) = \frac{1}{c} E[T_{\vec{p}}(n)]. \end{aligned}$$

To treat the case $c > 1$ we first recall that $1 + x \leq \exp(x)$. Since we have required that $cp_j \leq 1/2$, we can thus bound

$$\frac{1}{1-cp_j} = 1 + \frac{cp_j}{1-cp_j} \leq \exp\left(\frac{cp_j}{1-cp_j}\right) \leq \exp(2cp_j).$$

Therefore, we obtain for all $n \in \mathbb{N}$ that

$$\begin{aligned} E[T_{\vec{q}}(n)] &= \frac{1}{2} \sum_{i=1}^n \frac{1}{q_i} \prod_{j<i} \frac{1}{1-q_j} = \frac{1}{2} \sum_{i=1}^n \frac{1}{cp_i} \prod_{j<i} \frac{1}{1-cp_j} \\ &\leq \frac{1}{2c} \sum_{i=1}^n \frac{1}{p_i} \prod_{j<i} \exp(2cp_j) < \frac{1}{2c} \sum_{i=1}^n \frac{1}{p_i} \exp\left(2c \sum_{j \in \mathbb{N}} p_j\right) \\ &= \frac{\exp(2cp)}{2c} \sum_{i=1}^n \frac{1}{p_i} \leq \frac{\exp(2cp)}{c} E[T_{\vec{p}}(n)]. \end{aligned}$$

□

6.2.2 No Separation Between Summable and Non-summable Sequence

Given the bounds proven so far, one may be tempted to believe the best performing $(1 + 1) EA_{\vec{p}}$ is based on a summable sequence \vec{p} . Furthermore, one may ask whether between summable and non-summable sequences there is a strict separation in the performance of the $(1 + 1) EA_{\vec{p}}$ on LEADINGONES in the sense that, for example, there exists a bound B such that for every non-summable sequence \vec{q} the expected optimization time of the $(1 + 1) EA_{\vec{q}}$ on LEADINGONES is greater than B while for some summable sequence \vec{p} the expected run time of the $(1 + 1) EA_{\vec{p}}$ is at most B . Regardless of how one makes such a claim precise, the following observations show that it cannot hold. More precisely, we show that for every summable sequence \vec{p}

there exists a non-summable sequence \vec{q} such that the expected performance of the $(1 + 1) EA_{\vec{q}}$ on LEADINGONES is of strictly smaller order than that of the $(1 + 1) EA_{\vec{p}}$. Less surprisingly, the converse also holds.

Theorem 30 (A) For any given summable sequence \vec{p} with positive terms $0 < p_n < 1$ there exists a non-summable sequence \vec{q} of terms $0 < q_n < 1$ such that the expected optimization times $T_{\vec{p}}$ and $T_{\vec{q}}$ of the $(1 + 1) EA_{\vec{p}}$ and $(1 + 1) EA_{\vec{q}}$, respectively, on the LEADINGONES instance of length n in the initial segment uncertainty model satisfy $E[T_{\vec{q}}] = o(E[T_{\vec{p}}])$.

(B) Likewise, for every non-summable sequence \vec{q} of positive terms $0 < q_n < 1$ there exists a summable sequence \vec{p} of elements $0 < p_n < 1$ such that $E[T_{\vec{p}}] = o(E[T_{\vec{q}}])$ for $T_{\vec{p}}$ and $T_{\vec{q}}$ defined as above.

To prove Theorem 30 we need a few auxiliary statements. The first lemma, informally, says that it suffices to regard those $RLS_{\vec{p}}$ whose underlying sequences are monotonically decreasing.

Lemma 31 Let $\vec{p} : \mathbb{N} \rightarrow (0, 1)$ be a summable sequence with $\sum_{i \in \mathbb{N}} p_i \leq 1$. Then there exists a monotonically decreasing summable sequence $\vec{q} : \mathbb{N} \rightarrow (0, 1)$ with $\sum_{i \in \mathbb{N}} q_i = \sum_{i \in \mathbb{N}} p_i$ such that, for all $n \in \mathbb{N}$, $E[T_{RLS_{\vec{q}}}(n)] \leq E[T_{RLS_{\vec{p}}}(n)]$, where we denote by $E[T_{RLS_{\vec{q}}}(n)]$ and $E[T_{RLS_{\vec{p}}}(n)]$ the expected run time of the $RLS_{\vec{q}}$ and $RLS_{\vec{p}}$, respectively, on the LEADINGONES instance of length n in the initial segment uncertainty model.

Proof Note that, since p is summable, there is a maximal value among the p_i . Thus, it is possible to sort the sequence in non-increasing order (by defining the next element of the sequence as the largest unused value). Let q be this sorted sequence.

Let $n \in \mathbb{N}$. Let p'_1, \dots, p'_n be an enumeration of p_1, \dots, p_n in non-increasing order. Then $p'_i \leq q_i$ for all $i \in [1..n]$ and thus

$$E[T_{RLS_{\vec{q}}}(n)] = \frac{1}{2} \sum_{i=1}^n \frac{1}{q_i} \leq \frac{1}{2} \sum_{i=1}^n \frac{1}{p'_i} = \frac{1}{2} \sum_{i=1}^n \frac{1}{p_i} = E[T_{RLS_{\vec{p}}}(n)].$$

□

The second auxiliary statement will allow us to assume that the sequence \vec{p} does not decrease extremely fast.

Lemma 32 Let $\vec{p} : \mathbb{N} \rightarrow (0, 1)$ be a summable sequence with $\sum_{i \in \mathbb{N}} p_i \leq 1$. Then there exists a summable sequence $\vec{q} : \mathbb{N} \rightarrow (0, 1)$ with $\sum_{i \in \mathbb{N}} q_i = \sum_{i \in \mathbb{N}} p_i$ such that, for all $n \in \mathbb{N}$, $q_n \leq \sum_{j>n} q_j$ and $E[T_{RLS_{\vec{q}}}(n)] \leq 2E[T_{RLS_{\vec{p}}}(n)]$, where $E[T_{RLS_{\vec{q}}}(n)]$ and $E[T_{RLS_{\vec{p}}}(n)]$ are defined as in Lemma 31.

Proof We show that we can shift probability mass to larger-indexed positions, if needed.

For all $n \in \mathbb{N}$ let $q_n := p_n$. We modify \vec{q} iteratively with a single pass starting from $i = 1$, maintaining the total sum of the sequence as an invariant. For all i , do the

following. If $q_i \leq \sum_{j>i} q_j$, we do not modify q_i and continue with the definition of q_{i+1} . If, on the other hand, $q_i > \sum_{j>i} q_j$, we let $c_i = q_i/2$ and update $q_i \leftarrow q_i - c_i$ and $q_{i+1} \leftarrow q_{i+1} + c_i$. Note that this maintains our invariant. Now we continue with checking q_{i+1} , possibly modifying q_{i+1} for a second time, but never changing it afterwards.

Clearly, these modifications converge (point-wise) in the limit to a sequence q with the same sum as the sequence p . Furthermore, for all $n \in \mathbb{N}$, we have $q_n \leq \sum_{j>n} q_j$, as any large q_i were cut in half and the other half was shifted to higher indices, guaranteeing that q_i is at most as large as the sum of the following values.

Since we have, for all $i \in \mathbb{N}$, $q_i \geq p_i/2$, we get

$$E[T_{\text{RLS}_{\tilde{q}}}(n)] = \frac{1}{2} \sum_{i=1}^n \frac{1}{q_i} \leq \frac{1}{2} \sum_{i=1}^n \frac{2}{p_i} = 2 E[T_{\text{RLS}_{\tilde{q}}}(n)].$$

□

A corresponding statement can be shown to hold for the $(1 + 1) \text{EA}_{\tilde{p}}$.

Lemma 33 *Let $\tilde{p} : \mathbb{N} \rightarrow (0, 1)$ be a summable sequence with $\sum_{i \in \mathbb{N}} p_i = c < 1$. Then there exists a summable sequence $\tilde{q} : \mathbb{N} \rightarrow (0, 1)$ such that, for all $n \in \mathbb{N}$, $q_n \leq \sum_{j>n} q_j$ and $E[T_{\tilde{q}}(n)] \leq 2 E[T_{\tilde{p}}(n)]$, where by $T_{\tilde{p}}(n)$ ($T_{\tilde{q}}(n)$) we denote the run time of the $(1 + 1) \text{EA}_{\tilde{p}}$ ($(1 + 1) \text{EA}_{\tilde{q}}$) on the LEADINGONES instance of length n in the initial segment uncertainty model, respectively. Moreover, it holds that $\sum_{i \in \mathbb{N}} q_i \leq c/(1 - c)$.*

Proof By Theorem 26 there exists a summable sequence $\tilde{p} = (\tilde{p}_i)_{i \in \mathbb{N}}$ such that the initial segment run time profile of the $\text{RLS}_{\tilde{p}}$ equals that of the $(1 + 1) \text{EA}_{\tilde{p}}$. The construction of \tilde{p} in the proof of Lemma 22 shows that, for all n , $\tilde{p}_n = p_n \prod_{j<n} (1 - p_j) < p_n$ and thus

$$\sum_{i \in \mathbb{N}} \tilde{p}_i \leq \sum_{i \in \mathbb{N}} p_i = c. \tag{12}$$

For this sequence \tilde{p} , by Lemma 32, there exists a summable sequence \tilde{q} with $\sum_{i \in \mathbb{N}} \tilde{q}_i = \sum_{i \in \mathbb{N}} \tilde{p}_i \leq \sum_{i \in \mathbb{N}} p_i = c$ and such that, for all $n \in \mathbb{N}$, it holds that $\tilde{q}_n \leq \sum_{j>n} \tilde{q}_j$ and $E[T_{\text{RLS}_{\tilde{q}}}(n)] \leq 2 E[T_{\text{RLS}_{\tilde{p}}}(n)]$. To this sequence \tilde{q} we apply Lemma 24 to obtain a sequence q such that the initial segment run time profile of $\text{RLS}_{\tilde{q}}$ and the $(1 + 1) \text{EA}_{\tilde{q}}$ are identical, which implies

$$q_n \prod_{k<n} (1 - q_k) = \tilde{q}_n \tag{13}$$

and

$$E[T_{\tilde{q}}(n)] = E[T_{\text{RLS}_{\tilde{q}}}(n)] \leq 2 E[T_{\text{RLS}_{\tilde{p}}}(n)] = 2 E[T_{\tilde{p}}(n)]$$

for all $n \in \mathbb{N}$.

From (13) and $\tilde{q}_n \leq \sum_{j>n} \tilde{q}_j$, we get that

$$q_n = \frac{\tilde{q}_n}{\prod_{k<n}(1 - q_k)} \leq \sum_{j>n} \frac{\tilde{q}_j}{\prod_{k<n}(1 - q_k)} \leq \sum_{j>n} \frac{\tilde{q}_j}{\prod_{k<j}(1 - q_k)} = \sum_{j>n} q_j.$$

It remains to show that \vec{q} is summable with $\sum_{i \in \mathbb{N}} q_i \leq \frac{c}{1-c}$. To show this, we use (12) and (13) and compute

$$\sum_{i=1}^n q_i = \sum_{i=1}^n \frac{\tilde{q}_i}{1 - \sum_{j<i} \tilde{q}_j} \leq \sum_{i=1}^n \frac{\tilde{q}_i}{1 - \sum_{j \in \mathbb{N}} \tilde{q}_j} \leq \frac{c}{1 - c}.$$

□

We are now ready to prove part (A) of Theorem 30.

Proof (of Theorem 30.(A)) Let \vec{p} be a summable sequence of positive terms $0 < p_n < 1$.

We first argue that, without loss of generality, we can assume that $\sum_{i \in \mathbb{N}} p_i = \exp(-1)$ and that, for all $n \in \mathbb{N}$, $p_n \leq \sum_{j>n} p_j$. Since \vec{p} is summable, $p := \sum_{i \in \mathbb{N}} p_i < \infty$. In a first step we set, for all $i \in \mathbb{N}$, $p_i^{(1)} := p_i / (2p)$ to obtain a summable sequence $\vec{p}^{(1)} : \mathbb{N} \rightarrow (0, 1/2)$ with $\sum_{i \in \mathbb{N}} p_i^{(1)} = 1/2$. To $\vec{p}^{(1)}$ we apply Lemma 33 to obtain a summable sequence $\vec{p}^{(2)} : \mathbb{N} \rightarrow (0, 1)$ with $c := \sum_{i \in \mathbb{N}} p_i^{(2)} \leq 1$ such that, for all $n \in \mathbb{N}$, $p_n^{(2)} \leq \sum_{j>n} p_j^{(2)}$ and $E[T_{\vec{p}^{(2)}}(n)] \leq 2E[T_{\vec{p}^{(1)}}(n)]$. We finally set $p_i^{(3)} := p_i^{(2)} / (c \exp(1))$ to obtain a summable sequence $\vec{p}^{(3)} : \mathbb{N} \rightarrow (0, 1/2)$ with $\sum_{i \in \mathbb{N}} p_i^{(3)} = 1/\exp(1)$. By construction, we also have $p_n^{(3)} \leq \sum_{j>n} p_j^{(3)}$ for all $n \in \mathbb{N}$. It remains to show that we did not harm the initial segment run time profile by more than a multiplicative $\Theta(1)$ factor. This follows from applying Lemma 29 twice; once to \vec{p} and $p^{(1)}$ and once to $p^{(2)}$ and $p^{(3)}$, respectively.

In the following, by replacing \vec{p} with $\vec{p}^{(3)}$ if needed, we may therefore assume that $\sum_{i \in \mathbb{N}} p_i = \exp(-1)$ and that, for all $n \in \mathbb{N}$, $p_n \leq \sum_{j>n} p_j$.

We are now ready to define sequence \vec{q} . To this end, for every $i \in \mathbb{N}$, set $r_i := \sum_{j \geq i} p_j$. Set

$$q_i := \frac{p_i}{2r_{i+1} \ln(1/r_{i+1})} = -\frac{p_i}{2r_{i+1} \ln(r_{i+1})}.$$

Note that, for $i \in \mathbb{N}_{\geq 2}$, we have $0 < r_i < \exp(-1)$ and thus $q_i > 0$. Since for all i we have $p_i \leq r_{i+1}$ by assumption (see our discussion above), and since $1/\ln(1/r_{i+1}) < 1$ (this follows from $r_{i+1} < \sum_{i \in \mathbb{N}} p_i = \exp(1)$ and the monotonicity of the function $x \mapsto 1/\ln(1/x)$), we also have $q_i \leq p_i / (2r_{i+1}) \leq 1/2$. Thus, $0 \leq q_i \leq 1/2$ for all $i \in \mathbb{N}$.

Next we prove that $(q_i)_{i \in \mathbb{N}}$ is non-summable. To this end, we first observe that the function $x \mapsto 1/(x \ln(1/x))$ is monotonically decreasing in the interval $(0, \exp(-1))$;

this can easily be seen by taking the derivative. We can therefore estimate, for all $n \in \mathbb{N}$,

$$\begin{aligned}
 q_i &= \frac{p_i}{2r_{i+1} \ln(1/r_{i+1})} = \frac{r_i - r_{i+1}}{2r_{i+1} \ln(1/r_{i+1})} \\
 &\geq \int_{r_{i+1}}^{r_i} \frac{1}{2x \ln(1/x)} dx = - \int_{r_{i+1}}^{r_i} \frac{1}{2x \ln(x)} dx.
 \end{aligned}
 \tag{14}$$

This shows

$$\begin{aligned}
 \sum_{i=1}^n q_i &\geq -\frac{1}{2} \int_{r_{n+1}}^{r_1} \frac{1}{x \ln(x)} dx = \frac{1}{2} \int_{\ln(r_1)}^{\ln(r_{n+1})} \frac{1}{x} dx = \frac{1}{2} [\ln(|x|)]_{\ln(r_{n+1})}^{\ln(r_1)} \\
 &= \frac{1}{2} \ln(|\ln(r_{n+1})|) - \ln(|\ln(r_1)|) = \frac{1}{2} \ln(\ln(1/r_{n+1}))
 \end{aligned}
 \tag{15}$$

where we have used in the second step the substitution rule $\int_a^b f(g(x))g'(x)dx = \int_{g(a)}^{g(b)} f(x)dx$ with $f(x) = 1/x$ and $g(x) = \ln(x)$ as well as the rule $-\int_{g(a)}^{g(b)} f(x)dx = \int_{g(b)}^{g(a)} f(x)dx$ and in the last step we have used that $r_1 = 1/e$ implies that $\ln(|\ln(r_1)|) = \ln(1) = 0$. We note that, since \vec{p} is summable, we have that $(r_i)_{i \in \mathbb{N}}$ converges to 0, which shows that $\ln(\ln(1/r_{n+1}))$ diverges. This proves that the partial sums $\sum_{i=1}^n q_i$ diverge and the series $\vec{q} = (q_i)_{i \in \mathbb{N}}$ is hence not summable.

It remains to show that $E[T_{\vec{q}}] = o(E[T_{\vec{p}}])$. To this end, we first recall from the proof of Lemma 27 that the expected run time of the $(1 + 1) EA_{\vec{q}}$ on LEADINGONES is at most

$$\frac{1}{2} \sum_{i=1}^n \frac{\exp(2S_{i-1}(\vec{q}))}{q_i},$$

where, for $i \in \mathbb{N}$ we denote by $S_i(\vec{q}) := \sum_{j=1}^i q_j$ the i -th partial sum of \vec{q} and $S_0(\vec{q}) := 0$. Likewise, we know that the expected run time of the $(1 + 1) EA_{\vec{p}}$ on LEADINGONES is at least

$$\frac{1}{2} \sum_{i=1}^n \frac{\exp(S_{i-1}(\vec{p}))}{p_i}.$$

As formalized in Lemma 45, it therefore suffices to show that

$$\frac{\exp(2S_{i-1}(\vec{q}))}{q_i} = o\left(\frac{\exp(S_{i-1}(\vec{p}))}{p_i}\right) = o\left(\frac{1}{p_i}\right),$$

where the last step uses the summability of \vec{p} .

We recall from (14) that $q_i = \frac{r_i - r_{i+1}}{2r_{i+1} \ln(1/r_{i+1})}$. Using that $\frac{1}{x \ln(1/x)} \leq \frac{2}{2x \ln(1/(2x))}$ for $0 < x < 1/2$ and that $r_j = r_{j+1} + p_j \leq 2r_{j+1}$, we can thus bound $q_i \leq \int_{r_{i+1}}^{r_i} \frac{1}{x \ln(1/x)} dx$. With this inequality at hand, the same algebraic transformations as in (15) show that

$$S_{i-1}(\vec{q}) = \sum_{j=1}^{i-1} q_j \leq \ln(\ln(1/r_i)).$$

We thus obtain

$$\begin{aligned} \frac{\exp(2S_{i-1}(\vec{q}))}{q_i} &\leq \frac{2r_{i+1} \ln(1/r_{i+1}) \exp(2 \ln(\ln(1/r_i)))}{p_i} \\ &= \frac{2r_{i+1} \ln(1/r_{i+1})(\ln(1/r_i))^2}{p_i} = o(1/p_i), \end{aligned}$$

where the last step follows from the fact that the term $r_{i+1} \ln(1/r_{i+1})(\ln(1/r_i))^2 \leq r_{i+1}(\ln(1/r_{i+1}))^3$, which converges to zero for $i \rightarrow \infty$ (since for all $s \in \mathbb{R}$ the term $x(\ln(x))^s$ converges to zero for $x \rightarrow 0$ and $r_{i+1} \rightarrow 0$ for $i \rightarrow \infty$). \square

We now prove the second part of Theorem 30.

Proof (of Theorem 30.(B)) Let \vec{q} be a non-summable sequence of positive terms $0 < q_n < 1$. For all $n \in \mathbb{N}$, set $S_n := \sum_{i=1}^n q_i$. Let $p_1 := q_1/2$ and, for all $n \geq 2$, let $p_n := q_n/(2S_n^2)$. It is easily seen that $0 < p_n < 1/2$ for all n .

To see that $\vec{p} := (p_i)_{i \in \mathbb{N}}$ is summable, we estimate, for $n \geq 2$

$$p_n = \frac{S_n - S_{n-1}}{2S_n^2} \leq \frac{1}{2} \int_{S_{n-1}}^{S_n} \frac{1}{x^2} dx$$

and, thus,

$$\sum_{i=1}^n p_i \leq \frac{1}{2} \left(q_1 + \int_{S_1}^{S_n} \frac{1}{x^2} dx \right) \leq C$$

for some constant C .

Using the summability of \vec{p} , from Lemma 27 we obtain that, for some constant $C' > 0$,

$$E[T_{\vec{p}}(n)] \leq C' \sum_{i=1}^n \frac{1}{p_i} \leq 2C' \left(\frac{1}{q_1} + \sum_{i=2}^n \frac{S_i^2}{q_i} \right)$$

while

$$E[T_{\vec{q}}(n)] \geq \frac{1}{2} \sum_{i=1}^n \frac{\exp(S_{i-1})}{q_i} \geq \frac{1}{2} \sum_{i=1}^n \frac{\exp(S_i - 1)}{q_i}.$$

Since \vec{q} is non-summable, we have that S_i diverges to infinity. This gives $S_i^2 = o(\exp(S_i))$. The desired statement $E[T_{\vec{p}}] = o(E[T_{\vec{q}}])$ now follows from a direct application of Lemma 45. \square

6.2.3 Limiting Behavior

The previous results show, in particular, that there is no best possible $(1 + 1) EA_{\vec{p}}$ for LEADINGONES: whatever the sequence underlying the $(1 + 1) EA_{\vec{p}}$ looks like, there exists another one giving strictly better performance. It is nevertheless interesting to understand which absolute performance guarantees can be achieved. This is the aim of the following statement.

Theorem 34 *Let $b > 1$. For every sequence \vec{p} of positive terms $0 < p_n < 1$ and for all $s \in \mathbb{N}$ the run time $T_{\vec{p}}$ of the $(1 + 1) EA_{\vec{p}}$ on the LEADINGONES instance of length n in the initial segment uncertainty model satisfies $E[T_{\vec{p}}(n)] = \omega(n/p_b^{s,0}(n)) = \omega(n^2 \log_b(n) \log_b^{(2)}(n) \dots \log_b^{(s)}(n))$.*

For $b \geq e$ it also holds that $E[T_{\vec{p}}(n)] = \omega(n/p_b^\infty(n)) = \omega(n^2 \log_b(n) \log_b^{(2)}(n) \dots)$.

Proof Let \vec{p} be a sequence of positive terms $0 < p_n < 1$ and $s \in \mathbb{N}$. If \vec{p} is non-summable, Theorem 30.(B) tells us that there exists a summable sequence \vec{q} of terms $0 < q_n < 1$ such that $E[T_{\vec{p}}] = \omega(E[T_{\vec{q}}])$. We can therefore assume without loss of generality that \vec{p} is summable.

Lemma 27 allows to bound

$$E[T_{\vec{p}}(n)] \geq \frac{1}{2} \sum_{i=1}^n \frac{\exp(S_{i-1})}{p_i} = \Omega\left(\sum_{i=1}^n \frac{1}{p_i}\right).$$

By Lemma 42 it holds that $1/p_i = \omega(1/p_b^{s,0}(i)) = \omega(i \log_b(i) \log_b^{(2)}(i) \dots \log_b^{(s)}(i))$. Lemma 45 therefore implies that

$$\begin{aligned} E[T_{\vec{p}}(n)] &= \omega\left(\sum_{i=1}^n i \log_b(i) \log_b^{(2)}(i) \dots \log_b^{(s)}(i)\right) \\ &= \omega\left(n^2 \log_b(n) \log_b^{(2)}(n) \dots \log_b^{(s)}(n)\right), \end{aligned}$$

where we have used in the last step that, for all $i \geq n/2$, $i \log_b(i) \log_b^{(2)}(i) \dots \log_b^{(s)}(i) = \Theta(n \log_b(n) \log_b^{(2)}(n) \dots \log_b^{(s)}(n))$ holds.

The second statement follow similarly using Corollary 44 instead of Lemma 42. \square

As mentioned in the beginning of Sect. 6.2, Theorem 26 implies that all lower bounds proven for initial segment run time profile of the $(1 + 1) EA_{\vec{p}}$ immediately apply to the $(1 + 1) EA_Q$ with $Q = Q(\vec{p})$. The following corollary makes this explicit.

Corollary 35 *Let $b > 1$. For every sequence \vec{p} of positive terms $0 < p_n < 1$ and for all $s \in \mathbb{N}$ the run time $T_{Q(\vec{p})}$ of the $(1 + 1) EA_Q$ on the LEADINGONES instance of length n in the initial segment uncertainty model satisfies $E[T_{Q(\vec{p})}(n)] = \omega(n/p_b^{s,0}(n)) = \omega(n^2 \log_b(n) \log_b^{(2)}(n) \dots \log_b^{(s)}(n))$.*

For $b \geq e$ it also holds that $E[T_{Q(\bar{p})}(n)] = \omega(n/p_b^\infty(n)) = \omega(n^2 \log_b(n) \log_b^{(2)}(n) \dots)$.

7 Lower Bounds for OneMax in the Unrestricted Uncertainty Model

In this section we give lower bounds for the case of ONEMAX in the unrestricted uncertainty model (see Theorem 39). However, we start with a few technical lemmas. The first two lemmas are essentially just applications of Chernoff bounds, the third is a combination of the other two.

Lemma 36 ([10, Proposition 2]) *Consider the $(1 + 1)$ EA with mutation probability p on ONEMAX instance of length n in the unrestricted uncertainty model and a current search point with k incorrectly set bits. If $k \leq 0.6n$, then the probability that the search point of the next iteration has at most $k/2$ incorrectly set bits is $\exp(-\Omega(k))$.*

Proof Suppose the current search point is $x \in \{0, 1\}^n$ with $k \leq 0.6n$ incorrectly set bits. The next search point is now derived by flipping each bit in x independently with probability p . Thus, an application of [10, Proposition 2] gives the desired result. \square

Lemma 37 *Consider the $(1 + 1)$ EA with mutation probability p on ONEMAX instance of length n in the unrestricted uncertainty model and a current search point with $k \leq n/4$ incorrectly set bits. Then the probability that the offspring of mutation is accepted is at most $\exp(-cpn)$ for some constant c . Furthermore, the bound also holds when conditioning on a specific incorrect bit flipping.*

Proof The claim follows using the Chernoff bound as follows. The expected number of correct bits that are flipped is $p(n - k)$, while the expected number of incorrect bits that are flipped is pk . Using Chernoff bounds (and a union bound) we see that the probability to flip less than $pk/2$ correct bits or more than $pk/2$ incorrect bits is $\exp(-cpn)$, for some constant c . In order for the offspring to be accepted, we need to flip at least as many incorrect bits as correct bits. When conditioning on a specific incorrect bit flipping, only lower order terms change. \square

Lemma 38 *Consider the $(1 + 1)$ EA with mutation probability p on ONEMAX instance of length n in the unrestricted uncertainty model and a current search point with $k \leq \sqrt[3]{n}$ incorrectly set bits. Fix an incorrectly set bit in the current search point and let X be a 0-1-random variable which is 1 only if the bit is corrected by mutation applied to the current search point and the mutated individual is accepted. We have, for some constant c ,*

$$E(X) \leq p \exp(-cpn).$$

Proof We have that p is the probability of flipping the bit. Conditional on this bit flipping we have a probability of accepting the offspring of $\exp(-cpn)$, as given in Lemma 37. \square

The main theorem of this section is our lower bound for optimizing ONEMAX in the unrestricted uncertainty model.

Theorem 39 *Let Q be a discrete distribution over mutation probabilities from $(0, 1)$. Let $(d_i)_{i \in \mathbb{N}} \in (0, 1)^{\mathbb{N}}$ be any non-summable sequence.*

For infinitely many $n \in \mathbb{N}$, the expected run time of the $(1 + 1)$ EA $_Q$ on ONEMAX instance of length n in the unrestricted uncertainty model is at least $c \log(n)/d_n$ for some constant c .

The idea behind the proof is to assign a *quality* $q(n)$ of how suitable a distribution Q is for optimizing a ONEMAX instance of length n . Clearly, whenever Q chooses a bit flip probability of around $1/n$, optimization can proceed well; smaller probabilities are slower in optimizing ONEMAX, higher probabilities are even disruptive. We show that (a) the sequence $q(n)$ is summable, showing that it is infinitely often smaller than any given non-summable sequence; and (b) the expected progress (i.e., the drift) is at most $kq(n)$ when still k bits are missing. Together this gives the claimed lower bound.

Proof (of Theorem 39) Let c be as given by Lemma 38 and let D be the set of values assumed by Q . Define $q : (0, \infty) \rightarrow \mathbb{R}$ such that, for all $x > 0$,

$$q(x) = \sum_{p \in D} p \exp(-cpx) P(Q = p).$$

We will now show that the integral over q is finite. For this we will exchange an integral with a (possibly infinite) sum, which we can do with Fubini’s Theorem, given that the function we integrate over is non-negative and we show that one of the two terms is finite, which will be provided by the remainder of the chain of inequalities.

$$\begin{aligned} \int_0^\infty q(x) dx &= \int_0^\infty \sum_{p \in D} p \exp(-cpx) P(Q = p) dx \\ &= \sum_{p \in D} \int_0^\infty p \exp(-cpx) P(Q = p) dx \\ &= \sum_{p \in D} \left[\frac{p}{-cp} \exp(-cpx) \right]_0^\infty P(Q = p) \\ &= \sum_{p \in D} \frac{1}{c} P(Q = p) \\ &= \frac{1}{c}. \end{aligned}$$

Note that the function q is monotonically decreasing. This and the fact that the integral is finite imply that $\sum_{n=1}^\infty q(n)$ is finite. Thus, there are infinitely many n such that $q(n) \leq d_n$, since $(d_n)_n$ is a non-summable sequence. Let one such n be given.

Consider the optimization of $(1 + 1)$ EA $_Q$ on ONEMAX and suppose we currently have k wrong bits. Using Lemma 36 we know that, with high probability, the process

will have between $\sqrt[3]{n}$ and $\sqrt[3]{n}/2$ incorrect bits at some point. We consider the process starting from such a point in time and show that the remaining time is as large as desired.

Let M be the set of incorrect bit positions and fix a mutation probability p . For each $i \in M$ let X_i be the 0-1-variable indicating that this bit position is fixed in the next iteration. Using Lemma 38, we have, for all $i \in M$, $E(X_i) \leq p \exp(-cpn)$. Thus, the total drift in incorrect bit positions is bounded from above by

$$E\left(\sum_{i \in M} X_i\right) = \sum_{i \in M} E(X_i) \leq kp \exp(-cpn).$$

Consider now drawing a mutation probability $p \sim Q$. Regarding the drift Δ in incorrect bit positions we have

$$E(\Delta) \leq \sum_{p \in D} kp \exp(-cpn) P(Q = p) \leq kq(n) \leq kd_n.$$

We now want to use the lower bound multiplicative drift theorem from [31]. The main requirement of this theorem is the upper bound on the expected drift on the number of incorrect bits as derived above; thus, in the language of [31, Theorem 2.2], we chose $\delta = d_n$. The requirement of not making big jumps is fulfilled with $\beta = 1/2$ by Lemma 36 while we have $k \geq \sqrt[3]{n}/4$. □

From Theorem 39, Lemma 42, and Corollary 44 we obtain the following result.

Corollary 40 *For all $b \geq e$ and all distributions Q over $(0, 1)$ there exists a constant $c > 0$ such that for infinitely many $n \in \mathbb{N}$ the expected run time of the $(1 + 1)$ EA $_Q$ on a ONEMAX instance of length n in the unrestricted uncertainty model is at least $c \log(n)/p_b^\infty(n) = \Omega(n(\log_b(n))^2 \log_b^{(2)}(n) \log_b^{(3)}(n) \dots)$.*

Likewise, for all $s \in \mathbb{N}$ and all distributions Q over $(0, 1)$, there exists a constant $c > 0$ such that, for infinitely many $n \in \mathbb{N}$, the expected run time of the $(1 + 1)$ EA $_Q$ on a ONEMAX instance of length n in the unrestricted uncertainty model is at least $c \log(n)/p_b^{(s,0)}(n) = \Omega(n(\log(n))^2 \log^{(2)}(n) \dots \log^{(s)}(n))$.

In contrast to the case of LEADINGONES, our results for ONEMAX only pertain to the unrestricted uncertainty model, but not the initial segment uncertainty model. This latter model turns out to be harder to give good lower bounds for because of the following phenomenon. While for LEADINGONES early bits must not flip for making progress, for ONEMAX early bits may flip in exchange of later flips. If these later bits flip more rarely (and are thus more costly to optimize), such an exchange may actually be beneficial. It remains open how this effect can be analyzed.

8 Summary and Outlook

We have analyzed the performance of variants of the $(1 + 1)$ EA in the presence of unknown solution lengths. We studied the use of position-dependent mutation proba-

bilities and showed that, when optimizing LEADINGONES, they are unnecessary if the distribution of the solution length is known and has a light tail (in the sense that no asymptotic gain is possible). We have also shown that position-dependent mutation rates are crucial in settings in which we do not have any knowledge about the solution length. Surprisingly, in the latter situation, a sequence of (position-dependent) mutation probabilities exists such that the corresponding $(1 + 1)$ EA is almost optimal, simultaneously for all possible solution lengths. For optimizing ONEMAX we have shown analogous good upper bounds and conjecture that, just as for LEADINGONES, these bounds are asymptotically tight.

We have also investigated a setting in which the relevant bit positions can be arbitrary in number and position. Possibly even more surprisingly, even this can be handled quite efficiently by a $(1 + 1)$ EA variant for the two test functions ONEMAX and LEADINGONES.

Finally, we have proved the first non-trivial lower bounds for the regarded $(1 + 1)$ EA $_{\bar{p}}$ and $(1 + 1)$ EA $_Q$ algorithm classes. These were made possible by a result showing that both classes have the same performance when optimizing LEADINGONES in the initial segment uncertainty model. If this is not a particularity of the LEADINGONES function but rather a general phenomenon, then this would suggest to rather use the easier to understand $(1 + 1)$ EA $_Q$ class. Note that, for a $(1 + 1)$ EA $_Q$ algorithm, for many problems the run time on a subinstance of length n can be upper bounded by the reciprocal of the probability that the mutation rate is in $[1/2n, 2n]$ times the worst-case run time of the $(1 + 1)$ EA with mutation rate in the interval $[1/2n, 2n]$ on this instance (without uncertainty). This estimate is valid if additional iterations with other mutation rates cannot be harmful. This is the case, e.g., for run time analyses using the fitness level method. The next step towards increasing our understanding on the relation of these algorithm classes would be to prove tight bounds for the $(1 + 1)$ EA $_{\bar{p}}$ class on the ONEMAX function in the initial segment uncertainty model, a problem that we unfortunately could not solve.

We believe the setting of unknown solution length to be relevant for numerous real-world applications. As a next step toward a better understanding of how this uncertainty can be tackled efficiently with evolutionary algorithms, we suggest to investigate more challenging function classes, e.g., starting with the class of all linear functions. It is not clear a priori if bounds similar to the ones presented in Sect. 4 can be achieved for such problems.

Acknowledgements Parts of this work have been done while Timo Kötzing was visiting the École Polytechnique. This work was supported in part by the German Research Foundation under Grant FR 2988 (TOSU), by a public grant as part of the Investissement d’avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH, and from the Gaspard Monge Program for Optimization and Operations Research (PGMO) of the Jacques Hadamard Mathematical Foundation (FMJH).

Appendix A: Summable Sequences

In this section we recall definitions and results regarding summable sequences.

A sequence $\vec{p} = (p_i)_{i \in \mathbb{N}}$ is *summable* if the sequence $(S_n)_{n \in \mathbb{N}}$ of partial sums of absolute values $S_n := \sum_{i=1}^n |p_i|$ converges.

The run time guarantees in Theorems 15, 16, 19, and 20 all decrease with increasing sequence \vec{p} . In the hope of finding best possible parameters for the $(1 + 1)$ EA $_{\vec{p}}$ and the $(1 + 1)$ EA $_Q$ it is therefore natural to ask for a largest summable sequence p . The following statement shows that such a sequence does not exist. More precisely, for every summable sequence \vec{p} , there is a summable sequence which decreases more slowly than \vec{p} . Such constructions are well known in the mathematics literature, cf. [2] for an easily-accessible example proving Theorem 41.

Theorem 41 (Folklore) *For every summable sequence $\vec{p} := (p_n)_{n \in \mathbb{N}}$ of positive terms $p_n > 0$ there exists a summable sequence $\vec{q} := (q_n)_{n \in \mathbb{N}}$ such that $\vec{q} = \omega(\vec{p})$.*

Likewise, for every non-summable sequence \vec{q} there exists a non-summable sequence \vec{p} such that $\vec{p} = o(\vec{q})$.

It is interesting to note that, in both statements of Theorem 41, the sequence \vec{q} can be chosen in a way that the expected run times of the $(1 + 1)$ EA $_{\vec{p}}$ when fed with sequence \vec{p} and \vec{q} , respectively, satisfy $E[T_{\vec{q}}] = o(E[T_{\vec{p}}])$ (this is not automatically the case when $\vec{q} = \omega(\vec{p})$). But since this result also follows from Theorem 30, we do not prove these statements explicitly.

We have introduced in equation (5) (Sect. 4.2) the sequences $(p_b^{s,\varepsilon}(n))_{n \in \mathbb{N}}$. We have also discussed that it was shown in [21, page 48] that for every $\varepsilon > 0$ and every $s \geq 1$, the sequence $(p_2^{s,\varepsilon}(n))_{n \in \mathbb{N}}$ is summable. In the same work, Hardy also mentions that, for all $s \in \mathbb{N}$, the sequence $(p_2^{s,0}(n))_{n \in \mathbb{N}}$ is not summable; he attributes this result to De Morgan and Bertrand. From this result we easily get the following statement, which we need in the proof of our lower bounds in Sect. 6.2.3.

Lemma 42 *Let $b > 1$ and let \vec{p} be a summable sequence of positive terms $0 < p_n < 1$. Then, for all $s \in \mathbb{N}$, $1/p_n = \omega(1/p_b^{s,0}(n))$; i.e., $1/p_n = \omega(n \log_b(n) \log_b^{(2)}(n) \dots \log_b^{(s)}(n))$.*

Proof We show the statement for $b = 2$. The general case follows from observing that, for all $s \in \mathbb{N}$ and all $b > 1$, $1/p_b^{s,0}(n) = \Theta(1/p_2^{s,0}(n))$.

We thus need to show that for all summable sequences \vec{p} of positive terms $0 < p_n < 1$, all integers $s \in \mathbb{N}$, and all positive constants $C > 0$ there exists an integer $n_0 = n_0(\vec{p}, s, C)$ such that, for all $n \geq n_0$, it holds that $1/p_n \geq C/p_2^{s,0}(n) = Cn \log_2(n) \log_2^{(2)}(n) \dots \log_2^{(s)}(n)$.

For the sake of contradiction we assume that this is not the case. Then there exists a summable sequence \vec{p} of positive terms $0 < p_n < 1$, an integer $s \in \mathbb{N}$, and a constant $C > 0$ such that for all $n_0 \in \mathbb{N}$ there exists an integer $n \geq n_0$ such that $1/p_n \leq C/p_2^{s,0}(n) = Cn \log_2(n) \log_2^{(2)}(n) \dots \log_2^{(s)}(n)$. This implies that there exists a subsequence \vec{q} of \vec{p} such that $q_n \geq 1 / (Cn \log_2(n) \log_2^{(2)}(n) \dots \log_2^{(s)}(n))$. However, the result of De Morgan and Bertrand states that this sequence is not summable. Hence, \vec{p} cannot be summable. \square

Furthermore, we have mentioned in Sect. 4.2 that the summability of the sequence $(p_b^\infty(n))_{n \in \mathbb{N}}$ (cf. equation (7)) crucially depends on the base b of the logarithm. Indeed, using Cauchy’s condensation test, one can show the following.

Theorem 43 *The sequence $(p_b^\infty(n))_{n \in \mathbb{N}}$ is summable if and only if $b < e := \exp(1)$. More precisely,*

1. For all $b > 1$ and all $s \in \mathbb{N}$ it holds that $\sum_{i=1}^n p_b^{(s,0)}(i) = \Theta(\log^{(s+1)}(n))$.
2. For all $1 < b < e := \exp(1)$, $\sum_{i=1}^n p_b^\infty(i) = \Theta(1)$.
3. $\sum_{i=1}^n p_e^\infty(i) = \Theta(\log^*(n))$.
4. For all $b > e$, $\sum_{i=1}^n p_b^\infty(i) = \exp(\Theta(\ln^*(n)))$.

Proof For this proof, we set

$$e(s) := \exp^{(s)}(1) = \underbrace{\exp(\exp(\dots(\exp(1))))}_{s \text{ times}}$$

Let $s \in \mathbb{N}$ and $b > 1$. We apply integration by substitution, in the formulation of Theorem 46 below and with $\varphi = \exp_b$, $s + 1$ times to the function $x \mapsto p_b^{(s,0)}(x)$. Using $b^x p_b^{(s,0)}(b^x) = p_b^{(s-1,0)}(x)$ and that the derivative of \exp_b is \exp_b times $\ln(b)$ we obtain, for $n \geq e(s)$,

$$\int_{e(s)}^n p_b^{(s,0)}(x) dx = \int_0^{\ln^{(s+1)}(n)} (\ln(b))^{s+1} dx = \Theta(\ln^{(s+1)}(n)) = \Theta(\log^{(s+1)}(n)).$$

Thus, we also get (using p_b^∞ monotonically decreasing)

$$\begin{aligned} \sum_{i=2}^n p_b^\infty(i) &\leq \int_1^n p_b^\infty(x) dx \leq \sum_{j=0}^{\ln^*(n)} \int_{e(j)}^{e(j+1)} p_b^\infty(x) dx \\ &= \sum_{j=0}^{\ln^*(n)} \int_{e(j)}^{e(j+1)} p_b^{(j,0)}(x) dx = \sum_{j=0}^{\ln^*(n)} \int_0^1 (\ln(b))^{j+1} dx \\ &= \sum_{j=0}^{\ln^*(n)} (\ln(b))^{j+1}. \end{aligned}$$

Clearly, for $b < e$, this sum converges for $n \rightarrow \infty$. Furthermore, for $b = e$, this partial sum equals $\ln^*(n)$, while for $b > e$ it is $2^{O(\ln^*(n))}$ as desired. A corresponding lower bound can be found analogously. □

From Theorem 43 we also get the following result.

Corollary 44 *Let $b \geq e$. For every summable sequence \vec{p} it holds that $1/p_n = \omega(1/p_b^\infty(n)) = \omega(n \log_b(n) \log_b^{(2)}(n) \dots)$.*

Corollary 44 can be proven in the same way as Lemma 42.

Appendix B: Useful Tools

We use this section to recall two useful theorems.

The following well-known lemma is applied a few times in our proofs.

Lemma 45 *Let $\vec{a} = (a_n)_{n \in \mathbb{N}}$ and $\vec{b} = (b_n)_{n \in \mathbb{N}}$ be sequences of positive terms satisfying $\vec{a} = o(\vec{b})$. Set $A := (A_n)_{n \in \mathbb{N}}$ with $A_n := \sum_{i=1}^n a_i$ and $B := (B_n)_{n \in \mathbb{N}}$ with $B_n := \sum_{i=1}^n b_i$. Assume that $B = \omega(1)$. Then $A = o(B)$.*

Proof We need to show that for all $\varepsilon > 0$ there exists an integer $n(\varepsilon) \in \mathbb{N}$ such that $A_n/B_n \leq \varepsilon$. Let $\varepsilon > 0$. Since $\vec{a} = o(\vec{b})$ there exists an integer $n_\varepsilon \in \mathbb{N}$ such that $a_n/b_n < \varepsilon/2$ for all $n \geq n_\varepsilon$. Since $B = \omega(1)$ there exists $n_{\varepsilon,B} \in \mathbb{N}$ such that $\sum_{i=1}^{n_\varepsilon-1} a_i/B_n \leq \varepsilon/2$ for all $n \geq n_{\varepsilon,B}$. We therefore obtain for all $n \geq \max\{n_\varepsilon, n_{\varepsilon,B}\}$

$$\begin{aligned} \frac{A_n}{B_n} &= \frac{\sum_{i=1}^{n_\varepsilon-1} a_i}{B_n} + \frac{\sum_{i=n_\varepsilon}^n a_i}{B_n} \\ &\leq \frac{\varepsilon}{2} + \frac{\varepsilon \sum_{i=n_\varepsilon}^n b_i}{2B_n} \leq \varepsilon. \end{aligned}$$

□

When we apply integration by substitution, we refer to the following formulation.

Theorem 46 (Integration by Substitution) *Let $m < n$ and let I be an Interval. Let $f : I \rightarrow \mathbb{R}$ be continuous and $\varphi : [m, n] \rightarrow I$. Then we have*

$$\int_{\varphi(m)}^{\varphi(n)} f(x)dx = \int_m^n f(\varphi(x))\varphi'(x)dx.$$

References

1. Antipov, D., Doerr, B., Fang, J., Hetet, T.: Runtime analysis for the $(\mu + \lambda)$ EA optimizing OneMax. In: Genetic and Evolutionary Computation Conference (GECCO'18). ACM (2018) (to appear)
2. Ash, J.M.: Neither a worst convergent series nor a best divergent series exists. *Coll. Math. J.* **28**, 296–297 (1997)
3. Auger, A., Doerr, B.: *Theory of Randomized Search Heuristics*. World Scientific, Singapore (2011)
4. Bianchi, L., Dorigo, M., Gambardella, L., Gutjahr, W.: A survey on metaheuristics for stochastic combinatorial optimization. *Nat. Comput.* **8**, 239–287 (2009)
5. Bötcher, S., Doerr, B., Neumann, F.: Optimal fixed and adaptive mutation rates for the LeadingOnes problem. In: *Proceedings of Parallel Problem Solving from Nature (PPSN'10)*, pp. 1–10. Springer, Berlin (2010)
6. Cathabard, S., Lehre, P.K., Yao, X.: Non-uniform mutation rates for problems with unknown solution lengths. In: *Proceedings of Foundations of Genetic Algorithms (FOGA'11)*, pp. 173–180. ACM (2011)
7. Dietzfelbinger, M., Rowe, J.E., Wegener, I., Woelfel, P.: Tight bounds for blind search on the integers and the reals. *Comb. Probab. Comput.* **19**, 711–728 (2010)
8. Doerr, B.: Better runtime guarantees via stochastic domination. CoRR abs/1801.04487 (2018). <http://arxiv.org/abs/1801.04487>
9. Doerr, B.: Probabilistic tools for the analysis of randomized optimization heuristics. CoRR abs/1801.06733 (2018). <http://arxiv.org/abs/1801.06733>

10. Doerr, B., Doerr, C.: Optimal static and self-adjusting parameter choices for the $(1 + (\lambda, \lambda))$ genetic algorithm. *Algorithmica* **80**, 1658–1709 (2018)
11. Doerr, B., Doerr, C., Kötzing, T.: Solving problems with unknown solution length at (almost) no extra cost. In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'15)*, pp. 831–838. ACM (2015)
12. Doerr, B., Doerr, C., Kötzing, T.: The right mutation strength for multi-valued decision variables. In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'16)*, pp. 1115–1122. ACM (2016)
13. Doerr, B., Doerr, C., Kötzing, T.: Unknown solution length problems with no asymptotically optimal run time. In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'17)*, pp. 1367–1374. ACM (2017)
14. Doerr, B., Fouz, M., Witt, C.: Quasirandom evolutionary algorithms. In: *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference (GECCO'10)*, pp. 1457–1464. ACM (2010)
15. Doerr, B., Fouz, M., Witt, C.: Sharp bounds by probability-generating functions and variable drift. In: *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO'11)*, pp. 2083–2090. ACM (2011)
16. Doerr, B., Jansen, T., Witt, C., Zarges, C.: A method to derive fixed budget results from expected optimisation times. In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'13)*, pp. 1581–1588. ACM (2013)
17. Doerr, B., Johannsen, D., Winzen, C.: Multiplicative drift analysis. *Algorithmica* **64**, 673–697 (2012)
18. Doerr, B., Künnemann, M.: Optimizing linear functions with the $(1+\lambda)$ evolutionary algorithm—different asymptotic runtimes for different instances. *Theor. Comput. Sci.* **561**, 3–23 (2015)
19. Doerr, B., Le, H.P., Makhmara, R., Nguyen, T.D.: Fast genetic algorithms. In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'17)*, pp. 777–784. ACM (2017)
20. Droste, S., Jansen, T., Wegener, I.: On the analysis of the $(1+1)$ evolutionary algorithm. *Theor. Comput. Sci.* **276**, 51–81 (2002)
21. Hardy, G.H.: *Orders of Infinity*. Cambridge University Press, Cambridge (1910)
22. Hwang, H., Panholzer, A., Rolin, N., Tsai, T., Chen, W.: Probabilistic analysis of the $(1+1)$ -evolutionary algorithm. *Evol. Comput.* **26**, 299–345 (2018)
23. Jansen, T.: *Analyzing Evolutionary Algorithms—The Computer Science Perspective*. Springer, Berlin (2013)
24. Jansen, T., De Jong, K.A., Wegener, I.: On the choice of the offspring population size in evolutionary algorithms. *Evol. Comput.* **13**, 413–440 (2005)
25. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments—a survey. *IEEE Trans. Evol. Comput.* **9**, 303–317 (2005)
26. Ladret, V.: Asymptotic hitting time for a simple evolutionary model of protein folding. *J. Appl. Probab.* **42**, 39–51 (2005)
27. Lehre, P.K., Yao, X.: Runtime analysis of the $(1 + 1)$ EA on computing unique input output sequences. *Inf. Sci.* **259**, 510–531 (2014)
28. Neumann, F., Witt, C.: *Bioinspired Computation in Combinatorial Optimization—Algorithms and Their Computational Complexity*. Springer, Berlin (2010)
29. Sudholt, D.: A new method for lower bounds on the running time of evolutionary algorithms. *IEEE Trans. Evol. Comput.* **17**, 418–435 (2013)
30. Witt, C.: Runtime analysis of the $(\mu + 1)$ EA on simple pseudo-Boolean functions. *Evol. Comput.* **14**, 65–86 (2006)
31. Witt, C.: Tight bounds on the optimization time of a randomized search heuristic on linear functions. *Comb. Probab. Comput.* **22**, 294–318 (2013)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.