Unknown Solution Length Problems With No Asymptotically Optimal Run Time

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

Carola Doerr
CNRS and LIP6, Sorbonne Universités,
UPMC Univ Paris 06
Paris, France

Timo Kötzing
Hasso Plattner Institute
Potsdam, Germany

ABSTRACT

We revisit the problem of optimizing a fitness function of unknown dimension; that is, we face a function defined over bit-strings of large length N, but only $n \ll N$ of them have an influence on the fitness. Neither the position of these relevant bits nor their number is known. In previous work, variants of the (1 + 1) evolutionary algorithm (EA) have been developed that solve, for arbitrary $s \in \mathbb{N}$, such OneMax and LeadingOnes instances, simultaneously for all $n \in \mathbb{N}$, in expected time $O(n(\log(n))^2 \log\log(n) \ldots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$ and $O(n^2 \log(n) \log\log(n) \ldots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$, respectively; that is, in almost the same time as if n and the relevant bit positions were known.

In this work, we prove the first, almost matching, lower bounds for this setting. For LeadingOnes, we show that, for every $s \in \mathbb{N}$, the (1 + 1) EA with any mutation operator treating zeros and ones equally has an expected run time of $\omega(n^2 \log(n) \log\log(n) \ldots \log^{(s)}(n))$ when facing problem size n. Aiming at closing the small remaining gap, we realize that, quite surprisingly, there is no asymptotically best performance. 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.

CCS CONCEPTS

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

KEYWORDS

evolutionary algorithm, optimization under uncertainty, run time analysis, theory; unknown solution length

ACM Reference format:
Benjamin Doerr, Carola Doerr, and Timo Kötzing. 2017. Unknown Solution Length Problems With No Asymptotically Optimal Run Time. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071233

1 INTRODUCTION

We regard how well one can optimize small problems that are hidden in larger representations. Our aim is to design algorithms that on each such small problem have a performance comparable to the one that could be obtained with similar algorithms when the location of the small problem would be known. This line of research was started in [3] and extended in [4].

To model this kind of problem, 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 $I \subseteq [1..N] := \{1, \ldots, N\}$ 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 [3]; it is motivated there by applications in the design of testing sequences for finite state machines [7]. For the initial segment uncertainty model, evolutionary algorithms employing *position-dependent mutation rates* have been shown to give a good performance [3, 4].
(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 [3] whether there are evolutionary algorithms that can deal with this stronger type of uncertainty. This was answered affirmatively in [4], where it was shown that *choosing the mutation rate in each iteration randomly* according to a suitable distribution can give a good performance.

The results obtained in [3, 4], roughly speaking, show the following. The (1 + 1) EA with suitable position-dependent mutation rates or suitable random mutation rates finds the optimum of any OneMax or LeadingOnes problem of (unknown) length n in the initial segment model in an expected optimization time (number of fitness evaluations) of $O(n(\log(n))^{2+\varepsilon})$ and $O(n^2(\log(n))^{1+\varepsilon})$, respectively. This is only by a factor of $O((\log(n))^{1+\varepsilon})$ slower than the performance of the (1 + 1) EA on OneMax or LeadingOnes when the relevant bits are known to the algorithms. The same results hold in the unrestricted uncertainty model when the (1+1) EA with a suitable random mutation rate is used. Note 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.

These results show that the same performance can be achieved in the initial segment uncertainty model (using either position-dependent or random mutation rates) and the unrestricted model (using random mutation rates). Since the unrestricted model contains the initial segment model as a very small subcase (one OneMax and LeadingOnes instance of size n instead of $\binom{N}{n} \geq (N/en)^n$ OneMax and $n!\binom{N}{n}$ LeadingOnes instances), 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.

In this work, we show the first non-trivial lower bounds for unknown solution length problems. With the exception of the optimization of OneMax via position-dependent mutation rates in the initial segment uncertainty model, our bounds show that the run time analyses in [4] are very tight (e.g., up to an $O(\log^{(s)}(n))$ factor for arbitrary $s \in \mathbb{N}$; here we denote by $\log^{(s)}$ the s-fold iterated logarithm of 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 $T : \mathbb{N} \to \mathbb{N}$ such that there is an algorithm solving LeadingOnes instances with unknown length n in expected time at most $T(n)$, there is a function $T' : \mathbb{N} \to \mathbb{N}$ 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.

2 PROBLEM SETTING, ALGORITHMS, AND SUMMABLE SEQUENCES

We use this section to fix the notation used in the remainder, to make precise the models of uncertainty, to recall the definitions of the algorithms regarded, and to recall a few basic facts about summable sequences.

2.1 Basic Notation, Summability, OneMax, and LeadingOnes

The basic notation we use is standard. We write $\mathbb{N}$ to denote the positive integers. We write $[a..b]$ to denote the set of integers not less than a and not larger then b. We abbreviate $[n] := [1..n]$.

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

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

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

A *sequence* $\vec{p}$ is a mapping $\vec{p} : \mathbb{N} \to \mathbb{R}$, 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 well-known fitness functions 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, $\textsc{Om}(x) := \sum_{i=1}^n x_i$, and $\textsc{Lo}(x) := \max\{i \in [0..n] \mid \forall j \leq i : x_j = 1\}$.

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 $\tilde{f} : \{0,1\}^I \to \mathbb{R}$.

It turns out that the dimension N of the large search space is not very relevant for our results. For this reason, as in previous works, 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 outer 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, $f(x) = \max\{i \in [0..n] \mid \forall j \leq i : x_j = 1\}$.

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 $\sigma : [1..n] \to I$ such that $f(x) = \max\{i \in [0..n] \mid \forall j \leq i : x_{\sigma(j)} = 1\}$.

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 first idea, which makes sense only for the initial segment uncertainty model, is to use *position-dependent* mutation rates [3]. For a given sequence $\vec{p} : \mathbb{N} \to [0,1]$, the $(1+1)$ EA$_{\vec{p}}$ (cf. Algorithm 1) is a standard $(1+1)$ EA except that in the mutation step the offspring is generated from flipping each bit position i independently with position-dependent probability p_i.

Algorithm 1: The $(1+1)$ EA$_{\vec{p}}$ with position dependent mutation rates $(p_i)_{i\in\mathbb{N}}$ to maximize a pseudo-Boolean function $f : \{0,1\}^{\mathbb{N}} \to \mathbb{R}$ 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 $t = 1, 2, 3, \ldots$ do
3     $y \leftarrow x$;
4     for $i \in \mathbb{N}$ do with probability $p_i$ set $y_i \leftarrow 1 - x_i$;
5     compute $f(y)$;
6     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 [4]. 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 2 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.

In this work, as already in [4], we will mostly work with distributions Q that are concentrated 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 2: The $(1+1)$ EA$_Q$ with mutation rate distribution Q maximizing a pseudo-Boolean function $f : \{0,1\}^{\mathbb{N}} \to \mathbb{R}$ 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 $t = 1, 2, 3, \ldots$ do
3     $y \leftarrow x$;
4     Sample $p_t$ from $Q$;
5     for $i \in \mathbb{N}$ do with probability $p_t$ set $y_i \leftarrow 1 - x_i$;
6     compute $f(y)$;
7     if $f(y) \geq f(x)$ then $x \leftarrow y$;
```

2.4 Previous Results

The research question how to deal with unknown solution lengths was initiated in [3] and was motivated with the problem of finding unique input-output sequences. [3] only regards the initial segment uncertainty model. In addition, it is assumed that the unknown solution length n follows a truncated geometric distribution with mean $\Theta(q)$. In this situation, it is shown that the $(1+1)$ EA$_{\vec{p}}$ with $p_i = 1/(i+1)$ gives an expected run time of $O(q^{-2}\log N)$ for a OneMax instance with unknown solution length and of $O(q^{-3})$ for such a LeadingOnes instance. Here N is again the dimension of the outer search space and the expectation is taken in the product space of random instance length and random decisions of the algorithm. In [4], these expected run times were improved to $O(q^{-1}\log(q^{-1}))$ and $O(q^{-2})$, respectively.

However, it was also noted that the truncated geometric distribution is relatively strongly concentrated, so the assumption on the unknown solution length to follow such a distribution might be too strong in applications. Therefore, in [4] we adopted a worst-case view, that is, we tried to find algorithms that for each possible length n give a performance close to the performance that the $(1+1)$ EA on instances of this length has. Interestingly, this is possible.

In the initial segment uncertainty model, when taking a position dependent mutation rate $\vec{p}$ with $p_i = \Theta(1/(i(\log(i))^{1+\varepsilon}))$ and the implicit constant small enough, then the $(1+1)$ EA$_{\vec{p}}$ finds the optimum of a OneMax instance of length n in time $O(n(\log(n))^{2+\varepsilon})$ and solve the LeadingOnes instance of length n in time $O(n^2(\log(n))^{1+\varepsilon})$. Note that this is only a factor of $O((\log(n))^{1+\varepsilon})$ slower than the run times on instances of known length n.

These results can be improved by taking a sequence $\vec{p}$ that is summable, but decreases slower. If for some $s \in \mathbb{N}$ we have $p_i^{(s)} = \Theta(1/i\log(i)\log^{(2)}(i)\ldots(\log^{(s)}(i))^{1+\varepsilon})$, again with the implicit constant small enough, then these run times improve to $O(n(\log(n))^2\log^{(2)}(n)\ldots\log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$ and $O(n^2\log(n)\log^{(2)}(n)\ldots\log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$. For the most general result of [4], see Theorem 2.1 below.

In [3], the question was raised whether one can also reasonably well deal with problems in which not only the solution length, but also the relevant bit positions are unknown. This was answered in [4] by proposing the $(1+1)$ EA$_Q$ algorithm, which uses a random mutation rate in each iteration. When using a mutation rate of $1/i$ with probability $\Theta(1/(i \log i))$, here the implicit constant is determined by the normalization constraint that the probabilities have to add up to one, then any OneMax instance hidden on any n bits is optimized in time $O(n(\log(n))^{2+\varepsilon})$. Similarly, any LeadingOnes instance of length n, hidden on arbitrary n bits and in any permuted way, is optimized in time $O(n^2(\log(n))^{1+\varepsilon})$. Note that this uses the mutation rate distribution $Q(\vec{p})$ with $\vec{p}$ as above. If we use $Q(\vec{p}^{(s)})$ as defined above, we obtain the improved run times given above, now again for instances hidden on any n bits and in any permutation. These results raise the question to what extend the formally much harder problem of unrestricted uncertainty is algorithmically harder, a question we try to answer in the following.

The precise result shown in [4] is the following.

Theorem 2.1 ([4]). *Let $\vec{p} = (p_n)_{n\in\mathbb{N}}$ be a monotonically decreasing summable sequence. Let $\Sigma := \sum_{i=1}^{\infty} p_i$.*

(1) Assume that $\Sigma < 1$. Then the expected run time of the $(1+1)$ EA$_{\vec{p}}$ on the OneMax instance of length n in the initial segment uncertainty model is at most $\log n/(p_n(1-\Sigma)) = O(\log n/p_n)$. For LeadingOnes, the same run time is at most $n/(p_n(1-\Sigma)) = O(n/p_n)$.

(2) Let $Q := Q(\vec{p})$ be the mutation rate distribution choosing the uniform bit flip probability $1/i$ with probability p_i/Σ. Then the expected run time of the $(1+1)$ EA$_Q$ on a OneMax instance of length n in the unrestricted uncertainty model is $O(\log n/p_{2n})$. For arbitrarily permuted LeadingOnes instances, this run time is $O(n/p_{2n})$.

2.5 Our Results

The similar performance which the two discussed algorithms display in the two very different uncertainty models raises the desire for lower bounds, which ideally answer the question if the weaker initial segment model allows for a better performance than the unrestricted model or not.

For the LeadingOnes problem, we answer this question with surprising precision: The $(1+1)$ EA$_{\vec{p}}$ and the $(1+1)$ EA$_Q$ have exactly the same performance in the initial segment model. For any $\vec{p}$ there is a Q such that the $(1+1)$ EA$_Q$ for all $n \in \mathbb{N}$ solves the length-n LeadingOnes instance with exactly the same run time distribution as the $(1+1)$ EA$_{\vec{p}}$, and vice versa. Since the $(1+1)$ EA$_Q$ treats all bit positions identically, it has the same performance on the unique LeadingOnes instance in the initial segment model as it has on all LeadingOnes instances in the unrestricted model. Consequently, no $(1+1)$ EA$_{\vec{p}}$ can have a performance in the initial segment model that is superior to a performance achievable with an $(1+1)$ EA$_Q$ in the unrestricted model.

We show further that this equivalence of the algorithms extends to a very general class of algorithms, which is a superclass of the previous two. Consider any probability distribution $\mathbb{P}$ on the subsets of $\mathbb{N}$. Let $(1+1)$ EA$_{\mathbb{P}}$ be the $(1+1)$ EA which in each iteration samples from $\mathbb{P}$ a set $X \subseteq \mathbb{N}$ of bit positions and then creates the offspring by flipping exactly these bits. The class of $(1+1)$ EA$_{\mathbb{P}}$ algorithms consists of all variants of the $(1+1)$ EA which use a static mutation operator choosing the bits to flip without regarding the parent individual. We show that for any of these algorithms there is a $(1+1)$ EA$_{\vec{p}}$ and a $(1+1)$ EA$_Q$ which have exactly the same performance on LeadingOnes in the initial segment model. Consequently, to obtain a better performing algorithm than the three above, one would have to use a dynamic algorithm or one that chooses the mutation operator depending on the parent individual (this can obviously give an improvement, e.g., by using an operator that prefers flipping zeros to ones, but this seems very problem-specific).

The equivalence of these algorithms classes allows us to prove lower bounds for all of them simultaneously. 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 performance of any of these algorithms on LeadingOnes instances of length n. This nearly matches the $O(n^2 \log(n) \log^{(2)}(n) \dots \log^{(s-1)}(n)(\log^{(s)}(n))^{1+\varepsilon})$, $s \in \mathbb{N}$, upper bounds discussed in the previous subsection.

We have mildly tighter bounds that we will discuss in the following subsection, but we do not have asymptotically tight bounds. However, we have a good excuse for this. To our surprise, we observe that for the LeadingOnes problem there is no choice of the algorithm parameters giving an asymptotically best performance. Even stronger, there is no asymptotically best performance at all for these algorithms. For any $(1+1)$ EA$_{\vec{p}}$ solving the initial segment LeadingOnes problem of length n in expected time $T(n)$, there is another $(1+1)$ EA$_{\vec{p}}$ with run time bound $T'(n)$ such that $T' = o(T)$. By the above equivalences, this holds analogously for the $(1+1)$ EA$_Q$ and $(1+1)$ EA$_{\mathbb{P}}$ algorithm classes.

For the OneMax problem all bit positions are treated symmetrically. We were able to exploit this and the fact that the unrestricted uncertainty model treats positions symmetrically in order to show a lower bound for this model. However, this approach does not carry over to the initial segment model, since here position dependent mutation probabilities come into play; we have to leave open what the exact bounds for this setting are. Our bound for the unrestricted uncertainty model matches the upper bound in a similar manner as for LeadingOnes, that is, no $(1+1)$ EA$_Q$ can have an expected run time of $O(n(\log(n))^2 \log^{(2)}(n) \dots \log^{(s)}(n))$ on all length-n OneMax instances.

2.6 Summable Sequences

Theorem 2.1 shows that the bounds for the $(1+1)$ EA$_{\vec{p}}$ and the $(1+1)$ EA$_Q$ proven in [4] all crucially depend on the underlying summable sequence $\vec{p}$. Such sequences will also play a crucial role in the remainder of this work. We therefore use this section to recall and prove a few basic facts about these sequences. We also recall and extend particular sequences regarded in [4].

We first note that the run time guarantees in Theorem 2.1 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. Such constructions are well known in the mathematics literature, our construction follows that of [1].

Theorem 2.2 (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 $\vec{q} = \omega(\vec{p})$.*

A corresponding statement for non-summable sequences exists, i.e., for every non-summable sequence $\vec{p}$ there exists another non-summable sequence $\vec{q}$ such that $\vec{q} = o(\vec{p})$. In both cases the sequence $\vec{q}$ can be chosen in a way that the expected run time of the corresponding $(1+1)$ EA$_{\vec{p}}$ and $(1+1)$ EA$_{\vec{q}}$ satisfy $\mathrm{E}[T_{\vec{q}}] = o(\mathrm{E}[T_{\vec{p}}])$ Since this result also follows from Theorem 3.8 below, we do not prove these statements explicitly.

2.6.1 Sequences Based on Iterated Logarithms. In [4], sequences of iterated logarithms were shown to yield good performance guarantees for the $(1+1)$ EA$_{\vec{p}}$ and the $(1+1)$ EA$_Q$ on LeadingOnes and OneMax. We present here a generalized version of these sequences, which will be used later on to derive lower bounds for the performance of best-possible $(1+1)$ EA- and RLS-type algorithms in unknown solution length environments.

For every $b > 1$, $\varepsilon \geq 0$, $r > 0$, and all positive integers s set

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

It was noted in [4] and shown, for example, in Hardy [6, 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. Hardy also mentions that for all s 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, we easily get the following result.

Lemma 2.3. *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., $1/p_n = \omega(n \log_b(n) \log_b^{(2)}(n) \ldots \log_b^{(s)}(n))$.*

A natural question arising from the construction in (1) 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

$$p_b^\infty(n) := 1/\left(n \prod_{j=1}^{\infty} \log_b^{(j)}(n)\right). \tag{2}$$

It is known in the mathematics literature that the summability of the sequence $(p_b^\infty(n))_{n\in\mathbb{N}}$ crucially depends on the base b of the logarithm. Indeed, using Cauchy's condensation test, one can show the following.

Lemma 2.4. *The sequence $(p_b^\infty(n))_{n\in\mathbb{N}}$ is summable if and only if $b < e := \exp(1)$.*

Plugging these sequences $(p_b^\infty(n))_{n\in\mathbb{N}}$—or, more precisely, in case of the $(1+1)$ EA$_{\vec{p}}$, the sequences $(p_b^\infty(n)/\sum_{i\in\mathbb{N}} p_b^\infty(i))_{n\in\mathbb{N}}$—into Theorem 2.1, we obtain the following results.

Corollary 2.5. *Let $b < e$. The expected run time of the $(1+1)$ EA$_{\vec{p}}$ with $\vec{p} = (p_b^\infty(n))_{n\in\mathbb{N}}$ on the OneMax instance of length n in the initial segment uncertainty model is $O\left(\log(n)/p_b^\infty(n)\right) = O(n(\log_b(n))^2 \log_b^{(2)}(n)\ldots)$ and it is $O\left(n/p_b^\infty(n)\right) = O(n^2 \log_b(n) \log_b^{(2)}(n)\ldots)$ on LeadingOnes.*

Furthermore, for $Q := Q(\vec{p})$ as in Theorem 2.1, the expected run time of the $(1+1)$ EA$_Q$ on a OneMax instance of length n in the unrestricted uncertainty model is $O\left(\log(n)/p_b^\infty(2n)\right) = O(2n(\log_b(2n))^2 \log_b^{(2)}(2n)\ldots)$ and $O\left(n/p_b^\infty(2n)\right) = O(2n^2 \log_b(2n) \log_b^{(2)}(2n)\ldots)$ on LeadingOnes.

From Lemma 2.4 we also get the following result.

Corollary 2.6. *Let $b \geq e$. For every summable sequence $\vec{p}$ it holds that $1/p_n = \omega(1/p_b^\infty(n)) = \omega(n \log_b(n) \log_b^{(2)}(n)\ldots)$.*

2.7 Other Useful Tools

We recall the following well-known lemma, which will be needed several times in the subsequent sections.

Lemma 2.7. *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 $A = o(B)$.*

3 LEADINGONES

In this section, we prove very sharp lower bounds for the optimization of the LeadingOnes function in the two uncertainty models via the two algorithms 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. We then show that all three lead to the same run time profiles for the initial segment model. 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 on the performance of the $(1+1)$ EA$_{\vec{p}}$ in the initial segment uncertainty model. We show that for all $s \in \mathbb{N}$, $\Omega(n^2 \log(n) \log^{(2)}(n) \ldots \log^{(s)}(n))$ is a lower bound for the expected run time of the LeadingOnes instance of length n. By carefully constructing for each $\vec{p}$ a $\vec{q}$ 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$_Q$ class).

3.1 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$_Q$, 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 $a_1, \ldots, a_n \in \{0,1\}$, we call the set $C(a_1, \ldots, a_n) := \{x \in \Omega \mid \forall i \in [n] : x_i = a_i\}$ 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 $y : I \to \{0,1\}$. 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 $(\Omega, \mathcal{A})$ 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 $(1+1)$ 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 with 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 $(1+1)$ EA$_{\vec{p}}$ and $(1+1)$ EA$_Q$ classes are subclasses of the class of all algorithms $(1+1)$ EA$_{\mathbb{P}}$.

- Let $\vec{p} = (p_i)_{i \in \mathbb{N}}$ be any sequence in $[0,1]$. Then the $(1+1)$ 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 $(1+1)$ EA$_{\vec{p}}$ and the $(1+1)$ EA$_{\mathbb{P}}$ are the same algorithm.
- Let Q be a discrete distribution on $[0,1]$. Then the $(1+1)$ 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 $(1+1)$ EA$_Q$ and the $(1+1)$ EA$_{\mathbb{P}}$ are the same algorithm.

For this reason, any lower bound for the best performance achievable with an algorithm from the class $(1+1)$ EA$_{\mathbb{P}}$ immediately carries over to the subclasses $(1+1)$ EA$_{\vec{p}}$ and $(1+1)$ EA$_Q$. Now that we regard the same class $(1+1)$ 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 $(1+1)$ 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.

Generalized Randomized Local Search on Infinite Bit Strings: RLS$_{\vec{p}}$. Let $p_1, p_2, \dots$ be non-negative numbers with $\sum_{n \in \mathbb{N}} p_n \le 1$. Then the algorithm RLS$_{\vec{p}}$ is a special case of the $(1+1)$ EA$_{\mathbb{P}}$ that flips exactly the ith bit with probability p_i and flips no bit with probability $1 - \sum_{n \in \mathbb{N}} p_n$. More formally, RLS$_{\vec{p}}$ equals $(1+1)$ EA$_{\mathbb{P}}$ for the measure $\mathbb{P}$ defined by $\mathbb{P}[C(a_1,\dots,a_n)] :=$
$$\begin{cases} p_i, & \text{if } a_i = 1 \text{ and } a_j = 0 \text{ for } j \in [n] \setminus \{i\}, \\ 0, & \text{if there are } i, j \in [n] \text{ with } i \ne j \text{ and } a_i = 1 = a_j, \\ 1 - \sum_{n \in \mathbb{N}} p_n, & a_j = 0 \text{ for } j \in [n]. \end{cases}$$

Lemma 3.1. *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 $\sum_{n \in \mathbb{N}} p_n \le 1$ such that, for LeadingOnes, the randomized local search algorithm RLS$_{\vec{p}}$ and $(1+1)$ 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 $(1+1)$ 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 [2, 5, 8] to analyze the optimization of fixed length LeadingOnes instances.

Lemma 3.2. *Consider a run of an $(1+1)$ 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 $(1+1)$ EA$_{\mathbb{P}}$ after the t-th iteration. Then, for any two search points $x, y \in \{0,1\}^n$ with $Lo(x) = Lo(y)$, we have $\Pr[X^{(t)} = x] = \Pr[X^{(t)} = y]$.*

To prove Lemma 3.1, for all $n \in \mathbb{N}$, 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 are disjoint, we have $\sum_{n \in \mathbb{N}} p_n \le 1$. Using Lemma 3.2 one can show, via an inductive proof, that the $(1+1)$ EA$_{\mathbb{P}}$ and RLS$_{\vec{p}}$ have an identical optimization behavior.

We note in the following lemma that, for LeadingOnes, the class of algorithms $(1+1)$ EA$_{\vec{p}}$ regarded in [4] is equally powerful as the class of algorithms RLS$_{\vec{p}}$. Together with Lemma 3.1, this shows that all three classes of algorithms are equally powerful, and in particular, that the subclass $(1+1)$ EA$_{\vec{p}}$ regarded in [4] was of maximal power.

Lemma 3.3. *For any sequence of non-negative numbers $\vec{p} = (p_1, p_2, \dots)$ with $\sum_{n \in \mathbb{N}} p_n \le 1$ 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 $1 - \sum_{j<n} p_j \le p_n$, this also shows that $q_n \le 1$. 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 3.1 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$_{\vec{p}}$. □

An advantage of the class $\mathrm{RLS}_{\vec{p}}$ is that the expected run times are easy to compute. The following result will be used in Lemma 3.6 to bound the run time of the $(1+1)$ $\mathrm{EA}_{\vec{p}}$.

Lemma 3.4. *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 $\frac{1}{2}\sum_{i=1}^{n} \frac{1}{p_i}$.*

The above insight allows us to completely describe the possible performances of $(1+1)$ $\mathrm{EA}_{\mathbb{P}}$, $(1+1)$ $\mathrm{EA}_{\vec{p}}$, and $\mathrm{RLS}_{\vec{p}}$ 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 $E : \mathbb{N} \to \mathbb{R}$ 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 3.5 (characterization of initial segment run time profiles). *Let $E : \mathbb{N} \to \mathbb{R}$. The following four properties are equivalent.*

(i) *There is a sequence $q : \mathbb{N} \to [0,1]$ 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 $\vec{p} : \mathbb{N} \to [0,1]$.*
(iv) *E is the initial segment run time profile of $RLS_{\vec{p}}$ for some $\vec{p} : \mathbb{N} \to [0,1]$ with $\sum_{n=1}^{\infty} p_n \leq 1$.*

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

3.2 Computing Concrete Lower Bounds for LeadingOnes

In the remainder of this section, we use Theorem 3.5 to compute concrete run time bounds for the $(1+1)$ $\mathrm{EA}_{\vec{p}}$. These bounds then immediately yield performance bounds for all the other settings covered by Theorem 3.5. We note that some (but not all) of the subsequent run time results can alternatively be obtained by analyzing the summable sequences described in Theorem 3.5.(i). Such an approach would, however, not give any additional insights into the sequences $\vec{p}$ underlying the $(1+1)$ $\mathrm{EA}_{\vec{p}}$ with initial segment run time profile E, thus motivating us to study this algorithm directly.

3.2.1 Initial Segment Run Time Profile of the $(1+1)$ $EA_{\vec{p}}$. Lemma 3.1 can be used to precisely determine the initial segment run time profile of the $(1+1)$ $\mathrm{EA}_{\vec{p}}$. From this we can easily compute the following upper and lower bounds. This result extends Theorem 15 of [4] where the upper bound mentioned in Theorem 2.1 is given for the case that $\vec{p}$ is monotonically decreasing and summable.

Lemma 3.6. *Let $\vec{p} = (p_n)_{n\in\mathbb{N}}$ be a sequence of positive terms $p_n > 0$. The expected optimization time $\mathrm{E}[T_{\vec{p}}(n)]$ of the $(1+1)$ $EA_{\vec{p}}$ on the LeadingOnes instance of length n in the initial segment uncertainty model can be bounded by*

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

where, for all n, $S_n := \sum_{i=1}^{n} p_i$ denotes the n-th partial sum of $\vec{p}$ and $S_0 := 0$.

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

From Lemma 3.6 it is not difficult to 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 3.7. *Let $\vec{p} : \mathbb{N} \to [0,1]$, $0 < c$, and $\vec{q} = (q_i)_{i\in\mathbb{N}}$ with $q_i := c \cdot p_i \in [0, 1/2]$. Then there exists a constant $C > 0$ such that, for all $n \in \mathbb{N}$, $\mathrm{E}[T_{\vec{q}}(n)] \leq C\,\mathrm{E}[T_{\vec{p}}(n)]$. When $c < 1$, C can be chosen as $1/c$ while for $c > 1$ we may chose $C := \exp(2cp)/c$.*

3.2.2 No Separation Between Summable and Non-Summable Sequence. Given the bounds proven so far, one may be tempted to believe the best performing $(1+1)$ $\mathrm{EA}_{\vec{p}}$ 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 $(1+1)$ $\mathrm{EA}_{\vec{p}}$ 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 $(1+1)$ $\mathrm{EA}_{\vec{q}}$ on LeadingOnes is greater than B while for some summable sequence $\vec{p}$ the expected run time of the $(1+1)$ $\mathrm{EA}_{\vec{p}}$ 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 $(1+1)$ $\mathrm{EA}_{\vec{q}}$ on LeadingOnes is of strictly smaller order than that of the $(1+1)$ $\mathrm{EA}_{\vec{p}}$. Less surprisingly, the converse also holds.

Theorem 3.8. *(A) For every summable sequence $\vec{p}$ of 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 $\mathrm{E}[T_{\vec{q}}] = o(\mathrm{E}[T_{\vec{p}}])$.*

(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 $\mathrm{E}[T_{\vec{p}}] = o(\mathrm{E}[T_{\vec{q}}])$ for $T_{\vec{p}}$ and $T_{\vec{q}}$ defined as above.

To prove Theorem 3.8.(A), 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 $n \in \mathbb{N}$, $p_n \leq \sum_{j>n} p_j$. The sequence $\vec{q}$ of the statement is defined by setting, for every $i \in \mathbb{N}$, $r_i := \sum_{j\geq i} p_j$ and

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

By bounding $q_i \geq \int_{r_{i+1}}^{r_i} \frac{1}{x\ln(1/x)}\mathrm{d}x$, one can show that $\vec{q}$ is not summable. The run time statement follows from applying Lemma 3.6 and Lemma 2.7 to $\vec{p}$ and $\vec{q}$ and the bounds on the run times, respectively.

Theorem 3.8.(B) can be verified by setting $p_1 := q_1$ and, for all $n \geq 2$, $p_n := q_n/S_n^2$, where $S_n := \sum_{i=1}^{n} q_i$. It is then not difficult to show that $\sum_{i=1}^{n} p_i \leq q_1 + \int_{S_1}^{S_n} \frac{1}{x^2}\mathrm{d}x$, proving that $\vec{p}$ is indeed summable. Again we apply Lemma 3.6 and Lemma 2.7 to obtain the claimed run time statement.

3.2.3 Limiting Behavior. The previous results show, in particular, that there is no best possible $(1+1)$ $\mathrm{EA}_{\vec{p}}$ for LeadingOnes: whatever the sequence underlying the $(1+1)$ $\mathrm{EA}_{\vec{p}}$ 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 3.9. *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 $\mathrm{E}[T_{\vec{p}}(n)] = \omega(n/p_b^{s,0}(n)) = \omega(n^2 \log_b(n) \log_b^{(2)}(n) \dots \log_b^{(s)}(n))$.*

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

This theorem can be proven by first using Theorem 3.8 to argue that one can assume $\vec{p}$ to be summable and then using Lemma 3.6 to see that $\mathrm{E}[T_{\vec{p}}(n)] = \Omega(\sum_{i=1}^{n} \frac{1}{p_i})$. By Lemma 2.3 it holds that $1/p_i = \omega(1/p_b^{s,0}(i)) = \omega(i \log_b(i) \log_b^{(2)}(i) \dots \log_b^{(s)}(i))$. An application of Lemma 2.7 then concludes the proof. The second statement follow similarly; using Corollary 2.6 instead of Lemma 2.3.

4 ONEMAX

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

Theorem 4.1. *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.

From Theorem 4.1, Lemma 2.3, and Corollary 2.6 we obtain the following result.

Corollary 4.2. *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.

5 SUMMARY AND OUTLOOK

In this work, we proved the first non-trivial lower bounds for the $(1+1)$ $\mathrm{EA}_{\vec{p}}$ and $(1+1)$ EA_Q algorithm classes that were previously found useful to cope with unknown solution length scenarios. One particularly noteworthy result is 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 $(1+1)$ $\mathrm{EA}_{\vec{p}}$ class on the OneMax function in the initial segment uncertainty model, a problem that we unfortunately could not solve.

Acknowledgments. This research has been done while Timo Kötzing was an invited researcher of École Polytechnique. It was supported in part by the German Research Foundation under Grant FR 2988 (TOSU) and by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH.

REFERENCES

[1] M. J. Ash. Neither a worst convergent series nor a best divergent series exists. *College Mathematics Journal*, 28:296–297, 1997.
[2] S. Böttcher, B. Doerr, and F. Neumann. Optimal fixed and adaptive mutation rates for the LeadingOnes problem. In *Proc. of Parallel Problem Solving from Nature (PPSN'10)*, pages 1–10. Springer, 2010.
[3] S. Cathabard, P. K. Lehre, and X. Yao. Non-uniform mutation rates for problems with unknown solution lengths. In *Proc. of Foundations of Genetic Algorithms (FOGA'11)*, pages 173–180. ACM, 2011.
[4] B. Doerr, C. Doerr, and T. Kötzing. Solving problems with unknown solution length at (almost) no extra cost. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'15)*, pages 831–838. ACM, 2015.
[5] B. Doerr, T. Jansen, C. Witt, and C. Zarges. A method to derive fixed budget results from expected optimisation times. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'13)*, pages 1581–1588. ACM, 2013.
[6] G. H. Hardy. *Orders of infinity.* Cambridge University Press, 1910.
[7] P. K. Lehre and X. Yao. Runtime analysis of the $(1+1)$ EA on computing unique input output sequences. *Information Sciences*, 259:510–531, 2014. An extended abstract appeared in the *Proc. of IEEE Congress on Evolutionary Computation (CEC'07)*.
[8] D. Sudholt. A new method for lower bounds on the running time of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 17:418–435, 2013.