Runtime Analysis of the Univariate Marginal Distribution Algorithm under Low Selective Pressure and Prior Noise[☆]

Per Kristian Lehre & Phan Trung Hai Nguyen

School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K.

Abstract

We perform a rigorous runtime analysis for the Univariate Marginal Distribution Algorithm on the LeadingOnes function, a well-known benchmark function in the theory community of evolutionary computation with a high correlation between decision variables. For a problem instance of size n, the currently best known upper bound on the expected runtime is $\mathcal{O}\big(n\lambda \log \lambda + n^2\big)$ (Dang and Lehre, GECCO 2015), while a lower bound necessary to understand how the algorithm copes with variable dependencies is still missing. Motivated by this, we show that the algorithm requires a $e^{\Omega(\mu)}$ runtime with high probability and in expectation if the selective pressure is low; otherwise, we obtain a lower bound of $\Omega(\frac{n\lambda}{\log(\lambda-\mu)})$ on the expected runtime. Furthermore, we for the first time consider the algorithm on the function under a prior noise model and obtain an $\mathcal{O}\big(n^2\big)$ expected runtime for the optimal parameter settings. In the end, our theoretical results are accompanied by empirical findings, not only matching with rigorous analyses but also providing new insights into the behaviour of the algorithm.

Keywords: Univariate marginal distribution algorithm, leadingones, noisy optimisation, running time analysis, theory

1. Introduction

Estimation of Distribution Algorithms (EDAs) [1, 2, 3] are black-box optimisation methods that search for optimal solutions by building and sampling from probabilistic models. They are known by various other names, including probabilistic model-building genetic algorithm or iterated density estimation algorithms. Unlike traditional evolutionary algorithms (EAs), which use standard genetic operators such as mutation and crossover to create variations, EDAs, on the other hand, achieve it via model building and model sampling. The workflow of EDAs is an iterative process. The starting model is a uniform distribution over

[☆]Preliminary version of this work will appear in the Proceedings of the 2019 Genetic and Evolutionary Computation Conference (GECCO 2019), Prague, Czech Republic.

the search space, from which the initial population of λ individuals is sampled. The fitness function then scores each individual, and the algorithm selects the $\mu < \lambda$ fittest individuals to update the model. The procedure is repeated many times and terminates when a threshold on the number of iterations is exceeded or a solution of good quality is obtained [4, 5]. We call the value λ the offspring population size, while the value μ is known as the parent population size of the algorithms.

Several EDAs have been proposed over the last decades. They differ in how they learn the variable interactions and build/update the probabilistic models over iterations. In general, EDAs can be categorised into two classes: univariate and multivariate. Univariate EDAs, which take advantage of first-order statistics (i.e. the means while assuming variable independence), usually represent the model as a probability vector, and individuals are sampled independently and identically from a product distribution. Typical EDAs in this class are the Univariate Marginal Distribution Algorithm (UMDA [1]), the compact Genetic Algorithm (cGA [6]) and the Population-Based Incremental Learning (PBIL [7]). Some ant colony optimisation algorithms like the λ-MMAS [8] can also be cast into this framework (also called n-Bernoulli-λ-EDA [9]). In contrast, multivariate EDAs apply statistics of order two or more to capture the underlying structures of the addressed problems. This paper focuses on univariate EDAs on discrete optimisation, and for that reason we refer the interested readers to [5, 10] for other EDAs on a continuous domain.

In the theory community, researchers perform rigorous analyses to gain insights into the runtime (synonymously, optimisation time), which is defined as the number of function evaluations of the algorithm until an optimal solution is found for the first time. In other words, theoretical work usually addresses the unlimited case when we consider the run of the algorithm as an infinite process. Considering function evaluations is motivated by the fact that these are often the most expensive operations, whereas other operations can usually be executed very quickly. Steady-state algorithms like the simple $(1+1)$ EA have the number of function evaluations equal the number of iterations, whereas for univariate EDAs the former is larger by a factor of the offspring population size λ than the latter. Runtime analyses give performance guarantee of the algorithms for a wide range of problem instance sizes. Due to the complex interplay of variables and limitations on the state-of-the-art tools in algorithmics, runtime analysis is often performed on simple (artificial) problems such as OneMax, LeadingOnes and BinVal, hoping that this provides valuable insights into the development of new techniques for analysing search heuristics and the behaviour of such algorithms on easy parts of more complex problem spaces [11]. By 2015, there had been a handful of runtime results for EDAs [12, 9], since then this class of algorithms have constantly drawn more attention from the theory community as evidenced in the increasing number of EDA-related publications recently [9, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24].

Droste [25] in 2006 performed the first rigorous analysis of the cGA, which works on a population of two individuals and updates the probabilistic model additively via a parameter K (also referred to as the hypothetical population

size of a genetic algorithm that the cGA is supposed to model) and obtained a lower bound $\Omega(K\sqrt{n}) = \Omega(n^{1+\varepsilon})$ on the expected runtime for the cGA on any pseudo-Boolean function for any small constant $\varepsilon > 0$. Each component in the probabilistic model (also called marginal) of the cGA considered in [25] is allowed to reach the extreme values zero and one. Such an algorithm is referred to as an EDA without margins, since in contrast it is possible to reinforce the margins $[1/n, 1 - 1/n]$ (sometimes called borders) to keep it away from the extreme probabilities. Friedrich *et al.* [9], on consideration of univariate EDAs (without borders), conjectured that the cGA might not optimise the LeadingOnes function efficiently (i.e., within an $\mathcal{O}(n^2)$ expected runtime) as the algorithm is balanced but not stable. They then proposed a so-called stable cGA to overcome this, which requires an $\mathcal{O}(n \log n)$ expected runtime on the same function. Motivated by the same work, Doerr *et al.* [23] recently developed the significant cGA, which uses memory to determine when the marginals should be set to a value in the set $\{1/n, 1/2, 1-1/n\}$, and surprisingly the algorithm optimises the OneMax and LeadingOnes functions using an $\mathcal{O}(n \log n)$ expected runtime.

The UMDA is probably the most famous univariate EDA. In each so-called iteration, the algorithm updates each marginal to the corresponding frequency of 1s among the μ fittest individuals. In 2015, Dang and Lehre [12] via the level-based theorem [26] obtained an upper bound of $\mathcal{O}(n\lambda \log \lambda + n^2)$ on the expected runtime for the algorithm (with margins) on the LeadingOnes function when the offspring population size is $\lambda = \Omega(\log n)$ and the selective pressure $\mu/\lambda \leq 1/(1+\delta)e$ for any constant $\delta > 0$. For the optimal setting $\lambda = \mathcal{O}(n/\log n)$, the above bound becomes $\mathcal{O}(n^2)$, which emphasises the need of borders for the algorithm to optimise the LeadingOnes function efficiently compared to the findings in [9]. We also note that a generalisation of the UMDA is the PBIL [7], which updates the marginals using a convex combination with a smoothing parameter $\eta \in [0, 1]$ between the current marginals and the frequencies of 1s among the μ fittest individuals in the current population. Wu *et al.* [15] performed the first rigorous runtime analysis of the algorithm, where they argued that for a sufficiently large population size, the algorithm can avoid making wrong decisions early even when the smoothing parameter is large. They also showed an upper bound $\mathcal{O}(n^{2+\epsilon})$ on the expected runtime for the PBIL (with margins) on the LeadingOnes function for some small constant $\varepsilon > 0$. The required offspring population size yet still remains large [15]. Very recently, Lehre and Nguyen [20], via the level-based theorem with some additional arguments, obtained an upper bound $\mathcal{O}(n\lambda \log \lambda + n^2)$ on the expected runtime for the offspring population sizes $\lambda = \Omega(\log n)$ and a sufficiently high selective pressure. This result improves the bound in [15] by a factor of $\Theta(n^{\varepsilon})$ for the optimal parameter setting $\lambda = \mathcal{O}(n/\log n)$.

In this paper, we analyse the UMDA in order to, when combining with previous results [12, 19], completes the picture on the runtime of the algorithm on the LeadingOnes function, a widely used benchmark function with a high correlation between variables. We first show that under a low selective pressure the algorithm fails to optimise the function in polynomial runtime with high

probability and in expectation. This result essentially reveals the limitations of probabilistic models based on probability vectors as the algorithm hardly stays in promising states when the selective pressure is not high enough, while the global optimum cannot be sampled with high probability. On the other hand, when the selective pressure is sufficiently high, we obtain a lower bound of $\Omega(\frac{n\lambda}{\log(\lambda-\mu)})$ on the expected runtime for the offspring population sizes $\lambda = \Omega(\log n)$. Moreover, we introduce noise to the LeadingOnes function, where a uniformly chosen bit is flipped with (constant) probability $p < 1$ before evaluating the fitness (also called prior noise). Via the level-based theorem, we show that the expected runtime of the algorithm on the noisy function is still $\mathcal{O}(n^2)$ for an optimal population size $\lambda = \mathcal{O}(n/\log n)$. To the best of our knowledge, this is the first time that the UMDA is rigorously studied in a noisy environment, while the cGA is already considered in [17] under Gaussian posterior noise. Despite the simplicity of the noise model, this can be viewed as the first step towards understanding the behaviour of the algorithm in a noisy environment. In the end, we provide empirical results to support our theoretical analyses and give new insights into the run of the algorithm which the theoretical results do not cover. Moreover, many algorithms similar to the UMDA with a fitness proportional selection are popular in bioinformatics [27], where they relate to the notion of *linkage equilibrium* [28, 29] – a popular model assumption in population genetics. Therefore, studying the UMDA especially in the presence of variable dependence and mild noise solidifies our understanding of population dynamics.

The paper is structured as follows. Section 2 introduces the studied algorithm. Section 3 provides a detailed analysis for the algorithm on the LeadingOnes function in case of low selective pressure, followed by the analysis for a high selective pressure in Section 4. In Section 5, we introduce the LeadingOnes function with prior noise and show an upper bound $\mathcal{O}(n^2)$ on the expected runtime. Section 6 presents an empirical study to complement theoretical results derived earlier. The paper ends in Section 7, where we give our concluding remarks and speak of potential future work.

2. The algorithm

In this section we describe the studied algorithm. Let $\mathcal{X} = \{0,1\}^n$ be a finite binary search space with n dimensions, and each individual in $\mathcal{X}$ is represented as $x = (x_1, x_2, \ldots, x_n)$. The population of λ individuals in iteration t is denoted as $P_t := (x_t^{(1)}, \ldots, x_t^{(\lambda)})$. We consider the maximisation of an objective function $f : \mathcal{X} \to \mathbb{R}$.

The UMDA, defined in Algorithm 1, maintains a probabilistic model that is represented as an n-vector $p_t := (p_{t,1}, \ldots, p_{t,n})$, and each marginal $p_{t,i} \in [0,1]$ for $i \in [n]$ (where $[n] := [1,n] \cap \mathbb{N}$) is the probability of sampling a one at the i-th bit position in the offspring. The joint probability distribution of an individual $x \in \mathcal{X}$ given the current model p_t is formally defined as

$$\Pr(x \mid p_t) = \prod_{i=1}^{n} (p_{t,i})^{x_i} (1 - p_{t,i})^{1-x_i}. \tag{1}$$

The starting model is the uniform distribution $p_0 := (1/2, \ldots, 1/2)$. In an iteration t, the algorithm samples a population P_t of λ individuals, sorts them in descending order according to fitness and then selects the μ fittest individuals to update the model (also called the selected population). Let $X_{t,i}$ denote the number of 1s sampled at bit position $i \in [n]$ in the selected population. The algorithm updates each marginal using $p_{t+1,i} = X_{t,i}/\mu$. Each marginal is also restricted to be within the interval $[1/n, 1 - 1/n]$, where the values $1/n$ and $1 - 1/n$ are called lower and upper border, respectively. We call the ratio $\gamma^* := \mu/\lambda$ the selective pressure of the algorithm.

Algorithm 1: UMDA

```
t ← 0; initialise p_t ← (1/2, 1/2, ..., 1/2)
repeat
    for j = 1, 2, ..., λ do
        sample x_{t,i}^{(j)} ~ Bernoulli(p_{t,i}) for each i ∈ [n]
    sort P_t ← (x_t^{(1)}, x_t^{(2)}, ..., x_t^{(λ)}) such that
      f(x^{(1)}) ≥ f(x^{(2)}) ≥ ... ≥ f(x^{(λ)})
    for i = 1, 2, ..., n do
        p_{t+1,i} ← max{1/n, min{1 − 1/n, X_{t,i}/μ}}
    t ← t + 1
until termination condition is fulfilled
```

3. Low Selective Pressure

Recall that we consider the problem of maximising the number of leading 1s in a bitstring, which is defined by

$$\textsc{LeadingOnes}(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_j.$$

The bits in this particular function are highly correlated, so it is often used to study the ability of EAs to cope with variable dependency [30]. Previous studies [12, 19] showed that the UMDA optimises the function within an $\mathcal{O}(n^2)$ expected time for the optimal offspring population size $\lambda = \mathcal{O}(n/\log n)$.

Before we get to analysing the function, we introduce some notation. Let $C_{t,i}$ for all $i \in [n]$ denote the number of individuals having at least i leading 1s in iteration t, and $D_{t,i}$ is the number of individuals having $i - 1$ leading 1s, followed by a 0 at the block i. For the special case of $i = 1$, $D_{t,i}$ consists of those with zero leading 1s. Furthermore, let $(\mathcal{F}_t)_{t\in\mathbb{N}}$ be a filtration induced from the population $(P_t)_{t\in\mathbb{N}}$.

Once the population has been sampled, the algorithm invokes truncation selection to select the μ fittest individuals to update the probability vector. We

take this μ-cutoff into account by defining a random variable

$$Z_t := \max\{i \in \mathbb{N} : C_{t,i} \geq \mu\},$$

which tells us how many marginals, counting from position one, are set to the upper border $1 - 1/n$ in iteration t. Furthermore, we define another random variable

$$Z_t^* := \max\{i \in \mathbb{N} : C_{t,i} > 0\}$$

to be the number of leading 1s of the fittest individual(s). For readability, we often leave out the indices of random variables like when we write C_t instead of $C_{t,i}$, if values of the indices are clear from the context.

3.1. On the distributions of $C_{t,i}$ and $D_{t,i}$

In order to analyse the distributions of the random variables $C_{t,i}$ and $D_{t,i}$, we shall take an alternative view on the sampling process at an arbitrary bit position $i \in [n]$ in iteration $t \in \mathbb{N}$ via the *principle of deferred decisions* [31]. We imagine that the process samples the values of the first bit for λ individuals. Once this has finished, it moves on to the second bit and so on until the population is sampled. In the end, we will obtain a population that is sorted in descending order according to fitness.

We now look at the first bit in iteration t. The number of 1s sampled in the first bit position follows a binomial distribution with parameters λ and $p_{t,1}$, i.e., $C_{t,1} \sim \mathrm{Bin}(\lambda, p_{t,1})$. Thus, the number of 0s at the first bit position is $D_{t,1} = \lambda - C_{t,1}$.

Having sampled the first bit for λ individuals, and note that the bias due to selection in the second bit position comes into play only if the first bit is 1. If this is the case, then a 1 is more preferred to a 0. The probability of sampling a 1 is $p_{t,2}$; thus, the number of individuals having at least 2 leading 1s is binomially distributed with parameters $C_{t,1}$ and $p_{t,2}$, that is, $C_{t,2} \sim \mathrm{Bin}(C_{t,1}, p_{t,2})$, and the number of 0s equals $D_{t,2} = C_{t,1} - C_{t,2}$. Unlike the first bit position, there are still $D_{t,1}$ remaining individuals, since for these individuals the first bit is a 0, there is no bias between a 1 and a 0. The number of 1s follows a binomial distribution with parameters $D_{t,1}$ (or $\lambda - C_{t,1}$) and $p_{t,2}$.

We can generalise this result for an arbitrary bit position $i \in [n]$. The number of individuals having at least i leading 1s follows a binomial distribution with $C_{t,i-1}$ trials and success probability $p_{t,i}$, i.e., $C_{t,i} \sim \mathrm{Bin}(C_{t,i-1}, p_{t,i})$, and $D_{t,i} = C_{t,i-1} - C_{t,i}$. Furthermore, the number of 1s sampled among the $\lambda - C_{t,i-1}$ remaining individuals is binomially distributed with $\lambda - C_{t,i-1}$ trials and success probability $p_{t,i}$. If we consider the expectations of these random variables, by the tower rule [32] and noting that $p_{t,i}$ is $\mathcal{F}_{t-1}$-measurable, we then get

$$\mathbb{E}[C_{t,i} \mid \mathcal{F}_{t-1}] = \mathbb{E}[\mathbb{E}[C_{t,i} \mid C_{t,i-1}] \mid \mathcal{F}_{t-1}] = \mathbb{E}[C_{t,i-1} \mid \mathcal{F}_{t-1}] \cdot p_{t,i}, \tag{2}$$

and similarly

$$\mathbb{E}[D_{t,i} \mid \mathcal{F}_{t-1}] = \mathbb{E}[C_{t,i-1} \mid \mathcal{F}_{t-1}] \cdot (1 - p_{t,i}). \tag{3}$$

We aim at showing that the UMDA takes exponential time to optimise the LeadingOnes function when the selective pressure is not sufficiently high, as required in [19]. Later analyses are concerned with two intermediate values:

$$\alpha = \alpha(n) := \log(\gamma^*/(1-\delta))/\log(1-1/n) \tag{4}$$

$$\beta = \beta(n) := \log(\gamma^*/(1+\delta))/\log(1-1/n) \tag{5}$$

for any constant $\delta \in (0,1)$. Clearly, we always get $\alpha \leq \beta$. We also define a stopping time $\tau := \min\{t \in \mathbb{N} \mid Z_t \geq \alpha\}$ to be the first hitting time of the value α for the random variable Z_t. We then consider two phases: (1) until the random variable Z_t hits the value α for the first time ($t \leq \tau$), and (2) after the random variable Z_t has hit the value α for the first time ($t > \tau$).

3.2. Before Z_t hits value α for the first time

The algorithm starts with an initial population P_0 sampled from a uniform distribution $p_0 = (1/2, \ldots, 1/2)$. An initial observation is that the all-ones bitstring cannot be sampled in the population P_0 with high probability since the probability of sampling it from the uniform distribution is 2^{-n}, then by the union bound [31] it appears in the population P_0 with probability at most $\lambda \cdot 2^{-n} = 2^{-\Omega(n)}$ since we only consider the offspring population of size at most polynomial in the problem instance size n. The following lemma states the expectations of the random variables Z_0^* and Z_0.

Lemma 1. $\mathbb{E}[Z_0^*] = \mathcal{O}(\log \lambda)$, *and* $\mathbb{E}[Z_0] = \mathcal{O}(\log(\lambda - \mu))$.

The proof uses that the random variables Z_0^* and Z_0 denote the expected numbers of leading 1s of the fittest individual in populations of λ and $\lambda - \mu$ individuals, respectively, sampled from a uniform distribution and by a result in [33]. We now show that the value of the random variable Z_t never decreases during phase 1 with high probability by noting that its value gets decreased if the number of individuals with at least Z_t leading 1s in iteration $t+1$ is less than μ.

Lemma 2. $\Pr(\forall t \in [1,\tau] : Z_t \geq Z_{t-1}) \geq 1 - \tau e^{-\Omega(\mu)}$.

Proof. We will show via strong induction on time step t that the probability that there exists an iteration $t \in [1,\tau]$ such that $Z_t < Z_{t-1}$ is at most $\tau e^{-\Omega(\mu)}$. The base case $t = 1$ is trivial since $Z_{t-1} = Z_0$ and the probability of sampling at most μ individuals having at least Z_0 leading 1s is at most $e^{-\Omega(\mu)}$. This is because $Z_t < \alpha$ for all $t < \tau$, in expectation there are at least $(1-1/n)^{Z_t}\lambda \geq (1-1/n)^{\alpha}\lambda = \mu/(1-\delta)$ individuals with at least Z_t leading 1s sampled in iteration $t+1$. By a Chernoff bound [31], the probability of sampling at most $(1-\delta)\cdot\mu/(1-\delta) = \mu$ such individuals is at most $e^{-(\delta^2/2)\cdot\mu/(1-\delta)} = e^{-\Omega(\mu)}$ for any constant $\delta \in (0,1)$.

For the inductive step, we assume that the result holds for the first $t < \tau$ iterations, meaning that $\Pr(\exists t' \leq t : Z_{t'} < Z_{t'-1}) \leq te^{-\Omega(\mu)}$. We are left to show that it also holds for iteration $t+1$, that is, $\Pr(\exists t' \leq t+1 : Z_{t'} < Z_{t'-1}) \leq$

$(t+1)e^{-\Omega(\mu)}$. Again, by Chernoff bound, there are at most μ individuals with at least $Z_t < \alpha$ leading 1s in iteration $t+1$ with probability at most $e^{-\Omega(\mu)}$. By a union bound, this rare event does not happen during the first $t+1$ iterations with probability at most $(t+1)e^{-\Omega(\mu)}$, which completes the inductive step, and the lemma follows. □

3.3. After Z_t has hit value α for the first time

The preceding section shows that the random variable Z_t is non-decreasing during phase 1 with probability $1 - \tau e^{-\Omega(\mu)}$. The following lemma also shows that its value stays above α afterwards with high probability.

Lemma 3. *For any constant $k > 0$, it holds that*

$$\Pr(\forall t \in [\tau, \tau + e^{k\mu}] : Z_t \geq \alpha(n)) \geq 1 - e^{k\mu} \cdot e^{-\Omega(\mu)}.$$

Recall that we aim at showing an $e^{\Omega(\mu)}$ lower bound on the runtime, so we assume that the stopping time τ is at most $e^{\Omega(\mu)}$. Otherwise, if this assumption does not hold and the selective pressure is sufficiently low (as chosen below) such that $n - \alpha \geq n - \beta = \Omega(n)$, then we are done. The following lemma further shows that there is also an upper bound on the random variable Z_t.

Lemma 4. *For any constant $k > 0$, it holds that*

$$\Pr(\forall t \in [1, e^{k\mu}] : Z_t \leq \beta(n)) \geq 1 - e^{k\mu} \cdot e^{-\Omega(\mu)}.$$

Proof. It suffices to show that $\Pr(\exists t \in [1, e^{k\mu}] : Z_t > \beta) \leq e^{k\mu} \cdot e^{-\Omega(\mu)}$ via strong induction on time step t. The base case $t = 1$ is trivial. For the inductive step, we assume that the result holds for the first t iterations and need to show that it also holds for iteration $t+1$, meaning that $\Pr(\exists t' \leq t+1 : Z_{t'} > \beta) \leq (t+1)e^{-\Omega(\mu)}$. With probability $te^{-\Omega(\mu)}$, we get $Z_{t'} > \beta$ in an iteration $t' \leq t$, and if this rare event does not happen, then we obtain $Z_t \leq \beta$, meaning that in the best case the first β marginals are set to the upper border $1 - 1/n$. Then, in iteration $t+1$, the expected number of individuals with at least β leading 1s is $\lambda(1-1/n)^\beta = \mu/(1+\delta)$ for some constant $\delta \in (0,1)$. By Chernoff bound, the probability of sampling at least $(1+\delta) \cdot \mu/(1+\delta) = \mu$ such individuals is at most $e^{-(\delta^2/3)\cdot\mu/(1+\delta)} = e^{-\Omega(\mu)}$. By the union bound, the probability that $Z_{t'} > \beta$ for an iteration $t' \leq t+1$ is at most $(t+1)e^{-\Omega(\mu)}$. Thus, the inductive step is complete, and the lemma itself passes. □

Lemma 3 and Lemma 4 together give essential insights about the behaviour of the algorithm. The random variable Z_t will stay well below the threshold $\beta(n)$ for $e^{\Omega(\mu)}$ iterations with probability $1 - e^{-\Omega(\mu)}$ for a sufficiently large parent population size μ. More precisely, the random variable Z_t will fluctuate around an *equilibrium* value $\kappa = \kappa(n) := \log(\gamma^*)/\log(1-1/n)$. This is because when $Z_t = \kappa$, in expectation there are exactly $\lambda(1-1/n)^\kappa = \lambda\gamma^* = \mu$ individuals having at least κ leading 1s.

Furthermore, an exponential lower bound on the runtime is obtained if we can also show that the probability of sampling the $n - \beta$ remaining bits

correctly is exponentially small. We now choose the selective pressure γ^* such that $n - \beta \geq \varepsilon n$ for any constant $\varepsilon \in (0,1)$, that is equivalent to $\beta \leq n(1-\varepsilon)$. By (5) and solving for γ^*, we then obtain $\gamma^*/(1+\delta) \geq [(1-1/n)^n]^{1-\varepsilon}$. The right-hand side is at most $1/e^{1-\varepsilon}$ as $(1-1/n)^n \leq 1/e$ for all $n > 0$ [31], so the above inequality always holds if the selective pressure satisfies $\gamma^* \geq (1+\delta)/e^{1-\varepsilon}$ for any constants $\delta > 0$ and $\varepsilon \in (0,1)$.

The remainder of this section shows that the $n - (\beta+1) = \Omega(n)$ remaining bits cannot be sampled correctly in any polynomial number of iterations with high probability. We define $(Y_{t,i})_{i\in[n]}$ to be an offspring sampled from the probabilistic model p_t. The following lemma shows that the sampling process among the $\Omega(n)$ remaining bits are indeed independent.

Lemma 5. *For any $t \leq e^{k\mu}$ for any constant $k > 0$, with probability $1 - e^{k\mu} \cdot e^{-\Omega(\mu)}$ that the random variables $(Y_{t,i})_{i\geq\beta+2}$ are pairwise independent.*

The proof uses that the number of 1s sampled in the selected population at any bit position between $Z_t + 2 \leq \beta + 2$ (with high probability by Lemma 4) and n is binomially distributed with μ trials and its marginal probability. This is because conditional on the random variable $C_{t,i}$ for $i = Z_t + 1$ the number of 1s in the selected population in bit position $i+1$ can be written as $\mathrm{Bin}(C_{t,i}, p_{t,i+1}) + \mathrm{Bin}(\mu - C_{t,i}, p_{t,i+1}) = \mathrm{Bin}(\mu, p_{t,i+1})$, independent of the random variable $C_{t,i}$. The second term in the above sum results from the fact that there is no bias at bit position $i+1$ among the $\mu - C_{t,i}$ remaining individuals in the selected population.

For any $i \geq \beta + 1$, we always get $\mathbb{E}[Y_{t,i} \mid \mathcal{F}_{t-1}] = p_{t,i}$, and again by the tower rule $\mathbb{E}[Y_{t,i}] = \mathbb{E}[\mathbb{E}[Y_{t,i} \mid \mathcal{F}_{t-1}]] = \mathbb{E}[p_{t,i}]$. For the UMDA without margins, we obtain $\mathbb{E}[p_{t,i}] = 1/2$ since $(p_{t,i})_{t\in\mathbb{N}}$ is a martingale [9] and the initial value $p_{0,i} = 1/2$, resulting that $\mathbb{E}[Y_{t,i}] = 1/2$ for all $t \in \mathbb{N}$. However, when borders are taken into account, $\mathbb{E}[Y_{t,i}]$ no longer exactly equals but remains very close to the value $1/2$. The following lemma shows that the expectation of an arbitrary marginal $i \geq \beta + 2$ stays within $(1 \pm o(1))(1/2)$ for any $t \leq e^{\Omega(\mu)}$ with high probability.

Lemma 6. *Let $\mu \geq c \log n$ for a sufficiently large constant $c > 0$. Then, it holds with probability $1 - e^{-\Omega(\mu)}$ that $\mathbb{E}[p_{t,i}] = (1 \pm o(1))(1/2)$ for any $t \leq e^{\Omega(\mu)}$ and any $i \geq \beta + 2$.*

Lemma 6 gives us insights into the expectation of the marginal at any time $t \leq e^{\Omega(\mu)}$. One should not confuse the expectation with the actual value of the marginals. Friedrich *et al.* [9] showed that even when the expectation stays at $1/2$ (for the UMDA without borders), the actual value of the marginal in iteration t can fluctuate close to the trivial lower or upper border due to its large variance.

Lemma 7. *Let $\mu \geq c \log n$ for some sufficiently large constant $c > 0$ and $\gamma^* \geq (1+\delta)/e^{1-\varepsilon}$ for any constants $\delta > 0$ and $\varepsilon \in (0,1)$. Then, the $n - (\beta+1) = \Omega(n)$ remaining bits cannot be sampled as all 1s during any $e^{\Omega(\mu)}$ iterations with probability $1 - e^{-\Omega(\mu)}$.*

The proof makes use of the observation that the remaining bits are all sampled correctly if the sum $\sum_{i \geq \beta+2} Y_{t,i} = n - (\beta+1) = \Omega(n)$, and by Chernoff-Hoeffding bound [34]. We are ready to show our main result of the UMDA on the LeadingOnes function.

Theorem 1. *The runtime of the UMDA with the parent population size $\mu \geq c \log n$ for some sufficiently large constant $c > 0$ and the offspring population size $\lambda \leq \mu e^{1-\varepsilon}/(1+\delta)$ for any constants $\delta > 0$ and $\varepsilon \in (0,1)$ is $e^{\Omega(\mu)}$ on the* LeadingOnes *function with probability $1 - e^{-\Omega(\mu)}$ and in expectation.*

Proof. It suffices to show the high-probability statement as by the law of total expectation [31] the expected runtime is $e^{\Omega(\mu)}(1 - e^{-\Omega(\mu)}) = e^{\Omega(\mu)}$. Consider phase 1 and phase 2 as mentioned above. We also assume that phase 1 lasts for a polynomial number of iterations; otherwise, we are done and the theorem trivially holds.

During phase 2, we have observed that the random variable Z_t always stays below β for any $t \leq e^{\Omega(\mu)}$ with high probability, while the $\Omega(n)$ remaining bits cannot be sampled correctly in any iteration $t \leq e^{\Omega(\mu)}$ with probability $1 - e^{-\Omega(\mu)}$ by Lemma 7. Thus, the all-ones bitstring will be sampled with probability at most $e^{-\Omega(\mu)}$, the runtime of the UMDA on the LeadingOnes function is $e^{\Omega(\mu)}$ with probability $1 - e^{-\Omega(\mu)}$, which completes the proof. □

4. High selective pressure

When the selective pressure becomes higher such that the value of $\alpha = \alpha(n)$ exceeds the problem instance size n, phase 1 would end when the μ fittest individuals are all-ones bitstrings, i.e., the global optimum has been found. In order for this to be the case, by (4) we obtain the inequality $\gamma^*/(1-\delta) \leq (1-1/n)^n$ for any constant $\delta \in (0,1)$. The right-hand side is at least $(1-\delta)/e$ for any $n \geq (1+\delta)/\delta$ [35]. If we choose the selective pressure $\gamma^* \leq (1-\delta)^2/e$, the above inequality always holds. In this case, Dang and Lehre [12] have already shown that the algorithm requires an $\mathcal{O}\big(n\lambda \log \lambda + n^2\big)$ expected runtime on the function via the level-based theorem. We are now going to show a lower bound of $\Omega(\frac{n\lambda}{\log(\lambda-\mu)})$ on the expected runtime.

Lemma 8. *For any $t \in \mathbb{N}$ that $\mathbb{E}[Z_t^* - Z_t] = \mathcal{O}(\log \mu)$.*

Proof. Let $\delta_t := Z_t^* - Z_t$. We pessimistically assume that the Z_t first marginals are all set to one since we are only interested in a lower bound and this will speed up the optimisation process. We also define δ_t' to be the number of leading 1s of the fittest individual in a population of μ individuals each of length $n - Z_t - 1$. By the law of total expectation, we get

$$\begin{aligned}\mathbb{E}[\delta_t \mid Z_t] &= (1 + \mathbb{E}[\delta_t' \mid Z_t, X_{t,Z_t+1} = \mu]) \cdot \Pr(X_{t,Z_t+1} = \mu \mid Z_t) \\ &\leq 1 + \mathbb{E}[\delta_t' \mid Z_t, X_{t,Z_t+1} = \mu] = 1 + \mathbb{E}[\delta_t' \mid Z_t].\end{aligned}$$

We are left to calculate the expectation of δ_t', conditional on the random variable Z_t. Let $f :=$ LeadingOnes. For simplicity, we also denote $(p_i)_{i=1}^{n'}$

as the marginals of the bit positions from $Z_t + 2$ to n, respectively, where $n' := n - Z_t - 1$. The probability of sampling an individual with k leading 1s is $\Pr(f(x) = k) = (1 - p_{k+1}) \prod_{i=1}^{k} p_i$, then $\Pr(f(x) \leq k) = \sum_{j=0}^{k} \Pr(f(x) = j) = \sum_{j=0}^{k} (1 - p_{j+1}) \prod_{i=1}^{j} p_i$. Furthermore, the probability that all μ individuals have at most k leading 1s is $\Pr(\delta'_t \leq k) = \prod_{q=1}^{\mu} \Pr(f(x^{(q)}) \leq k)$, and $\Pr(\delta'_t > k) = 1 - \Pr(\delta'_t \leq k)$. Because $\mathbb{E}[Y] \leq \sum_{i=0}^{\infty} \Pr(Y > i)$ for any bounded integer-valued random variable Y, we then get

$$\mathbb{E}[\delta'_t \mid (p_i)_i, Z_t] \leq \sum_{k=0}^{\infty} \left(1 - \prod_{q=1}^{\mu} \sum_{j=0}^{k} (1 - p_{j+1}) \prod_{i=1}^{j} p_i \right).$$

Note that by Lemma 6, each marginal p_i has an expectation of $(1 \pm o(1))(1/2)$. By the tower property of expectation, linearity of expectation and independent sampling, we then obtain

$$\begin{aligned} \mathbb{E}[\delta'_t \mid Z_t] &= \mathbb{E}[\mathbb{E}[\delta'_t \mid (p_i)_i, Z_t]] \\ &\leq \sum_{k=0}^{\infty} (1 - (1 - 2^{-(k+1)})^{\mu}) + o(1) = \mathcal{O}(\log \mu). \end{aligned}$$

The final bound follows from [33], which completes the proof. □

Lemma 8 gives the important insight that the random variables Z_t and Z_t^* only differ by a logarithmic additive term at any point in time in expectation. Clearly, the global optimum is found when the random variable Z_t^* obtains the value of n. We can therefore alternatively analyse the random variable Z_t instead of Z_t^*. In other words, the random variable Z_t, starting from an initial value Z_0 given in Lemma 1, has to travel an expected distance of $n - \mathcal{O}(\log \mu) - Z_0$ bit positions before the global optimum is found. We shall make use of the additive drift theorem (for a lower bound) [36] for a distance function $g(x) = n - x$ on the stochastic process $(Z_t)_{t \in \mathbb{N}}$. Let $\Delta_t := g(Z_t) - g(Z_{t+1}) = (n - Z_t) - (n - Z_{t+1}) = Z_{t+1} - Z_t$ be the single-step change (also called drift) in the value of the random variable Z_t. The following lemma provides an upper bound on the expected drift, which directly leads to a lower bound on the expected runtime.

Lemma 9. *For any $t \in \mathbb{N}$ and $Z_t \in [n-1] \cup \{0\}$, it holds that $\mathbb{E}[\Delta_t \mid \mathcal{F}_t] = \mathcal{O}(\log(\lambda - \mu))$.*

Proof. Consider bit $i = Z_t + 1$. The random variable Z_t does not change in value if $C_{t+1,i} < \mu$. Thus, the maximum drift is obtained when $C_{t+1,i} \geq \mu$. In this case, we can express $(\Delta_t \mid Z_t) \leq 1 + (\Delta'_t \mid Z_t)$, where the non-negative $\Delta'_t \mid Z_t$ denotes the difference between the number of leading 1s of the μ-th individual in iteration $t + 1$ and the value $Z_t + 1$. Here, we can take an alternative view that $\Delta'_t \mid Z_t$ is stochastically dominated by the number of leading 1s of the fittest individual in a population of $\lambda - \mu$ individuals each of length $n - Z_t - 1$, sampled from a product distribution where each marginal has an expectation of $(1 \pm o(1))/2$ (by Lemma 6). Following [33], the fittest individual in this population has $\mathcal{O}(\log(\lambda - \mu))$ leading 1s in expectation. □

We are ready to show a lower bound on the expected runtime of the UMDA on the LeadingOnes function.

Theorem 2. *The expected runtime of the* UMDA *with a parent population size* $\mu \geq c \log n$ *for some sufficiently large constant* $c > 0$ *and an offspring population size* $\lambda \geq \mu e/(1+\delta)^2$ *where the problem instance size is* $n \geq (1+\delta)/\delta$ *for any constant* $\delta \in (0,1)$ *is* $\Omega(\frac{n\lambda}{\log(\lambda-\mu)})$ *on the* LeadingOnes *function.*

Proof. Consider the drift Δ_t on the value of the random variable Z_t. By Lemma 9 we get $\mathbb{E}[\Delta_t \mid \mathcal{F}_t] = \mathcal{O}(\log(\lambda-\mu))$. Since the random variable Z_t has to travel an expected distance of $n - \mathcal{O}(\log \lambda) - Z_0$ before the global optimum is found, the additive drift theorem shows that the expected number of iterations until the global optimum is found is $\mathbb{E}[T \mid Z_0] = \mathcal{O}((n - Z_0)/\log(\lambda-\mu))$, which by the towel rule and noting that $\mathbb{E}[Z_0] = \mathcal{O}(\log \lambda)$ satisfies $\mathbb{E}[T] = \mathbb{E}[\mathbb{E}[T \mid Z_0]] = \Omega(n/\log(\lambda-\mu))$. The proof is complete by noting that there are λ fitness evaluations in each iteration of the UMDA. □

5. LeadingOnes with prior noise

We consider a prior noise model and formally define the problem for any constant $0 < p < 1$ as follows.

$$F(x_1,\ldots,x_n) = \begin{cases} f(x_1,\ldots,x_n), & \text{w.p. } 1-p, \text{ and} \\ f(\ldots,1-x_i,\ldots), & \text{w.p. } p, \text{ where } i \sim \text{Unif}([n]). \end{cases}$$

We denote F as the noisy fitness and f as the actual fitness. For simplicity, we also denote P_t as the population prior to noise. The same noise model is studied in [37, 38, 39] for population-based EAs on the OneMax and LeadingOnes functions.

We shall make use of the level-based theorem [26, Theorem 1] and first partition the search space $\mathcal{X}$ into $n+1$ disjoint subsets $A_0,\ldots,A_n$, where

$$A_j = \{x \in \mathcal{X} : \textsc{LeadingOnes}(x) = j\}. \tag{6}$$

We also denote $A_{\geq j} = \{x \in \mathcal{X} \mid \textsc{LeadingOnes}(x) \geq j\}$. We then need to verify three conditions (G1), (G2) and (G3) of the level-based theorem [26], where due to the presence of noise we choose the parameter $\gamma_0 = \gamma^*/((1-\varepsilon)(1-p))$ for any constant $\varepsilon \in (0,1)$ to leverage the impact of noise in our analysis. The following lemma tells us the number of individuals in the noisy population in iteration t which has fitness $F(x) = f(x) \geq j$.

Lemma 10. *Assume that* $|P_t \cap A_{\geq j}| \geq \gamma_0 \lambda$, *where* $\gamma_0 := \gamma^*/((1-p)(1-\delta))$ *for some constant* $\delta \in (0,1)$. *Then, there are at least* μ *individuals with the fitness* $F(x) = f(x) \geq j$ *in the noisy population with probability* $1 - e^{-\Omega(\mu)}$. *Furthermore, there are at most* $\varepsilon\mu$ *individuals with actual fitness* $f(x) \leq j-1$ *and noisy fitness* $F(x) \geq j$ *for some small constant* $\varepsilon \in (0,1)$ *with probability* $1 - e^{-\Omega(\mu)}$.

Proof. We take an alternative view on the sampling of the population and the application of noise. More specifically, we first sample the population, sort it in descending order according to the true fitness, and then noise occurs at an individual with probability p. Because noise does not occur at an individual w.p. $1-p$, amongst the $\gamma_0\lambda$ individuals in levels $A_{\geq j}$, in expectation there are $(1-p)\gamma_0\lambda = \gamma^*\lambda/(1-\delta) = \mu/(1-\delta)$ individuals unaffected by noise. Furthermore, by a Chernoff bound [31], there are at least $(1-\delta)\cdot\mu/(1-\delta) = \mu$ such individuals for some constant $0 < \delta < 1$ with probability at least $1 - e^{-(\delta^2/3)\cdot\mu/(1-\delta)} = 1 - e^{-\Omega(\mu)}$, which proves the first statement.

For the second statement, we only consider individuals with actual fitness $f(x) < j$ and noisy fitness $F(x) \geq j$ in the noisy population. If such an individual is selected when updating the model, it will introduce a 0-bit to the total number of 0s among the μ fittest individuals for the first j bits. Let B denote the number of such individuals. There are at most $(1-\gamma_0)\lambda$ individuals with actual fitness $f(x) < j$, the probability that its noisy fitness is at least $F(x) \geq j$ is at most p/n because a specific bit must be flipped in the prior noise model. Hence the expected number of these individuals is upper bounded by

$$\mathbb{E}[B] \leq (1-\gamma_0)\lambda p/n < \lambda p/n. \tag{7}$$

We now show by a Chernoff bound that the event $B \geq \varepsilon\mu$ for a small constant $\varepsilon \in (0,1)$ occurs with probability at most $e^{-\Omega(\mu)}$. We shall rely on the fact that $\lambda p/n \leq \mu\varepsilon/2$ for sufficiently large n, which follows from the assumption $\mu/\lambda = \Theta(1)$. We use the parameter $\delta := \varepsilon\mu/\mathbb{E}[B] - 1$, which by (7) and the assumption $\lambda p/n \leq \varepsilon\mu/2$ satisfies $\delta \geq \varepsilon\mu n/(p\lambda) - 1 \geq 1$. We also have the lower bound

$$\delta \cdot \mathbb{E}[B] = \varepsilon\mu - \mathbb{E}[B] \geq \varepsilon\mu - \lambda p/n \geq \varepsilon\mu/2.$$

A Chernoff bound [31] now gives the desired result

$$\Pr(B \geq \varepsilon\mu) = \Pr(B \geq (1+\delta)\mathbb{E}[\mathbb{B}]) \leq e^{-\delta\mathbb{E}[B]/3} = e^{-\varepsilon\mu/6}, \tag{8}$$

which completes the proof. □

We now derive upper bounds on the expected runtime of the UMDA on LeadingOnes in the noisy environment.

Theorem 3. *Consider a prior noise model with parameter $p < 1$. The expected runtime of the* UMDA *with a parent population size $\mu \geq c\log n$ for some sufficiently large constant $c > 0$ where $n \geq 1/(3/4-\varepsilon)$ for some small constant $\varepsilon \in (0, 3/4)$ and an offspring population size $\lambda \geq 4e(1+\delta)\mu$ is $\mathcal{O}\big(n\lambda\log\lambda + n^2\big)$ on the* LeadingOnes *function.*

Proof. We will make use of the level-based analysis, in which we need to verify the three conditions in the level-based theorem. Each level A_j for $j \in [n] \cup \{0\}$ is formally defined as in (6), and there are a total of $m := n+1$ levels.

Condition (G1) assumes that $|P_t \cap A_{\geq j}| \geq \gamma_0\lambda$, and we are required to show that the probability of sampling an offspring in levels $A_{\geq j+1}$ in iteration $t+1$ is

lower bounded by a value z_j. We choose the parameter $\gamma_0 = \gamma^*/((1-\delta)(1-p))$ for any constant $\delta \in (0,1)$ and the selective pressure $\gamma^* = \mu/\lambda$ (assumed to be constant). For convenience, we also partition the noisy population into four groups:

1. Individuals with the fitness $f(x) \geq j$ and $F(x) \geq j$.
2. Individuals with the fitness $f(x) \geq j$ and $F(x) < j$.
3. Individuals with the fitness $f(x) < j$ and $F(x) \geq j$.
4. Individuals with the fitness $f(x) < j$ and $F(x) < j$.

By Lemma 10, there are at least μ individuals in group 1 with probability $1 - e^{-\Omega(\mu)}$. The algorithm selects the μ fittest individuals according to the noisy fitness values to update the probabilistic model. Hence, unless the mentioned event does not happen, no individuals from group 2 or group 4 will be included when updating the model.

We are now going to analyse how individuals from group 3 impact the marginal probabilities. Let B denote the number of individuals in group 3. We pessimistically assume that the algorithm uses all of the B individuals in group 3 and $\mu - B$ individuals chosen from group 1 when updating the model. For all $i \in [j]$, let X_i be the number of individuals in group 3 which has a 1-bits in positions 1 through j, except for one position i where it has a 0-bit. By definition, we then have $\sum_{i=1}^{j} X_i = B$. The marginal probabilities after updating the model are

$$p_{t,i} = \begin{cases} 1 - X_i/\mu, & \text{if } X_i > 0, \\ 1 - X_i/\mu - 1/n, & \text{if } X_i = 0. \end{cases} \tag{9}$$

Following [40, 20, 15], we lower bound the probability of sampling an offspring x with actual fitness $f(x) \geq j$, by

$$\prod_{i=1}^{j} p_{t,i} \geq \prod_{i=1}^{j} q_i, \tag{10}$$

which holds for any vector $q := (q_1, \ldots, q_j)$ which majorises the vector $p := (p_{t,1}, \ldots, p_{t,j})$. Recall (see [41, 40]) that the vector q majorises the vector p if for all $k \in [j-1]$

$$\sum_{i=1}^{k} q_i \geq \sum_{i=1}^{k} p_{t,i}, \text{ and } \sum_{i=1}^{j} q_i = \sum_{i=1}^{j} p_{t,i}.$$

We construct such a vector q which by the definition majorises the vector p as follows.

$$q_i = \begin{cases} 1 - 1/n, & \text{if } i < j, \\ \sum_{k=1}^{j} p_{t,k} - (1-1/n)(j-1), & \text{if } i = j. \end{cases}$$

We now show that with high probability, the vector element q_j stays within the interval $[1-1/n-\varepsilon, 1-1/n]$, i.e., q_j is indeed a probability. Since $p_{t,i} \leq 1-1/n$ for all $i \leq j$, we have the upper bound $q_j \leq (1-1/n)j - (1-1/n)(j-1) = 1-1/n$.

For the lower bound, we note from (9) that $p_{t,i} \geq 1 - X_i/\mu - 1/n$ for all $i \leq j$ and any $X_i \geq 0$, so we also obtain

$$\begin{aligned} q_j &\geq \sum_{k=1}^{j}(1 - X_i/\mu - 1/n) - (1-1/n)(j-1) \\ &= 1 - 1/n - \sum_{k=1}^{j} X_i/\mu = 1 - 1/n - B/\mu. \end{aligned}$$

By Lemma 10, we have $B \leq \varepsilon\mu$ for some small constant $\varepsilon \in (0,1)$ with probability $1-e^{-\Omega(\mu)}$. Assume that this high-probability event actually happens, we therefore have $q_j \geq 1 - 1/n - \varepsilon$. From this result, the definition of the vector q and (10), we can conclude that the probability of sampling in iteration $t+1$ an offspring x with actual fitness $f(x) \geq j$ is

$$\prod_{i=1}^{j} p_{t,i} \geq \prod_{i=1}^{j} q_i \geq \left(1 - \frac{1}{n}\right)^{j-1}\left(1 - \frac{1}{n} - \varepsilon\right) \geq \frac{1}{4e} = \Omega(1)$$

since $(1-1/n)^{j-1} \geq 1/e$ for any $n > 0$, and by choosing $n \geq 1/(3/4-\varepsilon)$ for some positive constant $\varepsilon < 3/4$. Because we also have $p_{t,j+1} \geq 1/n$, the probability of sampling an offspring in levels $A_{\geq j+1}$ is at least $\Omega(1) \cdot (1/n) = \Omega(1/n)$. Thus, the condition (G1) holds with a value of $z_j = \Omega(1/n)$.

For the condition (G2), we assume further that $|P_t \cap A_{\geq j+1}| \geq \gamma\lambda$ for some value $\gamma \in (0, \gamma_0)$, and we are also required to show that the probability of sampling an offspring in levels $A_{\geq j+1}$ is at least $(1+\delta)\gamma$ for some small constant $\delta \in (0,1)$. Because the marginal $p_{t,j+1}$ can be lower bounded by $\gamma\lambda/\mu$, the above probability can be written as follows.

$$\prod_{i=1}^{j+1} p_{t,i} \geq p_{t,j+1} \cdot \prod_{i=1}^{j} p_{t,i} \geq \frac{\gamma\lambda}{\mu} \cdot \frac{1}{4e} \geq (1+\delta)\gamma,$$

where by choosing $\lambda/\mu \geq 4e(1+\delta)$ for some constant $\delta \in (0,1)$. Thus, the condition (G2) of the level-based theorem is verified.

The condition (G3) requires the offspring population size to satisfy

$$\lambda \geq \frac{4}{\gamma_0\delta^2} \ln\left(\frac{128m}{\delta^2 \cdot \min_j\{z_j\}}\right) = \Omega\left(\frac{1-p}{\gamma^*}\log n\right).$$

Having fully verified the three conditions (G1), (G2) and (G3), and noting that $\ln(\delta\lambda/(4+\delta z_j)) < \ln(3\delta\lambda/2)$, the level-based theorem now guarantees an upper bound of $\mathcal{O}\big(n\lambda\log\lambda + n^2\big)$ on the expected runtime of the UMDA on the noisy LeadingOnes function.

We note that our proof is not complete since throughout the proof we always assume the happening of the following two events in each iteration of the UMDA (see Lemma 10):

(A) The number of individuals in group 1 is at least μ with probability $1 - e^{-\Omega(\mu)}$.

(B) The number of individuals in group 3 is $B \leq \varepsilon\mu$ for some small constant $\varepsilon \in (0, 3/4)$ with probability $1 - e^{-\Omega(\mu)}$.

We call an iteration a *success* if the two events happen simultaneously; otherwise, we speak of a *failure*. By the union bound, an iteration is a failure with probability at most $2e^{-\Omega(\mu)}$, and a failure occurs at least once in a polynomial number of iterations with probability at most $\text{poly}(n) \cdot 2e^{-\Omega(\mu)}$. If we choose the parent population size $\mu \geq c \log n$ for some sufficiently large constant $c > 0$, then the above probability becomes $\text{poly}(n) \cdot 2e^{-c \log n} \leq n^{c'}$ for some other constant $c' > 0$. Actually, the upper bound $\mathcal{O}\big(n\lambda \log \lambda + n^2\big)$ given by the level-based theorem is conditioned on the event that there is no failure in any iteration. We can obtain the (unconditionally) expected runtime by splitting the time into consecutive phases of length $t^* = \mathcal{O}\big(n\lambda \log \lambda + n^2\big)$, and by [20, Lemma 5] the overall expected runtime is at most $4(1 + o(1))t^* = \mathcal{O}\big(n\lambda \log \lambda + n^2\big)$. □

As a final remark, we note that the exponential lower bound in Theorem 1 for the LeadingOnes function without noise should also hold for the noisy LeadingOnes function.

6. Experiments

In this section, we provide an empirical study in order to see how closely the theoretical results match the experimental results for reasonable problem sizes, and to investigate a wider range of parameters. Our analysis is focused on different regimes on the selective pressure in the noise-free setting.

6.1. Low selective pressure

We have shown in Theorem 1 that when the selective pressure $\gamma_0 \geq (1 + \delta)/e^{1-\varepsilon}$ for any constants $\delta > 0$ and $\varepsilon \in (0, 1)$, the UMDA requires $2^{\Omega(\mu)}$ function evaluations to optimise the LeadingOnes function with high probability. We now choose $\delta = 0.2$ and $\varepsilon = 0.1$, we then get $\gamma_0 \geq (1 + 0.2)/e^{1-0.1} \approx 0.4879$. Thus, the choice $\gamma_0 = 0.5$ should be sufficient to yield an exponential runtime. For the population size, we experiment with three different settings: $\mu = 5 \log n$ (small), $\mu = \sqrt{n}$ (medium) and $\mu = n$ (large) for a problem instance size $n = 100$. Substituting everything into (4) and (5), we then get $\alpha \approx 47$ and $\beta \approx 87$. The numbers of leading 1s of the fittest individual and the μ-th individual in the sorted population (denoted by random variables Z_t^* and Z_t respectively) are shown in Fig. 1 over an epoch of 5000 iterations. The dotted blue lines denote the constant functions of $\alpha = 47$ and $\beta = 87$. One can see that the random variable Z_t keeps increasing until it reaches the value of α during the early stage and always stays well under value β afterwards. Furthermore, Z_t^* does not deviate too far from Z_t that matches our analysis since the chance of sampling all ones from the $n - \beta$ remaining bits is exponentially small.

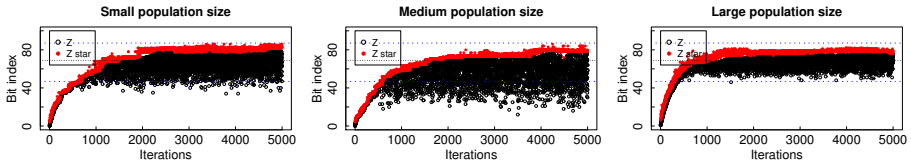

Figure 1: Low selective pressure over long-range time.

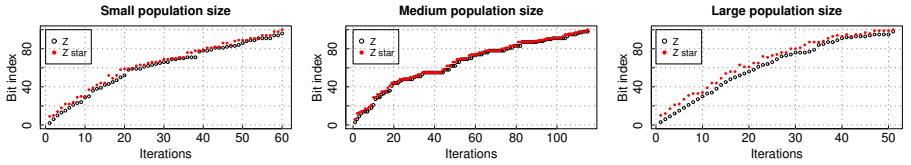

Figure 2: High selective pressure.

6.2. High selective pressure

When the selective pressure is sufficiently high, that is, $\gamma_0 \leq (1-o(1))(1-\delta)/e$ for any constant $\delta \in (0,1)$, there is an upper bound $\mathcal{O}\big(n^2 + n\lambda \log \lambda\big)$ on the expected runtime [12]. Theorem 2 yields a lower bound of $\Omega(\frac{n\lambda}{\log(\lambda-\mu)})$. We start by looking at how the values of random variable Z_t and Z_t^* change over time. Our analysis shows that it never decreases during the whole optimisation course with overwhelming probability and eventually reaches the value of n. Similarly, we consider the three different settings for population size and also note that our result holds for a parent population size $\mu \geq c \log n$, when the constant $c > 0$ must be tuned carefully; in this experiment, we set $c = 5$ (an integer larger than 3 should be sufficient). We then get $\gamma^* \leq (1-1/100)(1-0.1)/e \approx 0.1821$. Therefore, the choice of $\gamma_0 = 0.1$ should be sufficient and we then get $\alpha \approx 160 \gg n = 100$. The experiment outcomes are shown in Fig. 2. The empirical behaviours of the two random variables match our theoretical analyses.

Furthermore, we are also interested in the average runtime of the algorithm. We run some experiments using the same settings for the population size where $n \in \{100, 200, \ldots, 1000\}$. For each value of n, the algorithms are run 100 times, and the average runtime is computed. The empirical results are shown in Fig. 3. We then perform non-linear regression to fit the power model $y \sim a \cdot n^b$ to the empirical data. The fittest model and its corresponding coefficients a and b are also plotted. As seen in Fig. 3, the fittest models are all in the order of $n\lambda/\log(\lambda-\mu)$, which matches the expected runtime given in our theoretical analysis.

7. Conclusion and Future Work

In this paper, we perform rigorous analyses for the UMDA (with margins) on the LeadingOnes function in case of low selective pressure. We show that the algorithm requires a $2^{\Omega(\mu)}$ runtime with probability $1-2^{-\Omega(\mu)}$ and in expectation when $\mu \geq c \log n$ for a sufficiently large constant $c > 0$ and $\mu/\lambda \geq (1+\delta)/e^{1-\varepsilon}$

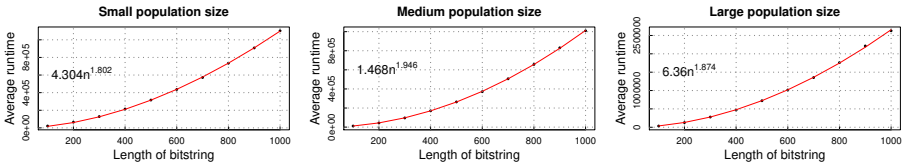

Figure 3: Average runtime under high selective pressure.

for any constant $\delta > 0$ and $\varepsilon \in (0, 1)$. The analyses reveal the limitations of the probabilistic model based on probability vectors as the algorithm hardly stays at promising states for a long time. This leads the algorithm into a non-optimal equilibrium state from which it is exponentially unlikely to sample the optimal all-ones bitstring. We also obtain the lower bound $\Omega(\frac{n\lambda}{\log(\lambda-\mu)})$ on the expected runtime when the selective pressure is sufficiently high. Furthermore, we study UMDA in noisy optimisation setting for the first time, where noise is introduced to the LeadingOnes function, causing a uniformly chosen bit is flipped with probability $p < 1$. We show that an $\mathcal{O}(n^2)$ expected runtime still holds in this case for the optimal offspring population size $\lambda = \mathcal{O}(n/\log n)$. Despite the simplicity of the noise model, this can be viewed as the first step towards broadening our understanding of the UMDA in a noisy environment.

For future work, the UMDA with an optimal offspring population size $\lambda = \mathcal{O}(n/\log n)$ needs $\mathcal{O}(n^2)$ expected time on the LeadingOnes function [12]. In this case, Theorem 2 yields a lower bound $\Omega(n^2/\log^2 n)$. Thus, it remains open whether this gap of $\Theta(\log^2 n)$ could be closed in order to achieve a tight bound on the runtime. Another avenue for future work would be to investigate the UMDA under a posterior noise model.

References

[1] H. Mühlenbein, G. Paaß, From recombination of genes to the estimation of distributions I. Binary parameters, 1996, pp. 178–187.

[2] M. Pelikan, D. E. Goldberg, F. G. Lobo, A survey of optimization by building and using probabilistic models, Computational Optimization and Applications 21 (1) (2002) 5–20.

[3] P. Larrañaga, J. A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Genetic Algorithms and Evolutionary Computation, Springer US, 2001.

[4] A. E. Eiben, J. E. Smith, Introduction to Evolutionary Computing, SpringerVerlag, 2003.

[5] M. Hauschild, M. Pelikan, An introduction and survey of estimation of distribution algorithms, Swarm and Evolutionary Computation 1 (3) (2011) 111–128.

[6] G. R. Harik, F. G. Lobo, D. E. Goldberg, The compact genetic algorithm, IlliGAL report No. 97006, University of Illinois at Urbana-Champaign.

[7] S. Baluja, Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning, Technical report, Carnegie Mellon University.

[8] T. Stützle, H. H. Hoos, Max-Min ant system, Future Generation Computer Systems 16 (8) (2000) 889 – 914.

[9] T. Friedrich, T. Kötzing, M. S. Krejca, EDAs cannot be balanced and stable, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '16, 2016, pp. 1139–1146.

[10] P. Larrañaga, H. Karshenas, C. Bielza, R. Santana, A review on probabilistic graphical models in evolutionary computation, Journal of Heuristics 18 (5) (2012) 795–819.

[11] B. Doerr, C. Doerr, The impact of random initialization on the runtime of randomized search heuristics, Algorithmica 75 (3) (2016) 529–553.

[12] D. C. Dang, P. K. Lehre, Simplified runtime analysis of estimation of distribution algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '15, 2015, pp. 513–518.

[13] M. S. Krejca, C. Witt, Lower bounds on the run time of the univariate marginal distribution algorithm on onemax, in: Proceedings of the Foundations of Genetic Algorithms Conference, FOGA '17, 2017, pp. 65–79.

[14] D. Sudholt, C. Witt, Update strength in edas and aco: How to avoid genetic drift, in: Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO '16, 2016, pp. 61–68.

[15] Z. Wu, M. Kolonko, R. H. Möhring, Stochastic runtime analysis of a cross entropy algorithm, IEEE Transactions on Evolutionary Computation 21 (4) (2017) 616–628.

[16] C. Witt, Upper bounds on the runtime of the univariate marginal distribution algorithm on onemax, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '17, 2017, pp. 1415–1422.

[17] T. Friedrich, T. Kötzing, M. S. Krejca, A. M. Sutton, The compact genetic algorithm is efficient under extreme gaussian noise, IEEE Transactions on Evolutionary Computation 21 (3) (2017) 477–490.

[18] P. K. Lehre, P. T. H. Nguyen, Improved runtime bounds for the univariate marginal distribution algorithm via anti-concentration, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '17, 2017, pp. 1383–1390.

[19] D. C. Dang, P. K. Lehre, P. T. H. Nguyen, Level-based analysis of the univariate marginal distribution algorithm, Algorithmica.

[20] P. K. Lehre, P. T. H. Nguyen, Level-based analysis of the population-based incremental learning algorithm, in: Proceedings of the International Conference on Parallel Problem Solving from Nature, PPSN XV, 2018, pp. 105–116.

[21] C. Witt, Domino convergence: why one should hill-climb on linear functions, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '18, 2018, pp. 1539–1546.

[22] J. Lengler, D. Sudholt, C. Witt, Medium step sizes are harmful for the compact genetic algorithm, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '18, 2018, pp. 1499–1506.

[23] B. Doerr, M. S. Krejca, Significance-based estimation-of-distribution algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '18, 2018, pp. 1483–1490.

[24] V. Hasenöhrl, A. M. Sutton, On the runtime dynamics of the compact genetic algorithm on jump functions, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '18, 2018, pp. 967–974.

[25] S. Droste, A rigorous analysis of the compact genetic algorithm for linear functions, Natural Computing 5 (3) (2006) 257–283.

[26] D. Corus, D. C. Dang, A. V. Eremeev, P. K. Lehre, Level-based analysis of genetic algorithms and other search processes, IEEE Transactions on Evolutionary Computation 22 (5) (2018) 707–719.

[27] R. Armañanzas, I. Inza, R. Santana, Y. Saeys, J. L. Flores, J. A. Lozano, Y. V. D. Peer, R. Blanco, V. Robles, C. Bielza, P. Larrañaga, A review of estimation of distribution algorithms in bioinformatics, BioData Mining 1 (1) (2008) 6.

[28] M. Slatkin, Linkage disequilibrium — understanding the evolutionary past and mapping the medical future, Nature Reviews Genetics 9 (6) (2008) 477–485.

[29] H. Mühlenbein, T. Mahnig, Evolutionary computation and wright's equation, Theoretical Computer Science 287 (2002) 145–165.

[30] M. S. Krejca, C. Witt, Theory of estimation-of-distribution algorithms, CoRR abs/1806.05392.

[31] R. Motwani, P. Raghavan, Randomised algorithms, Cambridge University Press, 1995.

[32] W. Feller, An introduction to probability theory and its applications, 3rd Edition, Vol. 1, John Wiley & Sons, Inc., 1968.

[33] B. Eisenberg, On the expectation of the maximum of iid geometric random variables, Statistics & Probability Letters 78 (2) (2008) 135 – 143.

[34] D. Dubhashi, A. Panconesi, Concentration of Measure for the Analysis of Randomized Algorithms, 1st Edition, Cambridge University Press, 2009.

[35] P. K. Lehre, P. S. Oliveto, Theoretical Analysis of Stochastic Search Algorithms, Springer International Publishing, 2018, pp. 1–36.

[36] J. He, X. Yao, Towards an analytic framework for analysing the computation time of evolutionary algorithms, Artificial Intelligence 145 (1) (2003) 59 – 97.

[37] C. Gießen, T. Kötzing, Robustness of populations in stochastic environments, Algorithmica 75 (3) (2016) 462–489.

[38] S. Droste, T. Jansen, I. Wegener, On the analysis of the (1+ 1) evolutionary algorithm, Theoretical Computer Science 276 (1-2) (2002) 51–81.

[39] D. Dang, P. K. Lehre, Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms, in: Foundations of Genetic Algorithms XIII, FOGA'15, 2015.

[40] P. J. Boland, F. Proschan, The reliability of k out of n systems, The Annals of Probability 11 (3) (1983) 760–764.

[41] L. J. Gleser, On the distribution of the number of successes in independent trials, The Annals of Probability 3 (1) (1975) 182–188.