



HAL
open science

Estimation-of-Distribution Algorithms for Multi-Valued Decision Variables

Firas Ben Jedidia, Benjamin Doerr, Martin S Krejca

► **To cite this version:**

Firas Ben Jedidia, Benjamin Doerr, Martin S Krejca. Estimation-of-Distribution Algorithms for Multi-Valued Decision Variables. Theoretical Computer Science, 2024, 1003, pp.114622:1-114622:16. 10.1016/j.tcs.2024.114622 . hal-04579634

HAL Id: hal-04579634

<https://hal.science/hal-04579634>

Submitted on 17 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Estimation-of-Distribution Algorithms for Multi-Valued Decision Variables

Firas Ben Jedidia^a, Benjamin Doerr^b, Martin S. Krejca^{b,*}

^a*École Polytechnique, Institut Polytechnique de Paris, Route de Saclay, Palaiseau, 91120, France*

^b*Laboratoire d'Informatique (LIX), CNRS, École Polytechnique, Institut Polytechnique de Paris, 1 rue Honoré d'Estienne d'Orves, Palaiseau, 91120, France*

Abstract

The majority of research on estimation-of-distribution algorithms (EDAs) concentrates on pseudo-Boolean optimization and permutation problems, leaving the domain of EDAs for problems in which the decision variables can take more than two values, but which are not permutation problems, mostly unexplored. To render this domain more accessible, we propose a natural way to extend the known univariate EDAs to this setting. Different from a naïve reduction to the binary case, our approach avoids additional constraints.

Since understanding genetic drift is crucial for an optimal parameter choice, we extend the known quantitative analysis of genetic drift to EDAs for multi-valued, categorical variables. Roughly speaking, when the variables take r different values, the time for genetic drift to become significant is r times shorter than in the binary case. Consequently, the update strength of the probabilistic model has to be chosen r times lower now.

To investigate how desired model updates take place in this framework, we undertake a mathematical runtime analysis on the r -valued LEADINGONES problem. We prove that with the right parameters, the multi-valued UMDA solves this problem efficiently in $O(r \ln(r)^2 n^2 \ln(n))$ function evaluations. This bound is nearly tight as our lower bound $\Omega(r \ln(r) n^2 \ln(n))$ shows.

Overall, our work shows that our good understanding of binary EDAs naturally extends to the multi-valued setting, and it gives advice on how to set the main parameters of multi-values EDAs.

Keywords: estimation-of-distribution algorithms, univariate marginal distribution algorithm, evolutionary algorithms, genetic drift, LeadingOnes benchmark

*Corresponding author

Email address: martin.krejca@polytechnique.edu (Martin S. Krejca)

1. Introduction

Estimation-of-distribution algorithms (EDAs [1]) are randomized search heuristics that evolve a probabilistic model of the search space (that is, a probability distribution over the search space). In contrast to solution-based algorithms such as classic evolutionary algorithms, which only have the choice between the two extreme decisions of keeping or discarding a solution, EDAs can take into account the information gained from a function evaluation also to a smaller degree. This less short-sighted way of reacting to new insights leads to several proven advantages, e.g., that EDAs can be very robust to noise [2, 3]. Since the evolved distributions often have a larger variance, EDAs can also be faster in exploring the search space, in particular, when it comes to leaving local optima, where they have been shown to significantly outperform simple evolutionary algorithms [4–9].

While EDAs have been employed in a variety of settings and to different types of decision variables [1, 10], they are very often presented and discussed for the binary domain. In fact, the number of results in which they have been used for discrete optimization problems with decision variables taking more than two values, other than permutation problems, is scarce [11–15]. All of these results have in common that they propose specific EDAs to deal with multi-valued problems. To the best of our knowledge, no systematic way to model EDAs for the multi-valued domain exists, even not for the easiest case of EDAs that do not model dependencies, so-called *univariate* EDAs (we note that multi-variate EDAs are much less understood, i.e., despite some theoretical works in this direction [16, 17], there are no proven runtime guarantees for these algorithms).

In order to improve our theoretical understanding in this domain, we undertake the first steps towards a framework of univariate EDAs for problems with decision variables taking more than two values (but different from permutation problems). We first note that the strong dependencies that distinguish a permutation problem from just a problem defined on $\{1, \dots, n\}^n$ have led to very particular EDAs for permutation problems. We did not see how to gain insights from these results for general multi-valued problems.

We therefore define EDAs for multi-valued decision variables without building on any related existing work. We note that, in principle, one could transform a multi-valued problem into a binary one by having, for each variable taking r different values, r binary variables, each indicating that the variable has the corresponding value. This would lead to a constrained optimization problem with the additional constraints that exactly one of these variables can take the value 1. This might be a feasible approach, but since such constraints generally impose additional difficulties, we propose a way that does not need an additional treatment of constraints (in other words, we set up our EDAs in a way that these constraints are satisfied automatically).

We defer the details to Section 4.2 and only sketch the rough idea of our approach here. For each variable taking r values, without loss of generality the values $\{0, \dots, r-1\}$, we have r sampling frequencies p_0, p_1, \dots, p_{r-1} that always add up to 1. When sampling a value for the variable, we do this mutually exclu-

sively, that is, the variable takes the value i with probability p_i . This appears to be a convenient (and in fact very natural) set-up for a multi-valued EDA. We note that there are some non-trivial technical questions to be discussed when working with frequency borders, such as $[\frac{1}{n}, 1 - \frac{1}{n}]$ in the classical binary case, but we also come up with a simple and natural solution for this aspect. Moreover, this model is well suited for *categorical* decision variables, i.e., variables whose different values do not exhibit any neighborhood property. For other kinds of variables, different, more concise models might be better suited, although our model is also applicable.

As a first step towards understanding this multi-valued EDA framework, we study how prone it is to genetic drift. Genetic drift in EDAs means that sampling frequencies not only move because of a clear signal induced by the objective function, but also due random fluctuations in the sampling process. This has the negative effect that even in the complete absence of a fitness signal, the EDA develops a preference for a particular value of this decision variable. From a long sequence of works, see Section 5 for the details, it is well understood how the time for this genetic-drift effect to become relevant depends on the parameters of the EDA [18]. Consequently, if one plans to run the EDA for a certain number of iterations, then this quantification tells the user how to set the parameters as to avoid genetic drift within this time period.

Since such a quantification is apparently helpful in the application of EDAs, we first extend it to multi-valued EDAs. When looking at the relatively general tools used in [18], this appears straightforward, but it turns out that such a direct approach does not give the best possible result. The reason is that for multi-valued decision variables, the martingale describing a frequency of a neutral variable over time has a lower variance (in the relevant initial time interval). To profit from this, we use a fairly technical martingale concentration result of McDiarmid [19], which, to the best of our knowledge, has not been used before in the analysis of randomized search heuristics. Thanks to this result, we show that the time for genetic drift to become relevant is (only) by a factor of r lower than in the case of binary decision variables (Theorem 3).

We use this result to conduct a mathematical runtime analysis of the multi-valued univariate marginal distribution algorithm (r -UMDA) on the r -valued LEADINGONES problem in the regime with low genetic drift. This problem returns, similar to the binary domain, the longest prefix of consecutive 0s in the input. It is interesting since a typical optimization process optimizes the variable sequentially in a fixed order. Consequently, in a run of an EDA on LEADINGONES, there is typically always one variable with undecided sampling frequency that has a strong influence on the fitness. Hence, this problem is suitable to study how fast an EDA reacts to a strong fitness signal.

Our runtime analysis shows that also in the multi-valued setting, EDAs can react quickly to a strong fitness signal. Since now the frequencies start at the value $\frac{1}{r}$, the time to move a frequency is a little longer, namely $\Theta(r \ln(r))$ instead of constant when the sample size λ is by a sufficient constant factor larger than the selection size μ . This still appears to be a small price for having to deal with r decision alternatives. This larger time also requires that the model update has

to be chosen more conservatively as to prevent genetic drift (for this, we profit from our analysis of genetic drift), leading to another $\ln(r)$ factor in the runtime. In summary, we prove (Theorem 6) that the UMDA can optimize the r -valued LEADINGONES problem in time $O(r \ln(r)^2 n^2 \ln(n))$, a bound that agrees with the one shown in [20] for the classical case $r = 2$. Our upper bound is tight apart from a factor logarithmic in r , that is, we prove a lower bound of order $\Omega(r \ln(r) n^2 \ln(n))$ in Theorem 10.

Overall, our work shows that r -valued EDAs can be effective problem solvers, suggesting to apply such EDAs more in practice.

This work extends our prior extended abstract [21] by adding a lower bound for the runtime of the r -valued UMDA on the r -valued LEADINGONES problem. Also, it contains all proofs that were omitted in the conference version for reasons of space. To avoid misunderstandings, we note that this work bears no similarity or overlap with the paper *Generalized Univariate Estimation-of-Distribution Algorithms* [22], which studies generalized update mechanisms for EDAs for binary decision variables.

This article is organized as follows. We describe previous works in the following section and set the notation in the subsequent section. In Section 4, we propose our multi-valued EDA framework. Our main technical results, the analysis of genetic drift and the runtime analysis for the LEADINGONES problem, can be found in Sections 5 and 6. The paper ends with a short conclusion.

2. Related Work

Since the technical sections of this work contain three relatively independent topics—the definition of multi-valued EDAs, genetic drift, and a runtime analysis on the LEADINGONES benchmark—we present the previous works relevant to these topics in the respective sections.

This being a theoretical work, we do not discuss in detail how EDAs have been successfully used to solve real-worlds optimization problems and refer to the surveys [1, 10].

Theoretically oriented works have accompanied the development and use of univariate binary EDAs for a long time, see, e.g., the early works on genetic drift described in Section 5. The first mathematical runtime analysis of an EDA was conducted by Droste [23]. This seminal work, showing an asymptotically tight bound for the runtime of the compact genetic algorithm on the ONEMAX benchmark, already contains many ideas that are now frequently used in the runtime analysis of EDAs. It also observed that EDAs optimize problems in a very different manner, visible from the different runtimes shown on two linear functions, which contrasts the famous analysis of how the $(1 + 1)$ EA optimizes linear functions by Droste, Jansen, and Wegener [24]. Interestingly, apart from the works of one research group [25–27], Droste’s ground-breaking work [23] was not followed up by other runtime analyses for around ten years. Since then, starting with works like [28–31], the runtime analysis of EDAs has become very active and has, despite the technical challenges in analyzing such complex algorithms, produced many fundamental results and a good understanding of some

of the working principles of EDAs. We refer to the recent survey [32] for more details.

An algorithmic concept related to EDAs is *ant colony optimization* (ACO) [33]. ACO lends itself well to combinatorial optimization, which is typically multi-valued in nature, and ACO has been analyzed theoretically to some degree, e.g., on the minimum-spanning tree problem [34], the traveling-salesman problem [35], and shortest-path problems [36]. ACO is related to EDAs in that ACO constructs solutions according to a probabilistic model, known as the *construction graph*. However, in contrast to EDAs, ACO usually stores a best-so-far solution, which helps enforce the probabilistic model. This in contrast to the EDAs we consider in this work. A more thorough overview on theoretical results for ACO is also provided in the same survey mentioned above [32].

3. Preliminaries

We denote by \mathbb{N} the set of all natural numbers, including 0, and by \mathbb{R} the set of all real numbers. Additionally, for $a, b \in \mathbb{N}$, let $[a..b] = [a, b] \cap \mathbb{N}$, and let $[a] = [1..a]$. When we say that a random process is a martingale and do not specify a filtration, then we mean that the process is a martingale with respect to its natural filtration. Further, for all $n \in \mathbb{N}_{\geq 1}$ and $p \in \mathbb{R}_{\geq 0}^n$, we denote the 1-norm of p , that is, the sum of the entries of p , by $\|p\|_1$. For a proposition P , we denote the Iverson bracket by $\mathbf{1}\{P\}$, which is 1 if P is true, and it is 0 otherwise.

Let $n \in \mathbb{N}_{\geq 1}$ and $r \in \mathbb{N}_{\geq 2}$. We consider the maximization of functions of the form $f: [0..r-1]^n \rightarrow \mathbb{R}$, which we call *r-valued fitness functions*. Whenever we mention an *r-valued fitness function*, we implicitly assume that its dimension n and the cardinality r of its domain are given. We call each $x \in [0..r-1]^n$ an *individual*, and we call $f(x)$ the *fitness* of x .

We say that a random variable Y *stochastically dominates* another random variable X , not necessarily defined on the same probability space, denoted by $X \preceq Y$, if and only if for all $\lambda \in \mathbb{R}$, we have $\Pr[X \leq \lambda] \leq \Pr[Y \leq \lambda]$.

4. Multi-Valued EDAs

In this section, we generalize the three common univariate EDAs for the binary decision variable to multi-valued decision variables. We do so in a manner that is consistent with the existing (empirical) literature on univariate EDAs [13, 15]. We call our EDA variants *multi-valued EDAs*. To this end, we briefly discuss the binary case in Section 4.1 before presenting our framework in Section 4.2. In our presentation, we concentrate on the UMDA [37] and then briefly present the generalizations of the other two common univariate EDAs.

We note that for classic evolutionary algorithms, multi-valued decision variables have been discussed to some extent [38–44]. Due to the very different working principles, we could not see how these results help in designing and analyzing multi-valued EDAs.

4.1. Binary EDAs

Binary EDAs refer to EDAs for *pseudo-Boolean* optimization, that is, the optimization of functions $f: \{0, 1\}^n \rightarrow \mathbb{R}$. This setting is a special case of optimizing r -valued fitness functions, for $r = 2$. The probabilistic model of univariate EDAs in this domain is a length- n vector p of probabilities (the *frequency vector*), where the probability (the *frequency*) at position $i \in [n]$ denotes the probability that a sample has a 1 at position i , independent of the other positions. Formally, for all $x, y \in \{0, 1\}^n$, it holds that $\Pr[x = y] = \prod_{i \in [n]} (p_i^{y_i} \cdot (1-p_i)^{1-y_i})$, where we assume that $0^0 = 1$.

Binary EDAs commonly take at least a parameter $\lambda \in \mathbb{N}_{\geq 1}$ (the *population size*) as well as a pseudo-Boolean fitness function f as input and optimize f as follows: Initially, the frequency vector p models the uniform distribution, that is, each frequency is $1/2$. Then, in an iterative manner, the algorithm produces λ samples (the *population*) independently via p , and it updates p based on these samples and their fitness. This process is repeated until a user-defined termination criterion is met.

In order to prevent frequencies from only producing a single value (which is the case if a frequency is 0 or 1), after the frequency vector is updated, it is typically restricted to the interval $[1/n, 1 - 1/n]$. That is, if the frequency is less than $1/n$, it is set to $1/n$, and if it is greater than $1 - 1/n$, it is set to $1 - 1/n$. The extreme values of this interval are referred to as the *borders*, and the value $1/n$ is called the *margin* of the algorithm.

UMDA. Algorithm 1 shows the *univariate marginal distribution algorithm* (UMDA) [37], which is a well established binary EDA, both in the empirical [1] and the theoretical [45] domain. In addition to the population size $\lambda \in \mathbb{N}_{\geq 1}$ and a fitness function, the UMDA also utilizes a parameter $\mu \in [\lambda]$, called the *selection size*. In each iteration, the UMDA selects μ out of the λ samples that have the best fitness (breaking ties uniformly at random). Each frequency is then set to the relative frequency of 1s at the respective position (line 6). Afterwards, the frequencies are restricted to lie within the frequency borders.

4.2. The Multi-Valued EDA Framework

We propose a framework for EDAs for optimizing r -valued fitness functions. We call the resulting EDAs r -valued EDAs. Our framework closely follows the one presented in Section 4.1. That is, an r -valued EDA starts with a probabilistic model initialized to represent the uniform distribution, and it then iteratively generates $\lambda \in \mathbb{N}_{\geq 1}$ samples independently, based on its model. This model is then updated and afterwards restricted such that it does not contain the extreme probabilities 0 and 1.

The difference to the framework for binary EDAs lies in how the probabilistic model of r -valued EDAs is represented and how it is restricted from containing extreme probabilities.

The probabilistic model. The probabilistic model of an r -valued EDA is an $n \times r$ matrix $(p_{i,j})_{(i,j) \in [n] \times [0..r-1]}$ (the *frequency matrix*), where each row $i \in [n]$ forms a vector $p_i := (p_{i,j})_{j \in [0..r-1]}$ (the *frequency vector at position i*)

Algorithm 1: The UMDA [37] with parameters $\lambda \in \mathbb{N}_{\geq 1}$ and $\mu \in [\lambda]$, maximizing a pseudo-Boolean fitness function f

```

1  $t \leftarrow 0$ ;
2  $p^{(0)} \leftarrow (\frac{1}{2})_{i \in [n]}$ ;
3 repeat // iteration  $t$ 
4    $P^{(t)} \leftarrow$  population of  $\lambda$  individuals, independently sampled
   from  $p^{(t)}$ ;
5    $\{x^{(t,k)}\}_{k \in [\mu]} \leftarrow$  multiset of  $\mu$  individuals from  $P^{(t)}$  with the highest
   fitness (breaking ties uniformly at random);
6   for  $i \in [n]$  do  $\bar{p}_i^{(t+1)} \leftarrow \frac{1}{\mu} \sum_{k \in [\mu]} x_i^{(t,k)}$ ;
7    $p^{(t+1)} \leftarrow$  values of  $\bar{p}^{(t+1)}$ , restricted to  $[\frac{1}{n}, 1 - \frac{1}{n}]$ ;
8    $t \leftarrow t + 1$ ;
9 until termination criterion met;

```

of probabilities (the *frequencies*) that sum to 1. As in the binary case, samples from p are created independently for each position. When creating an individual $x \in [0..r-1]^n$, then, for all $i \in [n]$ and all $j \in [r-1]$, the probability that x_i has value j is $p_{i,j}$. Formally, for all $x, y \in [0..r-1]^n$, it holds that $\Pr[x = y] = \prod_{i \in [n]} \prod_{j \in [0..r-1]} (p_{i,j})^{1_{\{y_i=j\}}}$, where we assume that $0^0 = 1$.

The frequency matrix p is initialized such that each frequency is $1/r$, representing the uniform distribution. When performing an update to p , it is important to make sure that each row sums to 1.

Restricting the probabilistic model. The aim of restricting the frequency matrix p is to clamp all frequencies, for some values $a, b \in [0, 1]$ (the *lower* and *upper border*, respectively) with $a \leq 1/r \leq b$, to $[a, b]$. That is, if a frequency q is less than a , it should be a after the restriction, and if it is greater than b , it should be b afterwards. For such a restriction, it is important for each row $i \in [n]$ that the frequency vector p_i sums to 1 after the restriction. This process is not straightforward. If $q \notin [a, b]$, and q is updated to $q' \in [a, b]$, then this creates a change in probability mass of $q' - q$. Hence, simply updating q to q' can result in all frequencies of p_i summing to a value other than 1 after the restriction.

We address the problem above as follows. To this end, let $a, b \in [0, 1]$ be the lower and upper border, respectively, with $a \leq 1/(r-1) - 1/(r(r-1))$ and $b = 1 - a(r-1)$. Further, let $i \in [n]$ be a row of the frequency matrix we wish to restrict, let $\bar{p}_i \in [0, 1]^n$ be the frequency vector after the update but before the restriction (with $\|\bar{p}_i\|_1 = 1$), and let $p_i^{\dagger} \in [a, b]^n$ be the vector \bar{p}_i after clamping it to $[a, b]$ but before taking care that the frequencies sum to 1. We define the *restriction of \bar{p}_i to $[a, b]$* , denoted by p_i' , to be the vector where each frequency's share above a is reduced by the surplus of the probability relative to the share

above a . Formally, for all $j \in [0..r-1]$, it holds that

$$p'_{i,j} = (p_{i,j}^+ - a) \frac{1 - ar}{\|p_i^+ - (a)_{k \in [n]}\|_1} + a. \quad (1)$$

Note that $1 - ar = \|\bar{p}_i - (a)_{k \in [n]}\|_1$ denotes how much probability mass *should* be in the frequency vector, above a . The resulting frequency vector p'_i sums to 1, since

$$\begin{aligned} \sum_{j \in [0..r-1]} p'_{i,j} &= \frac{1 - ar}{\|p_i^+ - (a)_{k \in [n]}\|_1} \sum_{j \in [0..r-1]} (p_{i,j}^+ - a) + \sum_{j \in [0..r-1]} a \\ &= 1 - ar + ar = 1. \end{aligned}$$

Further, each frequency is at least a , since this value is added at the end of eq. (1) and since $p_{i,j}^+ \geq a$ by definition of p_i^+ . Last, since each frequency is at least a after restricting, the largest a frequency can be is $1 - (r-1)a = b$.

In order to disallow the extreme frequencies 0 and 1 but to stay close to the binary case, we propose to choose the upper border as $1 - 1/n$. Following our ideas above, this implies that the lower border is $1/((r-1)n)$. This is consistent with the binary case but generalizes to the r -valued domain.

We say that an EDA is *without margins* if and only if the lower border is 0 and the upper border is 1. That is, the restriction of the frequencies does not take place.

r -UMDA. We generalize the UMDA (Algorithm 1) to the r -UMDA (Algorithm 2), utilizing our framework. This leads to the same generalization mentioned by Santana et al. [13]. Like the UMDA, the r -UMDA has three parameters, namely the population size $\lambda \in \mathbb{N}_{\geq 1}$, the selection size $\mu \in [\lambda]$, and the r -valued fitness function f . It also updates its frequencies analogously to the UMDA by choosing μ best individuals from the population of size λ and then setting each frequency at position $i \in [n]$ for value $j \in [0..r-1]$ to the relative frequency of value j at position i among the μ best individuals (line 7). We note that this results in a valid frequency vector for each row $i \in [n]$, since

$$\sum_{j \in [0..r-1]} \frac{1}{\mu} \sum_{k \in [\mu]} \mathbb{1}\{x_i^{(t,k)} = j\} = \frac{1}{\mu} \sum_{k \in [\mu]} \sum_{j \in [0..r-1]} \mathbb{1}\{x_i^{(t,k)} = j\} = \frac{1}{\mu} \sum_{k \in [\mu]} 1 = 1.$$

r -PBIL. Another popular univariate EDA is *population-based incremental learning* (PBIL [46]). It operates very similarly to the UMDA, with the only difference being in how it performs an update. In contrast to the UMDA, the PBIL does not set a frequency to the relative frequency of respective values at a position but, instead, computes the convex combination of the relative frequency with the current frequency value in its frequency vector. To this end, it utilizes a parameter $\rho \in [0, 1]$, the *scaling factor*.

We generalize the PBIL to the r -PBIL (Algorithm 3). Each frequency vector of the r -PBIL sums to 1 (before the restriction) because it is a convex combination of the r -UMDA's update (which sums to 1) and the current frequency vector (which also sums to 1).

Algorithm 2: The r -UMDA with parameters $\lambda \in \mathbb{N}_{\geq 1}$ and $\mu \in [\lambda]$, maximizing an r -valued fitness function f

```

1  $t \leftarrow 0$ ;
2  $p^{(0)} \leftarrow (\frac{1}{r})_{(i,j) \in [n] \times [0..r-1]}$ ;
3 repeat // iteration  $t$ 
4    $P^{(t)} \leftarrow$  population of  $\lambda$  individuals, independently sampled
   from  $p^{(t)}$ ;
5    $\{x^{(t,k)}\}_{k \in [\mu]} \leftarrow$  multiset of  $\mu$  individuals from  $P^{(t)}$  with the highest
   fitness (breaking ties uniformly at random);
6   for  $(i, j) \in [n] \times [0..r-1]$  do
7      $\bar{p}_{i,j}^{(t+1)} \leftarrow \frac{1}{\mu} \sum_{k \in [\mu]} \mathbf{1}\{x_i^{(t,k)} = j\}$ ;
8    $p^{(t+1)} \leftarrow$  restriction of  $\bar{p}^{(t+1)}$  to  $[\frac{1}{(r-1)n}, 1 - \frac{1}{n}]$ , as described in
   eq. (1);
9    $t \leftarrow t + 1$ ;
10 until termination criterion met;

```

r -cGA. Another popular univariate EDA is the *compact genetic algorithm* (cGA [47]). The cGA only has a single parameter $K \in \mathbb{R}_{>0}$, the *hypothetical population size*, and it creates only two samples each iteration. It ranks these two samples by fitness and then adjusts each frequency by $\frac{1}{K}$ such that the frequency of the value of the better sample is increased and that of the worse sample decreased.

We generalize the cGA to the r -cGA (Algorithm 4). Each frequency vector of the r -cGA sums to 1 after the update (before the restriction) because exactly one entry is increased by $\frac{1}{K}$ and exactly one value is decreased by this amount (noting that this can be the same frequency, in which case no change is made overall).

5. Genetic Drift

We prove an upper bound on the effect of genetic drift for r -valued EDAs (Theorem 3) in a similar fashion as Doerr and Zheng [18] for binary decision variables. This allows us to determine parameter values for EDAs that avoid the usually unwanted effect of genetic drift. The main novelty of our result over that by Doerr and Zheng [18] is that we use a slightly technical martingale concentration result due to McDiarmid [19] that allows one to profit from small variances. Such an approach is necessary. If one directly applies the methods presented by Doerr and Zheng [18], one obtains estimates for the genetic drift times that are by a factor of $\Theta(r)$ lower than ours (that is, the genetic-drift effect appears r times stronger).

In Sections 5.1 and 5.2, we first present a general introduction to the phenomenon of genetic drift. In Section 5.3, we then prove a concentration result

Algorithm 3: The r -PBIL with parameters $\lambda \in \mathbb{N}_{\geq 1}$, $\mu \in [\lambda]$, and $\rho \in [0, 1]$, maximizing an r -valued fitness function f

```

1  $t \leftarrow 0$ ;
2  $p^{(0)} \leftarrow (\frac{1}{r})_{(i,j) \in [n] \times [0..r-1]}$ ;
3 repeat // iteration  $t$ 
4    $P^{(t)} \leftarrow$  population of  $\lambda$  individuals, independently sampled
   from  $p^{(t)}$ ;
5    $\{x^{(t,k)}\}_{k \in [\mu]} \leftarrow$  multiset of  $\mu$  individuals from  $P^{(t)}$  with the highest
   fitness (breaking ties uniformly at random);
6   for  $(i, j) \in [n] \times [0..r-1]$  do
7      $\bar{p}_{i,j}^{(t+1)} \leftarrow (1 - \rho)p_{i,j}^{(t)} + \frac{\rho}{\mu} \sum_{k \in [\mu]} \mathbb{1}\{x_i^{(t,k)} = j\}$ ;
8      $p^{(t+1)} \leftarrow$  restriction of  $\bar{p}^{(t+1)}$  to  $[\frac{1}{(r-1)n}, 1 - \frac{1}{n}]$ , as described in
     eq. (1);
9      $t \leftarrow t + 1$ ;
10 until termination criterion met;

```

on neutral positions (Theorem 3). Last, in Section 5.4, we consider the setting of weak preference.

5.1. Introduction to Genetic Drift

In EDAs, *genetic drift* means that a frequency reaches the extreme values 0 or 1 due to random fluctuations from the stochasticity of the process and in the absence of a clear signal from the objective function.

While there is no proof that genetic drift is always problematic, the general opinion is that this effect should better be avoided. This is supported by the following observations and results: (i) When genetic drift is strong, many frequencies (in the binary case) approach the extreme values 0 and 1 and, consequently, the behavior of the EDA comes close to the one of a mutation-based EA, so the advantages of an EDA might be lost. (ii) The vast majority of the runtime results for EDAs, especially those for harder scenarios like noise [2] or multimodality [5], have only been shown in regimes with low genetic drift. (iii) For some particular situations, a drastic performance decrease from genetic drift was proven. For example, the UMDA with standard selection pressure but small population size $\lambda \in \Omega(\ln(n)) \cap o(n)$ has a runtime exponential in λ on the DECEPTIVELEADINGBLOCKS problem [16]. In contrast, when the population size is large enough to prevent genetic drift, here $\lambda = \Omega(n \ln(n))$, then the runtime drops to $O(\lambda n)$ with high probability.

Genetic drift in EDAs has been studied since the ground-breaking works of Shapiro [48–50], and it appears in many runtime analyses such as [8, 51–55]. Experimental evidences for the negative impact of genetic drift can further be found in [18, 32, 56]. The most final answer to the genetic-drift problem for univariate EDAs, including clear suggestions to choose the parameters as to

Algorithm 4: The r -cGA with parameter $K \in \mathbb{R}_{>0}$, maximizing an r -valued fitness function f

```

1  $t \leftarrow 0$ ;
2  $p^{(0)} \leftarrow (\frac{1}{r})_{(i,j) \in [n] \times [0..r-1]}$ ;
3 repeat // iteration  $t$ 
4    $x^{(t,1)}, x^{(t,2)} \leftarrow$  two individuals, independently sampled from  $p^{(t)}$ ;
5    $y^{(t,1)} \leftarrow$  individual with the higher fitness from  $\{x^{(t,1)}, x^{(t,2)}\}$ 
   (breaking ties uniformly at random);
6    $y^{(t,2)} \leftarrow$  individual from  $\{x^{(t,1)}, x^{(t,2)}\} \setminus \{y^{(t,1)}\}$ ;
7   for  $(i, j) \in [n] \times [0..r-1]$  do
8      $\bar{p}_{i,j}^{(t+1)} \leftarrow p_{i,j}^{(t)} + (\mathbb{1}\{y_{i,j}^{(t,1)} = j\} - \mathbb{1}\{y_{i,j}^{(t,2)} = j\}) \frac{1}{K}$ ;
9    $p^{(t+1)} \leftarrow$  restriction of  $\bar{p}^{(t+1)}$  to  $[\frac{1}{(r-1)n}, 1 - \frac{1}{n}]$ , as described in
   eq. (1);
10   $t \leftarrow t + 1$ ;
11 until termination criterion met;

```

avoid genetic drift, was given by Doerr and Zheng [18]. In the case of the UMDA (and binary decision variables, that is, the classic model), their work shows that a neutral frequency (defined in Section 5.2) stays with high probability in the middle range $[0.25, 0.75]$ for the first T iterations if $\mu = \omega(T)$. This bound is tight. When regarding n frequencies together, a value of $\mu = \Omega(T \ln(n))$ with implicit constant computable from [18, Theorem 2] ensures with high probability that all frequencies stay in the middle range for at least T iterations. Hence these bounds give a clear indication how to choose the selection size μ when aiming to run the UMDA for a given number of iterations. We note that the quantification of genetic drift can also be used to design automated ways to choose parameters, see the work by Zheng and Doerr [57], when no a-priori estimate on T is available.

Given the importance of a good understanding of genetic drift, we now analyze genetic drift for multi-valued EDAs, more specifically, for the r -UMDA. We are optimistic that, analogous to the work by Doerr and Zheng [18], very similar arguments can be applied for other main univariate EDAs.

5.2. Martingale Property of Neutral Positions

Genetic drift is usually studied via *neutral* positions of a fitness function. Let f be an r -valued fitness function. We call a position $i \in [n]$ (as well as, for an individual $x \in [0..r-1]^n$, its corresponding variable x_i and the associated frequencies of an EDA) *neutral* (w.r.t. to f) if and only if, for all $x \in [0..r-1]^n$, the value x_i has no influence on the value of f , that is, if and only if for all individuals $x, x' \in [0..r-1]^n$ such that for all $j \in [n] \setminus \{i\}$ it holds that $x_j = x'_j$, we have $f(x) = f(x')$.

An important property of neutral variables that we capitalize on in our analysis of genetic drift is that their frequencies in typical EDAs without margins form martingales [18]. This observation extends the corresponding one for EDAs for binary representations. We make this statement precise for the r -UMDA.

Lemma 1. *Let f be an r -valued fitness function, and let $i \in [n]$ be a neutral position of f . Consider the r -UMDA without margins optimizing f . For each $j \in [0..r - 1]$, the frequencies $(p_{i,j}^{(t)})_{t \in \mathbb{N}}$ are a martingale.*

Proof. Let $j \in [0..r - 1]$. Since the algorithm has no margins, in each iteration $t \in \mathbb{N}$, no restriction takes place, so it holds that $p_{i,j}^{(t+1)} = \frac{1}{\mu} \sum_{k \in [\mu]} \mathbb{1}\{x_i^{(t,k)} = j\}$. Since i is neutral, the selection of the μ best individuals is not affected by the values at position i of the λ samples. Consequently, for each $k \in [\mu]$, the value $x_i^{(t,k)}$ follows a Bernoulli distribution with success probability $p_{i,j}^{(t)}$. Hence, $\mathbb{E}[\mathbb{1}\{x_i^{(t,k)} = j\} \mid p_{i,j}^{(t)}] = p_{i,j}^{(t)}$. Further, by linearity of expectation, we get

$$\mathbb{E}\left[p_{i,j}^{(t+1)} \mid p_{i,j}^{(t)}\right] = \frac{1}{\mu} \sum_{k \in [\mu]} \mathbb{E}\left[\mathbb{1}\{x_i^{(t,k)} = j\} \mid p_{i,j}^{(t)}\right] = \frac{1}{\mu} \sum_{k \in [\mu]} p_{i,j}^{(t)} = p_{i,j}^{(t)},$$

proving the claim. \square

As in previous works on genetic drift, the martingale property of neutral frequencies allows to use strong martingale concentration results. Since in our setting the frequencies start at a value of $\frac{1}{r}$, we can only tolerate smaller deviations from this value, namely up to $\frac{1}{2r}$ in either direction. With the methods of Doerr and Zheng [18], this reduces the genetic drift by a factor of $\Theta(r^2)$. We therefore use a stronger martingale concentration result, namely [19, Theorem 3.15], which allows to exploit the lower sampling variance present at frequencies in $\Theta(\frac{1}{r})$. We note that we adjust the theorem by incorporating comments by McDiarmid, especially [19, eq. (41)], mentioning that the absolute value in eq. (41) should be around the sum, not around the maximum, as also observed by Doerr and Zheng [18].

Theorem 2 (Martingale concentration result based on the variance [19, Theorem 3.15 and eq. (41)]). *Let $(X_t)_{t \in \mathbb{N}}$ be a martingale with respect to a filtration $(\mathcal{F}_t)_{t \in \mathbb{N}}$. Further, for all $t \in \mathbb{N}_{\geq 1}$, denote the deviation by $\text{dev}_t := |X_t - X_{t-1}|$. In addition, let $b = \sup_{t \in \mathbb{N}} \text{dev}_t$, and assume that b is finite. Last, for all $t \in \mathbb{N}$, let $\hat{v}_t = \sup \sum_{s \in [t]} \text{Var}[X_s - X_{s-1} \mid \mathcal{F}_{s-1}]$. Then for all $t \in \mathbb{N}$ and all $\varepsilon \in \mathbb{R}_{\geq 0}$, it holds that*

$$\Pr\left[\max_{s \in [0..t]} |X_s - \mathbb{E}[X_0]| \geq \varepsilon\right] \leq 2 \exp\left(-\frac{\varepsilon^2}{2\hat{v}_t + 2b\varepsilon/3}\right).$$

5.3. Upper Bound on the Genetic-Drift Effect of a Neutral Position

By utilizing Theorem 2, we show for how long the frequencies of the r -UMDA at neutral positions stay concentrated around their initial value of $\frac{1}{r}$.

Theorem 3. *Let f be an r -valued fitness function, and let $i \in [n]$ be a neutral position of f . Consider the r -UMDA optimizing f . Let $T \in \mathbb{N}$ and $j \in [0..r-1]$. Then*

$$\Pr \left[\max_{s \in [0..T]} \left| p_{i,j}^{(s)} - \frac{1}{r} \right| \geq \frac{1}{2r} \right] \leq 2 \exp \left(- \frac{\mu}{12Tr + (4/3)r} \right).$$

Proof. We apply the same proof strategy as in the proof of [18, Theorem 1]. That is, we aim to apply Theorem 2. Naturally, one would apply the theorem to the sequence of frequencies $(p_{i,j}^{(t)})_{t \in \mathbb{N}}$. However, since the deviation of $p_{i,j}$ is very large, namely 1, we consider instead a more fine-grained process $(Z_t)_{t \in \mathbb{N}}$, which, roughly speaking, splits each iteration of the r -UMDA into μ sections, each of which denotes that an additional sample is added to the update. Formally, for all $t \in \mathbb{N}$ and $a \in [0..\mu-1]$, let

$$Z_{t\mu+a} = p_{i,j}^{(t)}(\mu - a) + \sum_{k \in [a]} \mathbb{1}\{x_i^{(t+1,k)} = j\}.$$

Note that, for all $t \in \mathbb{N}_{\geq 1}$, it holds that $Z_{t\mu} = \mu p_{i,j}^{(t)}$. Thus, the natural filtration $(\mathcal{F}_t)_{t \in \mathbb{N}}$ of Z allows us to measure $p_{i,j}$.

In order to apply Theorem 2, we check that its assumptions are met. To this end, we first show that Z is a martingale. Since i is neutral, the selection of the μ best individuals is not affected by the values at position i of the λ samples. Consequently, for all $k \in [\mu]$, the random variable $x_i^{(t,k)}$ follows a Bernoulli distribution with success probability $p_{i,j}^{(t)}$. Thus, we get for all $t \in \mathbb{N}$ and $a \in [0..\mu-2]$ that

$$\mathbb{E}[Z_{t\mu+a+1} - Z_{t\mu+a} \mid \mathcal{F}_{t\mu+a}] = -p_{i,j}^{(t)} + \mathbb{E}[\mathbb{1}\{x_i^{(t,a+1)} = j\} \mid \mathcal{F}_{t\mu+a}] = 0, \quad (2)$$

and further, by the definition of $p_{i,j}^{(t+1)}$, that

$$\begin{aligned} & \mathbb{E}[Z_{(t+1)\mu} - Z_{t\mu+\mu-1} \mid \mathcal{F}_{t\mu+\mu-1}] \\ &= \mu \mathbb{E}[p_{i,j}^{(t+1)} \mid \mathcal{F}_{t\mu+\mu-1}] - p_{i,j}^{(t)} - \mathbb{E}[\sum_{k \in [\mu-1]} \mathbb{1}\{x_i^{(t,k)} = j\} \mid \mathcal{F}_{t\mu+\mu-1}] \\ &= \sum_{k \in [\mu]} \mathbb{E}[\mathbb{1}\{x_i^{(t,k)} = j\} \mid \mathcal{F}_{t\mu+\mu-1}] - p_{i,j}^{(t)} \\ &\quad - \sum_{k \in [\mu-1]} \mathbb{E}[\mathbb{1}\{x_i^{(t,k)} = j\} \mid \mathcal{F}_{t\mu+\mu-1}] \\ &= \mathbb{E}[\mathbb{1}\{x_i^{(t,\mu)} = j\} \mid \mathcal{F}_{t\mu+\mu-1}] - p_{i,j}^{(t)} = 0, \end{aligned} \quad (3)$$

showing that Z is a martingale.

We take an alternative view of the event $\{\max_{s \in [0..T]} |p_{i,j}^{(s)} - \frac{1}{r}| \geq \frac{1}{2r}\}$, whose probability we aim to bound. Note that this event is equivalent to $\{\exists s \in [0..T]: |p_{i,j}^{(s)} - \frac{1}{r}| \geq \frac{1}{2r}\}$. A superset of this event is the event where we stop at the first iteration such that the inequality holds. To this end, let $S = \inf\{t \in \mathbb{N} \mid Z_t \notin [\frac{\mu}{2r}, \frac{3\mu}{2r}]\}$ be a stopping time (with respect to \mathcal{F}). From now on, we

consider the stopped process \tilde{Z} of Z with respect to S . That is, for all $t \in \mathbb{N}$, it holds that $\tilde{Z}_t = Z_{\min\{t, S\}}$. Since Z is a martingale, so is \tilde{Z} .

Let $t \in \mathbb{N}$, and let Y_t be a Bernoulli random variable with success probability $p_{i,j}^{(\lfloor t/\mu \rfloor)}$ that is \mathcal{F}_t -measurable. Note that by eqs. (2) and (3), disregarding the expected values, it holds that

$$\tilde{Z}_{t+1} - \tilde{Z}_t = (Y_t - p_{i,j}^{(\lfloor t/\mu \rfloor)}) \cdot \mathbf{1}\{t < S\}.$$

Thus, the maximum deviation b of \tilde{Z} is 1. Further, let \hat{v}_t denote the sum of variances, as defined in Theorem 2. Then, since $p_{i,j}^{(\lfloor t/\mu \rfloor)}$ and $\mathbf{1}\{t < S\}$ are \mathcal{F}_t -measurable and since, due to \tilde{Z} being stopped, it holds that $p_{i,j}^{(\lfloor t/\mu \rfloor)} \cdot \mathbf{1}\{t < S\} \in [\frac{1}{2r}, \frac{3}{2r}]$, we get

$$\begin{aligned} \text{Var}[\tilde{Z}_{t+1} - \tilde{Z}_t \mid \mathcal{F}_t] &= \text{Var}[Y_t \cdot \mathbf{1}\{t < S\} \mid \mathcal{F}_t] \\ &= p_{i,j}^{(\lfloor t/\mu \rfloor)} (1 - p_{i,j}^{(\lfloor t/\mu \rfloor)}) \cdot \mathbf{1}\{t < S\} \leq \frac{3}{2r}. \end{aligned}$$

Hence, $\hat{v}_t \leq \frac{3t}{2r}$.

Let \tilde{p} denote the stopped process of $p_{i,j}$ with respect to S . Applying Theorem 2 with $t = \mu T$ and our estimates above, noting that $\tilde{Z}_0 = \frac{\mu}{r}$, yields

$$\begin{aligned} \Pr\left[\max_{s \in [0..T]} \left| \tilde{p}_s - \frac{1}{r} \right| \geq \frac{1}{2r}\right] &= \Pr\left[\max_{s \in [0..T]} |\tilde{p}_s - \mathbb{E}[\tilde{p}_0]| \geq \frac{1}{2r}\right] \\ &= \Pr\left[\max_{s \in [0..T]} \frac{1}{\mu} |\tilde{Z}_{s\mu} - \mathbb{E}[\tilde{Z}_0]| \geq \frac{1}{2r}\right] \leq \Pr\left[\max_{s \in [0..t]} |\tilde{Z}_s - \mathbb{E}[\tilde{Z}_0]| \geq \frac{\mu}{2r}\right] \\ &\leq 2 \exp\left(-\frac{(\mu/(2r))^2}{2 \cdot 3\mu T/(2r) + (2/3)\mu/(2r)}\right) = 2 \exp\left(-\frac{\mu}{12Tr + (4/3)r}\right). \end{aligned}$$

Since we only need to consider the stopped process, as explained above, and since \tilde{p} is identical to $p_{i,j}$ until the process stops, the result follows. \square

5.4. Upper Bound for Positions with Weak Preference

A position is rarely neutral for a given fitness function. However, we prove that the results on neutral positions translate to positions where one value is better than all other values. This is referred to as *weak preference*. Formally, we say that an r -valued fitness function f has a *weak preference for a value* $j \in [0..r-1]$ at a position $i \in [n]$ if and only if, for all $x_1, \dots, x_n \in [0..r-1]$, it holds that

$$f(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) \leq f(x_1, \dots, x_{i-1}, j, x_{i+1}, \dots, x_n).$$

We now adapt Lemma 7 by Doerr and Zheng [18] to the r -UMDA.

Theorem 4. *Consider two r -valued fitness functions f, g to optimize using the r -UMDA, such that without loss of generality, the first position of f weakly prefers 0 and the first position of g is neutral.*

Let p correspond to the frequency matrix of f and q to the frequency matrix of g , both defined by the r -UMDA. Then, for all $t \in \mathbb{N}$, it holds that $q_{1,0}^{(t)} \leq p_{1,0}^{(t)}$.

Proof. We prove our claim by induction on the number of iterations t . For the base case $t = 0$, all frequencies are $1/r$. Hence, $q_{1,0}^{(0)} \preceq p_{1,0}^{(0)}$.

For the induction step, let $t \in \mathbb{N}_{\geq 1}$ and let $j \in [0..r-1]$. Further, let $Y_j \sim \text{Bin}(\mu, q_{0,j}^{(t)})$. Since 0 is a neutral position of g , the selection of the μ best individuals is not affected by the values at position 0 of the λ samples. Thus, $q_{1,j}^{(t+1)} = \frac{1}{\mu} Y$. Further, since f weakly prefers 0s, defining $Y'_j \sim \text{Bin}(\mu, p_{0,j}^{(t)})$, it holds that $p_{1,j}^{t+1} \gtrsim \frac{1}{\mu} Y'$.

Analogously to Doerr and Zheng [18], we note that since $p_{1,0}^{(t)}$ stochastically dominates $q_{1,0}^{(t)}$ by induction hypothesis, there exists a coupling of the two probability spaces that describe the states of the two algorithms at iteration t in such a way that $p_{1,0}^{(t)} \geq q_{1,0}^{(t)}$ for any point ω in the coupling probability space. For such an ω , it then follows that $Y_j \leq Y'_j$, as the success probability of the former is bounded from above by that of the latter. Hence, $q_{1,j}^{(t+1)} = \frac{1}{\mu} Y \leq \frac{1}{\mu} Y' \leq p_{1,j}^{(t+1)}$, which proves the claim. \square

We now apply Theorem 4 and extend Theorem 3 to positions with weak preference.

Theorem 5. *Let f be an r -valued fitness function with a weak preference for 0 at position $i \in [n]$. Consider the r -UMDA optimizing f . Let $T \in \mathbb{N}$. Then*

$$\Pr \left[\min_{s \in [0..T]} p_{i,0}^{(s)} \leq \frac{1}{2r} \right] \leq 2 \exp \left(- \frac{\mu}{12Tr + (4/3)r} \right). \quad (4)$$

Proof. Let g be an r -valued fitness function with neutral position i . Let q be the frequency matrix of the r -UMDA optimizing g . By Theorem 4, it follows for all $s \in \mathbb{N}$ that $p_{i,0}^{(s)}$ stochastically dominates $q_{i,0}^{(s)}$. Applying Theorem 3 to g for position i , we have

$$\Pr \left[\min_{s \in [0..T]} q_{i,0}^{(s)} \leq \frac{1}{2r} \right] \leq 2 \exp \left(- \frac{\mu}{12Tr + (4/3)r} \right).$$

Using the stochastic domination yields the tail bound for f . \square

6. Runtime Analysis of the r -UMDA

We analyze the runtime of the r -UMDA (Algorithm 2) on an r -valued variant of LEADINGONES. We start by describing the previous runtime results of EDAs on LEADINGONES (Section 6.1), then define the r -LEADINGONES problem formally (Section 6.2), and finally state and prove our main result (Theorem 6, Section 6.3).

6.1. Previous Runtime Analyses of EDAs on LEADINGONES

In contrast to ONEMAX (another popular theory benchmark function), LEADINGONES is not that extensively studied for EDAs. This is surprising, as LEADINGONES is interesting as a benchmark for univariate EDAs, since the function introduces dependencies among the different positions of a bit string, but the model of univariate EDAs assumes independence. However, since LEADINGONES only has a single local maximum, known runtime results are rather fast.

In a first mathematical runtime analysis of an EDA, however, using the unproven no-error-assumption (which essentially states that there is no genetic drift), it was shown that the UMDA optimizes the LEADINGONES benchmark in expected time $O(\lambda n)$. This was made rigorous by Chen et al. [27] with a proof that the UMDA with population size $\Omega(n^{2+\epsilon})$ optimizes LEADINGONES in time $O(\lambda n)$ with high probability. Here the relatively large required population stems from the, then, incomplete understanding of genetic drift.

In a remarkable work [28], Dang and Lehre prove a runtime of $O(n\lambda \ln(\lambda) + n^2)$, only assuming that the sample size λ is at least logarithmic. Hence this result applies both to regimes without and with genetic drift. In the regime with genetic drift, however, the dependence on λ is slightly worse than in the result by Chen et al. [27]. This was improved by Doerr and Krejca [20], where an $O(n\lambda \ln(\lambda))$ upper bound was shown for the whole regime $\lambda = \Omega(n \ln(n))$ of low genetic drift. More precisely, when $\mu = \Omega(n \ln(n))$ and $\lambda = \Omega(\mu)$, both with sufficiently large implicit constants, then the runtime of the UMDA on LEADINGONES is $O(n\lambda \ln(\frac{\lambda}{\mu}))$ with high probability. We note that the analysis by Doerr and Krejca [20] is technically much simpler than the previous ones, in particular, it avoids the complicated level-based method used by Dang and Lehre [28]. We note that also lower bounds [3, 20] and runtimes in the presence of noise have been regarded. Since we have no such results, we refer to the original works.

Besides the UMDA, LEADINGONES was considered in the analysis of newly introduced univariate EDAs. Interestingly, each of these algorithms optimizes LEADINGONES in $O(n \ln(n))$ with high probability. This runtime is faster by a factor of $n/\ln(n)$ when compared to classical EAs, and it suggests that LEADINGONES is a rather easy problem for EDAs. Friedrich, Kötzing, and Krejca [29] proved the first of these results for their *stable compact genetic algorithm* (scGA), which introduces an artificial bias into its update process that is overcome by the LEADINGONES function. However, it was later proven that the scGA fails on the typically easy ONEMAX function [58], highlighting that the scGA is not a good EDA in general.

The next result was proven by Doerr and Krejca [58], who introduce the *significance-based compact genetic algorithm* (sig-cGA). The sig-cGA saves a history of good individuals and only updates a frequency when the number of bits in the history of that position significantly deviates from its expectation. This algorithm also performs well on ONEMAX, i.e., exhibits an $O(n \ln(n))$ expected runtime.

The last result was proven recently by Ajimakin and Devi [59], who introduce the *competing genes evolutionary algorithm* (cgEA). The cgEA utilizes the Gauss–Southwell score as a quality metric for the positions of its samples. Iteratively, it picks the position i with the best score and creates a new population by letting each individual of the previous population compete against a copy of it where the bit at position i is flipped. Based on the best individuals created this way, the frequency at position i is immediately set to either 0 or 1, whichever value turns out to be better. This approach works very well for a variety of theory benchmarks, as proven by the authors. For example, for optimal parameter values, it exhibits a (deterministic) linear runtime on ONEMAX, and it optimizes the JUMP $_k$ benchmark with high probability in $O(4^k n \ln(n))$ time.

6.2. The r -LEADINGONES Benchmark

The r -LEADINGONES function (eq. (5)) is a generalization of the classical LEADINGONES benchmark [60] from the binary to the multi-valued domain. Before we define the generalization, we briefly present the LEADINGONES function.

LeadingOnes. LEADINGONES [60] is one of the most commonly mathematically analyzed benchmark functions, both in the general domain of evolutionary computation [45] as well as in the domain of EDAs [32]. For a bit string of length $n \in \mathbb{N}_{\geq 1}$, it returns the number of consecutive 1s, starting from the leftmost position. Formally, LEADINGONES: $\{0, 1\}^n \rightarrow [0..n]$ is defined as $x \mapsto \sum_{i \in [n]} \prod_{j \in [i]} x_j$. The function has a single local maximum at the all-1s string, which is also its global maximum.

r -LeadingOnes. Inspired by LEADINGONES from the binary domain, we define r -LEADINGONES: $[0..r - 1]^n \rightarrow [0..n]$ as the function that returns the number of consecutive 0s, starting from the leftmost position. Formally,

$$r\text{-LEADINGONES}: x \mapsto \sum_{i \in [n]} \prod_{j \in [i]} \mathbb{1}\{x_j = 0\}. \quad (5)$$

In contrast to the binary case, the single local optimum of r -LEADINGONES is the all-0s string, which is also its global optimum.

6.3. Runtime Results

We analyze the runtime of the r -UMDA (Algorithm 2) on the r -LEADINGONES benchmark (eq. (5)) in the regime with low genetic drift. For the upper bound (Theorem 6), compared to the binary case [20, Theorem 5], we get an extra factor of order $r \ln(r)^2$ in the runtime. The factor of r is a result of the increased waiting time to see a certain position out of r . The factor of $\ln(r)^2$ stems from the choice to stay in the regime with low genetic drift as well as for the time it takes a frequency to get to the upper border. For the lower bound, (Theorem 10), compared to the binary case [20, Theorem 6], we get an extra factor of order $r \ln(r)$.

Our two bounds differ by a factor in the order of $\ln(r)$ (for polynomial population sizes). We believe that our lower bound is missing a factor of $\ln(r)$, as we currently do not account for the time it takes a frequency to get from its starting value $\frac{1}{r}$ to $1 - \frac{1}{n}$ for this bound.

We prove the upper bound in Section 6.3.1 and the lower bound in Section 6.3.2. Both bounds are a generalization of the binary case.

6.3.1. Upper Bound

Our upper bound shows that the number of iterations until an optimum is found for the first time is almost linear in λ and in n , only adding a factor in the order of $\ln(r)$.

Theorem 6. *Let $s \in \mathbb{R}_{\geq 1}$. Consider the r -UMDA optimizing r -LEADINGONES with $\lambda \geq 3s\epsilon\mu$, $\mu \geq 24(n+1)r \ln(n)(1 + \ln_{2s}(r))$, and $n \geq 4r$. Then with a probability of at least $1 - \frac{2}{n} - \ln_{2s}(2r)n^{2-0.5n} =: p^{\text{succ}}$, the frequency vector corresponding to the value 0 is set to $(1 - \frac{1}{n})_{i \in [n]}$ in $n \ln_{2s}(2r)$ iterations.*

Hence, after $\lambda(n \ln_{2s}(2r) + 1)$ fitness function evaluations, the r -UMDA samples the optimum with probability at least $p^{\text{succ}}(1 - \exp(-\frac{\lambda}{2\epsilon}))$.

The basic premise for our proof is that for the entirety of the considered iterations, frequencies corresponding to the value 0 remain above a given threshold since r -LEADINGONES weakly prefers 0 at all positions. We define this threshold as $\frac{1}{2r}$, and we show that in a sequential manner, position by position, the frequencies corresponding to 0 are brought to $1 - \frac{1}{n}$ within a given number of iterations until all positions are covered.

First, we provide a guarantee on the concentration of all the probabilities during the entirety of the algorithm's runtime, in a way to avoid genetic drift and to remain above a minimal threshold for all frequencies.

Lemma 7. *Let $s \in \mathbb{R}_{\geq 1}$. Consider the r -UMDA with $\lambda \geq \mu \geq 24(n+1)r \ln(n)(1 + \ln_{2s}(r))$ optimizing a function that weakly prefers 0 at every position. Then with a probability of at least $1 - \frac{2}{n}$, for each $i \in [n]$, the frequency $p_{i,0}^{(t)}$ remains above $\frac{1}{2r}$ for the first $n(1 + \ln_{2s}(r))$ iterations.*

Proof. By Theorem 5 with $T = n(1 + \ln_{2s}(r))$, we have for all $i \in [n]$ that

$$\Pr \left[\min_{k=1, \dots, T} p_{i,0}^{(k)} \leq \frac{1}{2r} \right] \leq 2 \exp \left(- \frac{\mu}{12n(1 + \ln_{2s}(r))r + \frac{4r}{3}} \right).$$

Since $\mu \geq 24(n+1)r \ln(n)(1 + \ln_{2s}(r))$, we get

$$\begin{aligned} \Pr \left[\min_{k=1, \dots, T} p_{i,0}^{(k)} \leq \frac{1}{2r} \right] &\leq 2 \exp \left(- \frac{24(n+1)r \ln(n)(1 + \ln_{2s}(r))}{12n(1 + \ln_{2s}(r))r + \frac{4r}{3}} \right) \\ &\leq 2 \exp \left(- \frac{24(n+1) \ln(n)(1 + \ln_{2s}(r))}{12(n+1)(1 + \ln_{2s}(r))} \right) \\ &\leq 2 \exp(-2 \ln(n)). \end{aligned}$$

Hence, it follows that

$$\Pr \left[\min_{k=1, \dots, T} p_{i,0}^{(k)} \leq \frac{1}{2r} \right] \leq \frac{2}{n^2}.$$

Applying a union bound over all n positions yields the result. \square

In the proof of our next result, we apply the following Chernoff bound. We apply it in order to quantify the number of iterations necessary to converge every position $i \in [n]$.

Theorem 8 (Chernoff bound [61, Theorem 1.10.5]). *Let $k \in \mathbb{N}_{\geq 1}$, $\delta \in [0, 1]$, and let X be the sum of k independent random variables each taking values in $[0, 1]$. Then*

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

An important concept for our analysis, following the approach by Doerr and Krejca [20], is that a position is *critical*. Informally, a position is critical if and only if the frequencies for all smaller positions corresponding to value 0 are at the upper border. Our runtime proof relies on showing that the r -UMDA quickly increases the frequency of a critical position to the upper border, thus making the next position critical. Formally, let $t \in \mathbb{N}$. We call a position $i \in [n]$ *critical* for the r -UMDA on r -LEADINGONES in iteration t , if and only if for all $k \in [i - 1]$, it holds that $p_{k,0}^{(t)} = 1 - \frac{1}{n}$, and that $p_{i,0}^{(t)} < 1 - \frac{1}{n}$.

We now show that once a position $i \in [n]$ becomes critical, with high probability, with $s \in \mathbb{R}_{\geq 1}$ being an appropriate value separating λ from μ (that is, defining the selection pressure), it takes less than $n \ln_{2s}(r+1)$ iterations to bring the frequency of the value 0 to the upper border $1 - \frac{1}{n}$. We also prove that it remains there for a sufficient number of iterations until the convergence of the frequency matrix.

Lemma 9. *Let $s, u \in \mathbb{R}_{\geq 1}$. Consider the r -UMDA optimizing r -LEADINGONES with $\lambda \geq 3se\mu$ and $\mu \in \mathbb{N}_{\geq 1}$. Consider an iteration $t \in \mathbb{N}$ such that position $i \in [n]$ is critical, and let $b \in \mathbb{R}_{>0}$ such that $p_{i,0}^{(t)} \geq b \geq \frac{2}{n}$. Then with a probability of at least $1 - u \ln_{2s}(\frac{1}{b}) \exp(-\frac{s\mu b}{24})$, it holds for all $\theta \in [\ln_{2s}(\frac{1}{b})..u \ln_{2s}(\frac{1}{b})]$ that $p_{i,0}^{(t+\theta)} = 1 - \frac{1}{n}$.*

Proof. We start by proving that, for all $\theta \in [0..u \ln_{2s}(\frac{1}{b})]$, the frequency $p_{i,0}^{(t+\theta)}$ multiplies by at least $2s$ during an update, with high probability (and is then restricted). To this end, let $t' \in [t..t + \theta]$, and assume that $p_{i,0}^{(t')} \geq b$, and that position i or a position greater than i is critical (where we assume, for convenience, that if all frequencies for value 0 are $1 - \frac{1}{n}$, then position $n+1$ is critical). Furthermore, let X denote the number of sampled individuals in iteration t' that have at least i leading 0s. Note that $p_{i,0}^{(t')} \geq b$ by assumption as well as that i is critical in iteration t . We discuss later via induction why these assumptions also hold for iteration t' .

We consider the process of sampling a single individual. Since position at least i is critical, by definition, for all $k \in [i - 1]$, we have $p_{k,0}^{(t')} = 1 - \frac{1}{n}$. Hence, the probability that all these positions are sampled as 0 for this individual is $(1 - \frac{1}{n})^{i-1} \geq (1 - \frac{1}{n})^{n-1} \geq \frac{1}{e}$. This yields $\mathbb{E}[X] \geq \frac{\lambda p_{i,0}^{(t')}}{e}$, and since $\lambda \geq 3se\mu$, this yields $\mathbb{E}[X] \geq 3s\mu p_{i,0}^{(t')}$.

By the Chernoff bound (Theorem 8) and by the assumption $p_{i,0}^{(t')} \geq b$, we get

$$\begin{aligned} \Pr\left[X \leq \frac{5}{2}s\mu p_{i,0}^{(t')}\right] &\leq \Pr\left[X \leq \frac{5}{6}\mathbb{E}[X]\right] \leq \exp\left(-\frac{\mathbb{E}[X]}{72}\right) \\ &\leq \exp\left(-\frac{s\mu p_{i,0}^{(t')}}{24}\right) \leq \exp\left(-\frac{s\mu b}{24}\right). \end{aligned}$$

We consider $\bar{p}_{i,0}^{(t'+1)}$ as defined in Section 4.2, which is the updated frequency before being restricted to $[\frac{1}{(r-1)n}, 1 - \frac{1}{n}]$. Since $\bar{p}_{i,0}^{(t'+1)} \geq \min(\frac{X}{\mu}, 1)$ by the definition of the update of the r -UMDA, we have

$$\Pr\left[\bar{p}_{i,0}^{(t'+1)} \leq \min\left(\frac{5}{2}s\mu p_{i,0}^{(t')}, 1\right)\right] \leq \Pr\left[X \leq \frac{5}{2}s\mu p_{i,0}^{(t')}\right] \leq \exp\left(-\frac{s\mu b}{24}\right).$$

In order to update $p_{i,0}^{(t')}$, the frequency vector $\bar{p}_i^{(t'+1)}$ is restricted to the interval $[\frac{1}{(r-1)n}, 1 - \frac{1}{n}]$, which entails that the updated frequency $p_{i,0}^{(t'+1)}$ may reduce when compared to $\bar{p}_{i,0}^{(t'+1)}$. However, since the restriction adds at most the lower border (that is, $\frac{1}{(r-1)n}$) to a frequency, *any* restriction rule adds at most a probability mass of $\frac{1}{n}$ to the frequency vector. We assume pessimistically that, in order for the frequencies to sum to 1, this mass is entirely subtracted from $\bar{p}_{i,0}^{(t'+1)}$ during the restriction (noting that this does not take place once $\bar{p}_{i,0}^{(t'+1)} \geq 1 - \frac{1}{n}$, as this means that it is set to the upper border instead). Further, the assumption $p_{i,0}^{(t')} \geq b \geq \frac{2}{n}$ yields that $\frac{5}{2}s\mu p_{i,0}^{(t')} - \frac{1}{n} \geq 2s\mu p_{i,0}^{(t')}$. Hence, we get that

$$\begin{aligned} \Pr\left[p_{i,0}^{(t'+1)} < \min\left(2s\mu p_{i,0}^{(t')}, 1 - \frac{1}{n}\right)\right] \\ \leq \Pr\left[p_{i,0}^{(t'+1)} < \min\left(\frac{5}{2}s\mu p_{i,0}^{(t')} - \frac{1}{n}, 1 - \frac{1}{n}\right)\right] \leq \exp\left(-\frac{s\mu b}{24}\right). \end{aligned}$$

By induction on the iteration t' (starting at t), it follows that, with an additional failure probability of at most $\exp(-\frac{s\mu b}{24})$ per iteration, the assumptions that $p_{i,0}^{(t')} \geq b$ and that position at least i is critical are satisfied.

Starting from iteration t , a union bound over the next $u \ln_{2s}(\frac{1}{b})$ iterations yields that the frequency $p_{i,0}$ continues growing exponentially with a factor of $2s$ for the next $u \ln_{2s}(\frac{1}{b})$ iterations with probability at least $1 - u \ln_{2s}(\frac{1}{b}) \exp(-\frac{s\mu b}{24})$. Since, by assumption, $p_{i,0}^{(t)} \geq b$, it reaches $1 - \frac{1}{n}$ after at most $\ln_{2s}(\frac{1}{b})$ iterations during that time, concluding the proof. \square

We now prove our main result.

Proof of Theorem 6. Since r -LEADINGONES weakly prefers 0s at all positions $i \in [n]$, by Lemma 7, with a probability of at least $1 - \frac{2}{n}$, for all $i \in [n]$, the frequency $p_{i,0}$ remains above $\frac{1}{2r}$ for the first $n(1 + \ln_{2s}(r))$ iterations.

For each position $i \in [n]$, we apply Lemma 9 with $b = \frac{1}{2r}$ and $u = n$ as well as $\mu \geq 24nr \ln(n)$ and $s \geq 1$, noting that the assumption $b \geq \frac{2}{n}$ is satisfied, since we assume $n \geq 4r$. Hence, for each $i \in [n]$, with a probability of at least $1 - \ln_{2s}(2r)n^{1-0.5n}$, after at most $\ln_{2s}(2r)$ iterations, the frequency $p_{i,0}$ is set to $1 - \frac{1}{n}$ and remains there for at least $(n-1)\ln_{2s}(2r)$ iterations. Further, by a union bound over all n frequency vectors, the above holds for all frequency vectors, with probability at least $1 - \ln_{2s}(2r)n^{2-0.5n}$.

Combining everything, with probability at least $1 - \frac{2}{n} - \ln_{2s}(2r)n^{2-0.5n}$, it holds by induction on position i that once position i is critical, the frequency $p_{i,0}$ reaches $1 - \frac{1}{n}$ in at most $\ln_{2s}(2r)$ iterations and remains there until at least iteration $n \ln_{2s}(2r)$. Since position 0 is critical in iteration 0, it follows that the frequencies for value 0 are set, in increasing order of their position, to $1 - \frac{1}{n}$. After at most $n \ln_{2s}(2r)$ iterations, all such frequencies are at the upper border, which proves the first part of the claim.

For the second part, note that once for all $i \in [n]$ holds that $p_{i,0} = 1 - \frac{1}{n}$, which occurs with probability at least p^{succ} , as shown above, the r -UMDA creates the global maximum of r -LEADINGONES during the next iteration with probability at least $(1 - \frac{1}{n})^n \geq \frac{1}{2e}$ for each offspring. Since the algorithm creates λ offspring independently, the probability of not creating the global maximum within the next λ fitness evaluations is at most $(1 - \frac{1}{2e})^\lambda \leq \exp(-\frac{\lambda}{2e})$. Multiplying the complementary probability with the previous success probability p^{succ} thus concludes the proof. \square

6.3.2. Lower Bound

As the upper bound (Theorem 6), the lower bound shows an almost linear dependency of the number of iterations until the optimum is sampled for the first time with respect to λ and n , only adding a factor of order $\ln(r)$. The difference of $\ln(r)$ to the upper bound stems from the bound on μ , which is larger by a factor of around $\ln(r)$ in the upper bound.

Theorem 10. *Let $\delta \in (0, 1)$ be a constant. Consider the r -UMDA optimizing r -LEADINGONES with $\lambda \geq \mu \geq \max\{24(n+1)r \ln(n), 6\frac{1+\delta}{\delta^2} \ln(n)\}$. Furthermore, let $d = \lceil \log_{2r/3}((1+\delta)\frac{\lambda}{\mu}) \rceil = \lceil \frac{\ln((1+\delta)\lambda/\mu)}{\ln(2r/3)} \rceil$ and let $\xi = \lceil \log_{2r/3}(n^2\lambda) \rceil + 1$. Then with probability at least $1 - 4n^{-1}$, the r -UMDA does not sample the optimum in iteration $\lfloor \frac{n-\xi}{d} \rfloor - 1$ or earlier. This corresponds to more than $\lambda \lfloor \frac{n-\xi}{d} \rfloor$ fitness function evaluations until the optimum is sampled for the first time.*

Our proof of Theorem 10 follows closely the proof for a lower bound on the runtime of the UMDA on LEADINGONES in the binary case by Doerr and Krejca [20, Theorem 6]. The proof mainly relies on the leftmost position in a population that never had at least μ samples with a 0 so far. This position increases each iteration with high probability by only about $\ln(\frac{\lambda}{\mu})/\ln(r) =: d$. Before this position is sufficiently close to n , it is very unlikely that the r -UMDA samples the optimum of r -LEADINGONES. Hence, the runtime is with high probability in the order of $\frac{n}{d}$.

To make this outline formal, we say that a position $i \in [n]$ is *selection-relevant in iteration* $t \in \mathbb{N}$ (for r -LEADINGONES) if and only if the population in iteration t of the r -UMDA optimizing r -LEADINGONES has at least μ individuals with at least $i - 1$ leading 0s. Note that multiple positions can be selection-relevant in the same iteration, and that position 1 is always selection-relevant. Furthermore, for each iteration $t \in \mathbb{N}$, we say that position $i \in [n]$ is the *maximum selection-relevant position* if and only if i is the largest value among all selection-relevant positions in iteration t .

An important observation is that if position $i \in [n]$ is not selection-relevant up to (including) iteration $t \in \mathbb{N}$, then i is also neutral up to iteration t . The reason is that the selection of individuals is solely determined by positions up to the smallest position $j \in [n]$ of the μ best individuals where one of them contains a value different than 0. All following positions do not change the ranking of the μ best individuals. Hence, if $i > j$, then i is neutral.

The following lemma shows that the frequency for value 0 in positions that were not yet selection-relevant remain close to their starting value of $\frac{1}{r}$, as they are neutral up to that point.

Lemma 11. *Let $g \in \mathbb{N}_{\geq 1}$. Consider the r -UMDA optimizing r -LEADINGONES with $\lambda \geq \mu \geq 24(g+1)r \ln(g)$. For all $i \in [n]$, let T_i denote the first iteration such that position i is selection-relevant, and let $T_i^{\text{sel}} = \min\{T_i, g\}$. Then with probability at least $1 - 2ng^{-2}$, it holds for each $i \in [n]$ and each $t \in [0..T_i^{\text{sel}}]$ that $p_{i,0}^{(t)} \in (\frac{1}{2r}, \frac{3}{2r})$.*

Proof. Let $i \in [n]$. We show that the sequence $(p_{i,0}^{(t)})_{t \in \mathbb{N}}$ remains in $(\frac{1}{2r}, \frac{3}{2r})$ as long as $t \leq T_i^{\text{sel}}$ by aiming to apply Theorem 3. We then conclude the proof via a union bound of the failure probabilities (that is, the probabilities that a frequency does not remain in said interval) over all possible values for i .

Conditional on T_i^{sel} , since i only becomes selection-relevant the earliest in iteration T_i , position i is neutral up to (including) iteration T_i . That is, for all $t \in [0..T_i - 1]$, position i has no influence on the fitness of each individual in population $P^{(t)}$ (and thus on the updated frequency $p_{i,0}^{(t+1)}$). Hence, by Theorem 3, by $T_i^{\text{sel}} \leq g$, and by the lower bound on μ , we get that

$$\begin{aligned} \Pr \left[\max_{s \in [0..T_i^{\text{sel}}]} \left| p_{i,0}^{(s)} - \frac{1}{r} \right| \geq \frac{1}{2r} \mid T_i^{\text{sel}} \right] &\leq 2 \exp \left(- \frac{\mu}{12T_i^{\text{sel}}r + (4/3)r} \right) \\ &\leq 2 \exp \left(- \frac{\mu}{12(g+1)r} \right) \leq 2 \exp \left(- \frac{24(g+1)r \ln(g)}{12(g+1)r} \right) \leq 2g^{-2}. \end{aligned}$$

By the law of total probability, this bound also holds independently of the outcome of T_i^{sel} .

Taking the union bound of the above bound over all n values for i yields that the overall failure probability is at most $2ng^{-2}$, concluding the proof. \square

For the next lemma, we make use of the following Chernoff bound, which we apply in order to show that new offspring does not extend the prefix of

leading 0s by too much. It is a non-trivial extension of the typical Chernoff bound to the case where we have an upper bound on the expected value of the sum of independent Bernoulli random variables. This extension is non-trivial as the upper bound on the expectation also results in a stronger probability bound.

Theorem 12 (Chernoff bound [61, Theorem 1.10.21 (a) with Theorem 1.10.1]). *Let $k \in \mathbb{N}_{\geq 1}$, and let X be the sum of k independent random variables each taking values in $[0, 1]$. Moreover, let $\delta, \mu^+ \in \mathbb{R}_{\geq 0}$ such that $\mu^+ \geq \mathbb{E}[X]$. Then*

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

In the following lemma, we show that the maximum selection-relevant position increases each iteration with high probability by at most roughly $\log_r(\frac{\lambda}{\mu})$. To this end, we tie it to the concept of a critical position, as defined in Section 6.3.1. This proof is heavily inspired by the proof of Doerr and Krejca [20, Lemma 4], but we fix a mistake in their proof, where the penultimate estimate of the application of the Chernoff bound bounds the exponent in the wrong direction.

Lemma 13. *Let $\delta \in (0, 1)$ be a constant. Consider the r -UMDA optimizing r -LEADINGONES with $\mu \geq 6\frac{1+\delta}{\delta^2} \ln(n)$. Furthermore, consider an iteration $t \in \mathbb{N}$ such that position $i \in [n]$ is critical and that, for all positions $i' \in [i + 1, n]$, it holds that $p_{i',0}^{(t)} \leq \frac{3}{2}\frac{1}{r}$. Let $d = \lceil \log_{2r/3}((1 + \delta)\frac{\lambda}{\mu}) \rceil$. Then, with probability at least $1 - n^{-2}$, the maximum selection-relevant position in iteration t is at most $\min\{n, i + d\}$.*

Proof. We note that $\lambda \geq \mu$ by the definition of the r -UMDA and since $\delta > 0$, it holds that $d \geq 1$. Furthermore, we assume that $i < n - d$, that is, it holds that $\min\{n, i + d\} = i + d$. For $i \geq n - d$, we statement claims that the maximum selection-relevant position is at most n , which is trivially the case, as all positions are in $[n]$.

For a position $k \in [n]$ to become the maximum selection-relevant position in iteration t , by definition, it is necessary that at least μ individuals in population $P^{(t)}$ have at least $k - 1$ leading 0s. We show via Theorem 12 that it is very unlikely that such a prefix of leading 0s extends by much.

To this end, let $k = i + d$, and let X denote the number of individuals from $P^{(t)}$ with at least k leading 0s. Since we assume that each frequency of value 0 at a position larger than i is at most $\frac{3}{2}\frac{1}{r}$, as well as due to the independent sampling of the r -UMDA and due to the definition of d , it follows that

$$\mathbb{E}[X] \leq \lambda \left(\frac{3}{2}\frac{1}{r}\right)^d = \lambda \left(\frac{2}{3}\right)^{-d} \leq \lambda \frac{\mu}{(1 + \delta)\lambda} = \frac{\mu}{1 + \delta}.$$

Hence, by applying Theorem 12 with $\mu^+ = \frac{\mu}{1+\delta}$, recalling that $\delta \in (0, 1)$, and by applying the bound on μ , we get that

$$\begin{aligned} \Pr[X \geq \mu] &= \Pr\left[X \geq (1 + \delta) \frac{\mu}{1 + \delta}\right] \leq \exp\left(-\frac{1}{3} \min\{\delta^2, \delta\} \frac{\mu}{1 + \delta}\right) \\ &= \exp\left(-\frac{1}{3} \mu \frac{\delta^2}{1 + \delta}\right) \leq n^{-2}. \end{aligned}$$

Consequently, with probability at least $1 - n^{-2}$, the population $P^{(t)}$ contains fewer than μ offspring that have at least k leading 0s. That is, the largest position $k' \in [n]$ where at least μ offspring have at least k' leading 0s is at most $k - 1$, which is equivalent to the maximum selection-relevant position being at most k . \square

The next lemma is the last one before we prove our lower bound. The lemma shows that it is very unlikely for the r -UMDA to sample the optimum of LEADINGONES while many frequencies for value 0 are not high yet (which is measured by the critical position).

Lemma 14. *Consider the r -UMDA optimizing r -LEADINGONES, and consider an iteration $t \in \mathbb{N}$ and a position $i \in [n]$ such that, for all positions $i' \in [i + 1..n]$, it holds that $p_{i',0}^{(t)} \leq \frac{3}{2} \frac{1}{r}$. Then, with probability at least $1 - \lambda \left(\frac{3}{2} \frac{1}{r}\right)^{n-i}$, the r -UMDA does not sample the optimum in this iteration.*

Proof. We bound the probability for sampling the optimum this iteration from above. The probability for a single offspring to be the optimum is, due to the upper bound on the last $n - i$ frequencies, at most $\left(\frac{3}{2} \frac{1}{r}\right)^{n-i}$, as all positions need to be a 0. Taking a union bound over all λ samples of this iteration concludes the proof. \square

Lemmas 7, 13 and 14 are sufficient for proving Theorem 10.

Proof of Theorem 10. We only show the bound on the number of iterations. Since we start counting iterations at 0 and since the r -UMDA creates exactly λ offspring each iteration, the bound on the number of fitness function evaluations follows immediately.

For the entirety of the proof, we assume that during the first n iterations, all frequencies for value 0 remain in $\left(\frac{1}{2} \frac{1}{r}, \frac{3}{2} \frac{1}{r}\right)$ as long as they did not become selection-relevant yet. By Lemma 11 with $g = n$, noting that μ is sufficiently large, this occurs with probability at least $1 - 2n^{-1}$. Furthermore, we assume that $n - \xi \geq d$, as Theorem 10 yields a trivial lower bound of 0 otherwise.

We continue by proving via induction on $t \in [0..n]$ that with probability at least $1 - (t+1)n^{-2}$ it holds that each position $i \in [(t+1)d + 2..n]$ is not relevant up to (including) iteration t .

For the base case $t = 0$, by the definition of the r -UMDA, for all positions $i \in [n]$, it holds that $p_{i,0}^{(0)} = \frac{1}{r}$. This especially means that position 0 is critical this iteration. Applying Lemma 13, noting that the requirements for δ and μ are

met, proves the base case, as, with probability at least $1 - n^{-2}$, the maximum selection-relevant position in iteration 0 is d .

For the inductive step, assume that the inductive hypothesis holds up to (including) iteration $t \in [0..n-1]$. Hence, with probability at least $1 - (t+1)n^{-2}$, the maximum selection relevant-position in iteration t (and up to there) is at most $(t+1)d+1$. This implies that the critical position $k \in [n]$ in iteration $t+1$ is also at most $(t+1)d+1$. Furthermore, all frequencies for value 0 at positions greater than $(t+1)d+1$ have not been selection-relevant yet. Thus, by our argument at the beginning of the proof, these frequencies are at most $\frac{3}{2} \frac{1}{r}$. Overall, by Lemma 13, in iteration $t+1$, with probability at most n^{-2} , the maximum selection-relevant position in iteration $t+1$ is at least $k+d+1$. Via a union bound with the failure probability of the inductive hypothesis, this proves the claim, that is, with probability at least $1 - (t+2)n^{-2}$, the maximum-selection relevant position in iteration $t+1$ is at most $k+d \leq (t+2)d+1$.

This claim shows that, for $t' = \lfloor \frac{n-\xi}{d} \rfloor - 1 \leq n$, with probability at least $1 - n^{-1}$, each position greater than $n - \xi + 1$ is never selection-relevant up to (including) iteration t' . Hence, by our argument at the beginning of the proof, these frequencies are at most $\frac{3}{2} \frac{1}{r}$. Applying Lemma 14 with $i = n - \xi + 1$ then yields that the r -UMDA does not sample the optimum in each iteration up to t' with a probability of at least $1 - \lambda(\frac{3}{2} \frac{1}{r})^{n-i} = 1 - \lambda(\frac{3}{2} \frac{1}{r})^{\xi-1} \geq 1 - n^{-2}$ per iteration. A union bound over at most $t'+1 \leq n$ iterations then shows that with probability at least $1 - n^{-1}$, it holds that up to (including) iteration t' , the r -UMDA does not sample the optimum.

Last, a union bound over the three error probabilities of the three arguments above then shows that with probability at least $1 - 4n^{-1}$, the r -UMDA does not sample the optimum up to (including) iteration t' , concluding the proof. \square

7. Conclusion

We have proposed the first systematic framework of EDAs for problems with multi-valued decision variables. Our analysis of the genetic-drift effect and our runtime analysis on the multi-valued version of LEADINGONES have shown that the increase in decision values does not result in significant difficulties. Although there may be a slightly stronger genetic drift (requiring a more conservative model update, that is, a higher selection size μ for the UMDA) and slightly longer runtimes, these outcomes are to be expected given the increased complexity of the problem. We hope that our findings will inspire researchers and practitioners to embrace the benefits of EDAs for multi-valued decision problems, beyond the previously limited application to mostly permutations and binary decision variables.

An interesting question for future work is to analyze whether other model representations, especially for multi-valued problems that do not consider categorical variables, have a benefit over our model.

Acknowledgments

Thank you to Josu Ceberio for some useful discussions. This work also profited from many scientific discussions at the Dagstuhl Seminar 22182 “Estimation-of-Distribution Algorithms: Theory and Applications”. This work was supported by a public grant as part of the Investissements d’avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH.

References

- [1] M. Pelikan, M. Hauschild, F. G. Lobo, Estimation of distribution algorithms, in: J. Kacprzyk, W. Pedrycz (Eds.), Springer Handbook of Computational Intelligence, Springer, 2015, pp. 899–928.
- [2] 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 (2017) 477–490.
- [3] P. K. Lehre, P. T. H. Nguyen, Runtime analysis of the univariate marginal distribution algorithm under low selective pressure and prior noise, in: Genetic and Evolutionary Computation Conference, GECCO 2019, ACM, 2019, pp. 1497–1505.
- [4] V. Hasenöhr, A. M. Sutton, On the runtime dynamics of the compact genetic algorithm on jump functions, in: Genetic and Evolutionary Computation Conference, GECCO 2018, ACM, 2018, pp. 967–974.
- [5] B. Doerr, The runtime of the compact genetic algorithm on Jump functions, *Algorithmica* 83 (2021) 3059–3107.
- [6] S. Wang, W. Zheng, B. Doerr, Choosing the right algorithm with hints from complexity theory, in: International Joint Conference on Artificial Intelligence, IJCAI 2021, ijcai.org, 2021, pp. 1697–1703.
- [7] R. Benbaki, Z. Benomar, B. Doerr, A rigorous runtime analysis of the 2-MMAS_{ib} on jump functions: ant colony optimizers can cope well with local optima, in: Genetic and Evolutionary Computation Conference, GECCO 2021, ACM, 2021, pp. 4–13.
- [8] B. Doerr, M. S. Krejca, The univariate marginal distribution algorithm copes well with deception and epistasis, *Evolutionary Computation* 29 (2021) 543–563.
- [9] C. Witt, How majority-vote crossover and estimation-of-distribution algorithms cope with fitness valleys, *Theoretical Computer Science* 940 (2023) 18–42.
- [10] P. Larrañaga, J. A. Lozano (Eds.), Estimation of Distribution Algorithms, Springer, 2002.

- [11] R. Santana, P. Larrañaga, J. A. Lozano, Protein folding in simplified models with estimation of distribution algorithms, *IEEE Transactions on Evolutionary Computation* 12 (2008) 418–438.
- [12] R. Santana, P. Larrañaga, J. A. Lozano, Learning factorizations in estimation of distribution algorithms using affinity propagation, *Evolutionary Computation* 18 (2010) 515–546.
- [13] R. Santana, A. Ochoa-Rodriguez, M. Soto, Solving problems with integer representation using a tree based factorized distribution algorithm, in: *International NAISO Congress on Neuro Fuzzy Technologies*, 2002.
- [14] R. Santana, A. Mendiburu, Model-based template-recombination in markov network estimation of distribution algorithms for problems with discrete representation, in: *World Congress on Information and Communication Technologies, WICT 2013*, 2013, pp. 170–175.
- [15] H. Mühlenbein, The equation for response to selection and its use for prediction, *Evolutionary Computation* 5 (1997) 303–346.
- [16] P. K. Lehre, P. T. H. Nguyen, On the limitations of the univariate marginal distribution algorithm to deception and where bivariate EDAs might help, in: *Foundations of Genetic Algorithms, FOGA 2019*, ACM, 2019, pp. 154–168.
- [17] B. Doerr, M. S. Krejca, Bivariate estimation-of-distribution algorithms can find an exponential number of optima, *Theoretical Computer Science* 971 (2023) 114074.
- [18] B. Doerr, W. Zheng, Sharp bounds for genetic drift in estimation-of-distribution algorithms, *IEEE Transactions on Evolutionary Computation* 24 (2020) 1140–1149.
- [19] C. McDiarmid, Concentration, in: *Probabilistic Methods for Algorithmic Discrete Mathematics*, volume 16, Springer, Berlin, 1998, pp. 195–248.
- [20] B. Doerr, M. S. Krejca, A simplified run time analysis of the univariate marginal distribution algorithm on LeadingOnes, *Theoretical Computer Science* 851 (2021) 121–128.
- [21] F. Ben Jedidia, B. Doerr, M. S. Krejca, Estimation-of-distribution algorithms for multi-valued decision variables, in: *Genetic and Evolutionary Computation Conference, GECCO 2023*, ACM, 2023, pp. 230–238.
- [22] B. Doerr, M. Dufay, General univariate estimation-of-distribution algorithms, in: *Parallel Problem Solving From Nature, PPSN 2022, Part II*, Springer, 2022, pp. 470–484.
- [23] S. Droste, A rigorous analysis of the compact genetic algorithm for linear functions, *Natural Computing* 5 (2006) 257–283.

- [24] S. Droste, T. Jansen, I. Wegener, On the analysis of the (1+1) evolutionary algorithm, *Theoretical Computer Science* 276 (2002) 51–81.
- [25] T. Chen, K. Tang, G. Chen, X. Yao, On the analysis of average time complexity of estimation of distribution algorithms, in: *Congress on Evolutionary Computation, CEC 2007, IEEE, 2007*, pp. 453–460.
- [26] T. Chen, P. K. Lehre, K. Tang, X. Yao, When is an estimation of distribution algorithm better than an evolutionary algorithm?, in: *Congress on Evolutionary Computation, CEC 2009, IEEE, 2009*, pp. 1470–1477.
- [27] T. Chen, K. Tang, G. Chen, X. Yao, Analysis of computational time of simple estimation of distribution algorithms, *IEEE Transactions on Evolutionary Computation* 14 (2010) 1–22.
- [28] D. Dang, P. K. Lehre, Simplified runtime analysis of estimation of distribution algorithms, in: *Genetic and Evolutionary Computation Conference, GECCO 2015, ACM, 2015*, pp. 513–518.
- [29] T. Friedrich, T. Kötzing, M. S. Krejca, EDAs cannot be balanced and stable, in: *Genetic and Evolutionary Computation Conference, GECCO 2016, ACM, 2016*, pp. 1139–1146.
- [30] D. Sudholt, C. Witt, Update strength in EDAs and ACO: How to avoid genetic drift, in: *Genetic and Evolutionary Computation Conference, GECCO 2016, ACM, 2016*, pp. 61–68.
- [31] M. S. Krejca, C. Witt, Lower bounds on the run time of the univariate marginal distribution algorithm on OneMax, in: *Foundations of Genetic Algorithms, FOGA 2017, ACM, 2017*, pp. 65–79.
- [32] M. Krejca, C. Witt, Theory of estimation-of-distribution algorithms, in: B. Doerr, F. Neumann (Eds.), *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, Springer, 2020, pp. 405–442. Also available at <https://arxiv.org/abs/1806.05392>.
- [33] M. Dorigo, T. Stützle, *Ant colony optimization*, MIT Press, 2004.
- [34] F. Neumann, C. Witt, Ant colony optimization and the minimum spanning tree problem, *Theoretical Computer Science* 411 (2010) 2406–2413.
- [35] T. Kötzing, F. Neumann, H. Röglin, C. Witt, Theoretical analysis of two ACO approaches for the traveling salesman problem, *Swarm Intelligence* 6 (2012) 1–21. doi:10.1007/S11721-011-0059-7.
- [36] D. Sudholt, C. Thyssen, A simple ant colony optimizer for stochastic shortest path problems, *Algorithmica* 64 (2012) 643–672.
- [37] H. Mühlenbein, G. Paass, From recombination of genes to the estimation of distributions I. Binary parameters, in: *Parallel Problem Solving from Nature, PPSN 1996, Springer, 1996*, pp. 178–187.

- [38] B. Doerr, D. Johannsen, M. Schmidt, Runtime analysis of the (1+1) evolutionary algorithm on strings over finite alphabets, in: Foundations of Genetic Algorithms, FOGA 2011, ACM, 2011, pp. 119–126.
- [39] B. Doerr, S. Pohl, Run-time analysis of the (1+1) evolutionary algorithm optimizing linear functions over a finite alphabet, in: Genetic and Evolutionary Computation Conference, GECCO 2012, ACM, 2012, pp. 1317–1324.
- [40] B. Doerr, D. Sudholt, C. Witt, When do evolutionary algorithms optimize separable functions in parallel?, in: Foundations of Genetic Algorithms, FOGA 2013, ACM, 2013, pp. 48–59.
- [41] T. Kötzing, A. Lissovoi, C. Witt, (1+1) EA on generalized dynamic OneMax, in: Foundations of Genetic Algorithms, FOGA 2015, ACM, 2015, pp. 40–51.
- [42] Y. Yu, C. Qian, Z. Zhou, Switch analysis for running time analysis of evolutionary algorithms, *IEEE Transactions on Evolutionary Computation* 19 (2015) 777–792.
- [43] A. Lissovoi, C. Witt, MMAS versus population-based EA on a family of dynamic fitness functions, *Algorithmica* 75 (2016) 554–576.
- [44] B. Doerr, C. Doerr, T. Kötzing, Static and self-adjusting mutation strengths for multi-valued decision variables, *Algorithmica* 80 (2018) 1732–1768.
- [45] B. Doerr, F. Neumann (Eds.), *Theory of Evolutionary Computation—Recent Developments in Discrete Optimization*, Springer, 2020. Also available at http://www.lix.polytechnique.fr/Labo/Benjamin.Doerr/doerr_neumann_book.html.
- [46] S. Baluja, *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*, Technical Report, Carnegie Mellon University, 1994.
- [47] G. R. Harik, F. G. Lobo, D. E. Goldberg, The compact genetic algorithm, *IEEE Transactions on Evolutionary Computation* 3 (1999) 287–297.
- [48] J. L. Shapiro, The sensitivity of PBIL to its learning rate, and how detailed balance can remove it, in: Foundations of Genetic Algorithms, FOGA 2002, Morgan Kaufmann, 2002, pp. 115–132.
- [49] J. L. Shapiro, Drift and scaling in estimation of distribution algorithms, *Evolutionary Computing* 13 (2005) 99–123.
- [50] J. L. Shapiro, Diversity loss in general estimation of distribution algorithms, in: *Parallel Problem Solving from Nature, PPSN 2006*, Springer, 2006, pp. 92–101.

- [51] S. Droste, Not all linear functions are equally difficult for the compact genetic algorithm, in: Genetic and Evolutionary Computation Conference, GECCO 2005, ACM, 2005, pp. 679–686.
- [52] C. Witt, Domino convergence: why one should hill-climb on linear functions, in: Genetic and Evolutionary Computation Conference, GECCO 2018, ACM, 2018, pp. 1539–1546.
- [53] C. Witt, Upper bounds on the running time of the univariate marginal distribution algorithm on OneMax, *Algorithmica* 81 (2019) 632–667.
- [54] D. Sudholt, C. Witt, On the choice of the update strength in estimation-of-distribution algorithms and ant colony optimization, *Algorithmica* 81 (2019) 1450–1489.
- [55] J. Lengler, D. Sudholt, C. Witt, The complex parameter landscape of the compact genetic algorithm, *Algorithmica* 83 (2021) 1096–1137.
- [56] F. Neumann, D. Sudholt, C. Witt, The compact genetic algorithm struggles on Cliff functions, in: Genetic and Evolutionary Computation Conference, GECCO 2022, ACM, 2022, pp. 1426–1433.
- [57] W. Zheng, B. Doerr, From understanding genetic drift to a smart-restart mechanism for estimation-of-distribution algorithms, *Journal of Machine Learning Research* 24 (2023) 1–40.
- [58] B. Doerr, M. S. Krejca, Significance-based estimation-of-distribution algorithms, *IEEE Transactions on Evolutionary Computation* 24 (2020) 1025–1034.
- [59] A. D. Ajimakin, V. S. Devi, The competing genes evolutionary algorithm: Avoiding genetic drift through competition, local search, and majority voting, *IEEE Transactions on Evolutionary Computation* 27 (2023) 1678–1689.
- [60] G. Rudolph, *Convergence Properties of Evolutionary Algorithms*, Verlag Dr. Kováč, 1997.
- [61] B. Doerr, Probabilistic tools for the analysis of randomized optimization heuristics, in: B. Doerr, F. Neumann (Eds.), *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, Springer, 2020, pp. 1–87. Also available at <https://arxiv.org/abs/1801.06733>.