

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

Firas Ben Jedidia  
École Polytechnique  
Institut Polytechnique de Paris  
Palaiseau, France

Benjamin Doerr  
Laboratoire d'Informatique (LIX),  
CNRS, École Polytechnique  
Institut Polytechnique de Paris  
Palaiseau, France

Martin S. Krejca  
Laboratoire d'Informatique (LIX),  
CNRS, École Polytechnique  
Institut Polytechnique de Paris  
Palaiseau, France

## ABSTRACT

With apparently all research on estimation-of-distribution algorithms (EDAs) concentrated on pseudo-Boolean optimization and permutation problems, we undertake the first steps towards using EDAs for problems in which the decision variables can take more than two values, but which are not permutation problems. To this aim, we propose a natural way to extend the known univariate EDAs to such variables. Different from a naïve reduction to the binary case, it 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 variables. Roughly speaking, when the variables take  $r$  different values, the time for genetic drift to become critical 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  $O(r \log(r)^2 n^2 \log(n))$  function evaluations.

Overall, our work shows that EDAs can be adjusted to multi-valued problems and gives advice on how to set their parameters.

## CCS CONCEPTS

• Theory of computation → Evolutionary algorithms.

## KEYWORDS

Estimation-of-distribution algorithms, univariate marginal distribution algorithm, evolutionary algorithms, genetic drift, LeadingOnes benchmark.

### ACM Reference Format:

Firas Ben Jedidia, Benjamin Doerr, and Martin S. Krejca. 2023. Estimation-of-Distribution Algorithms for Multi-Valued Decision Variables. In *Genetic and Evolutionary Computation Conference (GECCO '23)*, July 15–19, 2023, Lisbon, Portugal. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3583131.3590523>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

GECCO '23, July 15–19, 2023, Lisbon, Portugal

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0119-1/23/07...\$15.00

<https://doi.org/10.1145/3583131.3590523>

## 1 INTRODUCTION

Estimation-of-distribution algorithms (EDAs [40]) 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 [25, 34]. 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 [3, 10, 27, 48].

In contrast to classic evolutionary algorithms, which have been extensively used for various types of search spaces, to the best of our knowledge, EDAs so far have been only used for problems with binary decision variables and for permutation problems.

Since this might be a lost opportunity, we undertake the first steps towards also using 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 therefore did not see how to gain any insights from these results for general multi-valued problems.

We therefore define univariate EDAs for multi-valued decision variables from scratch, that is, 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 is one. 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 exclusively, that is, the variable takes the value  $i$  with probability  $p_i$ . This mutual exclusion in the sampling immediately gives that the frequency

update does not violate the property that the frequencies add up to 1. Consequently, 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 frequencies 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.

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 EDAs [20]. 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 this quantification to multi-valued EDAs. When looking at the relatively general tools used Doerr and Zheng [20], 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 [37], 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 5.3).

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, that is, the range of parameters where the probability that a frequency gets close to a border goes asymptotically to zero. This problem 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 fast 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  $\Theta(r \log(r))$  instead of constant when the sample size  $\lambda$  is by a sufficient constant factor larger than the selection size  $\mu$ . 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  $\log(r)$  factor in the runtime. In summary, we prove that the UMDA can optimize the  $r$ -valued LEADINGONES problem in

time  $O(r(\log(r))^2 n^2 \log(n))$  (Theorem 6.1), a bound that agrees with the one shown Doerr and Krejca [14] for the classical case  $r = 2$ .

Overall, our work shows that  $r$ -valued EDAs can be effective problem solvers, and it detects no reason for the up-to-now hesitation to use such EDAs in practice.

This work 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. We hope that this eases the reading of this paper.

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 [32, 40].

Theoretically oriented works have accompanied the development and use of EDAs for a long time, see, e.g., the early works on genetic drift described in Section 5 or the report [4]. The first mathematical runtime analysis of an EDA was conducted by Droste [22]. 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 Droese, Jansen, and Wegener [23]. Interestingly, apart from the works of one research group [5–7], Droste’s ground-breaking work [22] was not followed up by other runtime analyses for around ten years. Since then, starting with works like [8, 24, 31, 45], 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 [30] for more details.

We note that there is a decent number of theoretical works on classic evolutionary algorithm for  $r$ -valued decision variables [11, 12, 17, 18, 29, 36, 49], however, we did not see how they would help us with our research question.

## 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  $\|p\|_1$ .

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  $X \leq Y$ , if and only if for all  $\lambda \in \mathbb{R}$ , we have  $\Pr[X \leq \lambda] \geq \Pr[Y \leq \lambda]$ .

## 4 MULTI-VALUED EDAS

In this section, we generalize the three common univariate EDAs for binary decision variables to multi-valued decision variables. We call these variants *multi-valued EDAs*. To this end, we briefly review the binary case in Section 4.1 before proposing our framework in Section 4.2. In our presentation, we concentrate on the UMDA [38] and then briefly discuss the generalizations of the other two common univariate 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 solutions  $x \in \{0, 1\}^n$  constructed by the frequency vector and all  $y \in \{0, 1\}^n$  (assuming that  $0^0 = 1$ ), it holds that  $\Pr[x = y] = \prod_{i \in [n]} (p_i^{y_i} \cdot (1 - p_i)^{1 - y_i})$ .

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  $\lambda$  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)* [38], which is a well established binary EDA, both in the empirical [40] and the theoretical [16] domain. Next 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  $\mu$  out of the  $\lambda$  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.

---

**Algorithm 1:** The UMDA [38] 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;
```

---

### 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 generates iteratively  $\lambda \in \mathbb{N}_{\geq 1}$  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*) 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  $y \in [0..r-1]^n$ , 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  $p_i^+ \in [a, b]^n$  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 relatively 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

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

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. 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  $\mu$  best individuals from the population of size  $\lambda$  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  $\mu$  best individuals (line 7). We note that this results in a valid frequency vector for each row  $i \in [n]$ , since

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

**$r$ -PBIL.** Another popular univariate EDA is *population-based incremental learning* (PBIL [2]). 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*.

---

**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      $\bar{p}_{i,j}^{(t+1)} \leftarrow \frac{1}{\mu} \sum_{k \in [\mu]} \mathbb{1}_{x_i^{(t,k)}=j}$ ;
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;

```

---



---

**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;

```

---

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).

**$r$ -cGA.** Another popular univariate EDA is the *compact genetic algorithm* (cGA [26]). 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

---

**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;

```

---

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 5.3) in a similar fashion as Doerr and Zheng [20] 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 [20] is that we use a slightly technical martingale concentration result due to McDiarmid [37] that allows one to profit from small variances. Such an approach is necessary. If one directly applies the methods presented by Doerr and Zheng [20], 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 on neutral positions (Theorem 5.3). Last, in Section 5.4, we consider the setting of weak preference.

Due to space limitations, some of our proofs are omitted in this paper. They can be found in the preprint [28].

### 5.1 Introduction to Genetic Drift

In EDAs, *genetic drift* means that a frequency does not reach or approach one of the extreme values 0 or 1 because of a clear signal from the objective function but due to random fluctuations from the stochasticity of the process.

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 [25] or multimodality [10], have only been shown in regimes with low genetic drift. (iii) For some particular situations, a drastic performance from genetic drift was proven. For example, the UMDA with standard selection pressure but small population size  $\lambda \in \Omega(\log(n)) \cap o(n)$  has a runtime exponential in  $\lambda$  on the DECEPTIVELEADINGBLOCKS problem [33]. In contrast, when the population size is large enough to prevent genetic drift, here  $\lambda = \Omega(n \log(n))$ , then the runtime drops to  $O(\lambda n)$  with high probability.

Genetic drift in EDAs has been studied explicitly since the ground-breaking works of Shapiro [42–44], and it appears implicitly in many runtime analyses such as [15, 21, 35, 45–47]. Experimental evidences for the negative impact of genetic drift can further be found in [20, 30, 39]. The most final answer to the genetic-drift problem for univariate EDAs, including clear suggestions to choose the parameters as to avoid genetic drift, was given by Doerr and Zheng [20]. 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  $\mu = \Omega(T \log(n))$  with implicit constant computable from [20, 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  $\mu$  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 Doerr and Zheng [19], 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 [20], 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 [20]. This observation extends the corresponding one for EDAs for binary representations. We make this statement precise for the  $r$ -UMDA.

**LEMMA 5.1 (NEUTRAL FREQUENCIES ARE MARTINGALES).** *Let  $f$  be an  $r$ -valued position, 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.*

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 [20], this reduces the time it takes a frequency to approach a border by a factor of  $\Theta(r^2)$ . We therefore use a stronger martingale concentration result, namely [37, 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 [37, 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 [20].

**THEOREM 5.2 (MARTINGALE CONCENTRATION RESULT BASED ON THE VARIANCE [37, 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  $\hat{v}_t = \sup_{s \in [t]} \text{Var}[X_s - X_{s-1} | \mathcal{F}_{s-1}]$ . 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 5.2, 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 5.3 (NEUTRAL FREQUENCIES STAY AROUND THEIR INITIAL VALUE).** *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*

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

### 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 [20] to the  $r$ -UMDA.

**THEOREM 5.4 (WEAK PREFERENCE STOCHASTICALLY DOMINATES NEUTRALITY).** *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  $q_{1,0}^{(t)} \leq p_{1,0}^{(t)}$ .*

We now apply Theorem 5.4 and extend Theorem 5.3 to positions with weak preference.

**THEOREM 5.5 (FREQUENCIES WITH WEAK PREFERENCE DO NOT GET TOO LOW).** *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*

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

**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 5.4, it follows for all  $s \in \mathbb{N}$  that  $p_{i,0}^{(s)}$  stochastically dominates  $q_{i,0}^{(s)}$ . Applying Theorem 5.3 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 6.1, 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. [7] with a proof that the UMDA with population size  $\Omega(n^{2+\varepsilon})$  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 [8], Dang and Lehre prove a runtime of  $O(n\lambda \log(\lambda) + n^2)$ , only assuming that the sample size  $\lambda$  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  $\lambda$  is slightly worse than in the result by Chen et al. [7]. This was improved by Doerr and Krejca [14], where an  $O(n\lambda \log(\lambda))$  upper bound was shown for the whole regime  $\lambda = \Omega(n \log(n))$  of low genetic drift. More precisely, when  $\mu = \Omega(n \log(n))$  and  $\lambda = \Omega(\mu)$ , both with sufficiently large implicit constants, then the runtime of the UMDA on LEADINGONES is  $O(n\lambda \log(\frac{\lambda}{\mu}))$  with high probability. We note that the analysis by Doerr and Krejca [14] is technically much simpler than the previous ones, in particular, it avoids the complicated level-based method

used by Dang and Lehre [8]. We note that also lower bounds [14, 34] 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  $O(n \log(n))$  with high probability. This runtime is faster by a factor of  $n/\log(n)$  when compared to classical EAs, and it suggests that LEADINGONES is a rather easy problem for EDAs. Friedrich, Kötzing, and Krejca [24] 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 [13], highlighting that the scGA is not a good EDA in general.

The next result was proven by Doerr and Krejca [13], 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.

Last, a result was recently proven by Ajimakin and Devi [1], 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.

## 6.2 The $r$ -LEADINGONES Benchmark

The  $r$ -LEADINGONES function (eq. (3)) is a generalization of the classical LEADINGONES benchmark [41] from the binary to the multi-valued domain. Before we define the generalization, we briefly present the LEADINGONES function.

**LEADINGONES.** LEADINGONES [41] is one of the most commonly mathematically analyzed benchmark functions, both in the general domain of evolutionary computation [16] as well as in the domain of EDAs [30]. 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,

$$r\text{-LEADINGONES}: x \mapsto \sum_{i \in [n]} \prod_{j \in [i]} \mathbb{1}_{\{x_j=0\}}. \quad (3)$$

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 Result

We analyze the runtime of the  $r$ -UMDA (Algorithm 2) on the  $r$ -LEADINGONES benchmark (eq. (3)) in the regime with low genetic drift. Compared to the binary case, we get an extra factor of order  $r \log(r)^2$  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  $\log(r)^2$  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. Our result is a generalization of the binary case.

**THEOREM 6.1 ( $r$ -UMDA EFFICIENTLY OPTIMIZES  $r$ -LEADINGONES).** *Let  $s \in \mathbb{R}_{\geq 1}$ . Consider the  $r$ -UMDA optimizing  $r$ -LEADINGONES with  $\lambda \geq 3se\mu$ ,  $\mu \geq 24(n+1)r \ln(n)(1 + \log_{2s}(r))$ , and  $n \geq 4r$ . Then with probability at least  $1 - \frac{2}{n} - \log_{2s}(2r)n^{2-0.5n} =: q$ , the frequency vector of the value 0 is  $(1 - \frac{1}{n})_{i \in [n]}$  after  $n \log_{2s}(2r)$  iterations.*

*This implies that with probability at least  $q$ , after  $\lambda n \log_{2s}(2r)$  fitness function evaluations, the  $r$ -UMDA samples the optimum.*

The premise for our proof is that for the entirety of the considered iterations, frequencies of 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 of 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 6.2 (FREQUENCIES DO NOT GET LOW UNDER WEAK PREFERENCE).** *Let  $s \in \mathbb{R}_{\geq 1}$ . Consider the  $r$ -UMDA with  $\lambda \geq \mu \geq 24(n+1)r \ln(n)(1 + \log_{2s}(r))$  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  $n(1 + \log_{2s}(r))$  iterations.*

**PROOF.** By Theorem 5.5 with  $T = n(1 + \log_{2s}(r))$ , we have for all  $i \in [n]$  that

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

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

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

Hence, it follows that  $\Pr[\min_{k=1, \dots, T} p_{i,0}^{(k)} \leq \frac{1}{2r}] \leq \frac{2}{n^2}$ . Applying a union bound over all  $n$  positions yields the result.  $\square$

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

**THEOREM 6.3 (CHERNOFF BOUND [9, 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  $\Pr[X \leq (1 - \delta) \mathbb{E}[X]] \leq \exp(-\frac{\delta^2 \mathbb{E}[X]}{2})$ .*

An important concept for our analysis, following the approach by Doerr and Krejca [14], is that a position is *critical*. Informally, a position is critical if and only if the frequencies corresponding to value 0 are for all smaller positions 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  $\lambda$  from  $\mu$  (that is, defining the selection pressure), it takes less than  $n \log_{2s}(r+1)$  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 6.4 (CRITICAL FREQUENCIES ARE QUICKLY MAXIMIZED AND REMAIN THERE FOR SOME TIME).** *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  $1 - u \log_{2s}(\frac{1}{b}) \exp(-\frac{s\mu b}{24})$ , it holds for all  $\theta \in [\log_{2s}(\frac{1}{b}) \cdot u \log_{2s}(\frac{1}{b})]$  that  $p_{i,0}^{(t+\theta)} = 1 - \frac{1}{n}$ .*

**PROOF.** We start by proving that, for all  $\theta \in [0..u \log_{2s}(\frac{1}{b})]$ , 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, and 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 6.3) 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

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

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

## 7 CONCLUSION

We have proposed the first EDAs to optimize 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  $\mu$  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 permutations and binary decision variables.

## 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] Adetunji David Ajimakin and V. Susheela Devi. The competing genes evolutionary algorithm: Avoiding genetic drift through competition, local search, and majority voting. *IEEE Transactions on Evolutionary Computation*, 2022. To appear.
- [2] Shumeet Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, Carnegie Mellon University, 1994.
- [3] Riade Benbaki, Ziyad Benomar, and Benjamin Doerr. A rigorous runtime analysis of the 2-MMAS<sub>ib</sub> on jump functions: ant colony optimizers can cope well with local optima. In *Genetic and Evolutionary Computation Conference, GECCO 2021*, pages 4–13. ACM, 2021.
- [4] Josu Ceberio Uribe, Benjamin Doerr, Carsten Witt, and Vicente P. Soloviev. Estimation-of-Distribution Algorithms: Theory and Applications (Dagstuhl Seminar 22182). *Dagstuhl Reports*, 12:17–36, 2022.
- [5] Tianshi Chen, Per Kristian Lehre, Ke Tang, and Xin Yao. When is an estimation of distribution algorithm better than an evolutionary algorithm? In *Congress on Evolutionary Computation, CEC 2009*, pages 1470–1477. IEEE, 2009.
- [6] Tianshi Chen, Ke Tang, Guoliang Chen, and Xin Yao. On the analysis of average time complexity of estimation of distribution algorithms. In *Congress on Evolutionary Computation, CEC 2007*, pages 453–460. IEEE, 2007.
- [7] Tianshi Chen, Ke Tang, Guoliang Chen, and Xin Yao. Analysis of computational time of simple estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation*, 14:1–22, 2010.
- [8] Duc-Cuong Dang and Per Kristian Lehre. Simplified runtime analysis of estimation of distribution algorithms. In *Genetic and Evolutionary Computation Conference, GECCO 2015*, pages 513–518. ACM, 2015.
- [9] Benjamin Doerr. Probabilistic tools for the analysis of randomized optimization heuristics. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, pages 1–87. Springer, 2020. Also available at <https://arxiv.org/abs/1801.06733>.
- [10] Benjamin Doerr. The runtime of the compact genetic algorithm on Jump functions. *Algorithmica*, 83:3059–3107, 2021.
- [11] Benjamin Doerr, Carola Doerr, and Timo Kötzing. Static and self-adjusting mutation strengths for multi-valued decision variables. *Algorithmica*, 80:1732–1768, 2018.
- [12] Benjamin Doerr, Daniel Johannsen, and Martin Schmidt. Runtime analysis of the (1+1) evolutionary algorithm on strings over finite alphabets. In *Foundations of Genetic Algorithms, FOGA 2011*, pages 119–126. ACM, 2011.
- [13] Benjamin Doerr and Martin S. Krejca. Significance-based estimation-of-distribution algorithms. *IEEE Transactions on Evolutionary Computation*, 24:1025–1034, 2020.
- [14] Benjamin Doerr and Martin S. Krejca. A simplified run time analysis of the univariate marginal distribution algorithm on LeadingOnes. *Theoretical Computer Science*, 851:121–128, 2021.
- [15] Benjamin Doerr and Martin S. Krejca. The univariate marginal distribution algorithm copes well with deception and epistasis. *Evolutionary Computation*, 29:543–563, 2021.
- [16] Benjamin Doerr and Frank Neumann, editors. *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](http://www.lix.polytechnique.fr/Labo/Benjamin.Doerr/doerr_neumann_book.html).
- [17] Benjamin Doerr and Sebastian 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*, pages 1317–1324. ACM, 2012.
- [18] Benjamin Doerr, Dirk Sudholt, and Carsten Witt. When do evolutionary algorithms optimize separable functions in parallel? In *Foundations of Genetic Algorithms, FOGA 2013*, pages 48–59. ACM, 2013.
- [19] Benjamin Doerr and Weijie Zheng. From understanding genetic drift to a smart-restart parameter-less compact genetic algorithm. In *Genetic and Evolutionary Computation Conference, GECCO 2020*, pages 805–813. ACM, 2020.
- [20] Benjamin Doerr and Weijie Zheng. Sharp bounds for genetic drift in estimation-of-distribution algorithms. *IEEE Transactions on Evolutionary Computation*, 24:1140–1149, 2020.
- [21] Stefan Droste. Not all linear functions are equally difficult for the compact genetic algorithm. In *Genetic and Evolutionary Computation Conference, GECCO 2005*, pages 679–686. ACM, 2005.
- [22] Stefan Droste. A rigorous analysis of the compact genetic algorithm for linear functions. *Natural Computing*, 5:257–283, 2006.
- [23] Stefan Droste, Thomas Jansen, and Ingo Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.
- [24] Tobias Friedrich, Timo Kötzing, and Martin S. Krejca. EDAs cannot be balanced and stable. In *Genetic and Evolutionary Computation Conference, GECCO 2016*, pages 1139–1146. ACM, 2016.
- [25] Tobias Friedrich, Timo Kötzing, Martin S. Krejca, and Andrew M. Sutton. The compact genetic algorithm is efficient under extreme Gaussian noise. *IEEE Transactions on Evolutionary Computation*, 21:477–490, 2017.
- [26] Georges R. Harik, Fernando G. Lobo, and David E. Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3:287–297, 1999.
- [27] Václav Hasenöhrl and Andrew M. Sutton. On the runtime dynamics of the compact genetic algorithm on jump functions. In *Genetic and Evolutionary Computation Conference, GECCO 2018*, pages 967–974. ACM, 2018.
- [28] Firas Ben Jedidia, Benjamin Doerr, and Martin S. Krejca. Estimation-of-distribution algorithms for multi-valued decision variables. *CoRR*, abs/2302.14420, 2023.
- [29] Timo Kötzing, Andrei Lissovoi, and Carsten Witt. (1+1) EA on generalized dynamic OneMax. In *Foundations of Genetic Algorithms, FOGA 2015*, pages 40–51. ACM, 2015.
- [30] Martin Krejca and Carsten Witt. Theory of estimation-of-distribution algorithms. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, pages 405–442. Springer, 2020. Also available at <https://arxiv.org/abs/1806.05392>.
- [31] Martin S. Krejca and Carsten Witt. Lower bounds on the run time of the Univariate Marginal Distribution Algorithm on OneMax. *Theoretical Computer Science*, 832:143–165, 2020.
- [32] Pedro Larrañaga and José Antonio Lozano, editors. *Estimation of Distribution Algorithms*. Springer, 2002.
- [33] Per Kristian Lehre and Phan Trung Hai 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*, pages 154–168. ACM, 2019.
- [34] Per Kristian Lehre and Phan Trung Hai Nguyen. Runtime analysis of the univariate marginal distribution algorithm under low selective pressure and prior noise. In *Genetic and Evolutionary Computation Conference, GECCO 2019*, pages 1497–1505. ACM, 2019.
- [35] Johannes Lengler, Dirk Sudholt, and Carsten Witt. The complex parameter landscape of the compact genetic algorithm. *Algorithmica*, 83:1096–1137, 2021.
- [36] Andrei Lissovoi and Carsten Witt. MMAS versus population-based EA on a family of dynamic fitness functions. *Algorithmica*, 75:554–576, 2016.
- [37] Colin McDiarmid. Concentration. In *Probabilistic Methods for Algorithmic Discrete Mathematics*, volume 16, pages 195–248. Springer, Berlin, 1998.
- [38] Heinz Mühlenbein and Gerhard Paass. From recombination of genes to the estimation of distributions I. Binary parameters. In *Parallel Problem Solving from Nature, PPSN 1996*, pages 178–187. Springer, 1996.
- [39] Frank Neumann, Dirk Sudholt, and Carsten Witt. The compact genetic algorithm struggles on Cliff functions. In *Genetic and Evolutionary Computation Conference, GECCO 2022*, pages 1426–1433. ACM, 2022.
- [40] Martin Pelikan, Mark Hauschild, and Fernando G. Lobo. Estimation of distribution algorithms. In Janusz Kacprzyk and Witold Pedrycz, editors, *Springer Handbook of Computational Intelligence*, pages 899–928. Springer, 2015.
- [41] Günter Rudolph. *Convergence Properties of Evolutionary Algorithms*. Verlag Dr. Kováč, 1997.
- [42] Jonathan L. Shapiro. The sensitivity of PBIL to its learning rate, and how detailed balance can remove it. In *Foundations of Genetic Algorithms, FOGA 2002*, pages 115–132. Morgan Kaufmann, 2002.
- [43] Jonathan L. Shapiro. Drift and scaling in estimation of distribution algorithms. *Evolutionary Computing*, 13:99–123, 2005.
- [44] Jonathan L. Shapiro. Diversity loss in general estimation of distribution algorithms. In *Parallel Problem Solving from Nature, PPSN 2006*, pages 92–101. Springer, 2006.
- [45] Dirk Sudholt and Carsten Witt. On the choice of the update strength in estimation-of-distribution algorithms and ant colony optimization. *Algorithmica*, 81:1450–1489, 2019.
- [46] Carsten Witt. Domino convergence: why one should hill-climb on linear functions. In *Genetic and Evolutionary Computation Conference, GECCO 2018*, pages 1539–1546. ACM, 2018.
- [47] Carsten Witt. Upper bounds on the running time of the univariate marginal distribution algorithm on OneMax. *Algorithmica*, 81:632–667, 2019.
- [48] Carsten Witt. How majority-vote crossover and estimation-of-distribution algorithms cope with fitness valleys. *Theoretical Computer Science*, 940:18–42, 2023.
- [49] Yang Yu, Chao Qian, and Zhi-Hua Zhou. Switch analysis for running time analysis of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 19:777–792, 2015.

## APPENDIX

We state the proofs that we omitted in the main paper, due to space limitations.

### Proof of Lemma 5.1

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  $\mu$  best individuals is not affected by the values at position  $i$  of the  $\lambda$  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,  $\mathbb{E}[\mathbb{1}_{x_i^{(t,k)}=j} | p_{i,j}^{(t)}] = p_{i,j}^{(t)}$ . Further, by linearity of expectation, we get

$$\begin{aligned} \mathbb{E}[p_{i,j}^{(t+1)} | p_{i,j}^{(t)}] &= \frac{1}{\mu} \sum_{k \in [\mu]} \mathbb{E}[\mathbb{1}_{x_i^{(t,k)}=j} | p_{i,j}^{(t)}] \\ &= \frac{1}{\mu} \sum_{k \in [\mu]} p_{i,j}^{(t)} = p_{i,j}^{(t)}, \end{aligned}$$

proving the claim.

### Proof of Theorem 5.3

We apply the same proof strategy as in the proof of [20, Theorem 1]. That is, we aim to apply Theorem 5.2. 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  $\mu$  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 5.2, 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  $\mu$  best individuals is not affected by the values at position  $i$  of the  $\lambda$  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

$$\mathbb{E}[Z_{t\mu+a+1} - Z_{t\mu+a} | \mathcal{F}_{t\mu+a}] = -p_{i,j}^{(t)} + \mathbb{E}[\mathbb{1}_{x_i^{(t,a+1)}=j} | \mathcal{F}_{t\mu+a}] = 0, \quad (4)$$

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

$$\begin{aligned} &\mathbb{E}[Z_{(t+1)\mu} - Z_{t\mu+\mu-1} | \mathcal{F}_{t\mu+\mu-1}] \\ &= \mu \mathbb{E}[p_{i,j}^{(t+1)} | \mathcal{F}_{t\mu+\mu-1}] - p_{i,j}^{(t)} \\ &\quad - \mathbb{E}[\sum_{k \in [\mu-1]} \mathbb{1}_{x_i^{(t,k)}=j} | \mathcal{F}_{t\mu+\mu-1}] \\ &= \sum_{k \in [\mu]} \mathbb{E}[\mathbb{1}_{x_i^{(t,k)}=j} | \mathcal{F}_{t\mu+\mu-1}] \\ &\quad - p_{i,j}^{(t)} - \sum_{k \in [\mu-1]} \mathbb{E}[\mathbb{1}_{x_i^{(t,k)}=j} | \mathcal{F}_{t\mu+\mu-1}] \\ &= \mathbb{E}[\mathbb{1}_{x_i^{(t,\mu)}=j} | \mathcal{F}_{t\mu+\mu-1}] - p_{i,j}^{(t)} = 0, \end{aligned} \quad (5)$$

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. (4) and (5), disregarding the expected values, by eq. (6), it holds that

$$\tilde{Z}_{t+1} - \tilde{Z}_t = (Y_t - p_{i,j}^{(\lfloor t/\mu \rfloor)}) \cdot \mathbb{1}_{t < S}. \quad (6)$$

Thus, the maximum deviation  $b$  of  $\tilde{Z}$  is 1. Further, let  $\hat{v}_t$  denote the sum of variances, as defined in Theorem 5.2. Then, since  $p_{i,j}^{(\lfloor t/\mu \rfloor)}$  and  $\mathbb{1}_{t < S}$  are  $\mathcal{F}_t$ -measurable and since, due to  $\tilde{Z}$  being stopped, it holds that  $p_{i,j}^{(\lfloor t/\mu \rfloor)} \cdot \mathbb{1}_{t < S} \in [\frac{1}{2r}, \frac{3}{2r}]$ , we get

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

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 5.2 with  $t = \mu T$  and our estimates above, noting that  $\tilde{Z}_0 = \frac{\mu}{r}$ , yields

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

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.

### Proof of Theorem 5.4

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

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  $\mu$  best individuals is not affected by the values at position 0 of the  $\lambda$  samples. Thus,  $q_{1,j}^{(t+1)} = \frac{1}{\mu} Y_j$ . Further, since  $f$  weakly prefers 0s, defining  $Y'_j \sim \text{Bin}(\mu, p_{0,j}^{(t)})$ , it holds that  $p_{1,j}^{t+1} \geq \frac{1}{\mu} Y'_j$ .

Analogously to Doerr and Zheng [20], 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  $w$  in the coupling probability space. For such a  $w$ , 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.

### Proof of Theorem 6.1

Since  $r$ -LEADINGONES weakly prefers 0s at all positions  $i \in [n]$ , by Lemma 6.2, 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  $n(1 + \log_{2s}(r))$  iterations.

For each position  $i \in [n]$ , we apply Lemma 6.4 with  $b = \frac{1}{2r}$  and  $u = n$ , 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  $1 - \log_{2s}(2r)n^{1-0.5n}$ , after at most  $\log_{2s}(2r)$  iterations,

the frequency  $p_{i,0}$  is set to  $1 - \frac{1}{n}$  and remains there for at least  $(n-1)\log_{2s}(2r)$  iterations. Further, by a union bound over all  $n$  frequency vectors, the above holds for all frequency vectors, with probability at least  $1 - \log_{2s}(2r)n^{2-0.5n}$ .

Combining everything, with probability at least  $1 - \frac{2}{n} - \log_{2s}(2r)n^{2-0.5n}$ , 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  $\log_{2s}(2r)$  iterations and remains there until at least iteration  $n\log_{2s}(2r)$ . 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  $n\log_{2s}(2r)$  iterations, all such frequencies are at the upper border, which proves the first part of the claim.

For the second part, note that once  $p_{n,0} = 1 - \frac{1}{n}$ , the population of the  $r$ -UMDA in that iteration contains at least  $(1 - \frac{1}{n})\mu$  times the optimum. Further, each iteration accounts for  $\lambda$  fitness function evaluations. This proves the second claim.