Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Unbiasedness of estimation-of-distribution algorithms

Tobias Friedrich, Timo Kötzing, Martin S. Krejca *

Hasso Platter Institute, University of Potsdam, Potsdam, Germany

ARTICLE INFO

Article history:
Received 3 July 2018
Received in revised form 1 November 2018
Accepted 1 November 2018
Available online xxxx
Communicated by C. Witt

Keywords:
Estimation-of-distribution algorithm
Unbiasedness
Theory

ABSTRACT

In the context of black-box optimization, black-box complexity is used for understanding the inherent difficulty of a given optimization problem. Central to our understanding of nature-inspired search heuristics in this context is the notion of unbiasedness. Specialized black-box complexities have been developed in order to better understand the limitations of these heuristics – especially of (population-based) evolutionary algorithms (EAs). In contrast to this, we focus on a model for algorithms explicitly maintaining a probability distribution over the search space: so-called estimation-of-distribution algorithms (EDAs). We consider the recently introduced n-Bernoulli-λ-EDA framework, which subsumes, for example, the commonly known EDAs PBIL, UMDA, λ-MMAS$_{\text{IB}}$, and cGA. We show that an n-Bernoulli-λ-EDA is unbiased if and only if its probability distribution satisfies a certain invariance property under isometric automorphisms of $[0,1]^n$. By restricting how an n-Bernoulli-λ-EDA can perform an update, in a way common to many examples, we derive conciser characterizations, which are easy to verify. We demonstrate this by showing that our examples above are all unbiased.

1. Introduction

We consider pseudo-Boolean optimization, that is, optimization of functions $f\colon \{0,1\}^n \to \mathbb{R}$, via randomized search heuristics (RSHs). These algorithms are commonly viewed as problem-agnostic solvers, and the only way to get any information about the problem at hand is to query a subroutine that serves as an oracle. Thus, the problem itself can be seen as a black box to the algorithm, and this scenario of optimization is called *black-box optimization*.

Black-box complexity. When optimizing in a black-box scenario, the number of calls to the subroutine is of great importance, since the other operations of an RSH are usually very cheap. Thus, the subroutine calls dominate the overall cost of the respective algorithm. Coming from this point of view, Droste et al. [1] introduced in 2006 a new complexity theory for RSHs: the so-called black-box complexity (BBC). In this model, a randomized search heuristic is assumed to solve a problem by querying the subroutine for potential solutions (modeled as bit strings) and then receiving feedback in form of the solution's quality. The worst-case complexity of a specific algorithm on a problem class is defined as the expected number of queries to the subroutine on the worst-case instance of this class; and the BBC of a class is the best possible worst-case complexity, taken over all black-box algorithms.

Although this definition captures the initial idea of subroutine calls very well, it does not prohibit the algorithm to be highly problem-specific. Droste et al. [2], for example, showed that the Max-Clique problem, which is NP-hard in classical complexity theory, has a polynomial BBC. This is due to the respective algorithm learning the instance via queries and

* Corresponding author.
E-mail addresses: tobias.friedrich@hpi.de (T. Friedrich), timo.koetzing@hpi.de (T. Kötzing), martin.krejca@hpi.de (M.S. Krejca).

https://doi.org/10.1016/j.tcs.2018.11.001
0304-3975/

then performing costly offline computations, which do not increase the BBC. This highlights that the BBC alone does not necessarily provide sufficient information about the true complexity of a problem. In addition to that, Anil and Wiegand [3] showed that the BBC of OneMax – a common benchmark problem in the run time analysis of evolutionary algorithms (EAs) – is $\Theta(n/\log n)$,[1] whereas traditional EAs, such as the $(1+1)$-EA, usually have an expected complexity of $\Theta(n \log n)$ on OneMax [2].

In order to focus on more problem-agnostic solvers, Lehre and Witt [5] restricted the BBC model originally introduced by Droste et al. [1] such that the algorithms considered can only make use of *unbiased* variation operators when querying the next bit string and the algorithms are not allowed to see the representational structure of solutions during selection. This restricted variant is called *unbiased black-box complexity*. *Unbiased* means that the respective algorithm performs the same when given two problem instances where one is a perturbation of the other, restricted to permutations and bit flips. Hence, the algorithm is unbiased with respect to 0s and 1s as well as their positions, and it has to treat them the same way. Rowe and Vose [6] propose a more general definition of being *unbiased*, which focuses on the sequence of queried bit strings (the *trace*) of the algorithm under consideration. They call an algorithm unbiased if it behaves the same under any permutation of the search space that the problem class is closed under. When considering problem classes closed under the aforementioned perturbations, this definition results in the unbiased BBC from Lehre and Witt [5] without the restriction on the selection process. Further, Rowe and Vose [6] show that, for any BB algorithm on a problem class, there exists an unbiased algorithm (in their sense) that is at least as good. This means that the BBC of a problem class is the same as its unbiased BBC. Hence, it suffices to study unbiased algorithms when one is interested in the BBC of a class.

Doerr et al. [7] show that the unbiased BBC in the sense of Lehre and Witt [5] of different subproblems of the NP-complete Partition problem is polynomial when using higher-arity operators. This has lead to other restricted BBCs [8,9] or combinations of them [10]. However, all of these models have their drawbacks, such that the unbiased BBC model is still considered an important complexity measure up to this date [11–15].

Estimation-of-distribution algorithms. Surprisingly, all of the results for unbiased BBC so far only considered population-based EAs. That is, a query to the black box always results in a potential solution (an *individual*) that is either added to the current set of individuals (the *population*) or discarded. That means that there are no unbiased BBC results for any other class of black-box optimization algorithms. One such class ignored so far are *estimation-of-distribution algorithms* (EDAs [16]): RSHs that maintain an explicit probability distribution over the search space and only sample individuals (according to the algorithm's current distribution) to update this distribution instead of maintaining an explicit population.

While there has been an increased amount of theoretical run time results of EDAs lately [17–25], structural results on EDAs remain scarce. The run time results proving lower bounds [18,19,25] give some structural insights into the behavior of EDAs by showing how the update to the probability distribution of an EDA affects its run time. However, these results only consider specific algorithms. We are interested in general structural properties of EDAs.

Friedrich et al. [26] observed that the algorithms considered in these papers above can be subsumed into a concise framework, which they call n-Bernoulli-λ-EDA. Such an algorithm maintains a multivariate binomial distribution, which is represented by a single vector $\boldsymbol{p} \in [0,1]^n$ – each component of the vector corresponding to a bit position of the individuals being sampled. This means that the probability of a sampled individual having a 1 at position $i \in \{1,\dots,n\}$ is given by $\boldsymbol{p}_i$ and is drawn independently of the other bits. The function that performs the update of $\boldsymbol{p}$ is called the *update scheme*. Friedrich et al. [26] show that the commonly investigated EDAs PBIL [27], UMDA [28], λ-MMAS$_{\text{ib}}$ [29,30], cGA [31], and even the $(1,\lambda)$-EA [32] – normally considered a population-based EA – all fall into the n-Bernoulli-λ-EDA framework.

Interestingly, to the best of our knowledge, all common EDAs are unbiased. One possible explanation for this may be that the sample procedure is unbiased and that the samples are not altered afterwards. Thus, a bias could only be introduced via the way the probability distribution is updated. However, as we show in Section 4, at least for the n-Bernoulli-λ-EDA framework, such an update usually exhibits a property that strongly favors unbiasedness.

Our results. We combine the up to now unrelated fields of unbiased BBC and EDAs, and we characterize unbiasedness for the entire class of n-Bernoulli-λ-EDAs. This characterization (Theorem 6) shows strong similarities to the original definition of unbiased algorithms, introduced by Lehre and Witt [5], and it provides insights into how the properties of an unbiased population-based algorithm extend to EDAs. We then restrict the update scheme of our considered algorithms to better suit the more specific update schemes of the previously mentioned examples, leading to characterizations that are easier to verify than the general one. We especially find a very concise characterization for *locally updating* n-Bernoulli-λ-EDAs – a concept that was introduced by Friedrich et al. [26] but not considered in depth. We also prove that all of the examples mentioned above are unbiased.

Overview. This paper is structured as follows: in Section 2, we introduce some notation, explain the n-Bernoulli-λ-EDA framework, give some examples, and talk about automorphisms of $[0,1]^n$, which we need for our characterization. Section 3 contains our main result: Theorem 6, which characterizes the unbiasedness of n-Bernoulli-λ-EDAs. In Section 4, we consider n-Bernoulli-λ-EDAs that use the same update function for each component. We characterize unbiasedness in this setting and prove for some examples that they are unbiased. In Section 5, we restrict the class of algorithms even further and only

[1] Actually, this result dates back to Erdős and Rényi [4], who proved it in the context of information theory.

Please cite this article in press as: T. Friedrich et al., Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.11.001

Algorithm 1: The n-Bernoulli-λ-EDA with a given update scheme φ.

1 $t \leftarrow 0$;
2 $\boldsymbol{p}^{(t)} \leftarrow \frac{1}{2}$;
3 **repeat**
4 $D \leftarrow \emptyset$;
5 **for** $j \in [\lambda]$ **do**
6 $\boldsymbol{x} \leftarrow$ offspring sampled with respect to $\boldsymbol{p}^{(t)}$;
7 $D \leftarrow D \cup \{\boldsymbol{x}\}$;
8 $\boldsymbol{p}^{(t+1)} \leftarrow \varphi\left(\boldsymbol{p}^{(t)}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)$;
9 $t \leftarrow t+1$;
10 **until** optimum is in D;

consider locally updating n-Bernoulli-λ-EDAs. We characterize their unbiasedness and prove it for some examples. Last, we conclude our work in Section 6.

2. Preliminaries

We call an objective function $f\colon \{0,1\}^n \to \mathbb{R}$ a *fitness function*. The dimension of such a function's domain is denoted by $n \in \mathbb{N}$. For any $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1,\dots,n\}$. For any number $a \in [0,1]$, we denote the all-a vector of length n by $\mathbf{a}$, that is, $\mathbf{a} = (a)_{i\in[n]}$. Further, for any vector $\boldsymbol{x} \in [0,1]^n$, we denote the i-th component ($i \in [n]$) by $\boldsymbol{x}_i$. Given a fitness function f and a bit string $\boldsymbol{x} \in \{0,1\}^n$ (also called an *individual*), we call $f(\boldsymbol{x})$ the *fitness* of $\boldsymbol{x}$. When considering two random variables X and Y, we let $X = Y$ denote that both random variables take the same value with the same probability for each value possible.

2.1. Framework

Throughout this paper, we focus on the n-Bernoulli-λ-EDA framework (Algorithm 1), introduced by Friedrich et al. [26]. An n-Bernoulli-λ-EDA maintains a *frequency vector* $\boldsymbol{p}$, whose components we call *frequencies*. Each iteration, the algorithm samples λ offspring whose number of 1s follows a Poisson binomial distribution induced by the frequency vector. That is, bit $i \in [n]$ of an individual $\boldsymbol{x}$ from the *offspring population* D is 1 with probability $\boldsymbol{p}_i$ independently from all its other bits: $\forall \boldsymbol{x} \in D, i \in [n]\colon \Pr[\boldsymbol{x}_i = 1] = \boldsymbol{p}_i \wedge \Pr[\boldsymbol{x}_i = 0] = 1 - \boldsymbol{p}_i$.

The essential part of an n-Bernoulli-λ-EDA is its *update scheme* $\varphi\colon [0,1]^n \times (\{0,1\}^n \times \mathbb{R})^\lambda \to [0,1]^n$ (line 7 of Algorithm 1). This update scheme is used to update the algorithm's frequency vector each iteration, based on the old frequency vector, its samples, and their respective fitness. Note that an update scheme fully characterizes an n-Bernoulli-λ-EDA and that φ is a random variable for our considerations, as the population D sampled on every iteration is random. For our examples below, this comes into play when considering tie-breaking rules when comparing the fitness of samples.

Given an offspring population D, we say that $\boldsymbol{x} \in D$ has a *rank* $i \in [\lambda]$ (denoted as $\boldsymbol{x}^{(i)}$) if $\boldsymbol{x}$ has the i-th best fitness in D.[2] Since we want ranks to be unique, we assume that ties are broken uniformly at random. Note that a common way for an algorithm to determine the rank of an individual is to sort the population by fitness.

Friedrich et al. [26] call an n-Bernoulli-λ-EDA *locally updating* if its update scheme can be split into two functions 'move': $(\{0,1\} \times \mathbb{R})^\lambda \to \{\text{up}, \text{stay}, \text{down}\}$ and 'set': $[0,1] \to [0,1]$ such that, for all $i \in [n]$, abbreviating $v_i = \text{move}\left(\left(\boldsymbol{x}_i, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)$,

$$\varphi\left(\boldsymbol{p}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)_i = \begin{cases} \text{set}(\boldsymbol{p}_i) & \text{if } v_i = \text{up}\,; \\ \boldsymbol{p}_i & \text{if } v_i = \text{stay}\,; \\ 1 - \text{set}(1-\boldsymbol{p}_i) & \text{if } v_i = \text{down}\,. \end{cases}$$

However, in contrast to Friedrich et al. [26], we additionally assume that $\text{set}(\boldsymbol{p}_i) \geq \boldsymbol{p}_i$. This way, we enforce that 'up' does not decrease the frequency. Note that this also entails that 'down' does not increase the frequency. Without this assumption, 'down', for example, could take the role of 'up', which would result in convoluted case distinctions in proofs. By assuming $\text{set}(\boldsymbol{p}_i) \geq \boldsymbol{p}_i$, we avoid such overlaps.

Example 1. Friedrich et al. [26] mention five algorithms that are commonly analyzed in theory. We briefly state their update schemes, since we prove later that these algorithms are all unbiased. The update scheme is always shown for a single component $i \in [n]$, since the same update is used for every component.

[2] Note that *best* is relative to whether the objective is maximization or minimization. For maximization, for example, the best fitness would be the highest value.

Please cite this article in press as: T. Friedrich et al., Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.11.001

PBIL [27]: The Population-based Incremental Learning algorithm uses two different parameters: $\rho \in [0,1]$ and $\mu \leq \lambda$.

$$\varphi\left(\boldsymbol{p}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)_i = (1-\rho)\boldsymbol{p}_i + \rho\frac{1}{\mu}\sum_{j=1}^{\mu}\boldsymbol{x}_i^{(j)}\,.$$

If $\rho = 1$, the resulting algorithm is called the Univariate Marginal Distribution Algorithm (UMDA [28]); and if $\mu = 1$, then the resulting algorithm is called λ-AS$_{\text{IB}}$.[3] Note that the PBIL is not locally updating for $\mu > 1$, since the amount by which $\boldsymbol{p}_i$ is updated is not only dependent on $\boldsymbol{p}_i$ itself but additionally on the bit values at position i of all of the sampled individuals, which results in more than three cases.

cGA [31]: The Compact Genetic Algorithm uses a single parameter $K \in \mathbb{N}^+$ and an offspring size $\lambda = 2$. We state the update scheme with respect to the cGA being locally updating [26]:

$$\text{move}\left(\left(\boldsymbol{x}_i, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right) = \begin{cases} \text{up} & \text{if } \boldsymbol{x}_i^{(1)} > \boldsymbol{x}_i^{(2)}\,, \\ \text{down} & \text{if } \boldsymbol{x}_i^{(1)} < \boldsymbol{x}_i^{(2)}\,, \\ \text{stay} & \text{if } \boldsymbol{x}_i^{(1)} = \boldsymbol{x}_i^{(2)}\,; \end{cases}$$

$$\text{set}(\boldsymbol{p}_i) = \min\left\{\boldsymbol{p}_i + \frac{1}{K}, 1\right\}\,.$$

$(1,\lambda)$-EA [32]: The $(1,\lambda)$-Evolutionary Algorithm is typically not considered to be an EDA, but it fits into the (locally updating) n-Bernoulli-λ-EDA framework, as observed by Friedrich et al. [26]. Its parameter λ is the same as the offspring population size λ of an n-Bernoulli-λ-EDA.

$$\text{move}\left(\left(\boldsymbol{x}_i, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right) = \begin{cases} \text{up} & \text{if } \boldsymbol{x}_i^{(1)} = 1\,, \\ \text{down} & \text{if } \boldsymbol{x}_i^{(1)} = 0\,; \end{cases}$$

$$\text{set}(\boldsymbol{p}_i) = 1 - \frac{1}{n}\,.$$

Friedrich et al. [26] also state that λ-AS$_{\text{IB}}$ has the same 'move' function as the $(1,\lambda)$-EA but the following 'set' function: $\text{set}(\boldsymbol{p}_i) = \boldsymbol{p}_i + \rho(1-\boldsymbol{p}_i)$.

Given an n-Bernoulli-λ-EDA A, let $P^{(t)}(A)$ denote the set of all possible frequency vectors of A in iteration t, and let $P(A) := \bigcup_{t=0}^{\infty} P^{(t)}(A)$ denote the set of all possible frequency vectors of A. Analogously, $P_i^{(t)}(A) \subseteq [0,1]$ denotes the set of possible values of the frequency $\boldsymbol{p}_i^{(t)}$, and $P_i(A)$ denotes the range of all possible values of $\boldsymbol{p}_i$. For example, for the UMDA, $P(A)$ would consist of all the vectors from $[0,1]^n$ whose components are of the form j/μ, where $j \in \{0\} \cup [\mu]$; that is, $P_i(A) = \{1/2\} \cup \{j/\mu\}_{j=0}^{\mu}$ for all $i \in [n]$.

For an n-Bernoulli-λ-EDA A and a fitness function f, let $\boldsymbol{p}^{(t)}(A, f)$ denote the random variable of all frequency vectors that A can take in iteration t when optimizing f, given the frequency vector from the previous iteration.

2.2. Automorphisms of the hypercube

Lehre and Witt [5] define unbiasedness with respect to variation operators over $\{0,1\}^n$. These operators draw bit strings according to probability distributions that are invariant under automorphisms of $\{0,1\}^n$: permutations and $\oplus$ (XOR) operations on bit strings. We extend these functions to $[0,1]^n$ to be able to use them directly on probability distributions.

Let $\overline{\sigma}$ be a permutation of $[n]$. We overload this notation and call a function $\sigma : [0,1]^n \to [0,1]^n$ a *permutation* if and only if it rearranges the elements of its input according to $\overline{\sigma}$ such that for all $\boldsymbol{p} \in [0,1]^n$ and all $i, j \in [n]$ with $\overline{\sigma}(i) = j$, $\sigma(\boldsymbol{p})_j = \boldsymbol{p}_i$. In this case, we say that σ maps position i to position j. Further, we call a function $\chi : [0,1]^n \to [0,1]^n$ a *complementation* if and only if, for each position, it either takes the complement of the value at this position or does not change it. That is, for all $\boldsymbol{p} \in [0,1]^n$ and all $i \in [n]$, $\chi(\boldsymbol{p})_i = 1 - \boldsymbol{p}_i$ or $\chi(\boldsymbol{p})_i = \boldsymbol{p}_i$.

We call any bijection $\alpha : \{0,1\}^n \to \{0,1\}^n$ that preserves the Hamming metric d_{H} a *Hamming automorphism*. It is well-known that there are exactly $2^n n!$ Hamming automorphisms.[4] Note that this means that any Hamming automorphism α can

[3] Originally introduced by Stützle and Hoos [29] as MIN–MAX Ant System (MMAS), the variant with *iteration-best* update was analyzed by Neumann et al. [30]. Friedrich et al. [26] dropped the MM part if the update is unrestricted on $[0,1]$.

[4] This can be seen as follows: there are 2^n possible choices for $\alpha(\mathbf{0})$. After $\alpha(\mathbf{0})$ is fixed, the n 1-neighbors of $\mathbf{0}$ can be mapped in an arbitrary (but bijective) manner to the n 1-neighbors of $\alpha(\mathbf{0})$, for which there are $n!$ choices. After these mappings are determined, the Hamming automorphism is completely determined.

Please cite this article in press as: T. Friedrich et al., Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.11.001

be denoted as the composition of a permutation σ and a complementation χ over $\{0,1\}^n$, that is, $\alpha = \chi \circ \sigma$. We are now interested in the superspace $[0,1]^n$ of $\{0,1\}^n$ and use the metric d defined as follows: $\forall \boldsymbol{x}, \boldsymbol{y} \in [0,1]^n : d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^n |\boldsymbol{x}_i - \boldsymbol{y}_i|$. This corresponds to the 1-metric. It naturally extends the Hamming metric on $\{0,1\}^n$. We say that $\alpha : [0,1]^n \to [0,1]^n$ is an *isometric automorphism* (of $[0,1]^n$) if it is bijective and *distance-preserving*, that is, for all $\boldsymbol{x}, \boldsymbol{y} \in [0,1]^n$, $d(\boldsymbol{x}, \boldsymbol{y}) = d\big(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{y})\big)$.

The following lemma shows that the Hamming automorphisms are in a one-to-one correspondence with the isometric automorphisms.

Lemma 2. *For any isometric automorphism α, let α_{H} denote α restricted to $\{0,1\}^n$. Let two isometric automorphisms α and β be given. We have that*

1. *α_{H} is a Hamming automorphism,*
2. *if $\alpha_{\mathrm{H}} = \beta_{\mathrm{H}}$, then $\alpha = \beta$, and that*
3. *for every Hamming automorphism η, there exists an isometric automorphism α such that $\alpha_{\mathrm{H}} = \eta$.*

Proof. Regarding (1), it suffices to show that α maps any $\boldsymbol{x} \in \{0,1\}^n$ to $\{0,1\}^n$. Let $\overline{\boldsymbol{x}}$ be the component-wise complement of $\boldsymbol{x}$. We then have $d(\boldsymbol{x}, \overline{\boldsymbol{x}}) = n$. Thus, $d\big(\alpha(\boldsymbol{x}), \alpha(\overline{\boldsymbol{x}})\big) = n$. Since only pairs of points from $\{0,1\}^n$ can be of distance n in $[0,1]^n$, we get that $\alpha(\boldsymbol{x}) \in \{0,1\}^n$, as desired.

Regarding (2), for all $j \in [n]$, we use $\mathbf{e}^{(j)}$ to denote the bit string which is 0 everywhere except in position j. We now show the following claim:

$$\forall \boldsymbol{x}, \boldsymbol{y} \in [0,1]^n : \big(\forall \boldsymbol{z} \in \{0,1\}^n : d(\boldsymbol{x}, \boldsymbol{z}) = d(\boldsymbol{y}, \boldsymbol{z})\big) \to \boldsymbol{x} = \boldsymbol{y} \,. \tag{1}$$

Let $\boldsymbol{x}, \boldsymbol{y} \in [0,1]^n$ such that for all $\boldsymbol{z} \in \{0,1\}^n$, $d(\boldsymbol{x}, \boldsymbol{z}) = d(\boldsymbol{y}, \boldsymbol{z})$. We now get for all $\boldsymbol{a} \in [0,1]^n$ and $j \in [n]$,

$$d(\boldsymbol{a}, \mathbf{0}) - d\big(\boldsymbol{a}, \mathbf{e}^{(j)}\big) = \sum_{i=1}^n \boldsymbol{a}_i - \left(\sum_{i=1}^{j-1} \boldsymbol{a}_i + (1 - \boldsymbol{a}_j) + \sum_{i=j+1}^n \boldsymbol{a}_i\right) = 2\boldsymbol{a}_j - 1 \,.$$

Thus, we get for all $j \in [n]$,

$$2\boldsymbol{x}_j - 1 = d(\boldsymbol{x}, \mathbf{0}) - d\big(\boldsymbol{x}, \mathbf{e}^{(j)}\big) = d(\boldsymbol{y}, \mathbf{0}) - d\big(\boldsymbol{y}, \mathbf{e}^{(j)}\big) = 2\boldsymbol{y}_j - 1 \,,$$

which shows $\boldsymbol{x} = \boldsymbol{y}$.

Now, suppose $\alpha_{\mathrm{H}} = \beta_{\mathrm{H}}$. Let $\boldsymbol{x} \in [0,1]^n$ be given. We have for all $\boldsymbol{z} \in \{0,1\}^n$, $d\big(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{z})\big) = d(\boldsymbol{x}, \boldsymbol{z}) = d\big(\beta(\boldsymbol{x}), \beta(\boldsymbol{z})\big)$, because α and β are isometric. Since $\alpha_{\mathrm{H}} = \beta_{\mathrm{H}}$ are Hamming automorphisms, we have for all $\boldsymbol{z} \in \{0,1\}^n$, $d\big(\alpha(\boldsymbol{x}), \boldsymbol{z}\big) = d\big(\beta(\boldsymbol{x}), \boldsymbol{z}\big)$. Using (1), we see that $\alpha(\boldsymbol{x}) = \beta(\boldsymbol{x})$, as desired.

Regarding (3), let η be given. We construct a distance-preserving extension α of η to $[0,1]^n$ such that $\alpha_{\mathrm{H}} = \eta$. For all $\boldsymbol{x} \in [0,1]^n$, let

$$\alpha(\boldsymbol{x}) = \eta(\mathbf{0}) + \sum_{i=1}^n \big(\eta\big(\mathbf{e}^{(i)}\big) - \eta(\mathbf{0})\big) \cdot \boldsymbol{x}_i \,. \tag{2}$$

Note that $\eta\big(\mathbf{e}^{(i)}\big)$ differs from $\eta(\mathbf{0})$ in exactly one position, since η is distance-preserving. This position does not necessarily have to be i, since η can perform a permutation. Assume that this position is $j \in [n]$. Then, for any $k \in [n]$, $\big(\eta\big(\mathbf{e}^{(i)}\big) - \eta(\mathbf{0})\big)_k$ is 1 or -1 if $k = j$ and 0 otherwise. Note further that for each unit vector, a component different from all other unit vectors is non-zero in $\eta\big(\mathbf{e}^{(i)}\big) - \eta(\mathbf{0})$, since η is bijective. Overall (assuming that position i is mapped to position j), for any $\boldsymbol{x} \in [0,1]^n$, this means that $\alpha(\boldsymbol{x})_j$ is completely determined by $\eta(\mathbf{0})_j + \big(\eta\big(\mathbf{e}^{(i)}\big) - \eta(\mathbf{0})\big)_j \cdot \boldsymbol{x}_i$. This shows that α is bijective (albeit not necessarily on $[0,1]^n$), since the inverse can be computed uniquely, due to the unique partition of the sum into the different components and due to η being a bijection.

We now show that α **maps bijectively to** $[0,1]^n$. Again, assume that position i is mapped to position j, and let $\boldsymbol{x} \in [0,1]^n$. Recall that η can only perform complementations additionally to permutations. We show that $\eta(\mathbf{0})_j + \big(\eta\big(\mathbf{e}^{(i)}\big) - \eta(\mathbf{0})\big)_j \cdot \boldsymbol{x}_i$ is a bijection of $[0,1]$. It then follows that α is a bijection of $[0,1]^n$, since the components do not interfere with one another.

We make a case distinction with respect to whether η takes the complement at position j (after performing the permutation) or not.

1. $\eta(\boldsymbol{x})_j = \boldsymbol{x}_i$. Then, $\eta(\mathbf{0})_j = 0$ and $\big(\eta\big(\mathbf{e}^{(i)}\big) - \eta(\mathbf{0})\big)_j = 1$. Thus, $\alpha(\boldsymbol{x})_j = \boldsymbol{x}_i$, which is a bijection from $[0,1]$ to $[0,1]$.

2. $\eta(\boldsymbol{x})_j = 1 - \boldsymbol{x}_i$. Then, $\eta(\mathbf{0})_j = 1$ and $\big(\eta\big(\mathbf{e}^{(i)}\big) - \eta(\mathbf{0})\big)_j = -1$. Thus, $\alpha(\boldsymbol{x})_j = 1 - \boldsymbol{x}_i$, which is also a bijection from $[0,1]$ to $[0,1]$.

Combining both cases yields that $\alpha(\boldsymbol{x})_i$ is a bijection of $[0,1]$ for every $\boldsymbol{x}$ and every i. Thus, α is a bijection of $[0,1]^n$, as we discussed before.

Please cite this article in press as: T. Friedrich et al., Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.11.001

Now, we show that α **is distance-preserving.** For this, let $\boldsymbol{x}, \boldsymbol{y} \in [0,1]^n$, and let $\overline{\sigma}$ be the permutation of indices that η realizes. We get

$$\begin{aligned} d\big(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{y})\big) &= \sum_{i=1}^{n} \big|\alpha(\boldsymbol{x})_i - \alpha(\boldsymbol{y})_i\big| \\ &= \sum_{i=1}^{n} \Big|\eta(\mathbf{0})_{\overline{\sigma}(i)} + \big(\eta\big(\mathbf{e}^{(i)}\big) - \eta(\mathbf{0})\big)_{\overline{\sigma}(i)} \cdot \boldsymbol{x}_i - \eta(\mathbf{0})_{\overline{\sigma}(i)} - \big(\eta\big(\mathbf{e}^{(i)}\big) - \eta(\mathbf{0})\big)_{\overline{\sigma}(i)} \cdot \boldsymbol{y}_i\Big| \\ &= \sum_{i=1}^{n} \Big|\big(\eta\big(\mathbf{e}^{(i)}\big) - \eta(\mathbf{0})\big)_{\overline{\sigma}(i)} (\boldsymbol{x}_i - \boldsymbol{y}_i)\Big| \\ &= \sum_{i=1}^{n} |\boldsymbol{x}_i - \boldsymbol{y}_i| \,, \end{aligned}$$

since $d\big(\eta(\mathbf{e}^{(i)}), \eta(\mathbf{0})\big) = d\big(\mathbf{e}^{(i)}, \mathbf{0}\big) = 1$.

We now show that α_{H} **is equal to** η, that is, for all $\boldsymbol{x} \in \{0,1\}^n$, $\alpha(\boldsymbol{x}) = \eta(\boldsymbol{x})$. We do so by showing that the components are the same. We handle this case very similarly to the one where we showed that α is bijective on $[0,1]^n$.

Let $i \in [n]$ be an arbitrary index, let $\boldsymbol{x} \in \{0,1\}^n$, and consider that η maps position i to position $j \in [n]$. We now make a case distinction with respect to whether η takes the complement at position j or not.

1. $\eta(\boldsymbol{x})_j = \boldsymbol{x}_i$:

$$\alpha(\boldsymbol{x})_j = \eta(\mathbf{0})_j + \big(\eta\big(\mathbf{e}^{(i)}\big)_j - \eta(\mathbf{0})_j\big) \cdot \boldsymbol{x}_i = 0 + (1-0) \cdot \boldsymbol{x}_i = \boldsymbol{x}_i = \eta(\boldsymbol{x})_j \,.$$

2. $\eta(\boldsymbol{x})_j = 1 - \boldsymbol{x}_i$:

$$\alpha(\boldsymbol{x})_j = \eta(\mathbf{0})_j + \big(\eta\big(\mathbf{e}^{(i)}\big)_j - \eta(\mathbf{0})_j\big) \cdot \boldsymbol{x}_i = 1 + (0-1) \cdot \boldsymbol{x}_i = 1 - \boldsymbol{x}_i = \eta(\boldsymbol{x})_j \,.$$

Hence, for all $j \in [n]$, $\alpha(\boldsymbol{x})_j = \eta(\boldsymbol{x})_j$. This finishes the proof. □

It follows from Lemma 2 that each Hamming automorphism α has a unique distance-preserving extension to $[0,1]^n$. Let $\widehat{\alpha}$ denote this unique extension of α to an isometric automorphism.

3. Unbiased EDAs

We start by defining the general concept of unbiasedness of a black-box algorithm. Such an algorithm follows the very general framework of querying bit strings from an oracle, where each query (i.e., sample) can only depend on all bit strings and their fitness queried so far. The original definition of unbiased black-box complexity was introduced by Lehre and Witt [5]. Since then, the definition has been stated several times in different formulations [6,11,33]. Usually, the definition makes use of unbiased variation operators, which are operators that sample from distributions in a way that is invariant under permutations and complementations (thus, only implicitly capturing the time point of the query).

Rowe and Vose [6], however, take a different approach and define unbiasedness in terms of the sequence of queries to the oracle (the *trace*). In this definition, an algorithm is unbiased if its trace is invariant under permutations and complementations (thus, only implicitly capturing the distribution used for sampling).

We use the definition of Rowe and Vose [6], phrased in the sense of our previously introduced concepts of permutations and complementations of $[0,1]^n$.

Definition 3. A *black-box algorithm* A is a mapping that takes a sequence of bit strings with corresponding fitness values and returns a bit string. Since A may be randomized, a black-box algorithm A and a fitness function f together define a sequence of random variables $(X^{(i)})_{i \in \mathbb{N}}$, where $X^{(i)}$ denotes the i-th bit string computed by A:

$$\forall i \in \mathbb{N}\colon\ X^{(i)} = A\Big(\big(X^{(j)}, f\big(X^{(j)}\big)\big)_{j<i}\Big) \,.$$

For a given algorithm A and fitness function f, we denote the dependency of $X^{(i)}$ on A and f by writing for all $i \in \mathbb{N}$, $X^{(i)}(A, f)$.

We call a black-box algorithm A *unbiased* if and only if for all Hamming automorphisms α, all fitness functions f, and all $i \in \mathbb{N}$,

$$X^{(i)}(A, f) = \alpha\big(X^{(i)}(A, f \circ \alpha)\big) \,. \tag{3}$$

Intuitively, when given an unbiased black-box algorithm A, it performs the same when optimizing f or $f \circ \alpha$; the only difference is that A using f optimizes the unperturbed hypercube, whereas A using $f \circ \alpha$ optimizes the hypercube perturbed by α. This way, if A using f samples $\boldsymbol{x}$, A using $f \circ \alpha$ samples $\alpha^{-1}(\boldsymbol{x})$ with the same probability and queries (with the same probability) the same individual: $\boldsymbol{x} = \alpha(\alpha^{-1}(\boldsymbol{x}))$ (since $f(\boldsymbol{x}) = (f \circ \alpha)(\alpha^{-1}(\boldsymbol{x}))$). Note that an n-Bernoulli-λ-EDA A is a black-box algorithm, as it samples λ bit strings every iteration, which can trivially be sequentialized. The probability vector $\boldsymbol{p}$ of A is implicitly captured in the history of search points $X^{(i)}(A, f)$, where those with indices $i \in \{\lambda t, \dots, \lambda(t+1)-1\}$ share the same distribution, which is $\boldsymbol{p}^{(t)}$. We now focus on an invariance property of the update scheme of an n-Bernoulli-λ-EDA.

Definition 4. Let A be an n-Bernoulli-λ-EDA with update scheme φ, and let $\mathcal{H}$ be a family of bijections over $[0,1]^n$ with fixed point $\mathbf{1/2}$ such that for all $h \in \mathcal{H}$, $h^{-1} \in \mathcal{H}$. We say that A is *$\mathcal{H}$-invariant* if and only if, for all $h \in \mathcal{H}$, all fitness functions f, all possible frequency vectors $\boldsymbol{p} \in P(A)$, and all offspring populations D that can be sampled by $\boldsymbol{p}$,

$$h\left(\varphi\left(\boldsymbol{p}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)\right) = \varphi\left(h(\boldsymbol{p}), \left(h(\boldsymbol{x}), f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right) .$$

Note that the right-hand side of this equation can be written as $\varphi\left(h(\boldsymbol{p}), \left(\boldsymbol{x}, (f \circ h^{-1})(\boldsymbol{x})\right)_{\boldsymbol{x}\in h(D)}\right)$, that is, $h(\boldsymbol{p})$ can sample $h(D)$ (since $\boldsymbol{p}$ can sample D). In the following, if $\mathcal{H}$ is the class of all permutations, we call an $\mathcal{H}$-invariant n-Bernoulli-λ-EDA *permutation-invariant*. If $\mathcal{H}$ is the class of all complementations, we call A *complementation-invariant*. And if $\mathcal{H}$ is the class of all isometric automorphisms, we call A *automorphism-invariant*. Note that any function in any of these families has $\mathbf{1/2}$ as a fixed point. This is important in order to ensure that A has an initial distribution that is equally fair to any function $h \in \mathcal{H}$.

Lemma 5. *Let A be an $\mathcal{H}$-invariant n-Bernoulli-λ-EDA. Then, for all $h \in \mathcal{H}$, all time steps t, and all $\boldsymbol{p}^{(t)} \in P^{(t)}(A)$,*

$$h\left(\boldsymbol{p}^{(t)}\right) \in P^{(t)}(A) .$$

Proof. Let h be any function of $\mathcal{H}$. We prove this lemma by induction over t.

For the **base case**, we have $\boldsymbol{p}^{(0)} = \mathbf{1/2}$, which is equal to $h(\mathbf{1/2})$, since $\mathbf{1/2}$ is a fixed point for any $h \in \mathcal{H}$. Hence, $h\left(\boldsymbol{p}^{(0)}\right) \in P^{(0)}(A)$.

For the **inductive step**, we assume that $h\left(\boldsymbol{p}^{(t)}\right) \in P^{(t)}(A)$ holds. We now show that $h\left(\boldsymbol{p}^{(t+1)}\right) \in P^{(t+1)}(A)$ holds.

Due to the definition of an n-Bernoulli-λ-EDA, we have that $\boldsymbol{p}^{(t+1)} = \varphi\left(\boldsymbol{p}^{(t)}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)$. And since A is $\mathcal{H}$-invariant, it holds that

$$h\left(\varphi\left(\boldsymbol{p}^{(t)}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)\right) = \varphi\left(h\left(\boldsymbol{p}^{(t)}\right), \left(h(\boldsymbol{x}), f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right) .$$

Thus, $h\left(\boldsymbol{p}^{(t+1)}\right) = \varphi\left(h\left(\boldsymbol{p}^{(t)}\right), \left(h(\boldsymbol{x}), f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)$. Due to the induction hypothesis, it holds that $h\left(\boldsymbol{p}^{(t)}\right) \in P^{(t)}(A)$. By definition of an update, it follows that $\varphi\left(h\left(\boldsymbol{p}^{(t)}\right), \left(h(\boldsymbol{x}), f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right) \in P^{(t+1)}(A)$, which completes the proof. □

Lemma 5 shows how heavily the update scheme of an n-Bernoulli-λ-EDA is already restricted when assuming that it is invariant. We now state our main theorem, which characterizes when an n-Bernoulli-λ-EDA is unbiased. Recall that we defined $\widehat{\alpha}$ to be the unique extension of a Hamming automorphism α after Lemma 2.

Theorem 6. *Let A be an n-Bernoulli-λ-EDA. The following are equivalent:*

1. *A is unbiased.*
2. *For all fitness functions f, all isometric automorphisms $\widehat{\alpha}$, and all $t \in \mathbb{N}$,*

$$\boldsymbol{p}^{(t)}(A, f) = \widehat{\alpha}\left(\boldsymbol{p}^{(t)}(A, f \circ \alpha)\right) . \tag{4}$$

3. *A is automorphism-invariant.*
4. *A is permutation- and complementation-invariant.*

Proof. (1) ⇒ (2). Let f be any fitness function, and let $\widehat{\alpha}$ be any isometric automorphism. Note that $\alpha\left(X^{(i)}(A, f \circ \alpha)\right) = \widehat{\alpha}\left(X^{(i)}(A, f \circ \alpha)\right)$, since $\widehat{\alpha}$ maps bit strings to bit strings, according to Lemma 2.

As mentioned before, the random variables $X^{(i)}(A, f)$ with indices $i \in \{\lambda t, \dots, \lambda(t+1)-1\}$ follow the same distribution, which is $\boldsymbol{p}^{(t)}$. This means that $\boldsymbol{p}^{(t)}(A, f)$ describes the distribution of (amongst others) $X^{(\lambda t)}(A, f)$, as A samples bit strings according to its frequency vector, due to the definition of an n-Bernoulli-λ-EDA.

Since we assume that A is unbiased, the random variable $X^{(\lambda t)}(A, f)$ is equal to $\widehat{\alpha}\left(X^{(\lambda t)}(A, f \circ \alpha)\right)$ for any t, and, thus, their distributions are equal as well. This means that Equation (4) holds.

Please cite this article in press as: T. Friedrich et al., Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.11.001

(1) ⇐ (2). Since Equation (4) holds and since $\boldsymbol{p}^{(t)}(A, f)$ and $\boldsymbol{p}^{(t)}(A, f\circ\alpha)$ describe distributions over bit strings, it follows that $X^{(\lambda t)}(A, f) = \widehat{\alpha}\big(X^{(\lambda t)}(A, f\circ\alpha)\big)$ for all t.

Due to an n-Bernoulli-λ-EDA sampling λ individuals every iteration, it follows that $X^{(\lambda t)}(A, f)$ is equal to $X^{(i)}(A, f)$, where $i \in \{\lambda t+1, \dots, \lambda(t+1)-1\}$. This shows Equation (3) for the remaining indices i and finishes this direction.

(2) ⇒ (3). Let f be any fitness function, let $\widehat{\alpha}$ be any isometric isomorphism, let $t \in \mathbb{N}$ be any iteration, let $\boldsymbol{p}^{(t)} = \boldsymbol{p}^{(t)}(A, f)$, and let $\widetilde{\boldsymbol{p}}^{(t)} = \boldsymbol{p}^{(t)}(A, f\circ\alpha)$. Note that due to the definition of $P(A)$, we cover all $\boldsymbol{p} \in P(A)$ by choosing an arbitrary t and considering $\boldsymbol{p}^{(t)}$. We now show that A is automorphism-invariant.

Since Equation (4) holds, we have that $\boldsymbol{p}^{(t+1)} = \widehat{\alpha}\big(\widetilde{\boldsymbol{p}}^{(t+1)}\big)$. Due to the definition of an update, this means that

$$\varphi\left(\boldsymbol{p}^{(t)}, \big(\boldsymbol{x}, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right) = \widehat{\alpha}\left(\varphi\left(\widetilde{\boldsymbol{p}}^{(t)}, \big(\boldsymbol{x}, (f\circ\alpha)(\boldsymbol{x})\big)_{\boldsymbol{x}\in \widetilde{D}}\right)\right), \tag{5}$$

where D should be sampled by $\boldsymbol{p}^{(t)}$ and $\widetilde{D}$ should be sampled by $\widetilde{\boldsymbol{p}}^{(t)}$.

Due to Equation (4), we further have that $\boldsymbol{p}^{(t)} = \widehat{\alpha}\big(\widetilde{\boldsymbol{p}}^{(t)}\big)$, which is equivalent to $\widehat{\alpha}^{-1}\big(\boldsymbol{p}^{(t)}\big) = \widetilde{\boldsymbol{p}}^{(t)}$, since $\widehat{\alpha}$ is invertible.

Substituting this into Equation (5) and inverting $\widehat{\alpha}$ leads to

$$\widehat{\alpha}^{-1}\left(\varphi\left(\boldsymbol{p}^{(t)}, \big(\boldsymbol{x}, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right)\right) = \varphi\left(\widehat{\alpha}^{-1}\big(\boldsymbol{p}^{(t)}\big), \big(\boldsymbol{x}, (f\circ\alpha)(\boldsymbol{x})\big)_{\boldsymbol{x}\in \widetilde{D}}\right).$$

Note that $\widetilde{D} = \widehat{\alpha}^{-1}(D)$, since $\widetilde{\boldsymbol{p}}^{(t)} = \widehat{\alpha}^{-1}\big(\boldsymbol{p}^{(t)}\big)$ and that $f\circ\alpha = f\circ\widehat{\alpha}$, since α and $\widehat{\alpha}$ are equal on bit strings. Thus, the above equation is equivalent to A being automorphism-invariant, as we discussed after Definition 4.

(3) ⇒ (2). For this direction, we use the same notation as in the previous direction. We now show that Equation (4) holds for all t.

For the **base case**, we have $\boldsymbol{p}^{(0)} = \widetilde{\boldsymbol{p}}^{(0)} = \mathbf{1/2}$ by the initialization step of an n-Bernoulli-λ-EDA. Permutations obviously do not change this equality, since all frequencies are the same. Complementations also do not change anything, since $1 - 1/2 = 1/2$. Hence, $\boldsymbol{p}^{(0)} = \widehat{\alpha}\big(\widetilde{\boldsymbol{p}}^{(0)}\big)$, as we desire.

For the **inductive step**, we assume that Equation (4) holds up to a $t \in \mathbb{N}$, which is equivalent to $\widehat{\alpha}^{-1}\big(\boldsymbol{p}^{(t)}\big) = \widetilde{\boldsymbol{p}}^{(t)}$. When making an update, we get

$$\widetilde{\boldsymbol{p}}^{(t+1)} = \varphi\left(\widetilde{\boldsymbol{p}}^{(t)}, \big(\boldsymbol{x}, (f\circ\alpha)(\boldsymbol{x})\big)_{\boldsymbol{x}\in\alpha^{-1}(D)}\right) = \varphi\left(\widehat{\alpha}^{-1}\big(\boldsymbol{p}^{(t)}\big), \big(\alpha^{-1}(\boldsymbol{x}), f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right),$$

where D is a random variable denoting an offspring population that can be sampled by $\boldsymbol{p}^{(t)}$, and where we used the induction hypothesis and that $(f\circ\alpha)\big(\alpha^{-1}(\boldsymbol{x})\big) = f(\boldsymbol{x})$.

Since A is automorphism-invariant, we can pull $\widehat{\alpha}^{-1}$ in front of φ. Hence, we get

$$\varphi\left(\widehat{\alpha}^{-1}\big(\boldsymbol{p}^{(t)}\big), \big(\alpha^{-1}(\boldsymbol{x}), f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right) = \widehat{\alpha}^{-1}\left(\varphi\left(\boldsymbol{p}^{(t)}, \big(\boldsymbol{x}, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right)\right),$$

which is equivalent to $\widetilde{\boldsymbol{p}}^{(t+1)}$, as we have seen previously.

Substituting $\varphi\left(\boldsymbol{p}^{(t)}, \big(\boldsymbol{x}, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right)$ with $\boldsymbol{p}^{(t+1)}$, we get

$$\widetilde{\boldsymbol{p}}^{(t+1)} = \widehat{\alpha}^{-1}\big(\boldsymbol{p}^{(t+1)}\big) \Leftrightarrow \boldsymbol{p}^{(t+1)} = \widehat{\alpha}\big(\widetilde{\boldsymbol{p}}^{(t+1)}\big),$$

which is what we wanted to show.

(3) ⇒ (4). This direction is trivial, since every permutation and every complementation is an isometric automorphism.

(4) ⇒ (3). Let $\widehat{\alpha}$ be any isometric automorphism, and let f be any fitness function. Let σ be a permutation and χ be a complementation such that $\widehat{\alpha} = \chi\circ\sigma$.

Since A is permutation-invariant, we get for any $\boldsymbol{p} \in P(A)$ and any D sampled by $\boldsymbol{p}$,

$$\begin{aligned}\sigma\left(\varphi\left(\boldsymbol{p}, \big(\boldsymbol{x}, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right)\right) &= \varphi\left(\sigma(\boldsymbol{p}), \big(\sigma(\boldsymbol{x}), f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right)\\ &= \varphi\left(\sigma(\boldsymbol{p}), \big(\boldsymbol{x}, (f\circ\sigma^{-1})(\boldsymbol{x})\big)_{\boldsymbol{x}\in\sigma(D)}\right),\end{aligned}$$

where $\sigma(\boldsymbol{p}) \in P(A)$, due to Lemma 5, and $\sigma(D)$ is an offspring population that can be sampled by $\sigma(\boldsymbol{p})$. Thus, since A is also complementation-invariant and since $(f\circ\sigma^{-1})$ is also a fitness function, we get

$$\begin{aligned}&\chi\left(\varphi\left(\sigma(\boldsymbol{p}), \big(\boldsymbol{x}, (f\circ\sigma^{-1})(\boldsymbol{x})\big)_{\boldsymbol{x}\in\sigma(D)}\right)\right) = \varphi\left(\chi\big(\sigma(\boldsymbol{p})\big), \big(\chi(\boldsymbol{x}), (f\circ\sigma^{-1})(\boldsymbol{x})\big)_{\boldsymbol{x}\in\sigma(D)}\right)\\ &= \varphi\left(\chi\big(\sigma(\boldsymbol{p})\big), \big(\chi\big(\sigma(\boldsymbol{x})\big), f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right) = \varphi\left(\widehat{\alpha}(\boldsymbol{p}), \big(\alpha(\boldsymbol{x}), f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right).\end{aligned}$$

We now combine both cases and get

$$\begin{aligned}\widehat{\alpha}\left(\varphi\left(\boldsymbol{p}, \big(\boldsymbol{x}, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right)\right) &= \chi\left(\sigma\left(\varphi\left(\boldsymbol{p}, \big(\boldsymbol{x}, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right)\right)\right)\\ &= \varphi\left(\widehat{\alpha}(\boldsymbol{p}), \big(\alpha(\boldsymbol{x}), f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\right),\end{aligned}$$

which means that A is automorphism-invariant.

Overall, we showed all equivalences that we stated. □

4. Decomposability

Theorem 6 characterizes the unbiasedness of any n-Bernoulli-λ-EDA, that is, its statement is very general, and some of the requirements may be difficult to check. Common n-Bernoulli-λ-EDAs have a more specific update scheme than the one presented in the framework (Algorithm 1). Thus, we can give more precise statements on unbiasedness if we focus on meaningful subsets of update schemes.

In the following, we consider a major subclass of n-Bernoulli-λ-EDAs whose update schemes can be split into separate functions. We call such EDAs *decomposable*. Within this class, we look at algorithm that we call *singularly decomposable* and *self-complementary*. The former (which includes all locally updating algorithms) are permutation-invariant, the latter are complementation-invariant. Hence, an algorithm having both properties are unbiased.

Definition 7. Let A be an n-Bernoulli-λ-EDA with update scheme φ. We say that A is *decomposable* if and only if there exists a set of n functions $\dot{\varphi}_i : [0,1] \times \left(\{0,1\} \times \mathbb{R}\right)^{\lambda} \to \mathbb{R}$, indexed by $i \in [n]$, such that for all fitness functions f, all possible frequency vectors $\boldsymbol{p} \in P(A)$, all offspring populations D that can be sampled by $\boldsymbol{p}$, and $i \in [n]$,

$$\varphi\left(\boldsymbol{p}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)_i = \dot{\varphi}_i\left(\boldsymbol{p}_i, \left(\boldsymbol{x}_i, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right).$$

If there is a single function $\dot{\varphi}$ such that for all $i \in [n]$, $\dot{\varphi} = \dot{\varphi}_i$, we say that A is *singularly decomposable*. We then drop the index i.

A decomposable n-Bernoulli-λ-EDA performs an update per frequency independently of values at other positions. However, the update function used may differ from frequency to frequency.

Theorem 8. *Let A be a decomposable n-Bernoulli-λ-EDA. The following are equivalent:*

1. *A is singularly decomposable.*
2. *A is permutation-invariant.*

Proof. **(1) ⇒ (2).** Let $\dot{\varphi}$ denote the update function used by A for every frequency, and let σ be any permutation. Consider that σ maps position $i \in [n]$ to position $j \in [n]$.

Since A is singularly decomposable, we get

$$\begin{aligned}\varphi\left(\sigma(\boldsymbol{p}), \left(\sigma(\boldsymbol{x}), f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)_j &= \dot{\varphi}\left(\sigma(\boldsymbol{p})_j, \left(\sigma(\boldsymbol{x})_j, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)\\ &= \dot{\varphi}\left(\boldsymbol{p}_i, \left(\boldsymbol{x}_i, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right) = \sigma\left(\varphi\left(\boldsymbol{p}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)\right)_j.\end{aligned}$$

And since i and j are arbitrary, it follows that A is permutation-invariant.

(2) ⇒ (1). We prove this direction by contraposition. Hence, assume that A is not singularly decomposable. Since A is still decomposable, there exist at least two different update functions $\dot{\varphi}_i$ and $\dot{\varphi}_j$.

Let f^* be a fitness function, $\boldsymbol{p}^* \in P(A)$ a possible frequency vector, and D^* an offspring population that can be sampled by $\boldsymbol{p}^*$ such that

$$\dot{\varphi}_i\left(\boldsymbol{p}^*_i, \left(\boldsymbol{x}_i, f^*(\boldsymbol{x})\right)_{\boldsymbol{x}\in D^*}\right) \neq \dot{\varphi}_j\left(\boldsymbol{p}^*_i, \left(\boldsymbol{x}_i, f^*(\boldsymbol{x})\right)_{\boldsymbol{x}\in D^*}\right). \tag{6}$$

Note that only the index of $\dot{\varphi}$ changes, since the functions are different for the same input.

Let σ^* be a permutation that maps i to j. Similar to the direction before, we have

$$\sigma^*\left(\varphi\left(\boldsymbol{p}^*, \left(\boldsymbol{x}, f^*(\boldsymbol{x})\right)_{\boldsymbol{x}\in D^*}\right)\right)_j = \dot{\varphi}_i\left(\boldsymbol{p}^*_i, \left(\boldsymbol{x}_i, f^*(\boldsymbol{x})\right)_{\boldsymbol{x}\in D^*}\right) \text{ and}$$

$$\varphi\left(\sigma^*(\boldsymbol{p}^*), \left(\sigma^*(\boldsymbol{x}), f^*(\boldsymbol{x})\right)_{\boldsymbol{x}\in D^*}\right)_j = \dot{\varphi}_j\left(\boldsymbol{p}^*_i, \left(\boldsymbol{x}_i, f^*(\boldsymbol{x})\right)_{\boldsymbol{x}\in D^*}\right).$$

Using Inequality (6), we get by combining the two above equations that

$$\sigma^*\left(\varphi\left(\boldsymbol{p}^*, \left(\boldsymbol{x}, f^*(\boldsymbol{x})\right)_{\boldsymbol{x}\in D^*}\right)\right)_j \neq \varphi\left(\sigma^*(\boldsymbol{p}^*), \left(\sigma^*(\boldsymbol{x}), f^*(\boldsymbol{x})\right)_{\boldsymbol{x}\in D^*}\right)_j,$$

which means that A is not permutation-invariant. This finishes the proof. □

Theorem 8 provides a very strong guideline for creating n-Bernoulli-λ-EDAs: if your algorithm should be decomposable and unbiased, then it has to be singularly decomposable. This statement is helpful, since it is not hard to come up with a singularly decomposable n-Bernoulli-λ-EDA. But it also tells us that there is no sense in trying any more complicated update schemes if we want a decomposable one. A question that might now be asked is whether unbiasedness implies decomposability. If so, we would know that only singularly decomposable n-Bernoulli-λ-EDAs could be unbiased. Unfortunately, there exist algorithms that are not decomposable but still unbiased, as we will argue next.

Please cite this article in press as: T. Friedrich et al., Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.11.001

4.1. An unbiased non-decomposable EDA

The idea of the algorithm we are going to present is the following: the update scheme is closely related to the one of λ-AS$_{\text{IB}}$. However, the parameter ρ is adaptive with respect to the current frequency vector. The more each frequency is away from its center 1/2, the greater ρ gets. This means that the step size is increased the further the algorithm commits to a certain direction for its frequencies. Formally, for any $\rho \in [0,1]$ and any frequency vector $\boldsymbol{p}$, let $\rho_{\boldsymbol{p}} = \rho(\frac{1}{2} + \frac{1}{n}\sum_{k=1}^{n} |\boldsymbol{p}_k - \frac{1}{2}|)$. Consider the following update scheme:

$$\varphi\left(\boldsymbol{p}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)_i = \left(1-\rho_{\boldsymbol{p}}\right)\boldsymbol{p}_i + \boldsymbol{x}_i^{(1)} \cdot \rho_{\boldsymbol{p}}\ , \tag{7}$$

where $\boldsymbol{x}^{(1)}$ – the individual of rank 1 – is determined uniformly at random in the case of ties. Note that an n-Bernoulli-λ-EDA with such an update scheme is not decomposable, as $\rho_{\boldsymbol{p}}$ uses all frequencies.

Proposition 9. *Let A be the n-Bernoulli-λ-EDA with update scheme φ as given in Equation (7). A is unbiased.*

Proof. We use Theorem 6 and show that A is permutation- and complementation-invariant.

We start by showing that ***A* is permutation-invariant.** For this, let σ be an arbitrary permutation that maps i to j. We get

$$\begin{aligned}
\sigma\left(\varphi\left(\boldsymbol{p}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)\right)_j &= \left(1-\rho_{\boldsymbol{p}}\right)\boldsymbol{p}_i + \boldsymbol{x}_i^{(1)} \cdot \rho_{\boldsymbol{p}} \\
&= \left(1-\rho_{\sigma(\boldsymbol{p})}\right)\sigma(\boldsymbol{p})_j + \sigma\left(\boldsymbol{x}^{(1)}\right)_j \cdot \rho_{\sigma(\boldsymbol{p})} \\
&= \varphi\left(\sigma(\boldsymbol{p}), \left(\sigma(\boldsymbol{x}), f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)_j\ ,
\end{aligned}$$

as we can re-order the summands in $\rho_{\boldsymbol{p}}$ without changing its value.

Note that the update $\sigma\left(\varphi\left(\boldsymbol{p}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)\right)$ and the update $\varphi\left(\sigma(\boldsymbol{p}), \left(\sigma(\boldsymbol{x}), f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)$ are both random variables. Thus, $\boldsymbol{x}^{(1)}$ does not necessarily have to denote the same individual when looking at both cases. However, due to our assumptions on $\boldsymbol{x}^{(1)}$, we do not have to consider it changing in-between the equations, since $\boldsymbol{x}^{(1)}$ and $\sigma\left(\boldsymbol{x}^{(1)}\right)$ both have fitness $f\left(\boldsymbol{x}^{(1)}\right)$ and, thus, get chosen for an update with equal probability in both cases. This shows that A is permutation-invariant.

We now show that ***A* is complementation-invariant.** For this, let χ be an arbitrary complementation. Note that $|\chi(\boldsymbol{p})_k - 1/2| = |\boldsymbol{p}_k - 1/2|$ and, thus, $\rho_{\boldsymbol{p}} = \rho_{\chi(\boldsymbol{p})}$.

Consider that χ takes the complement at position i; the other case of χ not changing anything at position i can be shown analogously and more easily. Thus, we get

$$\begin{aligned}
\varphi\left(\chi(\boldsymbol{p}), \left(\chi(\boldsymbol{x}), f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)_i &= \left(1-\rho_{\chi(\boldsymbol{p})}\right)\chi(\boldsymbol{p})_i + \chi\left(\boldsymbol{x}^{(1)}\right)_i \cdot \rho_{\chi(\boldsymbol{p})} \\
&= \left(1-\rho_{\boldsymbol{p}}\right)(1-\boldsymbol{p}_i) + \left(1-\boldsymbol{x}_i^{(1)}\right) \cdot \rho_{\boldsymbol{p}} \\
&= 1 - \rho_{\boldsymbol{p}} - \left(1-\rho_{\boldsymbol{p}}\right)\boldsymbol{p}_i + \rho_{\boldsymbol{p}} - \boldsymbol{x}_i^{(1)} \cdot \rho_{\boldsymbol{p}} \\
&= 1 - \left((1-\rho_{\boldsymbol{p}})\boldsymbol{p}_i + \boldsymbol{x}_i^{(1)} \cdot \rho_{\boldsymbol{p}}\right) \\
&= \chi\left(\varphi\left(\boldsymbol{p}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)\right)_i\ ,
\end{aligned}$$

which means that A is complementation-invariant. This completes the proof. □

The update scheme presented in Equation (7) uses the entire frequency vector to update any frequency $\boldsymbol{p}_i$, but it only uses the i-th bit for each individual in the offspring population D. Thus, it may be that an operator that makes use of D in a non-bit-wise manner might make the algorithm biased. However, there exist operators that use D entirely and still result in unbiased algorithms.

One example is the operator $s(D)$, which we will define after introducing the following auxiliary function (for $i \in [n]$):

$$s(D,i) = \begin{cases} 1 & \text{if } \sum_{\boldsymbol{x}\in D} \boldsymbol{x}_i \in \{0, \lambda\}\,, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $s(D,i)$ is 1 if and only if all individuals have the same value at index i; or else, it is 0. Now, let $s(D) = (1/n)\sum_{k=1}^{n} s(D,k)$. Consider the update scheme

$$\varphi\left(\boldsymbol{p}, \left(\boldsymbol{x}, f(\boldsymbol{x})\right)_{\boldsymbol{x}\in D}\right)_i = \left(1-\rho\cdot s(D)\right)\boldsymbol{p}_i + \boldsymbol{x}_i^{(1)} \cdot \rho \cdot s(D)\ . \tag{8}$$

Analogous to Proposition 9, this update scheme is unbiased. Essentially, this boils down to $s(D)$ being permutation- and complementation-invariant.

Finally, one can combine the schemes of Equations (7) and (8), and the resulting n-Bernoulli-λ-EDA is still unbiased. Thus, the class of unbiased decomposable n-Bernoulli-λ-EDAs is a proper subset of the class of all unbiased n-Bernoulli-λ-EDAs. However, many theoretically analyzed n-Bernoulli-λ-EDAs – such as the ones in Example 1 – are singularly decomposable.

4.2. Unbiased decomposable EDAs

We now look at complementation invariance for decomposable n-Bernoulli-λ-EDAs. For this, we define the following property.

Definition 10. Let A be a singularly decomposable n-Bernoulli-λ-EDA with update scheme φ, using the update function $\dot{\varphi}$. We say that A is *self-complementary* if and only if for all fitness functions f, all possible frequency vectors $\boldsymbol{p} \in P(A)$, all offspring populations D that can be sampled by $\boldsymbol{p}$, and all indices i,

$$1 - \dot{\varphi}\big(\boldsymbol{p}_i, \big(\boldsymbol{x}_i, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big) = \dot{\varphi}\big(1 - \boldsymbol{p}_i, \big(1 - \boldsymbol{x}_i, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)\,.$$

The update of a self-complementary n-Bernoulli-λ-EDA works in the following way: increasing or decreasing a frequency at index i (the probability to sample a 1) is the same as decreasing or increasing, respectively, the complement of the frequency (the probability to sample a 0) in the same manner as the frequency would if all the bits at index i were inverted. This means that the scheme for decreasing follows from the scheme of increasing or vice versa. We can now state the following theorem.

Theorem 11. *Let A be a singularly decomposable n-Bernoulli-λ-EDA. The following are equivalent:*

1. *A is complementation-invariant.*
2. *A is self-complementary.*

Proof. **(1) ⇒ (2).** This direction trivially follows from Definition 4 by applying Definition 7 and using the complementation such that for all i, $\chi(\boldsymbol{p})_i = 1 - \boldsymbol{p}_i$.

(2) ⇒ (1). The update scheme φ is per component equal to $\dot{\varphi}$ by Definition 7 for any i. Thus, we can consider an arbitrary i. For any complementation χ with $\chi(\boldsymbol{p})_i = \boldsymbol{p}_i$, the equation in Definition 4 trivially holds. Thus, only the case $\chi(\boldsymbol{p})_i = 1 - \boldsymbol{p}_i$ needs to be considered, which holds by the assumption of point (2). □

For a singularly decomposable n-Bernoulli-λ-EDA, checking complementation-invariance is very straightforward. We can now give a more specific characterization of unbiasedness.

Corollary 12. *Let A be a decomposable n-Bernoulli-λ-EDA. The following are equivalent:*

1. *A is unbiased.*
2. *A is singularly decomposable and self-complementary.*

Proof. We show that point (2) is equivalent to A being permutation- and complementation-invariant. Theorem 6 then yields the equivalence to point (1).

Due to Theorem 8, A being singularly decomposable is equivalent to it being permutation-invariant; and due to Theorem 11, being self-complementary is equivalent to being complementation-invariant. This finishes the proof. □

4.2.1. Examples of unbiased decomposable EDAs

We now apply Corollary 12 to some of the algorithms mentioned in Example 1 to prove their unbiasedness. We discuss the remaining algorithms in the next section.

Theorem 13. *PBIL, the UMDA, and λ-AS$_{IB}$ are unbiased.*

Proof. We only show that PBIL is unbiased, since the UMDA and λ-AS$_{\text{IB}}$ are special instances of it. Note that PBIL is singularly decomposable. Thus, we only show that it is complementation-invariant. Then, applying Corollary 12 finishes the proof.

In the following, let $\dot{\varphi}$ denote the update function that PBIL uses for its update scheme. Further, recall that we assume a uniform tie-breaking rule, that is, if there is a tie in fitness, the winner is determined uniformly at random. Hence, for any fitness function f and any Hamming automorphism α, if an individual $\boldsymbol{x}$ is the winner of a tie-break using f, $\alpha(\boldsymbol{x})$ will be the winner of a tie-break using $f \circ \alpha^{-1}$ with equal probability.

Please cite this article in press as: T. Friedrich et al., Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.11.001

$$\dot{\varphi}\big(1-\boldsymbol{p}_i,\big(1-\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)=(1-\rho)(1-\boldsymbol{p}_i)+\rho\frac{\sum_{k=1}^{\mu}\big(1-\boldsymbol{x}_i^{(k)}\big)}{\mu}$$
$$=1-\rho-(1-\rho)\boldsymbol{p}_i+\rho\left(\frac{\mu}{\mu}-\frac{\sum_{k=1}^{\mu}\boldsymbol{x}_i^{(k)}}{\mu}\right)=1-\left((1-\rho)\boldsymbol{p}_i+\rho\frac{\sum_{k=1}^{\mu}\boldsymbol{x}_i^{(k)}}{\mu}\right)$$
$$=1-\dot{\varphi}\big(\boldsymbol{p}_i,\big(\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)\,.$$

Note that $1-\boldsymbol{x}^{(k)}$ and $\boldsymbol{x}^{(k)}$ have the same rank with equal probability, due to uniform tie-breaking. Thus, the equalities shown above hold. □

5. Locally updating EDAs

Last, we look at locally updating n-Bernoulli-λ-EDAs. In Theorem 13, we skipped the cGA and the $(1,\lambda)$-EA. We did so because these algorithms are locally updating and, thus, their update schemes can be expressed very concisely. Hence, we want to give a characterization that is even easier to check than the one given in Corollary 12. Note that all locally updating n-Bernoulli-λ-EDAs are singularly decomposable by definition and, hence, already permutation-invariant by Theorem 8. We focus on a subclass of algorithms that *move symmetrically* and show that this property is equivalent to being complementation-invariant and, thus, unbiased.

Definition 14. Let A be a locally updating n-Bernoulli-λ-EDA. We say that A *moves symmetrically at position* $i\in[n]$ if and only if for all fitness functions f and all offspring populations D that can be sampled by a frequency vector from $P(A)$,

$$\mathrm{flip}\big(\mathrm{move}\big(\big(\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)_i\big)=\mathrm{move}\big(\big(1-\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)_i\,, \tag{9}$$

where $\mathrm{flip}\colon\{\mathrm{up},\mathrm{stay},\mathrm{down}\}\to\{\mathrm{up},\mathrm{stay},\mathrm{down}\}$ swaps 'up' and 'down' and maps 'stay' to 'stay', or $P_i(A)=\{1/2\}$. If A moves symmetrically at all positions $i\in[n]$, we say that it *moves symmetrically*.

A symmetrically moving n-Bernoulli-λ-EDA moves a frequency into the opposite direction if all the bits for an update at this position are flipped.

Theorem 15. *Let A be a locally updating n-Bernoulli-λ-EDA. The following are equivalent:*

1. *A is complementation-invariant.*
2. *A moves symmetrically.*

Proof. In this proof, we show that being self-complementary (Definition 10; here denoted by (S)) is equivalent to A moving symmetrically (this theorem's point (2)). The equivalence to A being complementation-invariant then follows by applying Theorem 11. In the following, we always consider any $i\in[n]$.

(S) ⇒ (2). We assume that $P_i(A)\supset\{1/2\}$ and show that Equation (9) holds. For this, let $v_i=\mathrm{move}\big(\big(\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)$, and let $\overline{v}_i=\mathrm{move}\big(\big(1-\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)$. We make a case distinction with respect to v_i.

First, assume that $v_i=\mathrm{up}$. Then, $1-\dot{\varphi}\big(\boldsymbol{p}_i,\big(\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)=1-\mathrm{set}(\boldsymbol{p}_i)$ by assumption and the definition of a locally updating n-Bernoulli-λ-EDA. Further, since we assume that A is self-complementary, we have

$$1-\dot{\varphi}\big(\boldsymbol{p}_i,\big(\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)=\dot{\varphi}\big(1-\boldsymbol{p}_i,\big(1-\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)$$

and, thus, $1-\mathrm{set}(\boldsymbol{p}_i)=\dot{\varphi}\big(1-\boldsymbol{p}_i,\big(1-\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)$.

We now make a case distinction with respect to $\overline{v}_i$ in the update on the right-hand side. For $\overline{v}_i=\mathrm{up}$, we get $\dot{\varphi}\big(1-\boldsymbol{p}_i,\big(1-\boldsymbol{x}_i,f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)=\mathrm{set}(1-\boldsymbol{p}_i)$ and, thus, $1-\mathrm{set}(\boldsymbol{p}_i)=\mathrm{set}(1-\boldsymbol{p}_i)$, which is only true if $\mathrm{set}(\boldsymbol{p}_i)=\boldsymbol{p}_i$. Now, $1/2$ is not a fixed point of 'set' due our assumption of $P_i(A)\supset\{1/2\}$. Note that D as well as its bit-wise complement can be sampled in iteration 0, i.e., when $\boldsymbol{p}^{(0)}=\mathbf{1/2}$, especially $\boldsymbol{p}_i^{(0)}=1/2$. Since 'move' decides independently of $\boldsymbol{p}_i$ in which direction to move, $\overline{v}_i$ cannot be 'up', because that would contradict A being complementation-invariant in iteration 0: $\boldsymbol{p}_i^{(1)}$ would be greater than $1/2$ (since $1/2$ is not a fixed point) but so would $\widetilde{\boldsymbol{p}}_i^{(1)}$ (the update of $\boldsymbol{p}^{(0)}$ with respect to the bit-wise complement of D). However, $\widetilde{\boldsymbol{p}}_i^{(0)}$ had to be smaller than $1/2$.

For $\overline{v}_i=\mathrm{stay}$, we get $1-\mathrm{set}(\boldsymbol{p}_i)=1-\boldsymbol{p}_i$, which, again, can only hold if $\mathrm{set}(\boldsymbol{p}_i)=\boldsymbol{p}_i$. We can argue analogously to before, the only difference being that $\widetilde{\boldsymbol{p}}_i^{(1)}$ would be $1/2$, whereas $\boldsymbol{p}_i^{(1)}$ would, again, be greater than $1/2$, which contradicts A being complementation-invariant.

For $\overline{v}_i=\mathrm{down}$, we get $1-\mathrm{set}(\boldsymbol{p}_i)=1-\mathrm{set}\big(1-(1-\boldsymbol{p}_i)\big)$, which is always true. Hence, if $v_i=\mathrm{up}$, then $\overline{v}_i=\mathrm{down}$.

The other two cases with respect to v_i can be done analogously and yield that $\overline{v}_i=\mathrm{stay}$ if $v_i=\mathrm{stay}$, and that $\overline{v}_i=\mathrm{up}$ if $v_i=\mathrm{down}$ (always assuming that $P_i(A)\supset\{1/2\}$). All in all, this results in $\mathrm{flip}(v_i)=\overline{v}_i$.

Please cite this article in press as: T. Friedrich et al., Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.11.001

(2) ⇒ (S). If $P_i(A) = \{1/2\}$, A is trivially complementation-invariant for that position, since its only frequency for this position is $1/2$, independent of any D sampled.

If $P_i(A) \supset \{1/2\}$, we show that $1 - \dot{\varphi}\big(\boldsymbol{p}_i, \big(\boldsymbol{x}_i, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)$ is equal to $\dot{\varphi}\big(1 - \boldsymbol{p}_i, \big(1 - \boldsymbol{x}_i, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)$. We start with the former and make, as before, a case distinction with respect to v_i. Due to A moving symmetrically, we have $\text{flip}(v_i) = \overline{v}_i$.

First, assume that $v_i = \text{up}$. We then get

$$\begin{aligned} 1 - \dot{\varphi}\big(\boldsymbol{p}_i, \big(\boldsymbol{x}_i, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big) &= 1 - \text{set}(\boldsymbol{p}_i) = 1 - \text{set}\big(1 - (1 - \boldsymbol{p}_i)\big) \\ &= \dot{\varphi}\big(1 - \boldsymbol{p}_i, \big(1 - \boldsymbol{x}_i, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big)\,, \end{aligned}$$

where the first and third equality follow from the definition of a locally updating n-Bernoulli-λ-EDA's update scheme, using that $v_i = \text{up}$ (first equality) and that $\overline{v}_i = \text{down}$ (third equality). The other two cases can be done analogously and yield the same result. This finishes the proof. □

Using Theorem 15, we get the following Corollary.

Corollary 16. *Let A be a locally updating n-Bernoulli-λ-EDA. The following are equivalent:*

1. *A is unbiased.*
2. *A moves symmetrically.*

Proof. We show that point (2) is equivalent to A being unbiased. Note that we assume A to be singularly decomposable, as it is locally updating.

Due to Theorem 15, A moving symmetrically is equivalent to it being complementation-invariant, which is equivalent to it being self-complementary (Theorem 11); and due to Corollary 12, this is equivalent to A being unbiased. □

For a locally updating n-Bernoulli-λ-EDA, only the 'move' function decides whether the algorithm is unbiased or not. The remaining update via 'set' is already set up to perform an unbiased update. This makes checking for unbiasedness very easy.

Theorem 17. *The cGA and the $(1, \lambda)$-EA are unbiased.*

Proof. We make use of Corollary 16 and show that both algorithms are moving symmetrically. Let $i \in [n]$ be an arbitrary index. As before, since we assume uniform tie-breaking, we can assume for all $\boldsymbol{x} \in D$ that the ranks of $\boldsymbol{x}$ and $1 - \boldsymbol{x}$ are the same with equal probability (as they have the same fitness $f(\boldsymbol{x})$, according to Equation (9)). Further, we use the same notation for the two 'move' values as in the proof of Theorem 15: v_i and $\overline{v}_i$.

For the **cGA**, $\text{move}\big(\big(\boldsymbol{x}_i, f(\boldsymbol{x})\big)_{\boldsymbol{x}\in D}\big) = v_i$ depends on the relation of $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$. If $\boldsymbol{x}_i^{(1)} = \boldsymbol{x}_i^{(2)}$, then $1 - \boldsymbol{x}_i^{(1)} = 1 - \boldsymbol{x}_i^{(2)}$, as well. This means that $v_i = \text{stay}$ and $\overline{v}_i = \text{stay}$. Thus, $\text{flip}(v_i) = \overline{v}_i$.

If $\boldsymbol{x}_i^{(1)} > \boldsymbol{x}_i^{(2)}$, then $1 - \boldsymbol{x}_i^{(1)} < 1 - \boldsymbol{x}_i^{(2)}$. This means that $v_i = \text{up}$ and $\overline{v}_i = \text{down}$. Hence, $\text{flip}(v_i) = \overline{v}_i$. The case $\boldsymbol{x}_i^{(1)} < \boldsymbol{x}_i^{(2)}$ follows completely analogously. Overall, the cGA moves symmetrically and is, thus, unbiased.

For the **$(1, \lambda)$-EA**, we only have to consider two cases, as its 'move' function never yields 'stay'. If $\boldsymbol{x}_i^{(1)} = 1$, then $1 - \boldsymbol{x}_i^{(1)} = 0$. Thus, $v_i = \text{up}$ and $\overline{v}_i = \text{down}$ and, hence, $\text{flip}(v_i) = \overline{v}_i$. The other case is, again, done completely analogously. This completes the proof. □

Since λ-AS_{IB} has the same 'move' function as the $(1, \lambda)$-EA, it follows from Theorem 17 and Corollary 16 that λ-AS_{IB} is unbiased. This observation can be even more generalized: Every locally updating n-Bernoulli-λ-EDA with a 'move' function as in the cGA or the $(1, \lambda)$-EA is unbiased. Thus, the framework of locally updating n-Bernoulli-λ-EDAs provides a powerful tool to easily come up with unbiased n-Bernoulli-λ-EDAs: a single (symmetrically moving) 'move' function can be used with arbitrary 'set' functions and will always result in an unbiased algorithm.

6. Conclusions

We analyzed the broad class of n-Bernoulli-λ-EDAs with respect to unbiasedness. That is, we proved when those algorithms behave the same, regardless of the problem encoding, which is a desirable property in true black-box optimization, as the optimization algorithm has no knowledge about the representation of the problem. We showed how this applies to certain algorithms by providing examples. In order to account for the simpler update schemes of those example algorithms, we provided conciser characterizations of certain subclasses, which can be verified more easily. Our results can especially be viewed as guidelines on how to create an update scheme when one wants to create an unbiased algorithm – or when not.

Please cite this article in press as: T. Friedrich et al., Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.11.001

Acknowledgements

We would like to thank the anonymous reviewers for their valuable feedback, which greatly improved this paper.

References

[1] S. Droste, T. Jansen, I. Wegener, Upper and lower bounds for randomized search heuristics in black-box optimization, Theory Comput. Syst. 39 (4) (2006) 525–544.
[2] S. Droste, T. Jansen, I. Wegener, On the analysis of the (1+1) evolutionary algorithm, Theoret. Comput. Sci. 276 (1–2) (2002) 51–81.
[3] G. Anil, R.P. Wiegand, Black-box search by elimination of fitness functions, in: Proc. of FOGA, 2009, pp. 67–78.
[4] P. Erdős, A. Rényi, On two problems of information theory, Magy. Tud. Akad. Mat. Kut. Intéz. Közl. 8 (1963) 229–243.
[5] P.K. Lehre, C. Witt, Black-box search by unbiased variation, in: Proc. of GECCO, 2010, pp. 1441–1448.
[6] J.E. Rowe, M.D. Vose, Unbiased black box search algorithms, in: Proc. of GECCO, 2011, pp. 2035–2042.
[7] B. Doerr, C. Doerr, T. Kötzing, The unbiased black-box complexity of partition is polynomial, Artificial Intelligence 216 (2014) 275–286.
[8] B. Doerr, C. Winzen, Towards a complexity theory of randomized search heuristics: ranking-based black-box complexity, in: Proc. of CSR'11, 2011, pp. 15–28.
[9] B. Doerr, C. Winzen, Playing Mastermind with constant-size memory, in: Proc. of STACS, vol. 14, 2012, pp. 441–452.
[10] C. Doerr, J. Lengler, OneMax in black-box models with several restrictions, in: Proc. of GECCO, 2015, pp. 1431–1438.
[11] B. Doerr, D. Johannsen, T. Kötzing, P.K. Lehre, M. Wagner, C. Winzen, Faster black-box algorithms through higher arity operators, in: Proc. of FOGA, 2011, pp. 163–172.
[12] B. Doerr, C. Winzen, Reducing the arity in unbiased black-box complexity, in: Proc. of GECCO, 2012, pp. 1309–1316.
[13] B. Doerr, C. Doerr, T. Kötzing, Unbiased black-box complexities of jump functions: how to cross large plateaus, in: Proc. of GECCO, 2014, pp. 769–776.
[14] G. Badkobeh, P.K. Lehre, D. Sudholt, Unbiased black-box complexity of parallel search, in: Proc. of PPSN, 2014, pp. 892–901.
[15] B. Doerr, C. Doerr, J. Yang, Optimal parameter choices via precise black-box analysis, in: Proc. of GECCO, 2016, pp. 1123–1130.
[16] M. Pelikan, M. Hauschild, F.G. Lobo, Estimation of distribution algorithms, in: Springer Handbook of Computational Intelligence, Springer, 2015, pp. 899–928.
[17] D.-C. Dang, P.K. Lehre, Simplified runtime analysis of estimation of distribution algorithms, in: Proc. of GECCO, 2015, pp. 513–518.
[18] D. Sudholt, C. Witt, Update strength in EDAs and ACO: how to avoid genetic drift, in: Proc. of GECCO, 2016, pp. 61–68.
[19] M.S. Krejca, C. Witt, Lower bounds on the run time of the Univariate Marginal Distribution Algorithm on OneMax, in: Proc. of GECCO, 2017, pp. 65–79.
[20] B. Doerr, M.S. Krejca, Significance-based estimation-of-distribution algorithms, in: Proc. of GECCO, 2018, pp. 1483–1490.
[21] C. Witt, Upper bounds on the runtime of the Univariate Marginal Distribution Algorithm on OneMax, in: Proc. of GECCO, 2017, pp. 1415–1422.
[22] P.K. Lehre, P.T.H. Nguyen, Improved runtime bounds for the Univariate Marginal Distribution Algorithm via anti-concentration, in: Proc. of GECCO, 2017, pp. 1383–1390.
[23] P.K. Lehre, P.T.H. Nguyen, Level-based analysis of the Population-Based Incremental Learning algorithm, in: Proc. of PPSN, 2018, pp. 105–116.
[24] V. Hasenöhrl, A.M. Sutton, On the runtime dynamics of the compact genetic algorithm on jump functions, in: Proc. of GECCO, 2018, pp. 967–974.
[25] J. Lengler, D. Sudholt, C. Witt, Medium step sizes are harmful for the compact genetic algorithm, in: Proc. of GECCO, 2018, pp. 1499–1506.
[26] T. Friedrich, T. Kötzing, M.S. Krejca, EDAs cannot be balanced and stable, in: Proc. of GECCO, 2016, pp. 1139–1146.
[27] S. Baluja, Population-Based Incremental Learning: a Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, Tech. rep., Carnegie Mellon University, Pittsburgh, PA, USA, 1994.
[28] H. Mühlenbein, G. Paaß, From recombination of genes to the estimation of distributions I. Binary parameters, in: Proc. of PPSN, 1996, pp. 178–187.
[29] T. Stützle, H.H. Hoos, MAX–MIN Ant System, Future Gener. Comput. Syst. 16 (9) (2000) 889–914.
[30] F. Neumann, D. Sudholt, C. Witt, A few ants are enough: ACO with iteration-best update, in: Proc. of GECCO, 2010, pp. 63–70.
[31] G.R. Harik, F.G. Lobo, D.E. Goldberg, The compact genetic algorithm, IEEE Trans. Evol. Comput. 3 (4) (1999) 287–297.
[32] H.-P. Schwefel, Evolution and Optimum Seeking: The Sixth Generation, John Wiley & Sons, Inc., 1993.
[33] B. Doerr, T. Kötzing, C. Winzen, Too fast unbiased black-box algorithms, in: Proc. of GECCO, 2011, pp. 2043–2050.

Please cite this article in press as: T. Friedrich et al., Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.11.001