The Univariate Marginal Distribution Algorithm Copes Well With Deception and Epistasis

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

Martin S. Krejca
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany
martin.krejca@hpi.de

ABSTRACT

In their recent work, Lehre and Nguyen (FOGA 2019) show that the univariate marginal distribution algorithm (UMDA) needs time exponential in the parent populations size to optimize the DeceptiveLeadingBlocks (DLB) problem. They conclude from this result that univariate EDAs have difficulties with deception and epistasis.

In this work, we show that this negative finding is caused by an unfortunate choice of the parameters of the UMDA. When the population sizes are chosen large enough to prevent genetic drift, then the UMDA optimizes the DLB problem with high probability with at most $\lambda(\frac{n}{2} + 2e \ln n)$ fitness evaluations. Since an offspring population size λ of order $n \log n$ can prevent genetic drift, the UMDA can solve the DLB problem with $O(n^2 \log n)$ fitness evaluations. Together with the result of Lehre and Nguyen, our result for the first time rigorously proves that running EDAs in the regime with genetic drift can lead to drastic performance losses.

This extended abstract summarizes our work "The Univariate Marginal Distribution Algorithm Copes Well with Deception and Epistasis", which appeared in the *Proceedings of Evolutionary Computation in Combinatorial Optimization (EvoCOP),* 2020, pp. 51–66, and won the conference's best-paper award.

CCS CONCEPTS

• **Theory of computation → Theory and algorithms for application domains**; *Theory of randomized search heuristics.*

KEYWORDS

Estimation-of-distribution algorithm; univariate marginal distribution algorithm; run time analysis; epistasis; theory

ACM Reference Format:
Benjamin Doerr and Martin S. Krejca. 2020. The Univariate Marginal Distribution Algorithm Copes Well With Deception and Epistasis. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3397487

1 INTRODUCTION

Estimation-of-distribution algorithms (EDAs) are randomized search heuristics that evolve a probabilistic model of the search space in an iterative manner. Starting with the uniform distribution, an EDA takes samples from its current model and then adjusts it such that better solutions have a higher probability of being generated in the next iteration. This method of refinement leads to gradually better solutions and performs well on many practical problems, often outperforming competing approaches [11].

Theoretical analyses of EDAs also often suggest an advantage of EDAs when compared to evolutionary algorithms (EAs); for an in-depth survey of run time results for EDAs, we refer to the article by Krejca and Witt [7].

The first, and so far only, result to suggest that EDAs can be drastically worse than EAs was recently stated by Lehre and Nguyen [8] via the DeceptiveLeadingBlocks function (DLB for short), which they introduce and which consists of blocks of size 2, each with a deceiving function value, that need to be solved sequentially. The authors prove that many common EAs optimize DLB within $O(n^3)$ fitness evaluations in expectation, whereas the univariate marginal distribution algorithm [10] (UMDA; an EDA) has a run time of $e^{\Omega(\mu)}$ (where μ is an algorithm-specific parameter that often is chosen as a small power of n) for a large regime of parameters. Only for extreme parameter values, the authors prove an expected run time of $O(n^3)$ also for the UMDA.

In the paper that this abstract is based on [2], we prove that the UMDA is, in fact, able to optimize DLB in time $O(n^2 \log n)$ with high probability if its parameters are chosen more carefully. Thus, we disagree with the statement of Lehre and Nguyen [8] that there are "inherent limitations of univariate EDAs against deception and epistasis". In addition to the improved run time, we derive our result using only tools commonly used in the analysis of EDAs and EAs, and we are optimistic that our analysis method can be useful also in other run time analyses of EDAs.

2 OUR RESULTS

We consider the UMDA, whose probabilistic model for optimizing a function defined on bit strings of length n is a vector of n probabilities p (each called a *frequency*), each frequency being initially $\frac{1}{2}$. When the UMDA samples an individual with respect to this model, each frequency denotes the probability to have a 1 at the respective position (independent of any other position). In each iteration, the UMDA samples λ individuals and chooses the μ best (breaking ties uniformly at random). Then, for each position $i \in [1, n] \cap \mathbb{N}$, the frequency p_i is set to the relative number of 1s at position i among the μ selected individuals. Last, all frequencies are capped to the interval $[\frac{1}{n}, 1 - \frac{1}{n}]$.

We are interested in how well the UMDA optimizes the DLB problem, which considers a bit string to consist of $\frac{n}{2}$ blocks of length 2, each block being a trap function, where 11 has the highest

GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico

ACM ISBN 978-1-4503-7127-8/20/07.
https://doi.org/10.1145/3377929.3397487

fitness, 00 the second highest, and the rest the lowest. We say that a block with a 11 is *correct*. DLB returns the longest prefix of correct blocks plus the value of the following *(critical)* block. Note that the critical block contributes with a deceptive value to the fitness and that the global optimum of DLB is the all-1s bit string.

Since positions after the critical block are irrelevant, these positions do not contribute to the selection process of the UMDA for a long time. We call such positions *neutral*. During this period, the probabilistic model of the UMDA starts degenerating at neutral positions [5]. Here, by *degenerate*, we mean that the sampling frequencies approach the boundary values $\frac{1}{n}$ and $1 - \frac{1}{n}$ without that this is justified by the fitness function. This leads to a probabilistic model that is strongly concentrated around a single, suboptimal search point. This effect is often called *genetic drift* [12]. While it appears natural to choose the parameters of an EDA as to prevent genetic drift, it also has been proven that genetic drift can lead to a complicated run time landscape and inferior performance (see Lengler et al. for the cGA [9]).

The key difference between the parameter regime of $\mu = \Omega(\log n) \cap o(n)$ and $\lambda = \Theta(\mu)$ used by Lehre and Nguyen [8] and our regime of $\mu = \Omega(n \log n)$ and $\lambda = \Omega(\mu)$ is that the UMDA is prone to genetic drift in the former regime but not in the latter. Choosing $\mu = \Omega(n \log n)$ results in frequencies at neutral positions to remain close to their initial value of $\frac{1}{2}$ for at least n iterations, with high probability [4]. Further, for a sufficiently large constant c, we choose $\lambda \geq c\mu$ in order to ensure that, during the selection step, the UMDA chooses μ individuals that extend the correct prefix of DLB by at least one block, also with high probability. Thus, after $\frac{n}{2}$ iterations, all frequencies are at their correct boundary values of $1 - \frac{1}{n}$, and it only takes an additional $2e \ln n$ iterations to sample the optimum, both with high probability.

A major problem for this analysis is that blocks right after the critical block can become not neutral and, consequently, be affected by the deceptive signal of DLB. Imagine that, at the start of an iteration, block i is the first block with frequencies less than their maximum of $1 - \frac{1}{n}$. Due to our choice of λ and μ, each of the μ selected individuals of the UMDA has at least i correct blocks. Let $j > i$ be the first position where not all μ selected individuals have a correct block. Then this block is *not* neutral during selection. Due to the deceptiveness of DLB, if the frequencies in block j are below $\frac{1}{2}$, it is more likely to sample 00s than 11s, which get selected, since they are almost as good. In this case, the frequencies in block j can decrease drastically, below the initial bounds we are guaranteed by our choice of $\mu = \Omega(n \log n)$.

Our analysis [2] takes this fact into account and rigorously calculates how far each frequency can drop at most, with high probability, once it is prone to deception. This value tells us how we should choose the constant c for $\lambda \geq c\mu$ to make sure that frequencies that drop so low are *still* optimized within a single iteration. Noting that each frequency can only be prone to deception at most once during the optimization process (since it is optimized afterward), yields that a new block is optimized in each iteration. Thus, the UMDA optimizes DLB within $O(n)$ iterations, with high probability, which translates into $O(\lambda n)$ fitness evaluations.

3 CONCLUSION

Our work shows that the UMDA, in contrast to what was suggested by Lehre and Nguyen [8], is actually very suitable to optimize the DLB problem. With the right parameters, which are very natural in the light of the analysis [4] (appeared after [8]), the UMDA solves the DLB problem in time $O(n^2 \log n)$ with high probability. Since for standard EAs no better runtime guarantee than $O(n^3)$ is known (and we strongly believe this to be the true order of magnitude), we rather observe an example where EDAs outperform classic EAs. Together with the high runtime proven by Lehre and Nguyen [8] in the regime with genetic drift, this constitutes the first example for which a drastic runtime difference could be proven depending on whether the EDA is run in the regime with or without genetic drift. This result shows how important the correct choice of the update strength of the probabilistic model in EDAs is. While for the compact genetic algorithm [6] (cGA, another EDA) two ways to automatically find a suitable parameter values have been proposed [1, 3] recently, such approaches have not yet been found for the UMDA.

ACKNOWLEDGMENTS

This research was supported by COST action CA15140 and by a public grant as part of the Investissement d'avenir project reference ANR-11-LABX-0056-LMH, LabEx LMH, in a joint call with Gaspard Monge Program for optimization, operations research and their interactions with data sciences.

REFERENCES

[1] Benjamin Doerr. 2019. A tight runtime analysis for the cGA on jump functions: EDAs can cross fitness valleys at no extra cost. In *Proc. of GECCO '19*. 1488–1496. https://doi.org/10.1145/3321707.3321747

[2] Benjamin Doerr and Martin S. Krejca. 2020. The univariate marginal distribution algorithm copes well with deception and epistasis. In *Proc. of EvoCOP'20*. 51–66. https://doi.org/10.1007/978-3-030-43680-3_4

[3] Benjamin Doerr and Weijie Zheng. 2020. A parameter-less compact genetic algorithm. In *Genetic and Evolutionary Computation Conference, GECCO 2020*. ACM. To appear.

[4] Benjamin Doerr and Weijie Zheng. 2020. Sharp bounds for genetic drift in estimation-of-distribution algorithms. *IEEE Transactions on Evolutionary Computation* (2020). https://doi.org/10.1109/TEVC.2020.2987361 To appear.

[5] Tobias Friedrich, Timo Kötzing, and Martin S. Krejca. 2016. EDAs cannot be balanced and stable. In *Proc. of GECCO '16*. 1139–1146. https://doi.org/10.1145/2908812.2908895

[6] Georges R. Harik, Fernando G. Lobo, and David E. Goldberg. 1999. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation* 3 (1999), 287–297.

[7] Martin S. Krejca and Carsten Witt. 2020. Theory of estimation-of-distribution algorithms. In *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, Benjamin Doerr and Frank Neumann (Eds.). Springer, 405–442. Also available at https://arxiv.org/abs/1806.05392.

[8] Per Kristian Lehre and Phan Trung Hai Nguyen. 2019. On the limitations of the univariate marginal distribution algorithm to deception and where bivariate EDAs might help. In *Proc. of FOGA '19*. 154–168. https://doi.org/10.1145/3299904.3340316

[9] Johannes Lengler, Dirk Sudholt, and Carsten Witt. 2018. Medium step sizes are harmful for the compact genetic algorithm. In *Proc. of GECCO '18*. 1499–1506. https://doi.org/10.1145/3205455.3205576

[10] Heinz Mühlenbein and Gerhard Paaß. 1996. From recombination of genes to the estimation of distributions I. Binary parameters. In *Proc. of PPSN '96*. 178–187. https://doi.org/10.1007/3-540-61723-X_982

[11] Martin Pelikan, Mark Hauschild, and Fernando G. Lobo. 2015. Estimation of distribution algorithms. In *Springer Handbook of Computational Intelligence*. Springer, 899–928. https://doi.org/10.1007/978-3-662-43505-2_45

[12] Dirk Sudholt and Carsten Witt. 2019. On the choice of the update strength in estimation-of-distribution algorithms and ant colony optimization. *Algorithmica* 81, 4 (2019), 1450–1489. https://doi.org/10.1007/s00453-018-0480-z