



HAL
open science

Runtime Analysis of the $(\mu + 1)$ GA: Provable Speed-Ups from Strong Drift towards Diverse Populations

Benjamin Doerr, Aymen Echarchaoui, Mohammed Jamal, Martin Krejca

► **To cite this version:**

Benjamin Doerr, Aymen Echarchaoui, Mohammed Jamal, Martin Krejca. Runtime Analysis of the $(\mu + 1)$ GA: Provable Speed-Ups from Strong Drift towards Diverse Populations. GECCO '24 Companion: Genetic and Evolutionary Computation Conference Companion, 2024, Melbourne, Australia. pp.35-36, 10.1145/3638530.3664065 . hal-04676962

HAL Id: hal-04676962

<https://hal.science/hal-04676962v1>

Submitted on 24 Aug 2024

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

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



Runtime Analysis of the $(\mu + 1)$ GA: Provable Speed-Ups from Strong Drift towards Diverse Populations

Benjamin Doerr

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

Aymen Echarchaoui

École Polytechnique,
Institut Polytechnique de Paris
Palaiseau, France

Mohammed Jamal

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

Most evolutionary algorithms used in practice heavily employ crossover. In contrast, the rigorous understanding of how crossover is beneficial is largely lagging behind. In this work, we make a considerable step forward by analyzing the population dynamics of the $(\mu + 1)$ genetic algorithm when optimizing the JUMP benchmark. We observe (and prove via mathematical means) that once the population contains two different individuals on the local optimum, the diversity in the population increases in expectation. From this drift towards more diverse states, we show that a diversity suitable for crossover to be effective is reached quickly and, more importantly, then persists for a time that is at least exponential in the population size μ . This drastically improves over the previously best known guarantee, which is only quadratic in μ .

Our new understanding of the population dynamics easily gives stronger performance guarantees. In particular, we derive that population sizes logarithmic in the problem size n already suffice to gain an $\Omega(n)$ -factor runtime improvement from crossover (previous works achieved comparable bounds only with $\mu = \Theta(n)$ or via a non-standard mutation rate).

This paper for the hot-off-the-press track at GECCO 2024 summarizes the work *Benjamin Doerr, Aymen Echarchaoui, Mohammed Jamal, and Martin S. Krejca: Runtime analysis of the $(\mu + 1)$ GA: Provable speed-ups from strong drift towards diverse populations. Conference on Artificial Intelligence, AAAI 2024. AAAI Press, 20683–20691. DOI: 10.1609/aaai.v38i18.30055* [4].

CCS CONCEPTS

• Theory of computation → Theory of randomized search heuristics; • Mathematics of computing → Evolutionary algorithms.

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).
GECCO '24 Companion, July 14–18, 2024, Melbourne, VIC, Australia
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0495-6/24/07.
<https://doi.org/10.1145/3638530.3664065>

KEYWORDS

Theory, runtime analysis, genetic algorithm, crossover, Jump

ACM Reference Format:

Benjamin Doerr, Aymen Echarchaoui, Mohammed Jamal, and Martin S. Krejca. 2024. Runtime Analysis of the $(\mu + 1)$ GA: Provable Speed-Ups from Strong Drift towards Diverse Populations. In *Genetic and Evolutionary Computation Conference (GECCO '24 Companion)*, July 14–18, 2024, Melbourne, VIC, Australia. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3638530.3664065>

1 INTRODUCTION

The vast majority of the applications of evolutionary algorithms (EAs) uses crossover, that is, new offspring are generated from two parents. Surprisingly, there is very little rigorous evidence for the usefulness of crossover. This is not only a fundamental open question, but more importantly also indicates that our understanding of the working principles of crossover is very limited, which means that there is very little trustworthy advice how to best employ crossover in evolutionary algorithms.

In this work [4], we analyze crossover via theoretical means, more specifically, via mathematical runtime analyses, which have given many important insights and explanations in the past [1, 6, 8, 11, 13]. Unfortunately, when it comes to understanding crossover, the mathematical runtime analysis area was not very successful (but we note that other attempts to understand crossover, e.g., the building block hypothesis [7] also struggled to explain crossover [10]).

The main difficulty towards understanding crossover via mathematical tools are the usually complicated population dynamics. We note that even when only regarding mutation-based algorithms, still the majority of runtime analyses regards algorithms with trivial populations such as *randomized local search* or the $(1 + 1)$ EA. The particular difficulty when adding crossover is that we need to understand the diversity of the population. Crossover can only be effective when sufficiently different, good solutions are available.

This core problem is easily visible in the existing runtime analyses of crossover-based algorithms. That diversity, more precisely, arriving at a state with a diverse population, is the crucial ingredient for profiting from crossover was already discussed very clearly in the first runtime analysis of a crossover-based algorithm [9], which analyzes how the $(\mu + 1)$ genetic algorithm (GA) optimizes

the JUMP benchmark. Unfortunately, the positive effect of crossover could be shown here only for unrealistically small crossover rates. Doerr et al. [5], without taking uncommon assumptions, showed an advantage of crossover when solving the all-pairs-shortest-path problem. This problem, asking for a short path between any two vertices, already in the problem definition contains a strong diversity mechanism that in particular prevents the population from converging to a single genotype. Consequently, it is not clear how the insights gained in that work extend to algorithms without diversity mechanisms.

The maybe most convincing proof for an advantage from crossover was given by Dang et al. [2], who again regarded how the $(\mu + 1)$ GA optimizes JUMP functions. Interestingly, without particular modifications of the algorithm or non-standard parameter settings, the authors managed to show that the $(\mu + 1)$ GA repeatedly arrives at diverse populations and keeps this diversity for a moderate time. More precisely, once the population is concentrated on the local optima of the JUMP function, it takes an expected time of $O(n\mu + \mu^2 \log \mu)$ iterations to reach a population in which all genotype classes contain at most $\mu/2$ individuals. A diversity of this order of magnitude is kept for an expected number of $O(\mu^2)$ iterations. Given these numbers, the best runtime results are obtained for a relatively large population sizes of order $\tilde{\Theta}(n)$, where a runtime gain of roughly a factor of $n/\log n$ from crossover can be proven.

The key argument of Dang et al. [2] is that the size of the largest genotype class, once below μ , is at least as likely to decrease as to increase. This behavior resembles an unbiased random walk in the interval $[\mu/2, \mu]$ and is the reason why the diversity is shown to persist for $\Omega(\mu^2)$ iterations.

Our results. We analyze the random process describing the largest genotype class of the $(\mu + 1)$ GA on JUMP more carefully. We observe that there is not an unbiased random walk behavior, as pessimistically assumed in the previous work, but that instead it is more likely that the dominant genotype class reduces than increases [4, Lemma 5]. This advantage is strongest when the largest genotype class covers only a constant fraction of the population. Consequently, we show that once the largest genotype class contains at most $\mu/2$ individuals, a diversity of this order of magnitude is kept for an expected number of $\exp(\Omega(\mu))$ iterations.

This result is interesting in its own right as it shows that diverse parent populations are much easier to obtain than what previous works suggest. For the particular problem of how fast the $(\mu + 1)$ GA optimizes JUMP functions, we obtain a stronger advantage from crossover [4, Theorem 8], namely a speed-up by a factor of $\Omega(n)$, and this already for population sizes of order $\Omega(\ln n)$. For smaller values of μ , we still obtain a significant speed-up from crossover, namely by a factor of $\exp(\Omega(\mu))$, again comparing favorably with the speed-up of $\Omega(\mu)$ shown by Dang et al. [2]. In addition, our results hold for any constant crossover and mutation rate, further generalizing the results by Dang et al. [2].

We complement our theoretical upper bounds with an empirical analysis. Our experiments support our finding that once the population is not anymore dominated by a single genotype, this diversity lasts for a long time. However, our experiments also show that the diversity produced by the $(\mu + 1)$ GA is even stronger than

what our proofs show. Not only does no genotype class dominate, but also the typical Hamming distance between individuals grows. If such an effect could be proven, this would immediately lead to much stronger runtime guarantees.

Summary and conclusion. Our results show that also basic evolutionary algorithms without diversity mechanisms or other adjustments are able to reach and then keep a diverse population for a long time. This is clearly good news for the use of crossover, but this lasting diversity might also lead to other advantages such as a more effective exploration of the search space or a lower risk of getting stuck in local optima.

From the viewpoint of designing effective genetic algorithms, our work suggest to use crossover with constant crossover rate, and without deviating from standard parameter suggestions such as using a mutation rate of $\frac{1}{n}$. In our results, the greatest speed-up is achieved for population sizes μ that are logarithmic in n .

Subsequent work and outlook. Very recently Opris, Lengler, and Sudholt [12] considered a variant of the $(\mu + 1)$ GA on JUMP, which creates multiple offspring when performing crossover, choosing uniformly at random one of the best thus-generated individuals. They build on top of recent work that analyzed the diversity of various classes of EAs and GAs [3] and prove that the $(\mu + 1)$ GA variant optimizes JUMP with gap size k in expected time $O(\mu n \log k + 4^k)$, choosing optimal parameters, i.e., a linear population size and a constant crossover rate, bounded away from 1. This result is impressive and showcases the potential of crossover. Combining the insights of their result with ours might lead to a similar runtime bound of the original $(\mu + 1)$ GA on JUMP, even for smaller population sizes.

REFERENCES

- [1] Anne Auger and Benjamin Doerr (Eds.). 2011. *Theory of Randomized Search Heuristics*. World Scientific Publishing.
- [2] Duc-Cuong Dang, Tobias Friedrich, Timo Kötzing, Martin S. Krejca, Per Kristian Lehre, Pietro S. Oliveto, Dirk Sudholt, and Andrew M. Sutton. 2018. Escaping local optima using crossover with emergent diversity. *IEEE Transactions on Evolutionary Computation* 22 (2018), 484–497.
- [3] Duc-Cuong Dang, Andre Opris, Bahare Salehi, and Dirk Sudholt. 2023. A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. In *Conference on Artificial Intelligence, AAAI 2023*. AAAI Press, 12390–12398.
- [4] Benjamin Doerr, Aymen Echarchaoui, Mohammed Jamal, and Martin S. Krejca. 2024. Runtime Analysis of the $(\mu + 1)$ GA: Provable Speed-Ups from Strong Drift towards Diverse Populations. In *Conference on Artificial Intelligence, AAAI 2024*. AAAI Press, 20683–20691. <https://doi.org/10.1609/aaai.v38i18.30055>
- [5] Benjamin Doerr, Edda Happ, and Christian Klein. 2012. Crossover can provably be useful in evolutionary computation. *Theoretical Computer Science* 425 (2012), 17–33.
- [6] Benjamin Doerr and Frank Neumann (Eds.). 2020. *Theory of Evolutionary Computation—Recent Developments in Discrete Optimization*. Springer.
- [7] John H. Holland. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- [8] Thomas Jansen. 2013. *Analyzing Evolutionary Algorithms – The Computer Science Perspective*. Springer.
- [9] Thomas Jansen and Ingo Wegener. 2002. The analysis of evolutionary algorithms – a proof that crossover really can help. *Algorithmica* 34 (2002), 47–66.
- [10] Melanie Mitchell, John H. Holland, and Stephanie Forrest. 1993. When will a genetic algorithm outperform hill climbing. In *Advances in Neural Information Processing Systems, NIPS 1993*. Morgan Kaufmann, 51–58.
- [11] Frank Neumann and Carsten Witt. 2010. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Springer.
- [12] Andre Opris, Johannes Lengler, and Dirk Sudholt. 2024. A Tight $O(4^k/p_c)$ Runtime Bound for a $(\mu + 1)$ GA on Jump_k for Realistic Crossover Probabilities. *CoRR* abs/2404.07061 (2024). To be published at GECCO 2024.
- [13] Zhi-Hua Zhou, Yang Yu, and Chao Qian. 2019. *Evolutionary Learning: Advances in Theories and Algorithms*. Springer.