



Preface to the Special Issue on Theory of Genetic and Evolutionary Computation

Anne Auger¹ · Per Kristian Lehre²

Published online: 15 March 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

The articles collected in this special issue are extensions of conference papers published in the theory track of the ACM Genetic and Evolutionary Computation Conference (GECCO) 2018 conference. The papers have followed a thorough review process to meet the high standards of Algorithmica.

They reflect the state of research in theory of randomised search algorithms where many important results are presented at the GECCO conference. The seven articles cover analysis of evolutionary algorithms in discrete search spaces. The first four papers focus on how the design of the algorithms impacts their running times, while the last three articles consider the impact of various forms of uncertainty in the objective (fitness) functions.

Evolutionary algorithms are population-based methods. A population size greater than one is crucial in many difficult situations. Yet many theoretical analyses deal with algorithms with a trivial population (reduced to one) such as the (1+1) EA. In “A Tight Runtime Analysis for the $(\mu + \lambda)$ EA”, Denis Antipov and Benjamin Doerr derive the asymptotically tight bound of $\Theta(n \log n + \mu n + \lambda n \log \log(\lambda/\mu) / \log(\lambda/\mu))$ on the expected optimisation time of the population-based $(\mu + \lambda)$ EA on the ONEMAX problem.

Most evolutionary algorithms (EAs) have one or more parameters, such as mutation rate and population size. Understanding how the parameters of evolutionary algorithms affect their performance is a central theoretical question with practical relevance. In “The Complex Parameter Landscape of the Compact Genetic Algorithm”, Johannes Lengler, Dirk Sudholt, and Carsten Witt analyse rigorously the influence of the parameter $1/K$ (sometimes called the “step size”) on the runtime of the compact GA (cGA). Surprisingly, the expected runtime is a bimodal function of this parameter with two distinct high quality regions separated by a region of poor performance. In particular, the expected optimisation time of the

✉ Anne Auger
anne.auger@inria.fr

¹ Inria and Ecole Polytechnique, IP Paris, France

² University of Birmingham, Birmingham, UK

cGA on ONEMAX is exponential for $K \leq \log_{10}(n)$, it is in $\Omega(K^{1/3}n) \cap \mathcal{O}(Kn)$ for $K \in \Omega(\log n) \cap \mathcal{O}(\sqrt{n} \log n)$, and it is in $\Theta(K\sqrt{n})$ for $K = \Omega(\sqrt{n} \log n)$.

There is an increasing interest in designing parameter control mechanisms for EAs which allow online adaptation of their parameter settings without the intervention of the user. In the paper “*Runtime Analysis for Self-adaptive Mutation Rates*”, Benjamin Doerr, Carsten Witt, and Jing Yang study a variant of the $(1, \lambda)$ EA where offspring either increases or decreases their mutation rates relative to the parent by a fixed factor F . Assuming an offspring population size $\lambda = \Omega(\log n)$, they show that this simple adaptation scheme suffices to achieve a runtime of $\mathcal{O}(n\lambda / \log \lambda + n \log n)$ on the ONEMAX problem. This optimisation time is asymptotically optimal within the class of λ -parallel mutation-based unbiased black-box algorithms.

Some evolutionary algorithms, such as the $(\mu + \lambda)$ GA, use both a crossover and a mutation operator, while others, such as the $(\mu + \lambda)$ EA, use only a mutation operator. A fundamental problem in theory of evolutionary computation is to characterise when the crossover operator is beneficial. The article by Andrew Sutton titled “*Fixed-Parameter Tractability of Crossover: Steady-State GAs on the Closest String Problem*” considers this problem from a novel perspective, that of fixed parameter tractability (FPT). Sutton proves that the $(\mu + 1)$ GA solves the closest string problem for k strings and optimal value d in fixed-parameter tractable time $2^{\mathcal{O}(d^2 + d \log k)} \cdot t(n)$, assuming parent population size $\mu \in \Theta(d + k)$ and algorithm restarts every $t(n)$ iterations for $n \leq t(n) \in \text{poly}(n)$. To prove this result, Sutton shows that under the settings above, the $(\mu + 1)$ GA simulates the SolveCS algorithm by Gramm, Niedermeier, and Rossmanith (Algorithmica 37(1), 25–42, (2003)). In contrast, he shows that the $(\mu + 1)$ EA with parent population size $\mu \in \Theta(d + k)$ and the same restart scheme has expected optimisation time at least $n^{\Omega(\log(d+k))}$, which is not fixed-parameter tractable.

In applications of evolutionary algorithms to objective functions with stochastic noise, the “sampling approach” re-evaluates each individual multiple times, and estimates the true objective value by the average of the samples. The alternative “large population approach” evaluates each individual only once, but hopes that with a sufficiently large population size, the individuals will collectively aggregate sufficient information about the true objective values. It is an important problem to characterise when the large population approach outperforms the sampling approach. The article “*Analysis of Noisy Evolutionary Optimization When Sampling Fails*” by Chao Qian, Chao Bian, Yang Yu, Ke Tang and Xin Yao presents noisy variants of the ONEMAX objective function where the $(1+1)$ EA using re-sampling has exponential optimisation time, while the $(\mu+1)$ EA with parent population size $\mu = 3 \log n$ and offspring population size 1 has expected optimisation time $\mathcal{O}(n \log^3 n)$. Furthermore, they show that if the the parent population size is reduced to $\mu \leq \sqrt{\log n}/2$, then the expected optimisation time of the $(\mu+1)$ EA becomes exponential. They also show that large offspring populations can be beneficial. The $(1+\lambda)$ EA with parent population size 1 and offspring population size $\lambda = 8 \log n$ has expected optimisation time $\mathcal{O}(n \log^2 n)$. However, reducing the offspring population size to $\lambda \leq (\log n)/10$, increases the expected optimisation time of the $(1+\lambda)$ EA exponentially. Finally, they design an “adaptive sampling approach” which heuristically adapts the sample

size based on observed differences between the objective values of individuals. They construct a noisy objective function where the (1+1) EA with the adaptive sampling approach has expected optimisation time $\mathcal{O}(n^4 \log^2 n)$, while the classical sampling and large population approaches lead to exponential expected optimisation times.

In the article “*Analysing the Robustness of Evolutionary Algorithms to Noise: Refined Runtime Bounds and an Example Where Noise is Beneficial*”, Dirk Sudholt considers optimisation of noisy pseudo-Boolean functions $f : \{0, 1\}^n \rightarrow \mathbb{R}$. In the *prior noise model* with noise level $p \in (0, 1/2)$, the noisy objective value of any bit-string $x \in \{0, 1\}^n$ is $f(x)$ with probability $1 - p$ and $f(x')$ with probability p , where x' is a uniformly sampled Hamming neighbour of x . Previously, it has been shown that the (1+1) EA optimises ONE MAX in expected polynomial time if $p = \mathcal{O}((\log n)/n)$. Sudholt proves that the (1+1) EA has expected optimisation time $\Theta(n^2) \cdot e^{\Theta(\min(pn^2, n))}$ on the LEADING ONES problem, which is super-polynomial already for noise levels $p = \omega((\log n)/n^2)$. This implies that the (1+1) EA is less noise-tolerant on LEADING ONES than on ONE MAX. Increasing the offspring population size raises the noise tolerance. Sudholt shows that (1+ λ) EA with $\log_{\frac{e}{e-1/2}}(n) \leq \lambda = \mathcal{O}(n)$ has expected optimisation time $\mathcal{O}(n^2 e^{\mathcal{O}(pn/\lambda)})$. At first, one may assume the optimisation time always increases with the noise level p . Sudholt shows that this is not the case. He proves that the RLS (a simpler variant of the (1+1) EA) optimises the “rugged” optimisation problem HURDLE in expected time $\mathcal{O}(n^2/(pw^2) + n \log n)$ (here, w is a problem parameter). For $w \geq 2$ and $p = 0$, Nguyen and Sudholt have proved before that the RLS has infinite optimisation time on HURDLE. In this setting, adding noise makes it possible for RLS to overcome local optima.

Evolutionary Algorithms (EAs) are often applied to dynamic optimisation problems. It is therefore important to determine how the dynamics impact the optimisation of EAs. In the article “*Runtime performances of randomized search heuristics for the dynamic weighted vertex cover problem*”, Feng Shi, Frank Neumann, and Jianxin Wang consider the RLS and the (1+1) EA as approximation algorithms for the Weighted Vertex Cover Problem. In the *dynamic model* of this combinatorial optimisation problem, they ask: given a 2-approximate solution to a problem instance I , what is the expected time for the algorithm to find a 2-approximate solution to a perturbed instance I' ? They present adaptations of these algorithms that lead to pseudo-polynomial time to recover a 2-approximation.

We thank all authors for their submission, the reviewers for their time in providing thorough reviews and the Algorithmica team including Ming-Yang Kao for their support and patience.

Anne Auger and Per Kristian Lehre
Paris and Birmingham, October 2020