

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

Carola Doerr¹ · Francisco Chicano² 

Received: 10 January 2017 / Accepted: 16 January 2017 / Published online: 1 February 2017
© Springer Science+Business Media New York 2017

Evolutionary algorithms (EAs) are randomized heuristics that are very successfully applied in a broad range of industrial and academic optimization problems. EAs work in a trial-and-error fashion, that is, they sample potential solution candidates, evaluate these, and—based on their quality—adapt the distribution from which the next generation of search points is sampled. Key questions in the design of EAs concern the choice of parameters such as the population size, the strength of variation, or the selection pressure. While a large body of empirical works in evolutionary computation exists, analyzing these general-purpose optimizers by mathematical means is a rather young research domain. The theory track of the annual ACM conference on *Genetic and Evolutionary Computation Conference (GECCO)* is the first tier event for advances in this direction.

In this special issue seven selected papers from the 2015 edition of the GECCO theory track are collected, each one of them carefully revised and extended to meet the high quality standards of *Algorithmica*.

The satisfiability problem (SAT) is one of the most prominent NP-complete problems in Computer Science. Running time analyses of randomized search heuristics for the SAT problem are thus specially challenging and interesting. The work “Time complexity analysis of evolutionary algorithms on random satisfiable k -CNF formulas” by Doerr, Neumann, and Sutton presents such an analysis for a simple $(1 + 1)$

✉ Carola Doerr
Carola.Doerr@mpi-inf.mpg.de
Francisco Chicano
chicano@lcc.uma.es

¹ CNRS, LIP6 UMR 7606, Sorbonne Universités, UPMC Univ Paris 06, 4 Place Jussieu, 75005 Paris, France

² University of Málaga, E.T.S. Ingeniería Informática, Bulevar Louis Pasteur 35, 29071 Málaga, Spain

evolutionary algorithm (EA) solving random k -satisfiability instances. The authors show that the $(1 + 1)$ EA is able to solve such random instances of n variables in time at most $O(n \ln n)$ if the clause to variable ratio is at least logarithmic. In the case of low densities, the algorithm seems to be less effective but still a subexponential optimization time can be shown. These results are proven by a clever use of the fitness-distance correlation.

Most of the analyses of randomized search heuristics provide an asymptotic expression for the running time. However, asymptotic expressions always hide a constant and lower order terms that could be important in practice. Gießen and Witt provide a tight expression for the running time of a $(1 + \lambda)$ evolutionary algorithm solving OneMax in their work “The interplay of population size and mutation probability in the $(1 + \lambda)$ EA on OneMax”. The bit flip probability of the mutation operator is assumed to be c/n . The runtime bound depends on the parameter c and the offspring population size λ , allowing the authors to study the influence of both parameters on the running time. They conclude that for small offspring sizes $\lambda = o(\ln n \ln \ln n / \ln \ln \ln n)$, the running time is minimized for $c = 1$. Interestingly, for larger values of λ , the running time is, up to lower terms, independent of c .

As in classical algorithms theory, also in evolutionary computation an important counterpart to running time analysis is complexity theory. Reflecting the performance measure typically regarded in this field, *black-box complexity* measures, intuitively speaking, the number of function evaluations that are needed by any trial-and-error black-box algorithm to identify an optimal solution for the problem at hand. Many different black-box complexity models exist, each one imposing different restrictions on the algorithms (e.g., the amount of memory or the disability to use absolute function values rather than relative information). By comparing the complexity of a problem in these different models, we learn how certain algorithmic choices influence the performance of the respective algorithms. In their work “OneMax in black-box models with several restrictions”, Doerr and Lengler regard how the complexity of a classic optimization problem changes if several of previously regarded black-box models are combined.

In the context of parallel (or decentralized) evolutionary algorithms, the island model works by evolving several sub-populations in an isolated way. Individuals are exchanged among the sub-populations with a frequency that is determined by the migration interval. Such parallel evolutionary algorithms have been used with success for solving dynamic optimization problems; i.e., problems whose objective function changes with time. Dynamism is a quite common feature of many real-world problems. The work “A runtime analysis of parallel evolutionary algorithms in dynamic optimization” by Lissovoi and Witt is the first analyzing the running time of a parallel evolutionary algorithm using the island model for solving a dynamic optimization problem: the MAZE problem. They study how the number of islands and the migration interval impacts the ability of a parallel evolutionary algorithm for tracking optimal solutions.

Population size is suspected to be an important parameter when evolutionary algorithms are applied to dynamic optimization problems. In the work “Populations can be essential in tracking dynamic optima”, Dang, Jansen, and Lehre analyze the influence of the population size on the effectiveness of an evolutionary algorithm to track the

optimal solution of a dynamic optimization problem. The work focuses on a quite general family of fitness functions and proves that an evolutionary algorithm without population will only reach the (moving) optimal region with a very low probability, while a population-based algorithm is able to track the optimum efficiently. The population-based algorithm used in their work includes a non-elitist replacement strategy and requires a population size that increases at least linearly with the problem size. The result is proven for four different selection mechanisms.

In the work “Towards a runtime comparison of natural and artificial evolution” the authors Paixão, Pérez Heredia, Sudholt, and Trubenová apply some of the recently developed tools from the theory of evolutionary computation to an algorithm inspired by population-genetics. While many standard evolutionary algorithms are elitist in the sense that only the current-best solutions have a good chance of forming part of the next generation, in the Strong Selection Weak Mutation (SSWM) algorithm, this probability is positive also for search points that are not as good as the current-best ones. Paixão et al. analyze how sensitive SSWM is against changes of the parameters that characterize the survival probabilities and compare the obtained bounds with those of the traditional $(1 + 1)$ evolutionary algorithm (EA). It is demonstrated that SSWM can have advantages over the $(1 + 1)$ EA at crossing fitness valleys by using information of the fitness gradient.

For the development of evolutionary and other bio-inspired algorithms it is crucial to understand how typical representatives behave on rather easy optimization problems and what the more difficult fitness landscapes look like. This question is addressed in the work “On easiest functions for mutation operators in bio-inspired optimisation” by Corus, He, Jansen, Oliveto, Sudholt, and Zarges, where easiest and most difficult problems for the contiguous somatic hypermutation (CHM) operator used in artificial immune systems are presented. Since the easiest problem, MinBlocks, is among the most difficult ones for the commonly applied standard bit mutation (SBM) operator, it is also demonstrated that a hybridization of CMH and SBM yields an algorithm having good performance on both MinBlocks and the classical OneMax function (which is known to be an easiest non-trivial problem for SBM). Furthermore, it is also shown that an easiest function for the hybrid algorithm is not just a weighted combination of the respective easiest functions for each operator.

We hope that with this special issue we further increase the interest of the general algorithms research community into evolutionary computation methods. We thank all authors for their submissions, our reviewers for their helpful and detailed comments, and last but not least the *Algorithmica* team and the editor-in-chief Ming-Yang Kao for their great support.