Leveraging benchmarking data for informed one-shot dynamic algorithm selection

Ye, F.; Doerr, C.; Bäck, T.H.W.; Chicano, F.

Leveraging Benchmarking Data for Informed One-Shot Dynamic Algorithm Selection

Furong Ye
LIACS, Leiden University
Leiden, Netherlands
f.ye@liacs.leidenuniv.nl

Carola Doerr
Sorbonne Université, CNRS, LIP6
Paris, France
Carola.Doerr@lip6.fr

Thomas Bäck
LIACS, Leiden University
Leiden, Netherlands
t.h.w.baeck@liacs.leidenuniv.nl

ABSTRACT

A key challenge in the application of evolutionary algorithms in practice is the selection of an algorithm instance that best suits the problem at hand. What complicates this decision further is that different algorithms may be best suited for different stages of the optimization process. Dynamic algorithm selection and configuration are therefore well-researched topics in evolutionary computation. Two different settings are classically considered: hyper-heuristics and parameter control studies typically assume a setting in which the algorithm needs to be chosen and adjusted during the run, without prior information, other approaches such as hyper-parameter tuning and automated algorithm configuration assume the possibility of evaluating different configurations before making a final recommendation. In practical applications of evolutionary algorithms we are often in a middle-ground between these two settings, where one needs to decide upon the algorithm instance before the run ("oneshot" setting), but where we have (possibly lots of) data available on which we can base an informed decision.

We analyze in this work how such prior performance data can be used to infer informed dynamic algorithm selection schemes for the solution of pseudo-Boolean optimization problems. Our specific use-case considers a family of genetic algorithms.

CCS CONCEPTS

• **Theory of computation** → *Bio-inspired optimization.*

KEYWORDS

Genetic Algorithms, Dynamic Algorithm Selection, Black-Box Optimization, Evolutionary Computation

ACM Reference Format:
Furong Ye, Carola Doerr, and Thomas Bäck. 2021. Leveraging Benchmarking Data for Informed One-Shot Dynamic Algorithm Selection. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459578

1 INTRODUCTION

It is very well known that different algorithms or different instantiations of the same algorithm are best suited for different problems and even for different stages of the optimization process. Automated algorithm selection [3] as well as dynamic parameter selection [2] are therefore intensively studied meta-optimization problems in evolutionary computation. However, the former has a strong requirement on being able to run different algorithms (or algorithm configurations) prior to making a decision which algorithm to apply to the problem at hand. Parameter control and related concepts (including hyper-heuristics, adaptive operator control, etc.), in contrast, assume that the selection has to be made on the fly, without leveraging existing data from previous or related runs. With the rise of artificial intelligence methods, evolutionary computation is currently facing a paradigm shift, in that we aim to actively exploit existing performance data to select which algorithms to apply, and how to possibly adjust them during the run. We are, however, still far from achieving a fully automated informed online selection.

We study in this work how well we can predict from existing performance data which algorithm instances to combine for a given problem at hand. While we do allow for switching between different algorithms, the decision when to switch has to be made prior to the run, and depends, in our case, on the solution quality of the evaluated solution candidates. More precisely, we use the benchmarking data from our study [7] as starting point to investigate, for each of the 25 individual problems, how well we can predict which single-switch algorithm combinations would show good performance. For some functions we easily obtain algorithm combinations that outperform the best static algorithms. For other functions the results are rather mixed. On three functions, none of the 100 tested single-switch algorithm combinations was able to outperform the best static solver. The prediction quality of the approach suggested in [4] varies a lot between the different functions. While for LeadingOnes, for example, the performance predictions are rather accurate, large discrepancies between predicted and actual performance can be observed for more complex function. In particular for multi-model functions the approach can get trapped by a first algorithm that is very efficient in converging to a local optimum from which the second algorithm cannot escape easily.

Data availability: Our data is available at [6].

2 INFORMED 1-SWITCH DYNAMIC ALGORITHM SELECTION

We take as input the benchmarking data from [7], which comprise detailed performance records for 80 genetic algorithms on the 25 functions suggested in [1]. We focus on expected running time (ERT) as performance measure, i.e., the average time needed by an algorithm to reach a given solution quality, the *target value*, denoted

GECCO '21 Companion, July 10–14, 2021, Lille, France
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459578

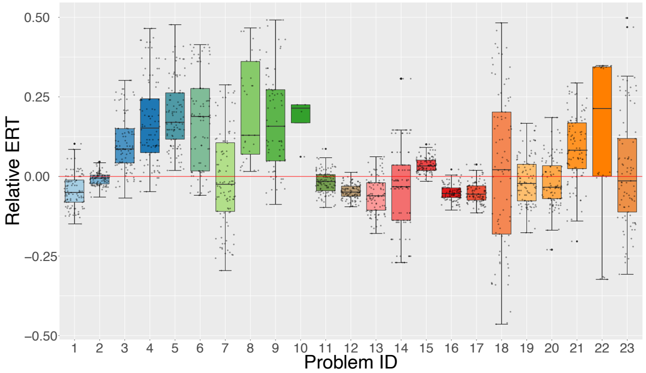

Figure 1: Relative ERT values of 100 **1-switch combinations** (A_1, A_2, ϕ_s) **(***dERT***) for 23 out of 25 IOHprofiler problems in dimension** $d = 100$**, compared to the ERT of the best static GAs according to [7] (***sERT***). Each black dot represents one ERT value. The relative deviation is calculated by** $(dERT - sERT)/sERT$ **so that negative values (below the red line) correspond to an advantage of the dynamic combination over the best static algorithm. We only display values between** -0.5 **and** 0.5 **so that the results of F24-F25 are missing here with values larger than** 1**. All ERT values are based on 100 independent runs.**

by ϕ_f in the following. More precisely, the ERT(A, P, ϕ) of an algorithm A on problem P for target ϕ is computed as $\frac{\sum_{i=1}^{r} \min\{t_i(A,P,\phi),B\}}{\sum_{i=1}^{r} \mathbb{1}\{t_i(A,P,\phi)<\infty\}}$, where B is the total budget of function evaluations that the algorithm can perform, $t_i(A, P, \phi)$ is the number of function evaluations that were needed in the i-th run to reach the target value ($t_i(A, P, \phi) = \infty$ if none of the evaluated solutions satisfies this quality constraint), and $\mathbb{1}(\mathcal{E}) = 1$ if event $\mathcal{E}$ is true and $\mathbb{1}(\mathcal{E}) = 0$, otherwise.

Following the approach suggested in [4] we compute a "theoretical" ERT value for all combinations (A_1, A_2, ϕ_s), where A_1 is the first algorithm, A_2 the second, and ϕ_s the target value at which we switch from A_1 to A_2. To this end, we simply compute ERT(A_1, P, ϕ_s) + ERT(A_2, P, ϕ_f) − ERT(A_2, P, ϕ_s), where all these ERT values are based on the performance recodes provided in [7]. In total, we consider 42 possible switching points ϕ_s, which we select within the interval $[\phi_m, \phi_f]$ between the smallest fitness value ϕ_m of the problem and the best found target ϕ_f according to [7, Table 1]. We consider evenly spaced targets, for the original and for the log-scaled interval, respectively. For each problem, we consider only algorithms that hit the final target value with probability at least 80% according to the data from [7]. Using this approach, we select for each problem the 100 best combinations (A_1, A_2, ϕ_s) and we then run the combination 100 independent times on the problem that they have been selected for.

In Figure 1 we compare the so-obtained ERT values with the best ERT value reported in [7], which we refer to as the *best static algorithm* (BSA). For combinations (A_1, A_2, ϕ_s) for which the parent population sizes μ_1 of A_1 is larger than the parent population size μ_2 of A_2 we selected the best μ_2 points to initialize the parent population of A_2. Where $\mu_1 < \mu_2$, the new parent population comprises all μ_1 points, as additional $\lfloor \mu_2/2 \rfloor - \mu_1$ copies of the best points, and $\lceil \mu_2/2 \rceil$ randomly added individuals.

For some of the problems (e.g., F1, F2, F7, F11-14, F16-23)), the ERT of several combinations (A_1, A_2, ϕ_s) outperform that of the BSA. For other functions, and in particular for F10, F24, and F25, none of the combinations (A_1, A_2, ϕ_s) is able to outperform the BSA. A few reasons and more detailed analyses can be found in the extended version of this poster, available at [5].

3 FUTURE WORK

We have investigated in this work possibilities to leverage existing benchmark data to derive switch-once dynamic algorithm selection policies. While for some cases the "theoretical" approach suggested in [4] could indeed predict combinations that outperformed the best static solver, the results are less positive for others. One obstacle that hinders an accurate performance prediction are local optima: when the first algorithm is very good at converging to a local optimum, it is likely to be chosen as A_1. It is then important, however, to continue the search with an algorithm that has a good enough exploration power to escape the local optimum. This ability, however, seems hard to infer from the pure performance profiles, and may require a "human in the loop".

Going forward, our long-term goal is the automated detection of situations in which switching from one algorithm to another one can be beneficial. To this end, we will further investigate efficient strategies to *warm-start* the algorithms by actively using the information accumulated thus far. In the here-presented study, we have used ERT values as performance measure and as indicator to select which algorithm combinations to execute. In future work we will consider other performance measures, and in particular those that measure the anytime performance of the algorithms.

ACKNOWLEDGMENTS

Our work is supported by the Chinese scholarship council (CSC No. 201706310143) and by the Paris Ile-de-France region.

REFERENCES

[1] Carola Doerr, Furong Ye, Naama Horesh, Hao Wang, Ofer M Shir, and Thomas Bäck. 2020. Benchmarking discrete optimization heuristics with IOHprofiler. *Applied Soft Computing* 88 (2020), 106027. https://doi.org/10.1016/j.asoc.2019.106027

[2] Giorgos Karafotias, Mark Hoogendoorn, and A.E. Eiben. 2015. Parameter Control in Evolutionary Algorithms: Trends and Challenges. *IEEE Transactions on Evolutionary Computation* 19 (2015), 167–187. https://doi.org/10.1109/TEVC.2014.2308294

[3] Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. 2019. Automated Algorithm Selection: Survey and Perspectives. *Evolutionary Computation* 27, 1 (2019), 3–45. https://doi.org/10.1162/evco_a_00242

[4] Diederick Vermetten, Hao Wang, Thomas Bäck, and Carola Doerr. 2020. Towards dynamic algorithm selection for numerical black-box optimization: investigating BBOB as a use case. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'20)*. ACM, 654–662. https://doi.org/10.1145/3377930.3390189

[5] Furong Ye, Carola Doerr, and Thomas Bäck. 2021. Leveraging Benchmarking Data for Informed One-Shot Dynamic Algorithm Selection. *CoRR* abs/2102.06481 (2021). arXiv:2102.06481 https://arxiv.org/abs/2102.06481

[6] Furong Ye, Carola Doerr, and Thomas Bäck. 2021. *Data Sets for the study "Leveraging Benchmarking Data for Informed One-Shot Dynamic Algorithm Selection"*. https://doi.org/10.5281/zenodo.4501275

[7] Furong Ye, Hao Wang, Carola Doerr, and Thomas Bäck. 2020. Benchmarking a $(\mu + \lambda)$ Genetic Algorithm with Configurable Crossover Probability. In *Proc. of Parallel Problem Solving from Nature (PPSN'20) (LNCS, Vol. 12270)*. Springer, 699–713. https://doi.org/10.1007/978-3-030-58115-2_49