Prof. Dr. Holger Hoos
Chair for AI Methodology
RWTH Aachen University

Project Proposal:
Towards Dynamic Switching in Per-Run Algorithm Selection

Type:
Bachelor Thesis ☒
Master Thesis ☐

Daily Supervisor:
Anja Jankovic
jankovic@aim.rwth-aachen.de
Theaterstrasse 35-39, 52062 Aachen, Room XXX

Overview and Related Work

This thesis covers automatic algorithm selection for the optimization of continuous black-box functions, i.e., finding the optimum of a function $f : \mathbb{R}^D \to \mathbb{R}$ within a domain $X \subset \mathbb{R}^D$. In the black-box setting, we are limited to evaluating the function at specific points in the search space; the analytical expression of a function cannot be used. This setting lets us apply optimization to various real-world scenarios, such as finding a set of hyperparameters that minimize the loss of a machine learning model. Thus, a large set of optimization algorithms has been developed over the years. These include deterministic methods, such as line search algorithms like BFGS, and stochastic approaches, such as population-based methods like CMA-ES [1]. The complementary strengths and complexity of these algorithms motivate an automated approach to selecting the most suitable algorithm for a given function f.

To this end, Exploratory Landscape Analysis (ELA) features, first introduced in [2], have been developed. ELA features aim to quantify the landscape of a function. In classical per-instance algorithm selection, ELA features are typically computed by sampling $c \cdot d$ points from the search space, where d is the dimensionality of the problem and c is a constant, often set to 50 [3]. Sampling is commonly performed using methods like Latin Hypercube Sampling to ensure a representative coverage of the search space. Machine learning models, most notably random forests, then map these features to the optimization algorithm predicted to perform best on the given instance.

While this approach performs well and is an important development in creating a mapping between a given function and the best optimization algorithm for that function, it bears a significant drawback: the computational cost of the initial function evaluations, required for the computation of the ELA features. Since black-box functions often model expensive real-world processes, these additional evaluations can be a significant overhead.

Per-run algorithm selection, as introduced in [4], aims to reduce this initial overhead. In this approach, the CMA-ES algorithm is initially run for a fixed budget of $\mathcal{A}1 = 30 \cdot d$ on each given function. The resulting function evaluations of that initial run are then used to compute ELA features, which serve as the basis for predicting the most promising algorithm for the remaining optimization budget $\mathcal{A}2$. This prediction is performed using a random forest model trained to select the algorithm that is expected to achieve the highest precision for the remaining budget. The selected algorithm is warm-started using information from the initial run, for example by using the best-so-far solution as the population center for population-based algorithms [5].

This strategy has shown promising results, achieving competitive performance while each function evaluation directly contributes to the optimization.

This thesis builds on the work of [4]. Specifically, we focus on one specific question regarding the per-run algorithm selection framework: How much of the overall budget should be allocated to the initial run ($\mathcal{A}1$) and how much to the warm-started second algorithm ($\mathcal{A}2$)? In [4], the same static $\mathcal{A}1$ budget was chosen across all functions and runs. We argue that the $\mathcal{A}1$ budget should be chosen dynamically for each run. This is supported by both literature ([6], [4]) and our own experiments:

In a preliminary study, we recorded runs on various functions, using twenty different switching points per run. For each switching point, we recorded the performance of the six warm-started algorithms used in [4]. For each run, we identified the optimal switching point as the point for which one of the six warm-started algorithms achieved maximum precision across all switching points and algorithms for that run. This study revealed that the static budget chosen in [4] was only optimal for 30% of runs. Furthermore, no budget was optimal for more than 45% of runs, indicating that there is no single $\mathcal{A}1$ budget that is consistently best across all runs and functions. A more detailed study revealed that, even across runs on the same function, optimal switching points vary greatly, underlining the necessity to use a dynamic approach that detects optimal switching points for each run individually.

In an additional study, we evaluated the quality of ELA features computed from CMA-ES samples across various evaluation budgets. Since these features are used to select the most suitable $\mathcal{A}2$ algorithm, an optimal switching point should provide consistent and informative features. However, the study revealed significant fluctuations in many ELA features across budgets and functions, indicating that the features are neither stable nor predictive. Thus, there is no single budget at which ELA features are reliably informative.

In this thesis, we want to pave the way for dynamic switching in per-run algorithm selection. We will properly define optimal switching points and analyze where these switching points lie across different runs and functions. Such a definition should consider both algorithm performance and feature quality. Building on this, we will investigate whether these optimal switching points can be detected during a run. Specifically, we will examine whether ELA features, computed from the samples generated by the initial algorithm, are informative enough to indicate whether an optimal switching point has been reached. To test this, we will train random forest models at different possible switching points and evaluate their performances. In addition to ELA features, we will also analzye if run-specific data that captures the internal state of CMA-ES as used in [4] can improve the performance of the random forests. Furthermore, we will compare two modeling strategies: Training a separate random forest for each checkpoint, versus using a single model across all checkpoints, with the number of function evaluations provided as an input feature. While the latter could enable truly dynamic switching, it also has to handle a more complex prediction task.

If we have enough time remaining and the ELA-based approach does not lead to improvements, we may consider experimenting with a surrogate-based switching criterion inspired by the automatic termination method proposed in [7]. In this criterion, we would initiate a switch if the difference between the best-so-far objective value f_{opt} and the predicted minimum of a surrogate model, $\min_{x \in \mathcal{X}} \mu(x)$, trained on the evaluations of CMA-ES, remains nearly constant over several iterations. Here, $\mu(x)$ denotes the predicted mean of a Gaussian process surrogate model. This criterion is only optional, as the main focus is to analyze and understand the ELA-based switching strategy in depth.

Thus, this thesis will contribute to a deeper understanding of switching behavior in per-run algorithm selection by analyzing the optimality of switching points, assessing the effectiveness of ELA features for their detection, and exploring the feasibility of supervised learning approaches to inform switching decisions.

Timeline

June 2025 Define optimal switching points and analyze their positions across runs and functions. Start with the assessment of ELA-based switching.

July 2025 Continue the assessment; train random forests at various checkpoints. Evaluate models using both ELA and CMA-ES internal state data.

August 2025 Compare per-checkpoint vs. unified model approaches. Start writing the thesis. Optionally: Try surrogate-based switching.

September 2025 Finalize experiments, complete thesis.

References

[1] Mario A. Muñoz et al. "Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges". In: *Information Sciences* 317 (2015), pp. 224–245. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2015.05.010. URL: https://www.sciencedirect.com/science/article/pii/S0020025515003680.

[2] Olaf Mersmann et al. "Exploratory landscape analysis". In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. GECCO '11. Dublin, Ireland: Association for Computing Machinery, 2011, pp. 829–836. ISBN: 9781450305570. DOI: 10.1145/2001576.2001690. URL: https://doi.org/10.1145/2001576.2001690.

[3] Pascal Kerschke. *Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-Package flacco*. https://arxiv.org/abs/1708.05258. arXiv preprint arXiv:1708.05258. 2017.

[4] Ana Kostovska et al. "Per-run algorithm selection with warm-starting using trajectory-based features". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2022, pp. 46–60.

[5] Anja Jankovic et al. "Trajectory-based Algorithm Selection with Warm-starting". In: *2022 IEEE Congress on Evolutionary Computation (CEC)*. 2022, pp. 1–8. DOI: 10.1109/CEC55065.2022.9870222.

[6] Dominik Schröder et al. *Chaining of Numerical Black-box Algorithms: Warm-Starting and Switching Points*. Apr. 2022. DOI: 10.48550/arXiv.2204.06539.

[7] Anastasiia Makarova et al. "Automatic termination for hyperparameter optimization". In: (2022). URL: https://www.amazon.science/publications/automatic-termination-for-hyperparameter-optimization.