Towards Dynamic Switching in Per-Run Algorithm Selection

Thesis Proposal

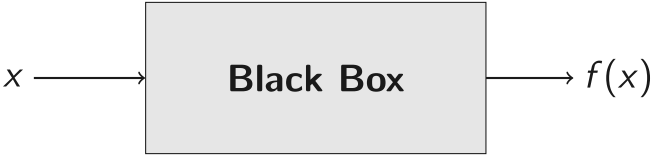
Gereon Geuchen

26^{th} May, 2025

RWTH AACHEN UNIVERSITY

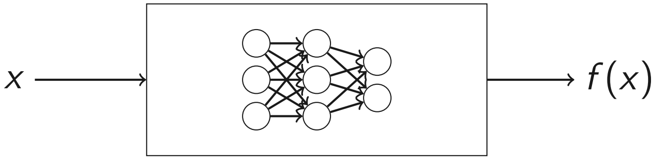
Black-Box Optimization

- Goal: Minimize a continuous function $f : \mathbb{R}^D \rightarrow \mathbb{R}$ within $X \subset \mathbb{R}^D$
- Black-Box: There is no closed-form expression, minimizing is limited to sampling

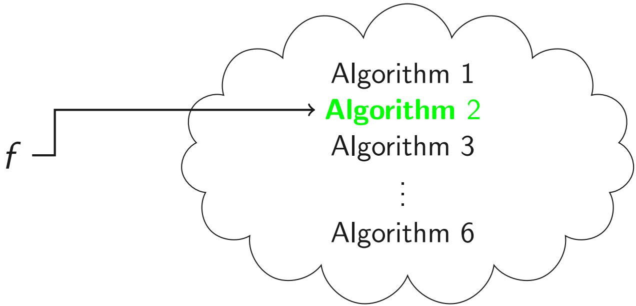


Black-Box Optimization

- Goal: Minimize a continuous function $f : \mathbb{R}^D \to \mathbb{R}$ within $X \subset \mathbb{R}^D$
- Black-Box: There is no closed-form expression, minimizing is limited to sampling

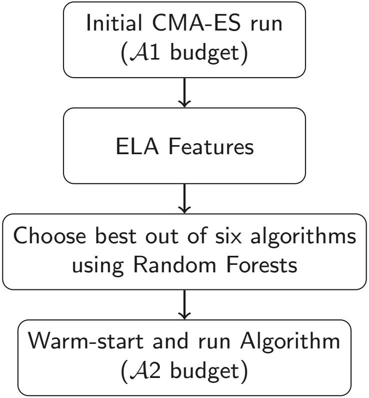


Algorithm Selection



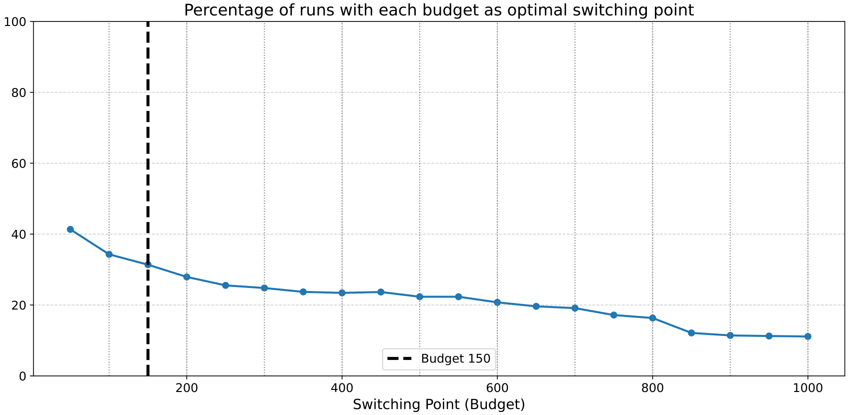
- Complex algorithm set, e.g. stochastic and deterministic algorithms
- Fixed-budget setting: Choose the algorithm that reaches the highest precision within a fixed number of function evaluations

Per-run algorithm selection (Kostovska et al. 2022)

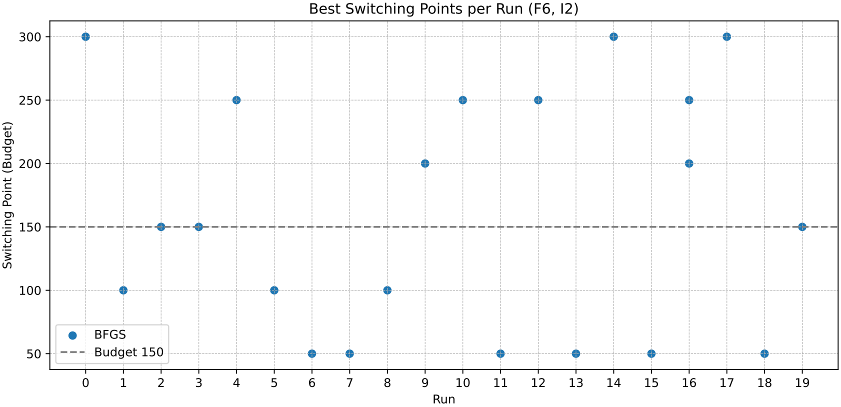


$\mathcal{A}1$ budget: Performance view

- Optimal switching point: Point at which one of the $\mathcal{A}2$ algorithms has reached maximum precision across all switching points for that run



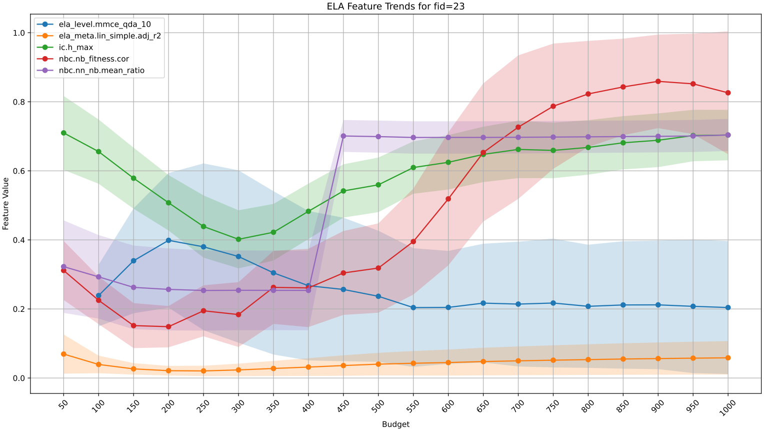
$\mathcal{A}1$ budget: Performance view



$\Rightarrow$ We need a dynamic approach that determines switching for each run individually

$\mathcal{A}1$ budget: Feature view

- Renau et al. 2020: ELA values are not absolute, but strongly depend on the sampling strategy



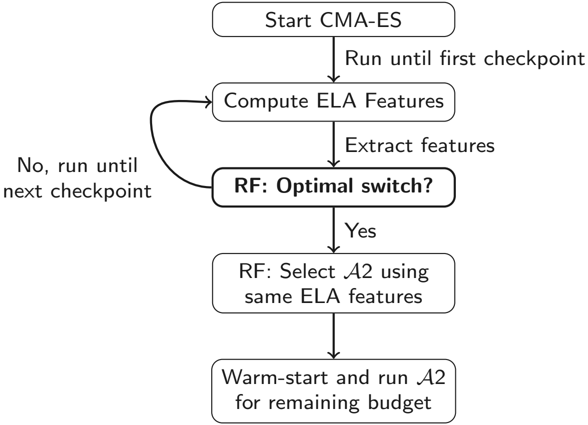
$\Rightarrow$ Unpredictable and inconsistent behavior over budgets

$\mathcal{A}1$ budget: Conclusion

- $\mathcal{A}1$ budget determines how much of the budget is allocated to the initial algorithm
- Should be chosen dynamically for each individual run
  - No clear best switching point based on performance
  - Inconsistent ELA features across budgets
- $\Rightarrow$ Goal of this thesis: Pave the way for dynamic switching
  - Motivate dynamic switching (done)
  - Build upon (Kostovska et al. 2022), analyze if we can use supervised learning to detect switching points

ELA-based switching

- Use ELA features to detect switching points



ELA-based switching: Research questions

- Does ELA-based dynamic switching lead to improvements over static switching?
- What is the best definition of optimality?
  - Should include both algorithm performance and feature quality
- Are ELA features able to detect switching points?
  - Are ELA features able to distinguish between runs or do they just pick up function-specific patterns?
- What are the best checkpoint placements?
- Can CMA-specific features lead to improvements? (CHECK WITH ANJA)
- Do we train one random forest for all checkpoints or do we train one random forest for each checkpoint?
- Optional: Try a model-based approach

References I

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

Renau, Quentin et al. (2020). "Exploratory Landscape Analysis is Strongly Sensitive to the Sampling Strategy". In: *Parallel Problem Solving from Nature – PPSN XVI*. Ed. by Thomas Bäck et al. Cham: Springer International Publishing, pp. 139–153. ISBN: 978-3-030-58115-2.

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

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

Surrogate-based switching

- Simple regret r_t after t iterations (Makarova et al. 2022):

$$r_t = f_{opt} - \min_{x \in \mathcal{X}} f(x)$$

- Train a surrogate (e.g. Gaussian process) to approximate f using the evaluations of CMA-ES:

$$\hat{r}_t := f_{opt} - \min_{x \in \mathcal{X}} \mu(x)$$

- Initiate switch if $\hat{r}_t$ remains nearly constant over a few iterations

Algorithm Set (Schröder et al. 2022)

- BFGS: Deterministic line-search, approximates the hessian
- DE: Random mutation of target solutions (population-based)
- MLSL: Random population, local search in clusters
- PSO: Swarm-based, velocity of particles gets adjusted based on own and global optimum
- CMA-ES elitism/non-elitism: Are all children the new population or all best candidates?