Partial Rankings of Optimizers

How to Benchmark Optimizers According to Multiple Criteria on Multiple Test Functions?

Julian Rodemann*, Hannah Blocher*

Department of Statistics, Ludwig-Maximilians-Universität München (LMU Munich), Germany

*Equal Contribution. Total order was enforced by fair coin flip.

Introduction

- Aim: Rank optimization algorithms on benchmark suites with regard to multiple criteria (giving rise to incomparibility) [4]
  - Multi-objective optimization [7, 3]
  - Single-objective optimizers evaluated w.r.t. several metrics [6]
- Problem: Impossible to aggregate a set of total orders to a single total order[1] [1]
- Common solution is to use "highly subjective" [5] consensus/weighting methods
- Our proposal: Consider *partial* orderings of optimizers on single test function
- Instead of aggregating, *describe* their distribution by depth functions → how central/outlying are the orderings?

Applications

Multi-Objective Evolutionary Algorithms

We compare DVC [7] against 6 competitors on 13 test functions w.r.t. mean inverted generational distance at four different phases (4 criteria). This results in 13 (unique) posets describing the relation between the optimizers' 4 performance criteria on each of the 13 test functions.

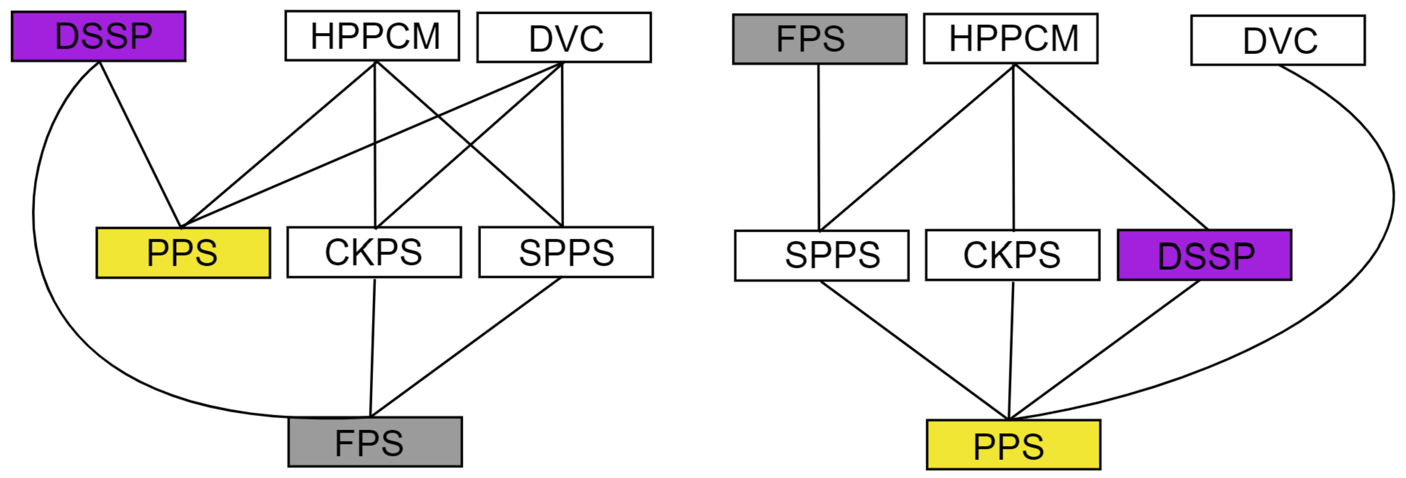


Figure: Orderings of optimizers corresponding to highest (0.39, left) and lowest (0.17, right) ufg depth. Edge between optimizers: The optimizer on top is not outperformed by the one below w.r.t. any of the 4 criteria.

`DeepOBS` [6]: Deep Learning Optimizers

We mimic the setup in [6, section 4] and compare vanilla stochastic gradient descent (SGD), adam, and momentum on 8 test functions with respect to performance (minimal test loss achieved in a fixed time budget) and speed (time required to achieve a given test loss). .

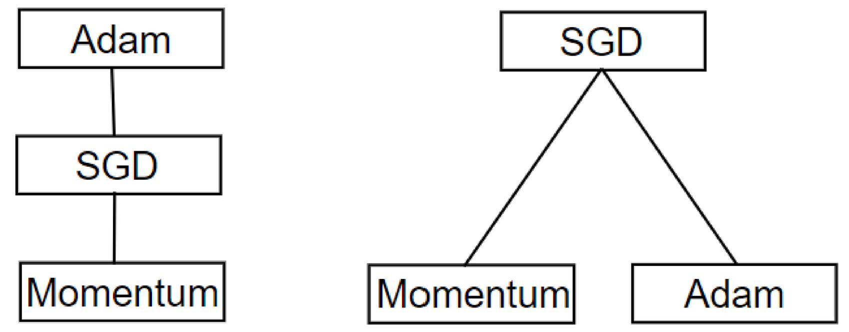


Figure: Orderings of optimizers corresponding to highest (0.65, left, duplicated) and lowest (0.29, right) ufg depth are shown below. The poset on the right can be seen as outlying. This means that the underlying problem (LSTM on *War and Peace*) produces an order structure that is atypical

Outlook

Design and Curation of Benchmarking Suites:

- Looks at the entire benchmarking suite (test function, evaluation criteria and the optimizers) at once
- Sensible tool to asses the diversity of the test function and produced partial orders

Limitations - Inferential Statements:

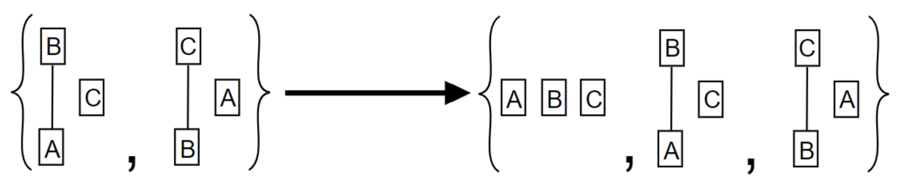
- "Which test functions produce more likely a typical partial ordering of the old and new optimizers combined, and which produce an atypical order?"
  → with no prior knowledge, this question cannot be answered (by any ranking) ↯
- "On another test function, what is the most likely poset structure to be observed?" → further research needed

Method: Adapt the Simplicial Depth to Partially Ordered Sets (Posets) [2]

UFG Depth

Define the Closure Operatur/System

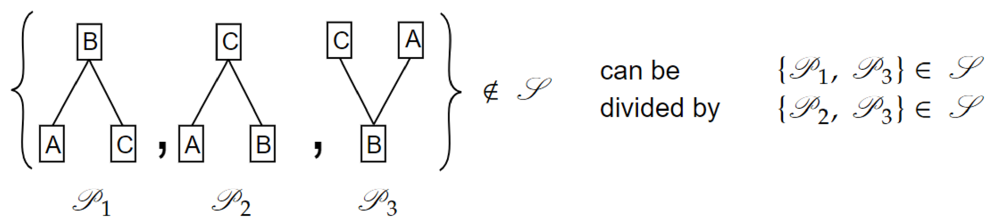
$$\gamma\colon\ 2^{\mathcal{P}} \to 2^{\mathcal{P}};\ P \mapsto \left\{ p \in \mathcal{P} \mid \bigcap_{\tilde{p}\in P} \tilde{p} \subseteq p \subseteq \bigcup_{\tilde{p}\in P} \tilde{p} \right\}.$$



Reduce the Input Set (It is still sufficient to describe the closure operator)

$$\mathscr{S} = \{ P \subseteq \mathcal{P} \mid \text{Condition } (C1) \text{ and } (C2) \text{ hold } \}$$

with
$$(C1):\ P \subsetneq \gamma(P),$$
$$(C2):\ \nexists\, (A_i)_{i\in\{1,\dots,\ell\}}\ \forall i \in \{1,\dots,\ell\}:\ A_i \subsetneq P \text{ and } \cup_{i\in\{1,\dots,\ell\}} \gamma(A_i) = \gamma(P).$$



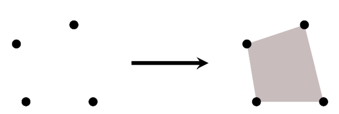
Define the Depth Function

$$D\colon\ \begin{array}{l} \mathcal{P} \times \mathcal{M} \to [0,1] \\ (p,\nu) \mapsto \begin{cases} 0, & \text{if for all } S \in \mathscr{S}:\ \prod_{\tilde{p}\in S} \nu(\tilde{p}) = 0 \\ c \sum_{S\in\mathscr{S}\,:\, p\in\gamma(S)} \prod_{\tilde{p}\in S} \nu(\tilde{p}), & \text{else} \end{cases} \end{array}$$

Simplicial Depth

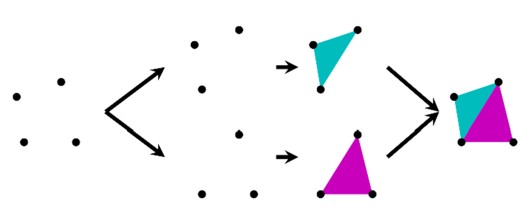
Define the Closure Operatur/System

$$\gamma_{\mathbb{R}^d}\colon\ 2^{\mathbb{R}^d} \to 2^{\mathbb{R}^d};\ A \mapsto \left\{ x \in \mathbb{R}^d \,\middle|\, \begin{array}{l} x = \sum_{i=1}^{k} \lambda_i a_i \text{ with } a_i \in A, \\ \lambda_i \in [0,1], \sum_{i=1}^{k} \lambda_i = 1, k \in \mathbb{N} \end{array} \right\}$$



Reduce the Input Set

$$\mathscr{S} = \{\{x_1,\dots,x_{d+1}\} \subseteq \mathbb{R}\}$$



Define the Depth Function

$$D\colon\ \begin{array}{l} \mathbb{R}^d \times \mathcal{M} \to [0,1], \\ (x,\nu) \mapsto \nu(x \in \gamma_{\mathbb{R}^d}\{X_1,\dots,X_{d+1}\}), \end{array}$$

with $X_1, \dots X_{d+1} \sim \nu$.

References

[1] Kenneth J Arrow. *Social choice and individual values.* Vol. 12. Yale university press, 2012.
[2] Hannah Blocher et al. "Depth functions for partial orders with a descriptive analysis of machine learning algorithms". In: *ISIPTA*. Vol. 215. PMLR, 2023, pp. 59–71.
[3] Alex Gu et al. "Min-max multi-objective bilevel optimization with applications in robust machine learning". In: *International Conference on Learning Representations (ICLR)*. 2023.
[4] Nikolaus Hansen et al. "Anytime performance assessment in blackbox optimization benchmarking". In: *IEEE Transactions on Evolutionary Computation* 26.6 (2022), pp. 1293–1305.
[5] Olaf Mersmann et al. "Analyzing the BBOB results by means of benchmarking concepts". In: *Evolutionary Computation* 23.1 (2015), pp. 161–185.
[6] Frank Schneider, Lukas Balles, and Philipp Hennig. "DeepOBS: A Deep Learning Optimizer Benchmark Suite". In: *International Conference on Learning Representations (ICLR)*. 2019.
[7] Fei Wu et al. "A dynamic multi-objective optimization method based on classification strategies". In: *Scientific Reports* 13.1 (2023), p. 15221.

Tiny Paper:

ICLR