Coupling the design of benchmark with algorithm in landscape-aware solver design

Johann Dréo, Carola Doerr, Yann Semet

Coupling the Design of Benchmark with Algorithm in Landscape-Aware Solver Design

Johann Dreo
Thales Research and Technology
johann.dreo@thalesgroup.com

Carola Doerr
Sorbonne Université, CNRS
Carola.Doerr@lip6.fr

Yann Semet
Thales Research and Technology
yann.semet@thalesgroup.com

ABSTRACT

Following a general trend in artificial intelligence, Evolutionary Computation has, in recent years, witnessed substantial performance gains from landscape-aware selection and parameter tuning of algorithmic modules. Such approaches, however, are critically relying on suitable benchmarks, or *training sets*, that provide the appropriate blend of performance and generality. With this position paper we argue that, on a landscape analysis basis, the benchmark design problem will form a substantial part of the next-generation of automated, on-demand algorithm design principles.

CCS CONCEPTS

• **Computing methodologies → Heuristic function construction**; **Randomized search**; • **Software and its engineering →** *Search-based software engineering*;

KEYWORDS

landscape-aware heuristics, algorithm engineering, algorithm design, black-box optimization, benchmarking, benchmark design

1 INTRODUCTION

The automation of the design of programs has been a classical concept in the evolutionary algorithm community from the start [4, 7] and was quickly applied to the design of algorithms themselves [5].

It was also stated very early that the performance of evolutionary algorithms is very parameter-sensitive [6] and that the tuning problem is in itself a (meta-)optimization problem [8]. Since then, problems related to the automated design of evolutionary algorithms have been commonly studied [1, 8]

In the broad sense, the automated *design* of an optimization solver may encompass any choice that has an impact on its behavior, namely the selection of the algorithm(s) [11, 14], the choice of its module(s) [15], or its parameter setting(s) [1]. Significant progress has been achieved on that side of the problem with automated parameter tuning methods [10] or hyper-heuristics [3].

More recently, the use of *fitness landscapes features* [12] as a mean of observation of the problems, from the algorithm perspective, prove to be particularly useful for design problems [2, 11].

2 LANDSCAPE-AWARE SOLVER: ALGORITHM + BENCHMARK

In order to design a good, specific, solver, two problems are crucial: algorithm design and benchmark design. The problem of designing the best algorithm is arguably the most studied in the literature, but we argue thatit should be coupled with the problem of designing the benchmark (cf. Figure 1) in order to provide adequate feature-wise representation of the intended use.

Modern search heuristics performances are always assessed first on a benchmark, most of the time a "synthetic" one, designed for fast execution and a sufficiently large "footprint" [9, 13], or representativity of potential instance types. As such, their design is very often influenced by the benchmark's design. This is even more true for heuristics that embed a landscape-aware process, where the given benchmark *fully determines* the general performance of the algorithm. Following [11, 13], we thus argue that having benchmarks with large "footprint" is a crucial challenge.

Going further, we argue that landscapes may be used for *automated benchmark design*. In that setting, the problem becomes to design benchmarks that "cover" the landscape space of a given optimization problem well enough, with as few instances as possible. That is to find a fixed set of problem instances from which reasonable performance prediction models can be built.

In the ideal setting, the benchmark generator is a configurable mean to produce problem instances with respect to specified landscape features based performance criteria. Hidden in that setting is the problem of selecting the subset of features that are usefulon the benchmark instances themselves.

Figure 1 shows the proposed coupling between the *algorithm design* and *benchmark design* optimization problems. The process is heavily using landscape-aware tools in order to produce an adaptive solver. The benchmark design optimization loop (top-right corner) adapt the instances set to a targeted *footprint* performance and output the related selected features as a byproduct.

3 CONCLUSION: COUPLED DESIGN NEEDED

We think that very significant gain can be expected from automating solver design for a specific problem and set of performance criteria by automatically generate the benchmark used by landscape-aware processes. The essential goal is the appropriate generalization level: maximum performance on foreseen use without counterproductive generalization. The long-term goal is an optimized solver generator in which an end-user would input her problem formalization, her desired benchmark footprint and performance criteria. Managing the interplay between features, benchmarks and algorithms is of course one of the biggest challenge in our suggested approach.

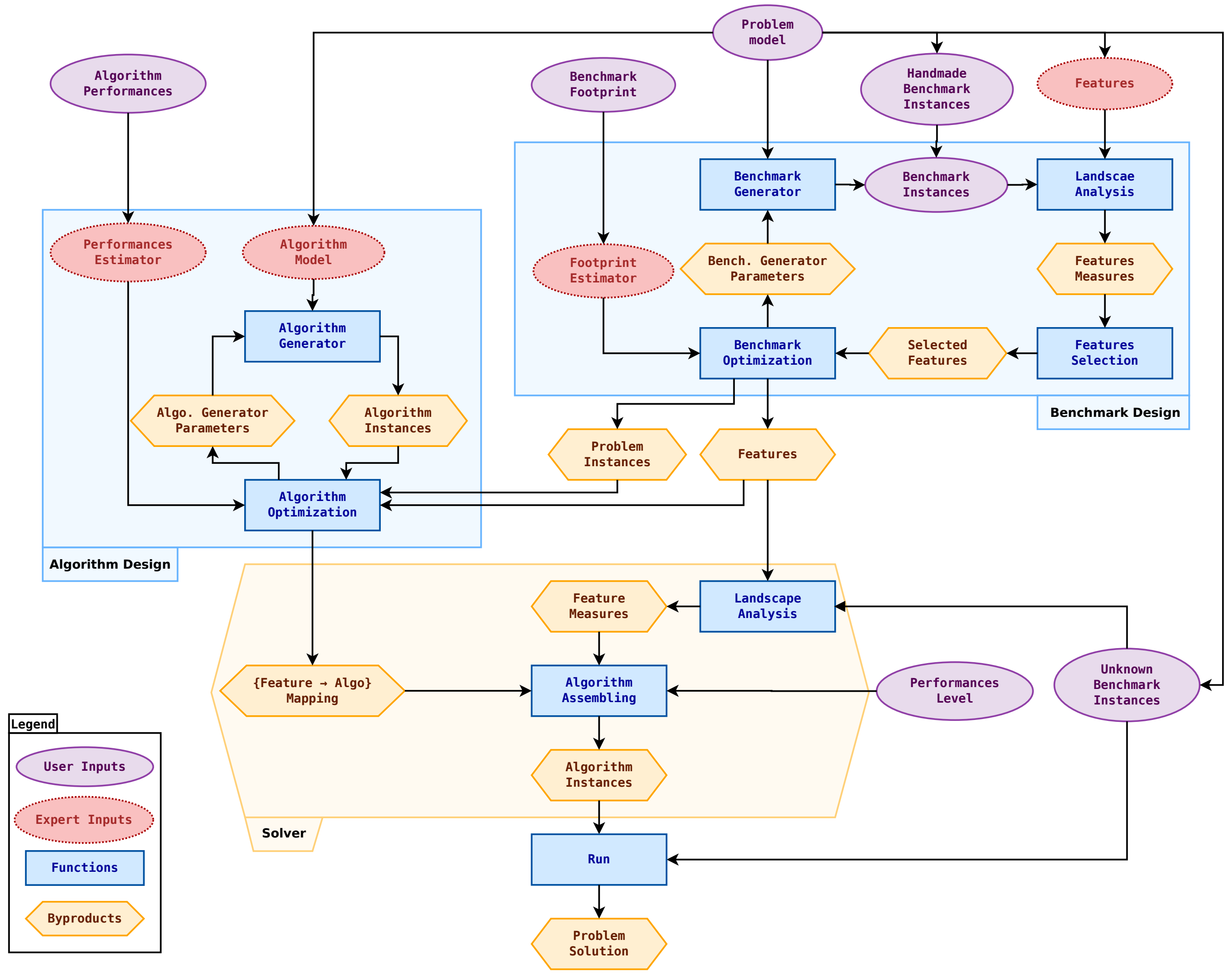

Figure 1: Diagram of the landscape-aware solver design scheme. The coupling between the algorithm design (top-left loop) and the benchmark design (top-right loop) occurs through the optimized set of problem instances and selected features (center). The output is a landscape-aware solver (bottom) that is able to adapt its behaviour to foreseen problem features.

REFERENCES

[1] Thomas Bartz-Beielstein, Marco Chiarandini, Luís Paquete, and Mike Preuss. 2010. *Experimental methods for the analysis of optimization algorithms.* Springer.

[2] Nacim Belkhir, Johann Dreo, Pierre Savéant, and Marc Schoenauer. 2016. Feature Based Algorithm Configuration: a Case Study with Differential Evolution. In *14th International Conference on Parallel Problem Solving from Nature.* to appear.

[3] Edmund K Burke, Michel Gendreau, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan, and Rong Qu. 2013. Hyper-heuristics: a survey of the state of the art. *Journal of the Operational Research Society* (2013).

[4] Nichael Lynn Cramer. 1985. A representation for the Adaptive Generation of Simple Sequential Programs. In *Proceedings of an International Conference on Genetic Algorithms and the Applications*, John J. Grefenstette (Ed.). Carnegie-Mellon University, Pittsburgh, PA, USA, 183–187.

[5] Dirk Dickmanns, Jürgen Schmidhuber, and Andreas Winklhofer. 1987. Der genetische algorithmus: Eine implementierung in prolog. *Fortgeschrittenenpraktikum, Institut fur Informatik, Universitat Munchen* (1987).

[6] Laurence Charles Ward Dixon. 1972. The choice of step length, a crucial factor in the performance of variable metric algorithms. *Numerical methods for non-linear optimization* (1972), 149–170.

[7] L.J. Fogel, A.J. Owens, and M.J. Walsh. 1966. *Artificial intelligence through simulated evolution.* Wiley, Chichester, WS, UK.

[8] John J Grefenstette. 1986. Optimization of control parameters for genetic algorithms. *IEEE Transactions on systems, man, and cybernetics* 16, 1 (1986), 122–128.

[9] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. 2010. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Proc. of the 12th GECCO.* ACM, 1689–1696.

[10] Frank Hutter, Holger H. Hoos, Kevin Leyton-Brown, and Thomas Stützle. 2009. ParamILS: An Automatic Algorithm Configuration Framework. *J. Artif. Int. Res.* 36, 1 (Sept. 2009), 267–306.

[11] Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. 2019. Automated Algorithm Selection: Survey and Perspectives. *Evol. Comp.* 27, 1 (2019), 3–45.

[12] Peter Merz and Bernd Freisleben. 1999. *New ideas in optimization.* McGraw-Hill Ltd., Chapter Fitness landscapes and memetic algorithm design, 245–260.

[13] Mario A. Muñoz and Kate A. Smith-Miles. 2017. Performance Analysis of Continuous Black-Box Optimization Algorithms via Footprints in Instance Space. *Evolutionary Computation* 25, 4 (2017), 529–554.

[14] Mario A Muñoz, Yuan Sun, Michael Kirley, and Saman K Halgamuge. 2015. Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Information Sciences* 317 (2015), 224–245.

[15] Sander Van Rijn, Carola Doerr, and Thomas Bäck. 2018. Towards an Adaptive CMA-ES Configurator. In *International Conference on Parallel Problem Solving from Nature.* Springer, 54–65.