Adaptive Landscape Analysis

Anja Janković
Sorbonne Université, Paris, France

Carola Doerr
Sorbonne Université, CNRS, Paris, France

ABSTRACT

Black-box optimization of a previously unknown problem can often prove to be a demanding task. In order for the optimization process to be as efficient as possible, one must first recognize the nature of the problem at hand and then proceed to choose the algorithm exhibiting the best performance for that type of problem. The problem characterization is done via underlying *fitness landscape features*, which allow to identify similarities and differences between various problems.

In this paper we present first steps towards an *adaptive landscape analysis*. Our approach is aimed at taking a closer look into how features evolve during the optimization process and whether this information can be used to discriminate between different problems. The motivation of our work is to understand if and how one could exploit the information provided by the features to improve on dynamic algorithm selection and configuration. Put differently, our goal is to leverage landscape analysis to adjust the choice of the algorithm on the fly, i.e., during the optimization process itself.

CCS CONCEPTS

• **Theory of computation → Theory of randomized search heuristics**;

ACM Reference format:
Anja Janković and Carola Doerr. 2019. Adaptive Landscape Analysis. In *Proceedings of Genetic and Evolutionary Computation Conference Companion, Prague, Czech Republic, July 13–17, 2019 (GECCO '19 Companion),* 4 pages.
https://doi.org/10.1145/3319619.3326905

1 INTRODUCTION

Optimization problems are encountered in many different real-world scenarios. Very often one is faced with finding the optimal solution for a problem at hand that is too complex to be analytically modeled. In those situations, the problem can be seen as a black box, as the exact relationship between problem inputs and its output is unknown. Algorithms solving black-box problems are commonly referred to as *black-box optimization algorithms* or *randomized search heuristics*. Black-box algorithms use only different pairs of the input-output values (where the output is the corresponding value of the function to be optimized for the given input) to guide the search towards a good estimate of the optimal solution.

Randomized search heuristics all come with a set of parameters that govern their behavior and performance during the run. Making existing heuristics more efficient and designing new and improved ones have long been at the core of various research axes. It is widely acknowledged today that designing a general optimization algorithm, which would be able to best solve all different kinds of problems, is futile: this is proved by the so-called *no free lunch theorem* [18]. As such, tackling a new and unknown optimization problem first amounts to the choice of the most appropriate algorithm (the *algorithm selection* problem, *AS*) and/or the choice of the best parameter settings for the chosen algorithm (the *algorithm configuration* problem, *AC*). While research on AS and AC has for a long time been studied independently, it has in recent years become common practice to regard the AS and AC problems as one.

Although extensively covered in literature, it seems that most of the previous research has focused on the *offline perspective* of algorithm selection and configuration, which consists in running different optimization algorithms entirely on different problem instances, and then matching measured performance to the given algorithm. Our ambition is to extend AS/AC techniques to the *online setting*, in which different algorithms and/or configurations can be used in different parts of the optimization process.

1.1 Exploratory Landscape Analysis

In order to make a clever choice of an algorithm and/or to configure it well for the given problem, it is important to have some knowledge of the problem beforehand. This can be done via some kind of a problem characterization that would allow us to group the problems according to their similarities. With this target in mind, Mersmann et al. [13] have introduced so-called *high-level properties*, that describe, for a certain function to be optimized, its *fitness landscape*: the degree of multimodality (i.e., if the function at hand has more than one local optimum), the underlying global structure, if the problem is separable or not, how its variables scale, the homogeneity of the search space and basin sizes, the contrast of global to local optima and whether the function landscape has plateaus.

However, these properties have a drawback of requiring expert knowledge in order to use them to classify optimization problems. This issue was also addressed by Mersmann et al. [12] by introducing the term *exploratory landscape analysis (ELA)* and with that come *low-level features* of continuous functions, which can be computed automatically based on a sample of observations (i.e., $(x, f(x))$ pairs) from the given problem instance. Since then, many new feature sets have been introduced for different computational needs and goals, and they fall into either of the 2 following classes: cheap or expensive, depending on their computational cost. *Cheap features* are computed using the fixed initial sample, while *expensive features* need additional sampling during the run, an overhead that makes them more inaccessible for practical use. For this reason, in this paper we only focus on cheap features.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326905

1.2 Online Algorithm Configuration

In recent years, it has been suggested in literature that non-static choices of algorithms [2, 4] and adaptive parameter configurations [3, 8] could be beneficial for the improvement of the overall performance of the optimization process. Extending that mindset to landscape-aware algorithm selection and configuration, we aim to investigate how problem features change locally depending on the quality of the already reached solution(s), in hope to understand if there is an underlying connection between how the fitness landscape looks locally and the performance of a certain optimization algorithm.

This paper presents some first steps in the study of such an *adaptive landscape analysis*. Here we take a closer look into the local feature structure, and focus in particular on gauging what features values tell us about the nature of the problem, and subsequently what information we can extract about how to adapt the algorithm choice during the optimization process, following the dynamic change in the fitness landscape. As part of an ongoing research, we plan to extend the so-called Per-Instance Algorithm Configuration (PIAC) approach in continuous optimization (successfully applied to the configuration of the well-known CMA-ES algorithm in [1]) to online algorithm selection and configuration relying on landscape feature values. Apart from parameter control, we can also target with the same approach an automated *algorithm design*. For instance, the modular CMA-ES presented in [16] could be an interesting test case. A preliminary analysis of its potential has recently been conducted [15, 17], but has not yet been combined with a feature-based selection.

2 EXPERIMENTAL SETTING

As in [1], we consider as heuristic of choice for our experiments the popular Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm [7]. The CMA-ES regularly updates its covariance matrix during the optimization process and thus adjusts the probability distribution from which the solution candidates are sampled. For our analysis, we consider the BBOB suite of functions on the COCO platform [6]. From the 24 functions provided in the BBOB noiseless testbed, we selected 3 (namely F1, F2 and F6), all in 5D, and focused on the first 5 instances for each of them.

The experiments are designed in a following way: while the optimizer (in our case, the CMA-ES) is running, we track the precision of the sampled points. Whenever a new target value 10^i, $i = \{-2, ..., 9\}$ is reached, 2000 additional points are sampled from the current distribution and stored along with their function evaluations (fitness values). These additional samples do not influence the behavior of the algorithm and are used only to compute the feature values of the fitness landscape currently seen by the CMA-ES. We run each optimization process 5 times, each time using the standard CMA-ES variant (no restarts, fixed default population size of 6 for 5D).

Landscape feature values are then computed at each target precision level and for each function and instance. To this end, we only considered features that do not require additional function evaluations for feature computation (i.e., the cheap features). We therefore considered 6 feature sets: 3 classical ELA ones (y-Distribution, Levelset and Meta-model), as well as Dispersion [11], Information Content [14] and Nearest-Better Clustering [9] feature sets, containing 68 features in total. Among them, 2 features by set (runtime and number of function evaluations) were excluded from further analysis as they do not provide insight into landscape characteristics, which leaves us with 56 features. For a comparison, we have also computed the features for the respective BBOB functions, using the same number of 2000 samples for each of the first five instances in 5D. These points are normally distributed in the range $[-5, 5]$ (which is equivalent to the domain of definition of BBOB functions). These latter are considered as the *global feature values* for the specific function/instance. For the purpose of this paper, we consider average feature values computed over 5 independent runs, and from hereon we focus exclusively on the first instance of each function.

The experimental code was built upon the original CMA-ES implementation provided in [5]. The feature computation was done in R using the *flacco* library [10].

3 RESULTS

In this section we present some of our preliminary findings for the adaptive landscape analysis, obtained through the analysis of 3 selected BBOB functions. We include one figure per function, each presenting results for different feature sets. All three figures show how the feature values evolve during the optimization process, with different features on x-axis, feature values on y-axis and target values as different plots within a figure. The scale has been normalized to the range $[0, 1]$, for a better visualization of the results. Lastly, in order to visualize the data more clearly and make the charts more accessible for understanding, we have purposefully omitted some target values (namely 10^{-2}, 10^{-4}, 10^{-6} and 10^{-8}) from all the figures.

Figure 1 shows Dispersion and Information Content feature sets for the function F2 (ellipsoid function). We observe monotonicity in the relationship between feature values and target values for certain features, which is most prominent in the case of features *IC:eps.ratio* and *IC:eps.s* from the Information Content feature set. IC feature set is closely related to the measure of ruggedness of the fitness landscape, and the monotonicity could be explained by the fact that F2 is a locally smooth function. The Dispersion feature set exhibits an overall similar monotonic behavior, although not as consistent at every feature; this set translates the notion of hardness of the problem and quantifies the proximity of more interesting regions of the search space. We remark that these 2 feature sets behave very similarly in the other 2 functions, albeit not shown here. This is in line with the previous comment about the monotonicity of IC features, as F1 and F6 are both smooth functions as well. It is worth noting that the curve of the global feature values is excluded from Figure 1 for reasons of scale. However, the general trend seems to be that global feature values usually differ significantly from local feature values, indicating that the fitness landscape as seen by the algorithm differs from that of uniform sampling.

On the other hand, both Figure 2 and Figure 3 display plots of their respective global feature values along with locally observed values, which for the most part show stark contrast between the two, as previously mentioned. Moreover, both of the figures demonstrate rather chaotic and inconsistent local behavior for most features. In Figure 2 we observe Meta-model and Nearest-Better Clustering feature sets for the function F1 (sphere function). Meta-model

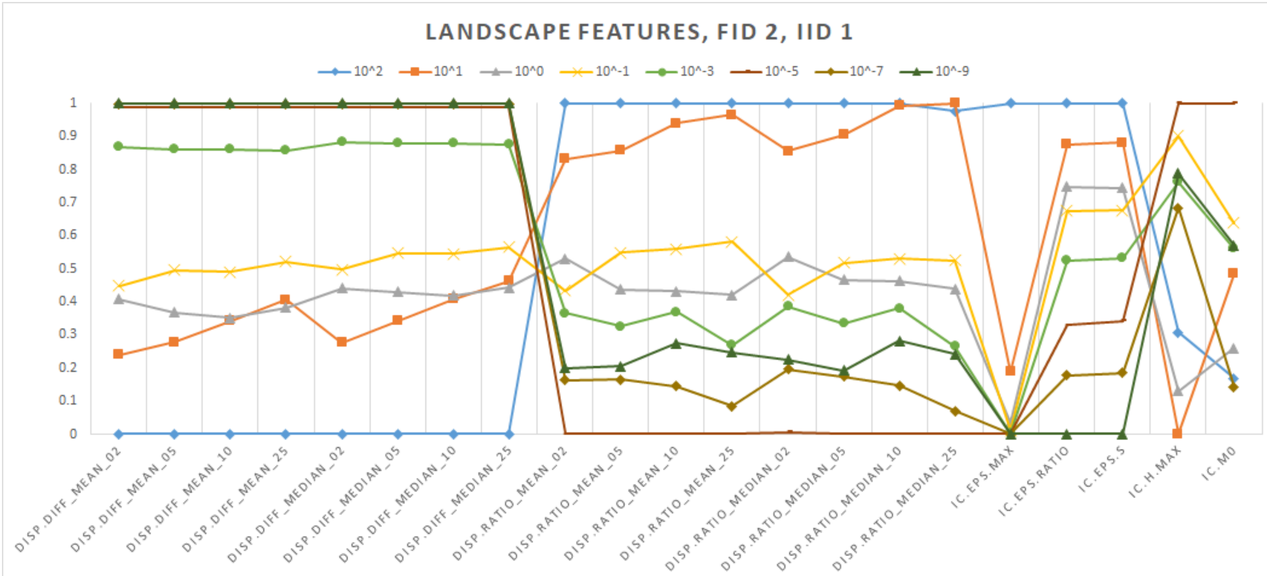

Figure 1: F2, IID1, normalized feature values for the Dispersion and Information Content feature sets.

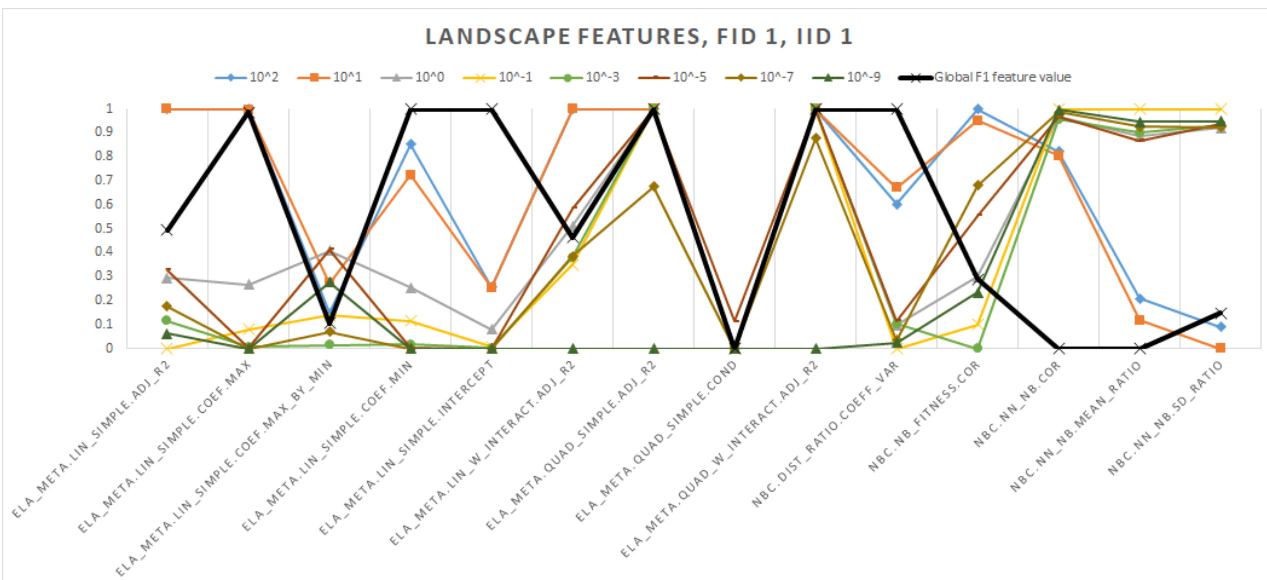

Figure 2: F1, IID1, normalized feature values for the Meta-model and Nearest-Better Clustering feature sets.

feature set aims to measure the ability to approximate the objective function with a linear, quadratic or regression model, while Nearest-Better Clustering feature set deals with recognizing single peaks within a multimodal landscape. Lastly, Figure 3 shows y-Distribution and Levelset feature sets for the function F6 (attractive sector function): the former contains features that measure ruggedness, symmetry and multimodality of the problem at hand, while the latter is especially useful when dealing with multimodal functions.

At this stage, the available data does not yet allow for an more complete intuitive interpretation of the dynamic change of feature values. Some of the important questions that guide our ongoing research activities are why these values evolve in such a fashion, whether these features capture the important knowledge about the problem instance and if yes, how to exploit that knowledge for the purpose of recognizing different problems.

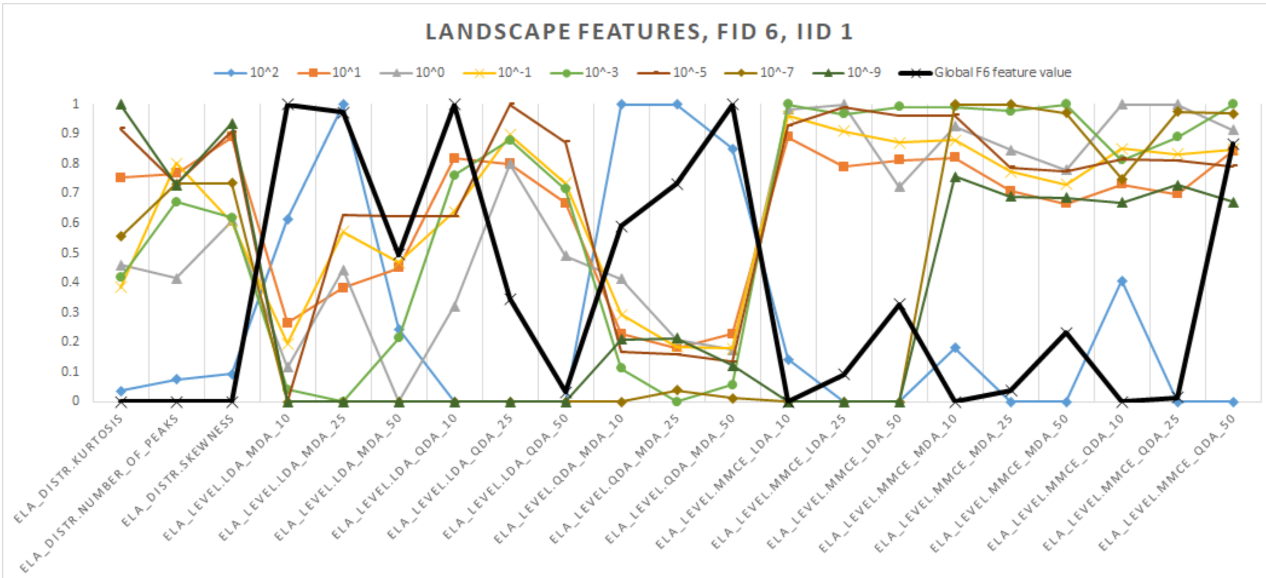

Figure 3: F6, IID1, normalized feature values for the y-Distribution and Levelset feature sets.

4 CONCLUSION AND FUTURE WORK

Motivated by the quest to design landscape-aware online algorithm selection and configuration techniques, we have analyzed in this work to what extent the fitness landscape, as seen by iterative black-box optimization heuristics, changes during the optimization process. To this end, we have computed 56 feature values of the fitness landscape, all belonging to the class of cheap features, as locally seen by CMA-ES at different target values $10^{-i}, i = -2, \dots, 9$. Our preliminary analysis focuses on 3 selected benchmark problems of the BBOB testbed. In an ongoing work we are extending our approach to the full set of 24 noiseless BBOB functions.

Our next step towards an online algorithm selection will be coupling feature information to performance of continuous black-box optimizers. Apart from studying the AS problem on standard solvers such as CMA-ES, Differential Evolution, EDAs etc., we also plan to build an online configurator for the modular CMA-ES proposed in [16]. A first indication that a dynamic configurator of this meta-model is likely to give additional performance gains has been demonstrated in [17].

ACKNOWLEDGMENTS

We are grateful to Pascal Kerschke, Johann Dréo, and Nacim Belkhir, for several inspiring discussions around landscape-aware algorithm configuration.

This work is supported by the Paris Ile-de-France Region.

REFERENCES

[1] Nacim Belkhir, Johann Dréo, Pierre Savéant, and Marc Schoenauer. 2017. Per instance algorithm configuration of CMA-ES with limited budget. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'17)*. ACM, 681–688.

[2] Edmund K. Burke, Michel Gendreau, Matthew R. Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan, and Rong Qu. 2013. Hyper-heuristics: a survey of the state of the art. *JORS* 64, 12 (2013), 1695–1724.

[3] Benjamin Doerr and Carola Doerr. 2018. Theory of Parameter Control for Discrete Black-Box Optimization: Provable Performance Gains Through Dynamic Parameter Choices. *CoRR* abs/1804.05650 (2018). arXiv:1804.05650

[4] Michael G. Epitropakis and Edmund K. Burke. 2018. Hyper-heuristics. In *Handbook of Heuristics*. Springer, 489–545.

[5] Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. 2019. CMA-ES/pycma on Github. Zenodo, doi:10.5281/zenodo.2559634. (Feb. 2019).

[6] Nikolaus Hansen, Anne Auger, Olaf Mersmann, Tea Tusar, and Dimo Brockhoff. 2016. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. *CoRR* abs/1603.08785 (2016). arXiv:1603.08785

[7] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9, 2 (2001), 159–195.

[8] Giorgos Karafotias, Mark Hoogendoorn, and Ágoston E. Eiben. 2015. Parameter Control in Evolutionary Algorithms: Trends and Challenges. *IEEE Trans. Evolutionary Computation* 19, 2 (2015), 167–187.

[9] Pascal Kerschke, Mike Preuss, Simon Wessing, and Heike Trautmann. 2015. Detecting Funnel Structures by Means of Exploratory Landscape Analysis. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'15)*. ACM, New York, NY, USA, 265–272.

[10] Pascal Kerschke and Heike Trautmann. 2016. The R-Package FLACCO for exploratory landscape analysis with applications to multi-objective optimization problems. In *Proc. of IEEE Congress on Evolutionary Computation (CEC'16)*. IEEE, 5262–5269.

[11] Monte Lunacek and Darrell Whitley. 2006. The Dispersion Metric and the CMA Evolution Strategy. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'06)*. ACM, 477–484.

[12] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. 2011. Exploratory landscape analysis. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'11)*. ACM, 829–836.

[13] Olaf Mersmann, Mike Preuss, and Heike Trautmann. 2010. Benchmarking Evolutionary Algorithms: Towards Exploratory Landscape Analysis. In *Proc. of Parallel Problem Solving from Nature (PPSN'10)*. Springer, 73–82.

[14] Mario A. Muñoz, Michael Kirley, and Saman K. Halgamuge. 2015. Exploratory Landscape Analysis of Continuous Space Optimization Problems Using Information Content. *IEEE Trans. Evolutionary Computation* 19, 1 (2015), 74–87.

[15] Sander van Rijn, Carola Doerr, and Thomas Bäck. 2018. Towards an Adaptive CMA-ES Configurator. In *Proc. of Parallel Problem Solving from Nature (PPSN'18)*. Springer, 54–65.

[16] Sander van Rijn, Hao Wang, Matthijs van Leeuwen, and Thomas Bäck. 2016. Evolving the structure of Evolution Strategies. In *Proc. of IEEE Symposium Series on Computational Intelligence (SSCI'16)*. IEEE, 1–8.

[17] Diederick Vermetten, Sander van Rijn, Thomas Bäck, and Carola Doerr. 2019. Online Selection of CMA-ES Variants. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'19)*. ACM. To appear.

[18] David H. Wolpert and William G. Macready. 1997. No free lunch theorems for optimization. *IEEE Trans. Evolutionary Computation* 1, 1 (1997), 67–82.