Expressiveness and robustness of landscape features (student workshop paper)

Quentin Renau, Johann Dréo, Carola Doerr, Benjamin Doerr

Expressiveness and Robustness of Landscape Features

Quentin RENAU
THALES Research & Technology
Palaiseau, France

Johann DREO
THALES Research & Technology
Palaiseau, France

Carola DOERR
Sorbonne Université, CNRS
Paris, France

Benjamin DOERR
Ecole Polytechnique
Palaiseau, France

ABSTRACT

Insights on characteristics of an optimization problem is highly important in order to select and configure the right algorithm. Some techniques called *features* are defined for analyzing the fitness landscape of a problem. Despite their successes, our understanding of which features are actually relevant for the discrimination between different optimization problems is rather weak, since in most applications the features are used in a black-box manner. Another aspect that has been ignored in the exploratory landscape analysis literature is the robustness of the feature computation against the randomness of sample points from which the feature values are estimated. Moreover, the influence of the number of sample points from which the feature values are estimated is also an aspect ignored by the literature. In this paper, we study these three aspects: the robustness against the random sampling, the influence of the number of sample points, and the expressiveness in terms of ability to discriminate problems. We perform such an analysis for 7 out of the 17 features sets covered by the flacco package. Our test bed are the 24 noiseless BBOB functions. We show that some of these features seems very well-fitted for the discrimination of the problems and quite robust whereas others lack robustness and/or expressiveness, and are therefore less suitable for an automated landscape-aware algorithm selection/configuration approach.

CCS CONCEPTS

• **Computing methodologies** → *Heuristic function construction*; *Continuous space search*; *Randomized search*;

KEYWORDS

Exploratory Landscape Analysis, Black-Box Optimization, Automated Algorithm Configuration

1 INTRODUCTION

When facing a new optimization problem to solve, insights on its characteristics can be leveraged to select a well-performing algorithm and a suitable instantiation of its parameters [4, 8]. Reliable indicators that give insight on the structure of the *fitness landscape* are therefore at high demand. The design and analysis of such *landscape features* is the subject of *exploratory landscape analysis* [9], and the landscaped-based selection and instantiation are commonly referred to as *landscape-aware* algorithm selection and configuration.

For a given problem, *fitness landscape analysis* aims at measuring problem characteristics through functions that assign to each problem a vector of real numbers. To date, most works on fitness landscape analysis concentrate on continuous optimization problems [1], but the concepts have also been very successfully applied to some NP-hard combinatorial problems, for example to the satisfiability problems [12] and to the travelling salesperson problem [5].

A description of the most relevant features for numerical problems can be found in [4] and can be conveniently computed using the R-package "flacco" [7]. *flacco* computes up to 343 features for each problem, clustered into 17 feature sets. A *feature* set regroups several *sub-features* corresponding to the same idea. For instance, the feature set *y-distribution* has 3 sub-features: *skewness, number of peaks and kurtosis*, which measure different aspects of the fitness value distribution.

While the previously mentioned success stories [1, 5, 12] demonstrate that the feature values can indeed be used for reliable performance predictions, we currently lack a good understanding of which features are most relevant for the accuracy of the performance models. With the long-term goal to investigate this question in more detail, we contribute with this work a first essential step towards a better understanding, by analysing *(1) the expressiveness of the various features and (2) the robustness of their computation.*

In light of the important impact that the existing applications of landscape-aware algorithm selection and configuration have demonstrated, it may be surprising that these two questions have not been rigorously addressed in the research literature. However, among the existing works, we could only identify one example that studies related questions. Morgan *et al.* [10] have done a study on the influence of the sample size and the dimension of the problem on a single feature, the so-called dispersion metric. They found that the expressiveness of the dispersion feature, measured by its ability to discriminate between different optimization problems, is negatively correlated with the dimension. Moreover, they found that as the dimensionality increases the dispersion of the sample will converge to fixed value which is the same for every considered problem.

In this paper, we focus on a fixed dimension $D = 5$ and we discuss the robustness of several *feature sets* against a random sample of points. Moreover, as mentioned above, we aim at finding which subset of features allows to extract useful information on the problem instance at hand. We analyse this question by generating different samples on several problems extracted from BBOB [2]. Then, we compute the values of the features and look at their distributions. Our key findings are that some features can not be used on this benchmark because of a low robustness or no gain of knowledge over the problem instance. Conversely, we found some features that could be used for landscape-aware heuristic design, since they show a high degree of expressiveness and are robust against both random sampling and the number of sample points.

2 DESIGN OF EXPERIMENT

From the 17 feature sets available in *flacco*, we ignore all of the cell-mapping-based metrics, since it was pointed out in [6] that they are not fitted for complex problems. When the budget is small, those features that do not require additional sampling (such features were coined *"cheap" features* in [1]) are preferable over *"expensive"* ones, which, for example, would require to perform a local search. In our experiments we therefore keep only the cheap features. Thus, overall, we are left with the following 7 feature sets:

- **Dispersion (disp)** which, intuitively speaking, aims at measuring the distance among points of a subset of the sample. These subsets are created with predefined threshold (2%, 5%, 10% and 25%);
- **Information Content (ic)** measuring the variety of information *objects* of the landscape (i.e. smoothness, ruggedness or neutrality);
- **Nearest Best Clustering (nbc)** which extract information based either on the nearest or the better neighbours;
- **Principal Component Analysis (pca)** converting a set of observation variables that could be correlated to a set of variables, called principal components, which are linearly uncorrelated;
- **Meta Model (ela_meta)** fitting linear and quadratic regression models to the initial design;
- **y-distribution (ela_distr)** which describe the distribution of the fitness values;
- **Level Set (ela_level)** splitting the initial data into two classes with a level defined by a threshold. The features are based on misclassification errors of each classifier;

The benchmark used is the Black-Box Optimisation Benchmark (BBOB) [3]. We study the first instance of each functions. We recall that we fix the dimension to $D = 5$. We choose this scale of problems to have meaningful results with convenient computational time. For each function we create 100 samples of $5^D = 3125$ search points,

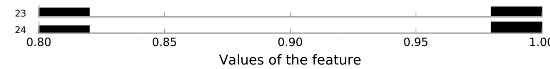

Figure 1: Distribution of the em Principal Component Analysis, *expl_var.cor_x* sub-feature for 3125 search points. The y axis is normalized for all problems.

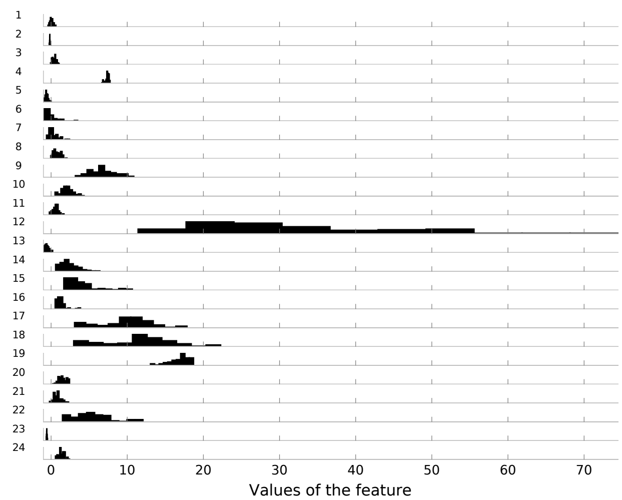

Figure 2: Distribution of the *ELA distribution, kurtosis* sub-feature for 3125 search points. The y axis is normalized for all problems.

all taken from the domain $[-5, 5]^D$. [1] Additionally, to measure the dependency of features to the number of search points, we also planned experiments for less points i.e. 300 and 30 on the same domain. In order to cover the decision space efficiently, we used a low-discrepancy sample generator [11] to sample the search points. Such *quasi-random* sampling methods avoid the cluster effect, and are thus more suited for a design of experiments where we seek to cover the search space well.

3 RESULTS

Once the features are computed, the feature values of the different samples are aggregated into histograms, one for each BBOB function. We then compare the distributions of the feature values, their averages and standard deviations.

3.1 Results for a fixed number of points

In this section, we look at the results for a number of points fixed at $5^D = 3125$.

Fig. 1 shows the distribution of a feature from the Principal Component Analysis set. As the distribution is the same for all 24 benchmark functions, only the two last functions are shown.

The central tendencies of the distributions of the feature in Fig. 2 appear to be different for a large fraction of function pairs. Nevertheless, the variability of the distributions is very large, when

[1] The BBOB functions are traditionally studied over the search domain $[-5, 5]$.

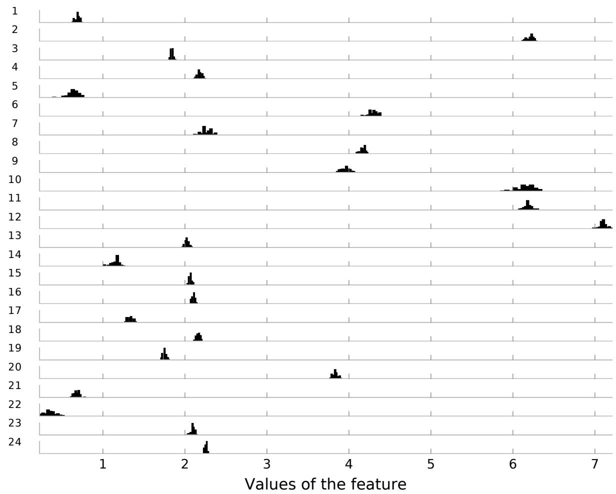

Figure 3: *Distribution of the Information Content, esp_ratio* sub-feature for 3125 search points. The y axis is normalized for all problems.

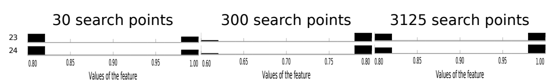

Figure 4: *Distribution of the Principal Component Analysis, expl_var.cor_x* sub-feature with different number of search points. The y axis is normalized for all problems.

measured against the value range and the differences between central tendencies. For instance, the standard deviation for function 12 is almost as large as the maximal value range of all 24 distributions.

In Fig. 3, we observe two main important properties: firstly, the variability of the distributions is very low for each function; secondly, from the visual analysis, the feature seems suitable to discriminate around 475 out of the $24 \times 23 = 552$ function pairs. However, functions 4 and 7 can hardly be distinguished, for instance. Measuring the discriminative power of each feature set through a rigorous statistical approach is one of the next steps to validate our findings.

3.2 Results for different number of points

Fig. 4, 5 and 6 represent the distributions of *expl_var.cor_x*, *kurtosis* and *esp_ratio* features over the BBOB functions for 30, 300 and 3125 search points. The values of the y axis are normalized and are the same for the 3 subplots of each features.

The results are comparable to one can expect. Indeed, when you reduce the number of search points, the variance of a measure almost always rise up. This phenomenon appears in Fig 5 with the *kurtosis*. For the fewer number of points, the variability of the values for some problems is as wide as the value range (problems 12,18,19 and 20 for instance). Only *expl_var.cor_x* (Fig. 4) shows almost the same values regarding either the different problems or the different sample sizes.

One can note in Fig 6 that the variance is only inching up compared to Fig. 5. Moreover, when it was hard to distinguish problems

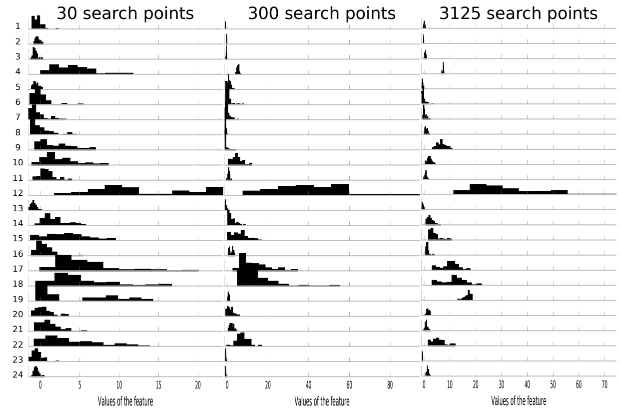

Figure 5: *Distribution of the ELA Distribution, kurtosis* sub-feature with different number of search points. The y axis is normalized for all problems.

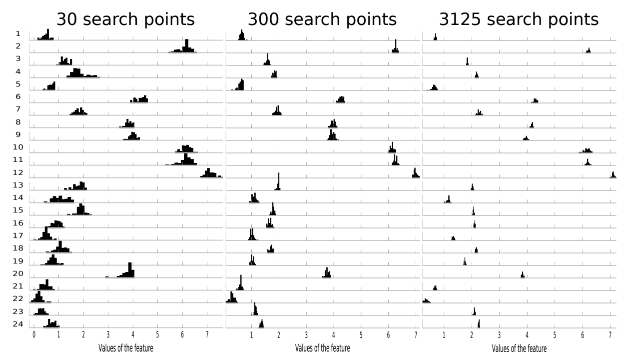

Figure 6: *Distribution of the Information Content, esp_ratio* sub-feature with different number of search points. The y axis is normalized for all problems.

with the *kurtosis* and 30 search points, it is still possible with the *esp_ratio* feature. Nevertheless, the discriminative power of this feature seems to fall with the diminution of search points.

4 DISCUSSION

4.1 Expressiveness

In our work, we define the *expressiveness* of a feature as its ability to distinguish several problems. Hence, the more expressive a feature is, the more problems it can distinguish.

For some features, like the one shown in Fig. 1, the distributions are identical or very similar, making it impossible to discriminate problems based on those features. Consequently, we should discard them in a feature-based algorithm selection and/or configuration, because they do not offer substantial insight into the characteristics of a problem.

In order to be able to identify a problem based on its feature values, the location of the central tendencies of the distribution and the variability of the values are key elements. As can be seen in Fig. 2, the central tendencies are not very similar but from this data we cannot claim that each pair of functions can be distinguished. If the variability of the values is too wide, two feature distributions of distinct problems can overlap. When a feature value falls into this overlapping range, several problems correspond to the measured feature value. We therefore conclude that a feature that is well-fitted to discriminate problems combines differences in central tendencies with a low variability on the different samples. Fig. 3 shows a good example for such a well-fitted feature.

4.2 Robustness

In this paper, we call a feature *robust* when its variance is not large over the range of values taken on all problems. Note here that we discuss in this paper the robustness of a feature to the sampling in two different ways :

- The robustness to the different samples of the same size. Only the locations of points are different;
- The robustness to the number of points generated the same way.

For instance, in case of Fig. 2, as the variance seems to be as big as the value range, it is possible to say that the kurtosis is not robust to the different samples. It also appears that kurtosis is not robust to the number of points since the variance seems to be increasing as the number of points is decreasing. As a matter of fact, this result is more or less generalized to nearly all the features. Nevertheless, for the classification of problems, the robustness of a feature is not sufficient. Fig. 4 is a perfect example of this statement. We can see that this is a very robust feature either regarding the size of the sample or regarding the different samples. For that reason, we should be looking at both expressiveness and robustness to define a well-fitted feature.

4.3 Expressive and robust features

Based on our computations, two of the most expressive and robust features are:

- eps_ratio feature of Information Content: this feature combine a low dispersion and central tendencies that are very well separated;
- lin_simple.intercept of ELA Meta-Model: as the first feature, it has a low dispersion and well separated central tendencies.

Conversely, the least expressive features are:

- expl_var.cov_x of Principal Component Analysis;
- expl_var.cor_x of Principal Component Analysis;
- expl_var.cov_init of Principal Component Analysis.

These 3 features have the same downside. The central tendencies and dispersions are exactly the same for all 24 functions.

Some features like the expl_var.cor_init of Principal Component Analysis or ratio_m_25 from Dispersion can be used to defined groups of functions. It seems that for these kind of features, the distributions are alike for only a couple of functions. It is quite interesting to note that these groups do not correspond to the commonly regarded grouping of the BBOB functions. Hence, it could be useful to define groups of functions from a feature point of view.

We also observe that the identification of the BBOB functions requires only a small number of features. As mentioned in Section 3, some pairs of functions can be distinguished with only one feature. For the mentioned example of functions 4 and 7, which cannot be discriminated by this feature, the kurtosis feature displayed in Fig. 2 allows to distinguish this pair. Identifying the minimal number of features needed to distinguish all pairs of functions is a promising research question, since it allows to reduce the computational overhead of the feature computation and will hopefully lead to a better interpretability of high-performing landscape-aware algorithm selectors and configurators.

5 CONCLUSION

Our study exhibits three categories of features. One category concerns the features that are not robust against the sampling, such as the example from Fig. 2. Another category concerns features that are not expressive, in the sense that the feature values are similar for several function pairs. The feature from Fig. 1 shows an extreme example of this case. A third category comprises the features that are well-fitted to discriminate between problems. As discussed above, these features are both expressive and robust.

Our results were mostly based on visual analysis, so that an obvious direction for future work is the application of suitable statistical tests to confirm our findings. Moreover, we plan on comparing the standard grouping of BBOB functions to those groups of suggested by the feature analysis.

REFERENCES

[1] Nacim Belkhir. 2017. *Per Instance Algorithm Configuration for Continuous Black Box Optimization.* phdthesis. Université Paris-Saclay.

[2] N. Hansen, A. Auger, O. Mersmann, T. Tušar, and D. Brockhoff. 2016. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. *ArXiv e-prints* arXiv:1603.08785 (2016).

[3] 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 *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation.* ACM, 1689–1696.

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

[5] Pascal Kerschke, Lars Kotthoff, Jakob Bossek, Holger H. Hoos, and Heike Trautmann. 2018. Leveraging TSP Solver Complementarity through Machine Learning. *Evolutionary Computation* 26, 4 (2018).

[6] Pascal Kerschke, Mike Preuss, Carlos Hernández, Oliver Schütze, Jian-Qiao Sun, Christian Grimme, Günter Rudolph, Bernd Bischl, and Heike Trautmann. 2014. Cell Mapping Techniques for Exploratory Landscape Analysis. In *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V.* Vol. 288. Springer International Publishing, Cham, 115–131. http://link.springer.com/10.1007/978-3-319-07494-8_9

[7] 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. Flacco is available at http://kerschke.github.io/flacco/.

[8] Pascal Kerschke and Heike Trautmann. 2019. Automated Algorithm Selection on Continuous Black-Box Problems by Combining Exploratory Landscape Analysis and Machine Learning. *Evolutionary Computation* 27, 1 (2019), 99–127.

[9] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. 2011. Exploratory Landscape Analysis. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO '11).* ACM, New York, NY, USA, 829–836. event-place: Dublin, Ireland.

[10] Rachael Morgan and Marcus Gallagher. 2014. Sampling Techniques and Distance Metrics in High Dimensional Continuous Landscape Analysis: Limitations and Improvements. *IEEE Transactions on Evolutionary Computation* 18, 3 (2014), 456–461. http://ieeexplore.ieee.org/document/6595542/

[11] François-Michel De Rainville, Christian Gagné, Olivier Teytaud, and Denis Laurendeau. 2012. Evolutionary optimization of low-discrepancy sequences. *ACM Trans. Model. Comput. Simul.* 22 (2012), 9:1–9:25.

[12] Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. 2008. SATzilla: Portfolio-based Algorithm Selection for SAT. *J. Artif. Intell. Res.* 32 (2008), 565–606.