# Comparing Solvability Patterns of Algorithms across Diverse Problem Landscapes

ANA NIKOLIKJ, Jozef Stefan International Postgraduate School and Jozef Stefan Institute, Slovenia
TOME EFTIMOV, Jozef Stefan Institute, Slovenia

In the field of continuous single-objective black-box optimization, understanding the varying performances of algorithms across different problem instances is crucial. A recent approach based on the concept of "algorithm footprint" identifies both easy and challenging problem instances for an algorithm. However, a major challenge persists — the lack of comparability among different algorithm footprints for effective benchmarking. This study introduces a solution through a multi-target regression model (MTR), which predicts the performance of multiple algorithms simultaneously, using a common set of problem landscape features. By establishing a common landscape feature set and using a single performance prediction model, not only can algorithm footprints be compared, but the explanations for the predicted algorithm performance derived with Explainable Artificial Intelligence (XAI) techniques can also be analyzed systematically. The methodology is applied to a set of three distinct algorithms, revealing their respective strengths and weaknesses on the Black-Box Optimization Benchmarking (BBOB) suite.

CCS Concepts: • **Computing methodologies** → **Machine learning**; **Learning latent representations**; **Supervised learning**; • **Theory of computation** → **Design and analysis of algorithms**.

Additional Key Words and Phrases: algorithm behavior, single-objective optimization, latent representations, supervised machine learning, explainability

**ACM Reference Format:**
Ana Nikolikj and Tome Eftimov. 2024. Comparing Solvability Patterns of Algorithms across Diverse Problem Landscapes. In *Genetic and Evolutionary Computation Conference (GECCO '24 Companion), July 14–18, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3638530.3654305

## 1 Introduction

In continuous single-objective optimization (SOO), no single algorithm outperforms all the others on every problem, making the selection of the most suitable algorithm for a new problem non-trivial [16, 19]. Thus, methodologies able to compare algorithms in a way that reveals their complementarity are of great importance.

The standard method for assessing algorithm efficacy involves analyzing their performance across a suite of benchmark problem instances [8], using statistical methods that focus on average performance results or comparing performance distributions [3]. A key limitation is that these methods rely solely on raw performance data without considering the properties of the problem instances, contributing to a black-box view of the algorithms.

Nikolikj et al. present a methodology for formulating an *algorithm footprint* consisting of problem instance sets that are either easy or difficult for the algorithm to solve. First, they train a single-target regression model (STR) to predict an algorithm's performance using the landscape feature representation of the benchmark problem

Authors' Contact Information: Ana Nikolikj, ana.nikolikj@ijs.si, Jozef Stefan International Postgraduate School and Jozef Stefan Institute, Ljubljana, Slovenia; Tome Eftimov, tome.eftimov@ijs.si, Jozef Stefan Institute, Ljubljana, Slovenia.


*GECCO '24, July 14–18, 2024, Melbourne, Australia*

ACM ISBN 979-8-4007-0495-6/24/07
https://doi.org/10.1145/3638530.3654305

instances. Then, the SHAP [11] method is employed to calculate local feature importance - how important is a feature on the level of a single problem instance in the dataset. The local feature importances are used to create meta-representations for each problem instance which encode the relationship between the landscape features and the algorithm's performance. By grouping the meta-representations with *deterministic clustering*, different performing regions for the algorithm are detected and further analyzed. The main challenge is that the methodology cannot be used to benchmark different algorithms based on their footprints. Each algorithm's footprint is created using a single-target regression model (STR) with unique features and hyper-parameters, resulting in footprints that exist in separate vector spaces. Deterministic clustering uses a single threshold to categorize algorithm performance, assumes uniformity within clusters. This can be misleading because instances near the threshold might be more similar to each other than to distant ones in the same cluster.

**Our contribution:** To enable benchmarking of algorithm footprints, this study employs multi-target regression models (MTR)[20] that predict the performance of multiple algorithms simultaneously, using a common set of problem landscape features suitable for predicting the performance of all algorithms, allowing for the generation of comparable footprints. Using this methodology, we analyzed the performance of three algorithms — Particle Swarm Optimization (PSO)[7], Estimation of Multivariate Normal Algorithm (EMNA)[10], and Random Search[1], across the 24 problems from the BBOB benchmark suite. The results showed similar performance on most of the problems, with notable exceptions like problem nine where EMNA excels, and problem 20 where PSO fails. Additionally, the post-hoc analysis revealed which landscape features make the problems easy or challenging for different algorithms.

**Outline:** The paper is structured as follows: Section 2 outlines the methodology for benchmarking algorithm footprints. Section 3 details the data and machine learning experiments. Results are presented in Section 4, and conclusions and future directions are in Section 5.

## 2 Methodology

To address the challenges from [15], in this study, a multi-target regression model (MTR) is trained that learns multiple tasks simultaneously (in our case, the performances of the algorithms in the portfolio) using a shared feature representation (the landscape feature representation of the problem instances). Next, the SHAP method for local explainability is applied to the trained model and determines the importance of landscape features in predicting the performance of each algorithm, on each problem instance from the test set. This results in generating multiple meta-representations (i.e. as many as target variables) for each problem instance in the test dataset, providing insights into how the same landscape features influence the different target variables. The meta-representations are further clustered using a clustering algorithm that automatically detects different regions concerning the algorithm performance across the problem landscape. We then order the clusters by the mean algorithm performance going from the best-performing cluster to the worst-performing one. Then, by comparing how the meta-representations that correspond to the same problem instance and different algorithms are distributed across the clusters, we can reveal similar or distinct algorithm behavior. As well as, regions that are easy or hard for the portfolio as a whole. Finally, we identify the landscape features that make problem instances easy or challenging for the algorithms by determining the most important features for each cluster using feature importance as a model explainability technique.

## 3 Experimental Design

We use the same problem portfolio as in [17], utilizing the Black-Box Optimization Benchmarking (BBOB) suite on the COCO platform [6], which comprises 24 noiseless SOO problem classes. Each class is transformed via translation, rotation, and scaling to create 50 instances, totaling 1,200 problem instances with a problem dimension $d$ of five ($d$=5). To numerically represent the problem instances, we employ 99 Exploratory Landscape Analysis

(ELA) features [12] from the same study [17]. The algorithm portfolio includes Particle Swarm Optimization (PSO) [7], Estimation of Multivariate Normal Algorithm (EMNA) [10], and Random Search [1], with their Nevergrad toolbox [1] implementation. The algorithm performance is assessed using a fixed-budget approach, by measuring the precision (distance to the optimum) of the best solution after 5.000 function evaluations. Each algorithm is executed 50 times per problem instance, then the median solution precision across the runs is calculated, and additionaly log-transformed to facilitate modeling with machine learning algorithms.

We experiment with three different ML algorithms, namely Random Forest (RF) [2] and Multi-Task Elastic Net (MTEN) [21] with their implementation in the *scikit-learn* package, and Neural Network (NN) [5] implemented in the *keras* package in Python. We need to note here that each ELA feature has been normalized in the range 0 to 1 before training the models, to bring them to the same scale. The scaler was fitted on the training set and then the test set was transformed.

To evaluate the ML models we perform a stratified train-test split, where the data are stratified based on the "problem class", with five instances from each problem class left out for testing (120 problem instances in total), while the other constitute the training set (980). We conduct a hyper-parameter search with the Tree-structured Parzen Estimator (TPE) algorithm from the *optuna* package in Python. The hyper-parameters with the best 5-fold stratified cross-validation score on the training set are selected as the best-performing. After tuning the hyper-parameters of the models, a Sequential Forward Feature Selection (SFFS) [4] is applied. We have performed this for the RF and MTEL models. We omit it for an NN as it can be a very time-consuming procedure. The feature subset that yielded the best 5-fold stratified cross-validation score on the train set was selected. Details on the hyper-parameter optimization and feature selection choices are documented on our GitHub repository [14]. Finally, the best-performing models for RF, NN, and MTEL (hyperparameters and features subset), the models are retrained on the entire training set, and tested on the held-out test set to evaluate their performance on unseen data. The model performance is measured with the Mean Absolute Error (MAE) and coefficient of determination (R2 score).

The SHAP method [11] determines the importance (i.e., the Shapley value) of each feature of a data instance, to the model's prediction on the data instance. In the context of ML, an additive explanation means that the output of the model for a data instance (i.e., its prediction) can be decomposed as the sum of the contributions (importance values) of the features.

We employ hierarchical clustering [13] as a clustering technique and measure cluster quality using the Silhouette coefficient. A higher score indicates better separation of clusters i.e., clusters are dense and well separated.

## 4 Results

**Algorithm Performance Prediction:** To find a good-performing model we evaluated three MTR models for algorithm performance prediction and compared them to a baseline model, which always predicts the average algorithm performance. The results indicate that all models significantly surpass the baseline in both metrics. The RF model after hyper-parameter optimization and trained on a subset of 22 features was the top performer for all algorithms.

**Clustering the Algorithm Behavior Meta-representations:** For the RF model, we generate local SHAP explanations for each algorithm, which results in generating three meta-representations for each problem instance in the test dataset. The different meta-representations provide insights into how the same landscape features influence the different algorithms. Next, a hierarchical clustering algorithm is applied to the meta-representations to identify the distinct regions of algorithm behavior. The clustering resulted in ten clusters with a silhouette score of around 0.6 and further clustering does not improve the clustering score.

Further, we compare the clusters across different algorithms, making the footprints comparable. To this end, we examine if the meta-representations corresponding to the same problem instance but different algorithms

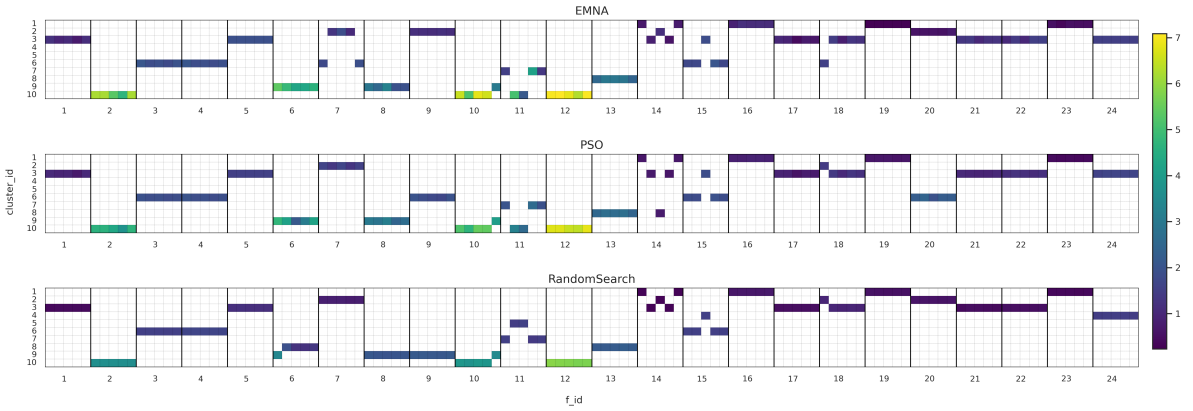


Fig. 1. Coverage matrix of the distribution of the meta-representations in the clusters, illustrating the results for 10 clusters.

are clustered together, indicating similar algorithm behavior. Otherwise, if they are in different clusters this means that the algorithms have different performance on the problem instances. The *coverage matrix* in Figure 1 visualizes this, with problem instances as columns, and clusters as rows. The color represents the ground truth algorithm performance achieved on the corresponding problem instance. The clusters are ranked by the average ground truth performance, with the first cluster representing the best and the last the worst performance. By examining the clustering patterns for the problem instance across the three algorithms, we can see that all five instances of the 1st, 5th, 16th, 17th, 19th, 21st, 22nd, 23rd, and 24th problems problem belong to the same clusters. This means that in these problems all the algorithms perform similarly, also visible by the ground truth algorithm performance. All the algorithms are highly effective in these problem instances, as the clusters (from one to three) align with the best-performing regions according to ground truth algorithm performance (dark blue). On the other hand, all the algorithms have poor performance in the problem instances from the 2nd, 8th, 12th, and 13th problems, whose clusters correspond to the worst ground truth algorithm performance regions (yellow). Overall, from this figure, we can conclude that the algorithms behave very similarly over all the benchmark problems as the clustering patterns of the problems are similar across algorithms. However, it is notable that EMNA is by far the best-performing algorithm on the 9th problem. Random Search is a significantly better performing algorithm on the 6th and the 11th problems than the others. Also for PSO is visible that is it the worst performing on the 20th problem.

**Post-hoc Explainability Analysis:** In Figure 2 the ten most important landscape features for the problem instances related to the 19th and 20th problems are illustrated. Each sub-figure in the plot presents a SHAP *decision plot*, depicting the local importance of each feature to the model's prediction on the problem instances. The y-axis displays the most important features in descending order, with the highest importance at the top, while the x-axis represents the model's prediction value. In the plot, each line corresponds to a distinct problem instance within the specified problem classes (five in total). The lines depict the cumulative sum of feature contributions, tracing a path from a base value to the final prediction made by the model. The base value corresponds to the average ground truth algorithm performance. At the top of the plot, each line intersects the x-axis at the predicted value determined by the model, with the color of the line indicating the predicted performance spectrum from dark blue (best) to yellow (worst). Each row of sub-figures corresponds to a problem class, and each column corresponds to a specific algorithm. Focusing on the 19th problem class, we detected that all the algorithms perform very well. We can see that the features also have a similar order of importance from top to bottom, with *ic.eps.ratio*,

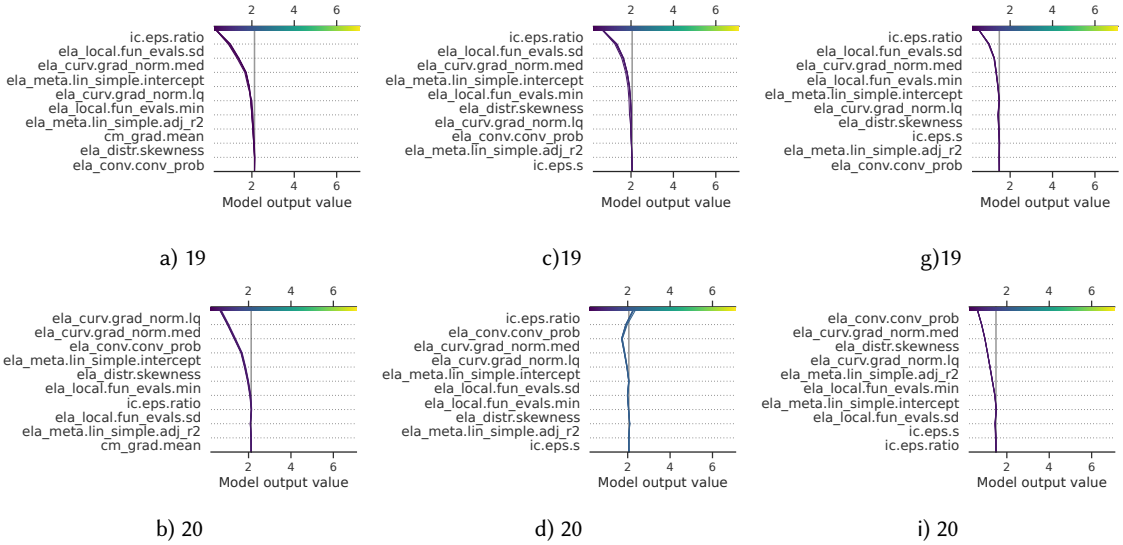


Fig. 2. Visualization of the 10 most important features for a-b) EMNA, c-d) PSO, and g-i) RS for the corresponding problem as indicated in the subfigure caption.

*ela_local.fun_evals.sd*, *ela_curv.grad_norm.med* appearing at the top in all three algorithms. Also, the lines for the different problem instances are almost identical, meaning that the algorithms behave very similarly on all the instances from this problem class. This suggests a straightforward relationship between the identified features and the algorithm behavior that happens in the first cluster (best-performing region). On the 20th problem, it becomes evident that EMNA and RS exhibit nearly identical performance, indicated by similar lines and a consistent order of features. This positions both EMNA and RS as optimal choices for addressing this problem. In contrast, PSO encounters challenges in solving this particular problem, leading to its placement in a less favorable performance region characterized by distinct feature patterns. Thus we can conclude that the landscape features have different importance on the algorithm performance in the different regions identified with the methodology.

## 5 Conclusions

This study developed a method for benchmarking algorithm footprints using multi-target regression models (MTR), allowing for simultaneous performance predictions of multiple algorithms using a subset of features universally suitable for performance prediction across all algorithms.This shared feature set, coupled with a unified performance prediction model, facilitated comprehensive comparisons of algorithmic footprints. Additionally, post-hoc explainable analysis revealed landscape features that affect problem difficulty. We applied this method to benchmark three different algorithms on the BBOB benchmark suite, providing insights into their complementarity and individual strengths. For future work, we plan to apply our methodology to analyze top-performing algorithms from the COCO competitions [6] and the Nevergard platform [1], assessing their performance across different problem types. We will also explore various benchmark suites, focusing on new ones like MA-BBOB [18]. Additionally, we intend to document our findings in the OPTION [9] ontology, to systematically trace insights from single-objective black-box optimization algorithms.

## Acknowledgments

The authors acknowledge the support of the Slovenian Research Agency through program grant P2-0098, Young Researcher grant No. PR-12897 to AN, and project No. J2-4460.

## References

[1] Pauline Bennet, Carola Doerr, Antoine Moreau, Jeremy Rapin, Fabien Teytaud, and Olivier Teytaud. 2021. Nevergrad: black-box optimization platform. *ACM SIGEVOlution* 14, 1 (2021), 8–15.

[2] Gérard Biau and Erwan Scornet. 2016. A random forest guided tour. *Test* 25 (2016), 197–227.

[3] Tome Eftimov, Peter Korošec, and Barbara Koroušić Seljak. 2017. A novel approach to the statistical comparison of meta-heuristic stochastic optimization algorithms using deep statistics. *Information Sciences* 417 (2017), 186–215.

[4] Francesc J Ferri, Pavel Pudil, Mohamad Hatef, and Josef Kittler. 1994. Comparative study of techniques for large-scale feature selection. In *Machine intelligence and pattern recognition*. Vol. 16. Elsevier, 403–413.

[5] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[6] Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. 2021. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software* 36, 1 (2021), 114–144.

[7] James Kennedy and Russell Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, Vol. 4. IEEE, 1942–1948.

[8] Pascal Kerschke, Holger H Hoos, Frank Neumann, and Heike Trautmann. 2019. Automated algorithm selection: Survey and perspectives. *Evolutionary computation* 27, 1 (2019), 3–45.

[9] Ana Kostovska, Diederick Vermetten, Carola Doerr, Sašo Džeroski, Panče Panov, and Tome Eftimov. 2023. OPTION: OPTImization Algorithm Benchmarking ONtology. *IEEE Transactions on Evolutionary Computation* 27, 6 (2023), 1618–1632. https://doi.org/10.1109/TEVC.2022.3232844

[10] Pedro Larrañaga and Jose A Lozano. 2001. *Estimation of distribution algorithms: A new tool for evolutionary computation*. Vol. 2. Springer Science & Business Media.

[11] Wilson E Marcílio and Danilo M Eler. 2020. From explanations to feature selection: assessing SHAP values as feature selection mechanism. In *2020 33rd SIBGRAPI conference on Graphics, Patterns and Images (SIBGRAPI)*. Ieee, 340–347.

[12] 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*. 829–836.

[13] Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms. *ArXiv* abs/1109.2378 (2011). https://api.semanticscholar.org/CorpusID:8490224

[14] Ana Nikolikj. 2024. *Footprints MTR*. https://github.com/anikolik/footprints_mtr.git

[15] Ana Nikolikj, Sašo Džeroski, Mario Andrés Muñoz, Carola Doerr, Peter Korošec, and Tome Eftimov. 2023. Algorithm Instance Footprint: Separating Easily Solvable and Challenging Problem Instances. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lisbon, Portugal) *(GECCO '23)*. 529–537.

[16] John R Rice. 1976. The algorithm selection problem. In *Advances in computers*. Vol. 15. Elsevier, 65–118.

[17] Risto Trajanov, Stefan Dimeski, Martin Popovski, Peter Korošec, and Tome Eftimov. 2021. Explainable landscape-aware optimization performance prediction. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 01–08.

[18] Diederick Vermetten, Furong Ye, Thomas Bäck, and Carola Doerr. 2023. MA-BBOB: Many-Affine Combinations of BBOB Functions for Evaluating AutoML Approaches in Noiseless Numerical Black-Box Optimization Contexts. *arXiv preprint arXiv:2306.10627* (2023).

[19] D.H. Wolpert and W.G. Macready. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1, 1 (1997), 67–82. https://doi.org/10.1109/4235.585893

[20] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering* 34, 12 (2021), 5586–5609.

[21] Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67, 2 (2005), 301–320.