2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Analyzing the generalizability of automated algorithm selection: a case study for numerical optimization

1st Urban Škvorc
Computer Systems Department
Jožef Stefan Institute,
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
urban.skvorc@ijs.si

2nd Tome Eftimov
Computer Systems Department
Jožef Stefan Institute
Ljubljana, Slovenia
tome.eftimov@ijs.si

3rd Peter Korošec
Computer Systems Department
Jožef Stefan Institute
Ljubljana, Slovenia
peter.korosec@ijs.si

***Abstract*—In numerical single-objective optimization, automated algorithm selection that uses exploratory landscape analysis to describe problem features has achieved great results when the machine learning models used for prediction are trained and tested on the same problem set. However, recent work has shown that the performance of such models decreases when the training and testing sets contain different problems. In this paper, we examine a recently developed algorithm selection model trained on a set of artificial problems and tested on a well-known set of hand-made benchmark problems. This model performed poorly when it was originally presented. Here, we provide an explanation for its poor performance by analyzing the feature importance of the model using Shapley Additive Explanations. We then compare these results to an alternative algorithm selection model that was both trained and tested on the same set of hand-made benchmark problems and achieved much higher performance. This allows us to determine which features each model considers as most significant for their predictions, and where they differ. We show that the original and the alternative model use different landscape features for their predictions, which explains the difference in their performance. Further, by plotting the SHAP values on a 2D plane, we show that the original model is unable to distinguish between certain types of problems. Finally, we show that regardless of their differences in utilizing the features both the original and the alternative models perform poorly on a specific group of problems.**

***Index Terms*—algorithm selection, exploratory landscape analysis, numerical optimization**

I. Introduction

The task of selecting the best performing algorithm for a given problem, called algorithm selection [1], is a popular area of research in optimization [2]. This is because algorithm performance depends heavily on the specific problem being solved, and no single algorithm exists that can solve all problems well. In recent years, techniques such as exploratory landscape analysis (ELA) [3] have been used with great success for algorithm selection in the domain of single objective numerical optimization, as they allow researchers to convert problem samples into real-valued descriptors called landscape features that can be used directly by machine learning models. In addition, these landscape features can easily be computed by utilizing programming libraries such as “flacco” [4].

However, the performance of such algorithm selection models is heavily dependent on the training and testing data used. If the data in the training and the testing sets are similar then the models produce great results. One example of such work is the study performed by Kerschke and Trautmann [5], which analyzed the performance of an automated algorithm selection model trained and tested on a set of 24 problems from the BBOB benchmark set (BBOB problems) [6], and reduced the overall runtime of the optimization procedure by half compared to only using a single best solver. In this experiment, the training and the testing data were similar to one another. However, other work has shown that ELA can have problems when the training and testing problems are different from one another in terms of their ELA representation [7], [8], [9].

In this paper, we examine a different group of models that we developed in our prior work [10], where we presented several algorithm selection models that differed based on the data used for training and testing. A performance evaluation of these models showed that the models that were trained and tested on similar data sets performed well, while the models that were trained on one set and tested on a different set performed poorly.

In this paper, we aim to expand on this analysis. To do this, we use Shapley Additive Explanations (SHAP) [11], a state-of-the-art approach for calculating feature importance by using concepts from game theory. The main goal of this paper is to compare the SHAP values of two different models introduced in our prior work. The first, which we will refer to as the original model, was trained and tested on different problem sets and achieved poor performance. The second, which we will call the alternative model, was trained and tested on only the BBOB problems. We analyze the SHAP values of the two models both by directly comparing the mean SHAP values and by visualizing them using t-sne [12] to reduce the dimensionality of the SHAP values to two.

This work was supported by the projects from the Slovenian Research Agency (project numbers No. P2-0098, Pr-08987, J2-4460, and N2-0239).

This paper is structured in the following way. In Section II we provide an overview of related work, including a brief overview of the original model presented in our prior work [10]. In Section III we present our experimental structure. In Section IV we present the results of our experiments. Finally, in Section V, we present the conclusion of our research.

II. Background

In our prior work [10] we presented several automated algorithm selection models that used exploratory landscape analysis and random forests, with the goal of determining how well ELA can generalize information between different problem sets. The models that we presented differed in the problems that they were trained and tested on, with five models examined in total. We concluded that models where the training data is different than the testing data in terms of their landscape features produce poor performance, while the models where the training and testing data are similar produce good performance.

In this paper, we aim to perform a more detailed analysis of these models' predictions in order to explain the performance of the models as it relates to the landscape features. Specifically, we want to explain the difference between two specific models out of the five introduced by the paper. The first, which we will call the original model, is trained on a set of artificially generated problems and evaluated on the BBOB problems. This model originally achieved poor algorithm selection performance because the data used to train the model differed greatly from the data used to test the model, with ELA unable to generalize between the two different datasets. The second, which we will call the alternative model, is trained and tested on the BBOB problems using cross-validation. This model performed much better than the original model. These two models were chosen because the former was the best-performing of the models examined, while the latter was the worst-performing, and because both of these models were evaluated on the same dataset. Both of the models are tasked with classifying which one out of a specific set of 10 algorithms achieves the best performance on a given problem, with the performance of the algorithms defined as the number of evaluations needed to reach a certain target precision.

The artificial problems are generated by a generator presented in [13]. The artificial problems are generated by using a tree structure which is randomly modified by the addition of different mathematical operators, as well as by the process of difficulty injection, which gives the function additional properties that are known to present difficulties to optimization algorithms, for example by introducing noise or multimodality to the problem. In addition, the generator evaluates the performance of 10 different optimization algorithms on each generated problem, and the best-performing algorithm is selected as the prediction class of each problem. The models use 50 problems for each of the 10 algorithms in order to ensure a balanced dataset, for a total of 500 problems.

The BBOB problems consist of 24 problems that are commonly used in numerical single-objective optimization benchmarking. For each of the 24 problems, the benchmarking platform allows the generation of an infinite amount of problem instances, which are small variations of the problem that should not affect algorithm performance. The models used 15 instances for each of the 24 problems, for a total of 360 instances. This data is much more imbalanced than the artificial problem data. For example, only seven of the 10 algorithms present in the training data were present in the testing data.

Both of the models used ELA to calculate the landscape features that were used for the machine learning models. Using ELA allows for the transformation of problem samples into numerical descriptors called landscape features which estimate the underlying properties of the problem instance, for example, its modality or its skewness. In total, 44 landscape features were used. Landscape features have shown great results when used for the task of algorithm selection [5], however, there has only been a limited amount of research into how well they generalize between different problem sets

To explain the difference in performance between the two algorithms, we use Shapley Additive Explanations [11], which allow us to calculate for each landscape feature its individual contribution towards a model's prediction for each problem instance. This approach has been used to explain the importance of ELA values in machine learning, for example in papers by [14], or [15]. The model we examine in this study differs from these by examining a classification rather than a regression model.

III. Experimental Setup

In order to more thoroughly evaluate the effect of the individual landscape features on the model's predictions, we examined the SHAP values on the level of predictions on each individual instance. The overview of this experiment is as follows:

1) Train the two algorithm selection models to be analyzed: the original model, and the alternative model. These two models use different training sets but are both evaluated on the same testing set of 360 BBOB instances.
2) For each model prediction, calculate the SHAP values that explain the feature importance that the model used for this prediction.
3) Plot the collected SHAP data on a 2D plane using t-sne.

In the first step, we train the two models that we will examine in this paper: the original model and the alternative model. To do this, we use the same training data and model definitions as in our original paper [10]. All parameters are unchanged from prior work, which used the default parameters for the random forest models with the number of trees set at 1000.

In the second step, we calculate the SHAP data for the two models to obtain explanations for the models' chosen predictions, based on the same training and testing used in our prior work [10]. The SHAP values are computed using the Tree SHAP algorithm [16] provided by the Python library shap [17] using the default parameters provided by the library.

For each prediction, we obtain 44 SHAP values, one value for each landscape feature that was used to train the algorithm selection learning model. The landscape features used are based on our prior work [8], which has shown these landscape features to be promising in terms of algorithm selection. For each instance, each of the 44 individual SHAP values represents the contribution of the corresponding landscape feature towards the model's decision to assign the selected class as the predicted class. The sum of all 44 SHAP values represents the probability that the model assigns to the class.

In the third step, as the 44 SHAP values cannot easily be visualized, we transform them into a two-dimensional representation using t-sne [12], and plot them visually on a 2D plane so that the predictions with similar SHAP values appear close together. The visualizations are performed using 10 000 iterations and a perplexity of 15. The number of iterations was fixed at 10 000 because further iterations produced no improvements. The perplexity parameter is conventionally set to a number between 5-50 but has to be determined manually, as it depends on the data being visualized. The parameter was set to 15 experimentally, as this produced good results, although other values produced roughly similar results. In order to achieve consistent visualizations and ensure repeatability, the random seed of the t-sne algorithm was set to 0 for all visualizations. As with the perplexity parameter, different seed values still produced similar results.

This approach does have several limitations. First, we are limited to the data, models, and methodology that we used in the original paper. Second, by performing dimensionality reduction using t-sne, some information is naturally lost. In addition, caution has to be used when interpreting t-sne visualizations, as the distances between clusters can be nondeterministic, and therefore do not carry significant information. As such, only the presence of clusters themselves should be considered significant, but not specific distances between them. Finally, the TreeSHAP algorithm used can only approximate the true SHAP values, as computing them precisely is computationally demanding.

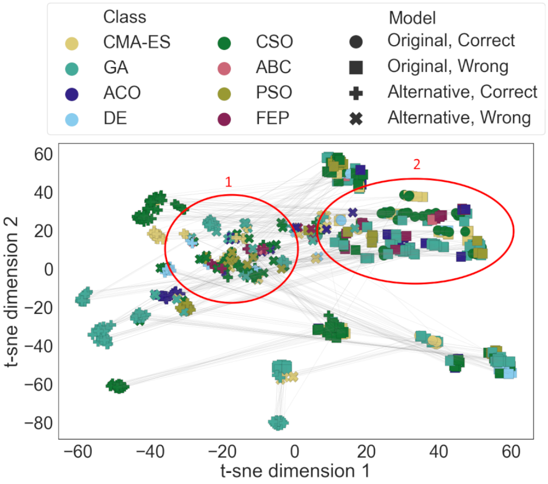

Fig. 1. A t-sne representation of the SHAP values of all predictions by both the alternative model and the original model, with each point representing a single instance that was predicted by the models. We highlight two groups of instances that we examine in more detail.

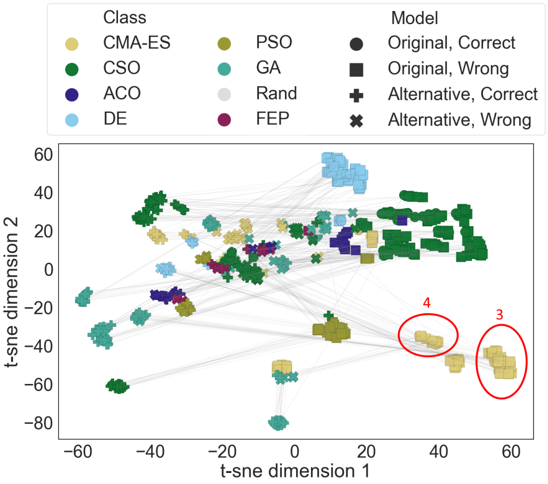

Fig. 2. A t-sne representation of the SHAP values of all predictions by both the instance split alternative model and the original model, with each point representing a single instance that was predicted by the models. We highlight two additional groups of instances that we examine in more detail.

IV. Results

In this Section, we present the results of our investigation. First, we examine the individual SHAP values of all model predictions. Then, we examine smaller subsets of predictions for a more detailed analysis. Finally, we aggregate the SHAP values across an entire prediction class to examine which specific landscape features the models consider important.

Figure 1 shows the t-sne visualization of the SHAP values of all predictions made by the two models, with the colors representing the true class of each prediction, and the shapes representing the model and whether or not the prediction was correct. The lines in the Figure connect the predictions from the original model to the predictions on the same instance from the alternative model. We can see that the SHAP values of the two models are very different overall, as the predictions of the two models are almost entirely separated in the 2D space, with the alternative model occupying the left side of the figure, and the original model occupying the right side.

We can see that some of the predictions form well-defined clusters on the edges of the visualization. The BBOB problem set that was used to calculate the SHAP values contains 24 problems and 15 instances per problem. As the instances are just small variations of a problem, an accurate algorithm selection model should be able to group individual instances into a single cluster. We also see that both models produce one large cluster of problems each. These two clusters are marked as 1 and 2 in the visualization. These clusters contain instances of many different problems with many different corresponding true classes. This indicates that both models are unable to distinguish between some types of problems.

In summary, from Figure 1, it appears that the predictions from both the alternative and the original model can be split into two groups. In the first group are the predictions made on the BBOB problems that can be easily distinguished and separated from the rest of the problems. In the second group are the BBOB problems that appear more difficult to categorize, and that the models categorize into one large cluster. A major difference between the original and the alternative models is that the alternative model is able to correctly predict the instances in the first group. On the other hand, the original model largely misclassifies the first group. Both models perform poorly on the second group, and this group represents the majority of misclassification for the alternative model.

Figure 2 shows the same visualization, but with the predictions colored by the predicted rather than the true class. When looking at the smaller, individual clusters, we can see that both the alternative and the original model predict them entirely as a single class, which is expected, since the models consider these instances as similar to one another. If we examine the original model, we can see that the large group on the top right is predicted entirely as CSO, while the smaller group at the top is predicted entirely as DE, and the group on the bottom right as CMA-ES.

In reality, most of the smaller clusters identified by the original model actually represent multiple different types of BBOB problems. We can determine this by examining the lines that map each of the predictions of the original model with a corresponding prediction of the alternative model. We can see that most of the clusters of the original model are connected to multiple smaller clusters of the alternative model. It appears that the original model has given more importance to landscape features which might not correctly group the BBOB problems and is not able to distinguish between different types of problems within this cluster in the same way that the alternative model is able to. To examine this in more detail, we will look at the clusters marked as 3 and 4 in Figure 2. As these relations are hard to see in the overall plot, we will examine the clusters separately.

Figure 3 shows a more detailed analysis of only the predictions belonging to cluster 3. We can see that the predictions of the original model map to several different groups of the alternative model. While the alternative model was able to separate these groups and predict them with the correct classes, the original model grouped all of these instances together and predicted them as a single class. In addition, the class chosen by the original model for the prediction, CMA-ES, is almost entirely incorrect, as it is only correct for a single prediction. This further indicates that the original model is unable to distinguish between these problems, while the alternative model can by considering different landscape features for making the prediction.

Figure 4 shows a similar visualization as Figure 3, but examining the cluster marked as 4 in Figure 2. Once again, we can observe similar results: the cluster belonging to the original model corresponds to multiple clusters of the alternative model, and the original model makes only a single prediction. In this case, the original model achieves slightly higher accuracy on this cluster, since its prediction matches more problems. However, the general pattern of behavior remains the same.

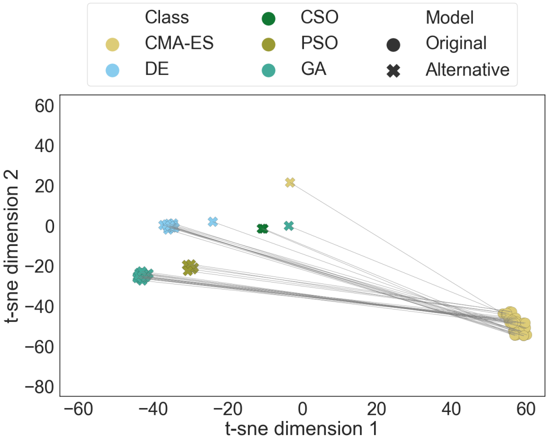

Fig. 3. A smaller subset of SHAP values showing just the bottom right group of the original model that was marked as cluster 3, as well as the paired alternative model predictions. Colors represent the predicted class of the instance.

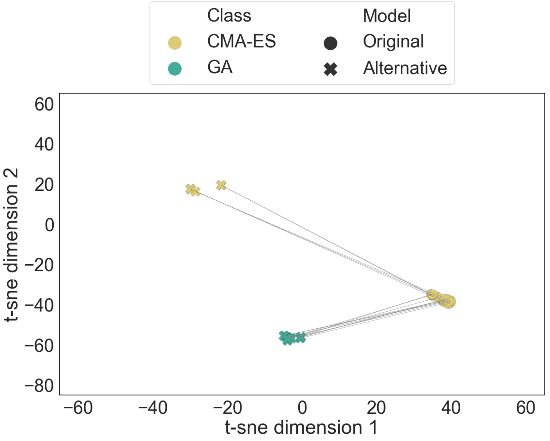

Fig. 4. A smaller subset of SHAP values showing just the bottom right group of the original model that was marked as cluster 4.

Figure 5 shows the same instances as Figure 4, but colored by whether or not the prediction of the model was correct. To make this figure more readable, we allow the t-sne algorithm to reposition the instances, placing them farther apart, so that we can better see the instance pairs. Interestingly, this plot shows that despite its poor performance, the original model is still able to correctly predict some instances that the alternative model does not.

Figure 6 shows the same instances as Figures 1 and 2, but with the individual instances colored by whether or not the prediction was correct. This allows us to more easily analyze where the models made mistakes. As we have seen previously, the alternative model is able to correctly predict most of its smaller clusters, while the original model predicts most of these clusters incorrectly. On the other hand, the two large

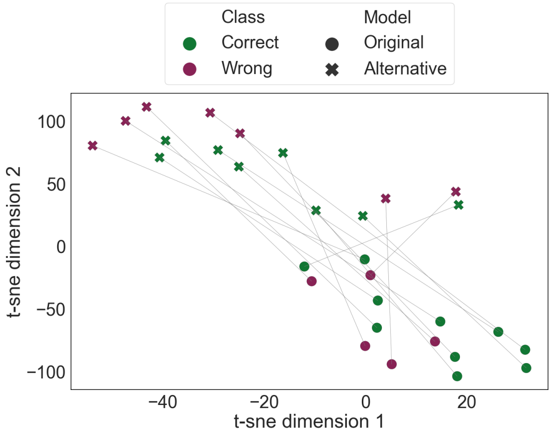

Fig. 5. The SHAP values from Figure 4, but visualized further apart, and colored by whether or not the model predictions were correct.

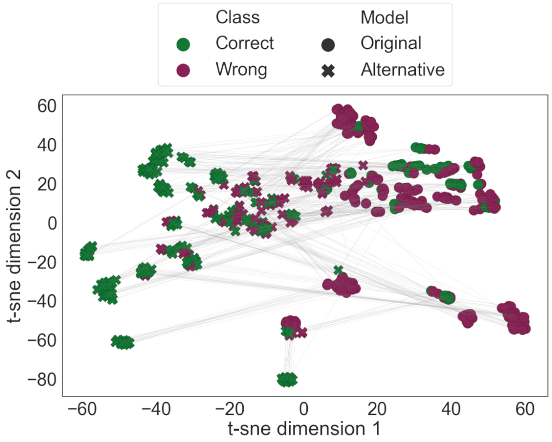

Fig. 6. A t-sne representation of all predictions by both the instance split alternative model and the original model, colored by whether or not the predictions were correct.

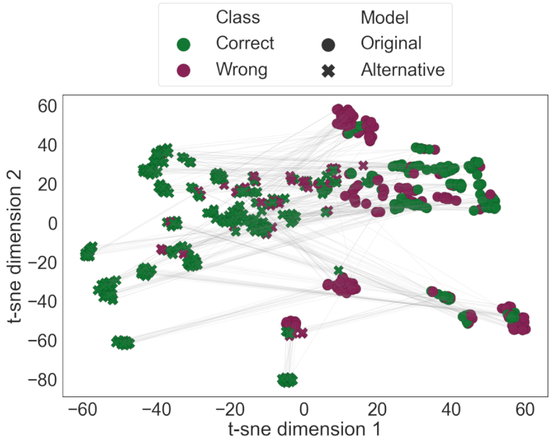

Fig. 7. A t-sne representation of all predictions by both the instance split alternative model and the original model, colored by whether or not the predictions were correct using the multi-label evaluation.

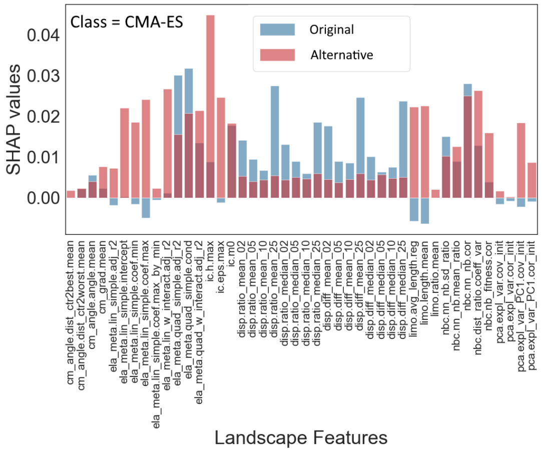

Fig. 8. Mean SHAP values of the original and the alternative models on the correct predictions of instances with the class CMA-ES.

clusters in the center contain a mix of correct and incorrect predictions.

Figure 7 shows similar data as Figure 6, but by following an alternate evaluation methodology described in [10] where a prediction is considered correct as long as the predicted algorithm achieves performance that similar to the absolute best performing algorithm from the perspective of statistical significance, as determined by a Wilcoxon statistical test. Such an evaluation naturally improves the results of the models. If we look at the data from the original model, it particularly improves its results in the large central group. These findings can also be observed in the alternative model. From this, we can conclude that the two large central groups contain instances that are solved well by multiple algorithms. On the other hand, the performance of the smaller outlying clusters is not improved for the original model.

Finally, in order to understand which specific landscape features the models consider important, Figure 8 shows the aggregated mean SHAP values of correct predictions of the original and the alternative models for the class CMA-ES. Each bar represents the mean individual contribution of a specific landscape feature toward the model's decision to classify the problem with the class CMA-ES. Plots for other classes are not included here due to space constraints but are included in the supplemental data of the paper. These other plots all show similar broad results, i.e., that the two models consider vastly different landscape features, but the exact landscape features considered important vary depending on the specific class.

We can see that there are large differences between the original and the alternative models, which is unsurprising given the poor performance of the original model. For the CMA-ES class, the landscape feature *ic.h.max* is the most important feature for the alternative model, while it is not considered important by the original model. The feature *ic.eps.max* is likewise important for the alternative model, but on average gives negative contributions to the original model. A similar result can be observed for the landscape features *lim* *o.avg_length.reg* and *limo.length.mean*. On the other hand, the original model considers the landscape features *disp.ratio _median_25*, *disp.diff_median_25*, and *disp.diff_mean_25* as much more important.

V. Conclusions and Future Work

In this paper, we have presented a case study of how SHAP values can be used to extract more information about an automated algorithm selection model based on landscape features. We used these values to compare two different automated algorithm selection models. The first was a model that was trained on one set of data, and tested on a different unrelated set of data. This model performed poorly. The second was a model that was both trained and tested on the same set of data using cross-validation. This model performed well.

We show that the explanations of both of these models separate the testing instances into multiple smaller clusters that correspond to individual classes. However, it appears that the original model does not distinguish between different groups of instances in the same way that the alternative model can, which leads to mispredictions of the problems belonging to these groups. Interestingly, the original model is able to correctly predict a small number of problems that the alternative model cannot. In addition, both models also contain one large cluster which seems to primarily correspond to instances that can be solved equally well by multiple algorithms. These findings reinforce the findings of the paper that originally introduced the model, as they show that training data that is different from the testing data in terms of how the problems are designed and the information that they contain in terms of landscape features produces poor model explanations. However, this paper represents only a preliminary investigation, and more work is needed in order to determine why certain features showed such large differences between the two models

Overall, these findings show the machine learning models that were used in our previous study could not generalize the information provided by the landscape features to the extent that it could have been used for performance prediction between completely different types of problems because the two models relied on entirely different features to make their predictions. A shift to different problem representations, or to different machine learning models could potentially improve these results.

In this paper, we only examined two out of five models that we presented in our original paper [10], as we were interested in comparing the worst-performing model with the best-performing model. This analysis could be expanded to other three models as well. In addition, metrics other than SHAP could be used for the analysis. As noted above, different problem representation methods might also provide better results and would require additional analysis to explain their usefulness, as well as to explain the complementarity between these new representation methods and existing landscape features. Other problem-generation methods should also be analyzed, as it is possible that the specific artificial problems used in this study contained very different problem landscapes from the BBOB problems. Finally, we have used a limited set of 44 landscape features which were shown to be promising in our prior work, but it is possible that the inclusion of additional features, including those that are not available in the flacco library, could produce different and more generalizable results.

References

[1] J. R. Rice, “The algorithm selection problem,” in *Advances in computers*. Elsevier, 1976, vol. 15, pp. 65–118.

[2] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann, “Automated algorithm selection: Survey and perspectives,” *Evolutionary computation*, vol. 27, no. 1, pp. 3–45, 2019.

[3] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, “Exploratory landscape analysis,” in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 829–836.

[4] P. Kerschke and H. Trautmann, “Comprehensive feature-based landscape analysis of continuous and constrained optimization problems using the r-package flacco,” in *Applications in Statistical Computing – From Music Data Analysis to Industrial Quality Improvement*, ser. Studies in Classification, Data Analysis, and Knowledge Organization, N. Bauer, K. Ickstadt, K. Lübke, G. Szepannek, H. Trautmann, and M. Vichi, Eds. Springer, 2019, pp. 93 – 123. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-25147-5_7

[5] ——, “Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning,” *Evolutionary computation*, vol. 27, no. 1, pp. 99–127, 2019.

[6] N. Hansen, S. Finck, R. Ros, and A. Auger, “Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions,” *[Research Report] RR-6829, INRIA. 2009. inria-00362633v2*, 2009.

[7] B. Lacroix and J. McCall, “Limitations of benchmark sets and landscape features for algorithm selection and performance prediction,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2019, pp. 261–262.

[8] U. Škvorc, T. Eftimov, and P. Korošec, “Understanding the problem space in single-objective numerical optimization using exploratory landscape analysis,” *Applied Soft Computing*, vol. 90, p. 106138, 2020.

[9] A. Kostovska, A. Jankovic, D. Vermetten, J. de Nobel, H. Wang, T. Eftimov, and C. Doerr, “Per-run algorithm selection with warm-starting using trajectory-based features,” in *Parallel Problem Solving from Nature–PPSN XVII: 17th International Conference, PPSN 2022, Dortmund, Germany, September 10–14, 2022, Proceedings, Part I*. Springer, 2022, pp. 46–60.

[10] U. Škvorc, T. Eftimov, and P. Korošec, “Transfer learning analysis of multi-class classification for landscape-aware algorithm selection,” *Mathematics*, vol. 10, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2227-7390/10/3/432

[11] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Proceedings of the 31st international conference on neural information processing systems*, 2017, pp. 4768–4777.

[12] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[13] Y. Tian, S. Peng, X. Zhang, T. Rodemann, K. C. Tan, and Y. Jin, “A recommender system for metaheuristic algorithms for continuous optimization based on deep recurrent neural networks,” *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 1, pp. 5–18, 2020.

[14] A. Nikolikj, R. Trajanov, G. Cenikj, P. Korošec, and T. Eftimov, “Identifying minimal set of exploratory landscape analysis features for reliable algorithm performance prediction,” in *2022 IEEE Congress on Evolutionary Computation (CEC)*, 2022, pp. 1–8.

[15] R. Trajanov, S. Dimeski, M. Popovski, P. Korošec, and T. Eftimov, “Explainable landscape-aware optimization performance prediction,” in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, pp. 01–08.

[16] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, “From local explanations to global understanding with explainable ai for trees,” *Nature Machine Intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.

[17] S. M. Lundberg, “shap,” https://github.com/slundberg/shap, 2018.