On Generalization of ELA Feature Groups

GAŠPER PETELIN, Computer Systems Department
Jožef Stefan Institute
Jožef Stefan International Postgraduate School, Slovenia
GJORGJINA CENIKJ, Computer Systems Department
Jožef Stefan Institute
Jožef Stefan International Postgraduate School, Slovenia

Algorithm selection, i.e., selecting the most suitable algorithm for a specific problem, is a vital task in continuous black-box optimization. A popular strategy used to address this task is to characterize optimization functions using Exploratory Landscape Analysis (ELA) features, which are then utilized to train a machine learning meta-model to select the appropriate algorithm for that function. A significant challenge with meta-models trained on current benchmarks is their often restricted ability to effectively generalize to new functions, limiting their practical application. In this study, we investigate which ELA feature groups are the best at generalizing to previously unseen functions when performing algorithm selection. Using the Comparing Continuous Optimizers functions, novel functions are generated through affine recombinations of existing functions. For each ELA feature group, a meta-model is developed on these functions, enabling it to rank various optimization algorithms. Subsequently, these trained meta-models are assessed using functions that are increasingly out-of-distribution to what was observed during training. We show that most ELA feature groups do not generalize well to out-of-distribution functions, implying reduced effectiveness of selecting algorithms for unfamiliar functions. In such situations, meta-models using different ELA features for algorithm ranking often do not outperform basic predictions based on average ranks.

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 selection, exploratory landscape analysis, evolutionary computation, black-box optimization

ACM Reference Format:
Gašper Petelin and Gjorgjina Cenikj. 2024. On Generalization of ELA Feature Groups. In *Genetic and Evolutionary Computation Conference (GECCO '24 Companion), July 14–18, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3638530.3654124

1 INTRODUCTION

A major challenge for practitioners in the optimization field is choosing the optimal algorithm for a specific objective function. Ideally, this selection should be automatic, aiming to identify the best-performing optimization algorithm tailored to the function at hand, a process known as automatic algorithm selection (AS). The AS task can be addressed by training a machine learning meta-model, which, provided with a set of features representing properties of the function to be solved, can determine the most appropriate optimization algorithm to use [11]. In the single-objective optimization domain, many features describing properties of optimization functions have already been proposed [6, 9]. However, the most widely used and known features are Exploratory Landscape Analysis (ELA) [4] features.

In the current literature, AS is primarily conducted using the established COmparing Continuous Optimizers (COCO) [3] benchmark suite. The COCO benchmark suite includes 24 classes of single-objective optimization problems. Each problem class within this benchmark contains several instances, which are derived by applying different transformations like scaling or shifting to the base problems. Problem instances within the same problem class typically share highly similar characteristics. However, when using the COCO benchmark for AS and performance prediction there are substantial differences in how evaluation is performed. Two commonly used approaches are referred to as “leave-one-instance-out" (LOIO) and “leave-one-problem-out" (LOPO). LOIO is a forgiving evaluation strategy where all problem classes are present in the training set, but only a few instances from each class are used for evaluation. This means the meta-model encounters functions with similar characteristics during training and evaluation. On the other hand, the LOPO validation strategy is much more challenging due to the diversity of problems in the COCO benchmark. During the evaluation phase, we may come across functions where a problem with similar characteristics has not been previously observed during training. Therefore, choosing which strategy to use can significantly influence the results [11].

Additionally, AS models trained using ELA features have been shown to have poor generalization capabilities to functions that were not observed during the training [5, 10]. Consequently, there is a need to determine which individual ELA feature groups contribute to extracting information used to construct meta-models that generalize to new problems and how far out of distribution can we go before ELA features fail to provide quality information used for AS.

Our contribution: In our paper we provide the answer to the following research question related to the generalizability of ELA feature groups: What is the effectiveness of individual ELA feature groups in generalizing performance compared to others? To what extent do individual feature groups demonstrate generalizability, and are there groups of ELA features that underperform relative to the single best solver (SBS) baseline?

Our research shows that using the LOIO methodology, most ELA feature groups outperform the SBS in ranking optimization algorithms effectively. However, the predictive power of meta-models declines notably when faced with new problems generated through affine transformations not encountered during training, particularly under the more rigorous LOPO strategy. In the more challenging evaluation scenario involving out-of-distribution (OOD) problems, hardly any ELA feature group significantly surpasses the simple baseline model.

Outline: The paper is structured as follows: Section 2 outlines the methodology, covering feature extraction and algorithm optimization. Section 3 discusses results, emphasizing algorithm generalization. Finally, Section 4 wraps up with concluding remarks.

Reproducibility: The experiments conducted can be replicated using the code available in the Gitlab repository, accessible at https://anonymous.4open.science/r/affine-ranking/README.md.

2 METHODOLOGY

In this section, we provide a detailed overview of the methodology for assessing each ELA feature group's ability to generalize to unseen instances. It involves the following steps: ***i)*** Create new problems by applying affine recombinations to the original 24 problems from the original COCO dataset. ***ii)*** Calculate ELA features and determine the ranks of optimization algorithms within the algorithm portfolio for all the newly created problems. ***iii)*** Split the dataset into two parts: one comprises a single COCO problem and all the affine functions derived from it, forming the test set, while the remaining 23 COCO functions and their affine combinations make up the training set. With this, the test set contains a set of progressively more difficult and OOD functions. ***iv)*** Train meta-models using individual ELA feature groups to predict the rankings of algorithms. ***v)*** Evaluate meta-models created using different ELA feature subsets on how well they generalize on the test set with completely new problem classes.

To illustrate how validation is conducted, let's consider a scenario where the first COCO problem class is chosen for the test data while the others are assigned to the training data. In this setup, problem instances 1-5, generated as affine combinations of the 23 problem classes in the training set, are utilized to train the meta-model. Subsequently, problem instances 6-10, where one of the base functions belongs to problem class 1, are employed to evaluate the performance of the meta-model. This approach ensures that test instances can progressively represent more OOD objective functions. When problem instances 6-10 are created from the 23 problem classes in the training set, the validation strategy resembles LOIO. Conversely, when they are the furthest from the training distribution, the validation strategy is similar to LOPO. This process is repeated for every ELA feature group and every COCO problem class. This ensures that each individual ELA feature group described in the next subsection is tested on its generalizability on each of the 24 problem classes. The following subsections provide a more detailed description of each component within the methodology.

2.1 Portfolio of Optimization Algorithms

In our research, we establish a portfolio comprising k optimization algorithms defined as $\mathcal{A} = \{a_1, \ldots, a_k\}$. The portfolio of algorithms in our study includes five algorithms sourced from the *pymoo* [1] framework version *0.5.0*. We evaluate the following optimization algorithms: Genetic Algorithm (GA); Differential Evolution (DE); Particle Swarm Optimization (PSO); Evolutionary Strategy (ES); Covariance Matrix Adaptation Evolution Strategy (CMA-ES). The algorithm performance is captured by comparing the best objective function value found by each algorithm after this execution budget, and assigning an integer rank in the range [1,5] to each algorithm.

2.2 Problem Creation and Feature Representation

As proposed in [2], new problems can be created by combining existing COCO problems, resulting in a problem set that is more diverse than the original 24 COCO problem classes. In addition to enhancing diversity, this approach also provides refined control over the generation of new problems, which can be designed to closely resemble existing problems. In our paper, we specifically employ the formulation from [12], where two COCO functions are combined in the following way:

$$\begin{aligned} &F(P_{i,m}, P_{j,n}, \alpha)(x) = \\ &\quad \exp(\alpha\, log(P_{i,m}(x) - P_{i,m}(O_{i,m})) + \\ &\quad\ (1-\alpha)\, log(P_{j,n}(x - O_{i,m} + O_{j,n}) - P_{j,n}(O_{j,n}))) \end{aligned} \tag{1}$$

In the definition provided, $P_{a,b}$ denotes the b-th instance of the a-th COCO problem class, with its optimum at $O_{a,b}$. All optimal solutions yield a value of zero ($P_{a,b}(O_{a,b}) = 0.0$). The parameter α controls the combination of functions, enabling precise control over the similarity between the new function and its parent functions.

To represent objective functions as numerical features, we employ ELA features [4], derived from fundamental metrics calculated on sampled points and their objective function values. The full ELA feature set is comprised of various individual features, which can be split into multiple feature groups. In our case, we analyze these specific feature groups: *cm_angle* (9), *cm_conv* (5), *cm_grad* (3), *disp* (17), *ela_distr* (4), *ela_level* (10), *ela_meta* (10), *ic* (6), *limo* (9), *nbc* (6) and *pca* (9). The parentheses indicate the number of features in each individual feature group.

Moreover, scaling samples can improve feature informativeness [7]. We employ min-max linear scaling, adjusting candidate solution values (x) within the range of -5 to 5 to 0 to 1, and function values ($f(x)$) are scaled to be between 0 and 1. For completes we also include a feature group that contains all the ELA features *ela*, its scaled counterpart *scaled_ela*, and concatenation of both groups marked with *concat_ela*. When computing ELA features, we utilize the *pflacco* library [8] with default hyperparameter values.

2.3 Meta-models and Error Metric

We employ two distinct meta-models to relate ELA feature groups to optimization algorithm ranks. These meta-models are designed as multi-target regression models, predicting rankings for all five algorithms simultaneously. Inputs are ELA feature groups, and outputs are five numerical values representing algorithm rankings. The meta-models utilized are: ***random-forest***: A random forest, an ensemble regression technique, constructs multiple decision trees during training and outputs their mean prediction as the final result. In this study, we used the implementation from *scikit-learn* version 1.3.2 with the default hyperparameters. ***mean***: This algorithm computes the mean ranking across the training set and makes predictions based on this average, ignoring individual problem features. It serves as the "best single solver" in ranking, reflecting the dataset's overall average rank.

This paper employs the following metric to assess the performance of meta-models, incorporating defined lower and upper bounds that captures the relationships in solution quality across all algorithms within the portfolio. We use the Pairwise Ranking Error (PRE) to assess and contrast the quality of various rankings. We use the PRE between two ranked lists as defined in [5]. PRE is determined in the following way:

$$PRE = \frac{1}{2\binom{|\mathcal{A}|}{2}} \sum_{a_i \in \mathcal{A}} \sum_{a_j \in \mathcal{A}} r(a_j, a_i) \tag{2}$$

$$r(a_j, a_i) = \begin{cases} 0 & \text{if } r_p(a_i, a_j) = r_g(a_i, a_j) \\ 1 & \text{if } r_p(a_i, a_j) \neq r_g(a_i, a_j) \end{cases} \tag{3}$$

In this scenario, a_i and a_j are algorithms from the portfolio with functions $r_p(a_i, a_j)$ and $r_g(a_i, a_j)$ describing the differences in ranks of optimization algorithms. The function $r_p(a_i, a_j)$ returns -1, 0, or 1, depending on whether the predicted rank of algorithm a_i is less than, equal to, or greater than the rank of algorithm a_i, respectively. Likewise, the function $r_g(a_i, a_j)$ works in a similar manner, with the key distinction that it considers the rankings of algorithms according to the ground truth. This implies that it yields -1, 0, or 1 based on the ground truth rankings, which are determined by executing the optimization algorithms 30 times.

The PRE assesses ranking algorithm effectiveness by measuring how accurately they predict the order of items compared to a true ranking. A PRE of 0.0 means the predicted order matches the true order perfectly. If rankings are random, the error is 0.5, showing half are correct. A PRE of 1.0 means all pairs are misordered, indicating the worst scenario where the predicted sequence is the opposite of the truth. Consider a set of optimization algorithms - GA, PSO, DE, CMA-ES, and ES - evaluated on a specific problem instance. After 30 runs, their average ranks are: [4.8, 2.23, 2.7, 1.06, 4.21]. CMA-ES ranks highest while GA performs the worst. Now, a regression meta-model predicts ranks of [4.6, 2.6, 2.39, 1.2, 4.21]. Despite most ranks differing from ground truth, only one pair of algorithms is predicted incorrectly: DE is wrongly placed above PSO. This discrepancy yields a PRE of 0.1, indicating one out of ten pairs is ordered incorrectly.

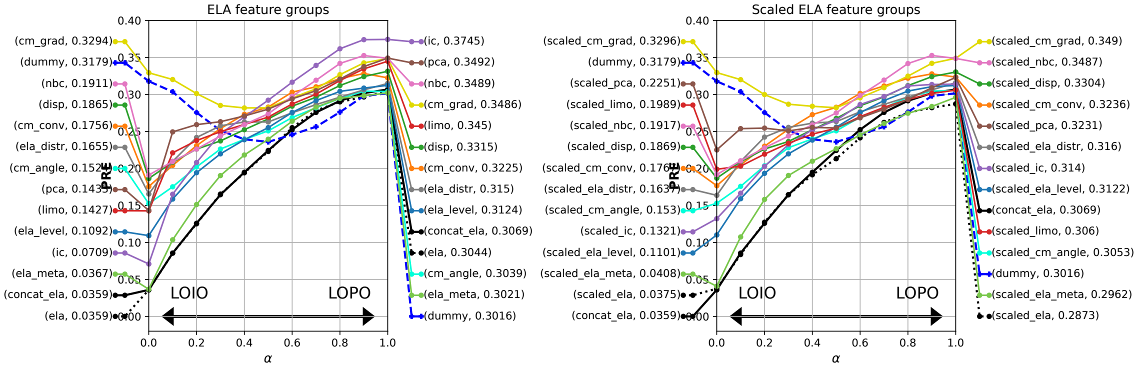

Fig. 1. Predictive power of individual ELA feature groups using the PRE metric averaged across all problem classes. The figures represent both the non-scaled and scaled versions. To enhance readability, the border of the plot displays the order of individual ELA feature groups alongside the precise PRE they attain in both LOIO and LOPO methodologies.

3 RESULTS

In this section, we assess how effectively specific groups of ELA features predict algorithm performance on increasingly OOD problems, following the methodology outlined in Section 2.

The experimental setup utilizes a fixed budget of $2000D$ function evaluations, where $D = 5$, and a population size of 20, focusing on objective functions within this dimensionality due to space and time limitations. ELA features are computed from a sample size of $500D$ obtained through LHS. To account for the stochastic nature of optimization algorithms, each algorithm in the portfolio undergoes 30 runs. For every problem class, 10 instances are employed, with α varied from 0.0 to 1.0 in increments of 0.1.

Figure 1 demonstrates the generalizability of distinct ELA feature groups in ranking algorithms when encountering progressively unfamiliar problems. In scenarios involving previously encountered problem classes ($\alpha = 0$ similar to LOIO), the *random-forest* meta-model can effectively rank optimization algorithms, surpassing the mean baseline meta-model with nearly all individual ELA feature groups. Notable exceptions include the *cm_grad* feature group and its scaled variant, both of which achieve slightly higher PRE than the baseline. Conversely, the most effective feature groups identified using LOIO methodology are, predictably, the comprehensive feature sets encompassing all ELA features. These are followed closely by feature sets such as *ela_meta*, *ic*, *ela_level*, and their scaled counterparts. However, this scenario mainly shows that the meta-model can memorize the association between features and performance, but does not guarantee this relation applies to new, unseen problems.

As the values of α increase, the predictive performance of all feature groups deteriorates. When $\alpha = 0.4$, individual meta-models are assessed on artificial problems, consisting of 0.6 of previously encountered problems and 0.4 of entirely new problems. In this scenario, the performance of most meta-models is inferior to that of the *mean* baseline meta-model. The only exemption from this rule applies to meta-models comprising all types of features (scaled, unscaled, or concatenated) or meta-models relying solely on the *ela_meta* feature group. Except for the *ela_meta* feature group, all other feature groups have a PRE ranging from 0.23 to 0.28, falling behind the baseline's PRE of 0.22. This suggests that while the baseline misranks 22% of the optimization algorithms, the feature-based model may misrank up to 28% of them. At $\alpha = 0.4$, most feature-based meta-models exhibit poor generalization capabilities, often underperforming when compared to the *mean* meta-model.

When considering the range of α values between 0.6 and 0.8, the comparison of feature-based meta-models with the *mean* baseline reveals an even more unfavorable situation for the feature-based models. Within the specified range, when ELA groups are computed on unscaled samples, the baseline consistently outperforms the feature-based models. When considering scaled ELA feature groups, similar outcomes are observed, with scaled *ela_meta* being the only feature group that exhibits performance close to the *mean* baseline. This further illustrates the challenge of achieving superior performance compared to the SBS.

When we evaluate the meta-model on completely new problems at $\alpha = 1$ that were not combined with any of the problems in the training set (similar to LOPO validation), the PRE of most meta-models increases further. In this case, only two meta-models slightly outperform the baseline. These two models are the ones that utilize either all features or *ela_meta* features both with prior scaling of the sample points.

A conclusion that can be drawn is that, with the current problem set and feature representation, the generalization of a feature-based meta-model is relatively poor, and they cannot perform substantially better than the simple baseline. The results, along with other literature [11], underscore notable differences between LOPO and LOIO strategies. In the AS model assessment, practitioners often favor the less changing LOIO approach. Yet, this method could misleadingly indicate the superior performance of meta-models, masking their difficulty in generalizing to new objectives. Consequently, their effectiveness may sharply decline when applied to new problems. For example, the *ic* feature set is highly influential in LOIO validation but underperforms a basic baseline in LOPO validation. It is crucial to note that feature significance depends on the validation strategy chosen. Important features in LOIO validation may not hold the same weight in LOPO, and vice versa.

4 CONCLUSION

This study investigates the generalizability of individual ELA feature groups when predicting the rankings of optimization algorithms. More precisely, we examine the abilities of a meta-model to effectively rank optimization algorithms, particularly when confronted with progressively OOD problems. It is crucial to explore this aspect for various reasons, including gaining insights into functions where meta-models may fail and to what extent, determining which feature groups exhibit the ability to generalize to new functions, and assessing whether such meta-models can surpass basic baseline models in performance. The findings indicate that employing the LOIO strategy with similar training and test data functions leads to strong performance across meta-models utilizing various ELA feature groups. This suggests that for previously encountered problems, the meta-model faces no difficulties in ranking optimization algorithms. Nevertheless, as meta-models confront increasingly OOD functions, their performance begins to decline. In the most extreme scenario, during LOPO validation, the majority of meta-models struggle to surpass the basic baseline in correctly ranking optimization algorithms. Furthermore, none of the individual ELA feature groups exhibit a substantial advantage over the baseline.

ACKNOWLEDGMENTS

This work was supported financially by the Slovenian Research Agency through core funding (No. P2-0098), as well as young researcher grants awarded to GP (No. PR-11263) and GC (No. PR-12393).

REFERENCES

[1] J. Blank and K. Deb. 2020. pymoo: Multi-Objective Optimization in Python. *IEEE Access* 8 (2020), 89497–89509.

[2] Konstantin Dietrich and Olaf Mersmann. 2022. Increasing the diversity of benchmark function sets through affine recombination. In *International Conference on Parallel Problem Solving from Nature*. Springer, 590–602.

[3] 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.

[4] 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.

[5] Gašper Petelin and Gjorgjina Cenikj. 2023. How Far Out of Distribution Can We Go With ELA Features and Still Be Able to Rank Algorithms?. In *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 341–346.

[6] Gašper Petelin, Gjorgjina Cenikj, and Tome Eftimov. 2023. TinyTLA: Topological landscape analysis for optimization problem classification in a limited sample setting. *Swarm and Evolutionary Computation* (2023), 101448.

[7] Raphael Patrick Prager and Heike Trautmann. 2023. Nullifying The Inherent Bias Of Non-Invariant Exploratory Landscape Analysis Features. In *Applications of Evolutionary Computation* (Brno, Czech Republic). Springer-Verlag, Berlin, Heidelberg, 411–425. https://doi.org/10.1007/978-3-031-30229-9_27

[8] Raphael Patrick Prager and Heike Trautmann. 2023. Pflacco: Feature-based landscape analysis of continuous and constrained optimization problems in Python. *Evolutionary Computation* (2023), 1–25.

[9] Raphael Patrick Prager, Moritz Vinzent Seiler, Heike Trautmann, and Pascal Kerschke. 2021. Towards Feature-Free Automated Algorithm Selection for Single-Objective Continuous Black-Box Optimization. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. 1–8. https://doi.org/10.1109/SSCI50451.2021.9660174

[10] Urban Škvorc, Tome Eftimov, and Peter Korošec. 2022. Transfer learning analysis of multi-class classification for landscape-aware algorithm selection. *Mathematics* 10, 3 (2022), 432.

[11] Ryoji Tanabe. 2022. Benchmarking Feature-Based Algorithm Selection Systems for Black-Box Numerical Optimization. *IEEE Transactions on Evolutionary Computation* 26, 6 (2022), 1321–1335.

[12] Diederick Vermetten, Furong Ye, and Carola Doerr. 2023. Using Affine Combinations of BBOB Problems for Performance Assessment. *arXiv preprint arXiv:2303.04573* (2023).