

This is the Author-Accepted Version of the paper:

Nikolikj, Ana, Gabriela Ochoa, and Tome Eftimov. "Customized Exploration of Landscape Features Driving Multi-Objective Combinatorial Optimization Performance." Proceedings of the Genetic and Evolutionary Computation Conference. 2025.

© ACM 2025. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in GECCO '25 Companion: Proceedings of the Genetic and Evolutionary Computation Conference Companion, <https://doi.org/10.1145/3712256.3726334>.



Customized Exploration of Landscape Features Driving Multi-Objective Combinatorial Optimization Performance

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

GABRIELA OCHOA, University of Stirling, United Kingdom

TOME EFTIMOV, Jozef Stefan Institute, Slovenia

We present an analysis of landscape features for predicting the performance of multi-objective combinatorial optimization algorithms. We consider features from the recently proposed compressed Pareto Local Optimal Solutions Networks (C-PLOS-net) model of combinatorial landscapes. The benchmark instances are a set of ρ mnk-landscapes with 2 and 3 objectives and various levels of ruggedness and objective correlation. We consider the performance of three algorithms – Pareto Local Search (PLS), Global Simple EMO Optimizer (GSEMO), and Non-dominated Sorting Genetic Algorithm (NSGA-II) – using the resolution and hypervolume metrics. Our tailored analysis reveals feature combinations that influence algorithm performance specific to certain landscapes. This study provides deeper insights into feature importance, tailored to specific ρ mnk-landscapes and algorithms.

CCS Concepts: • **Theory of computation** → **Design and analysis of algorithms**; • **Applied computing** → **Multi-criterion optimization and decision-making**; • **Computing methodologies** → **Supervised learning**; **Cluster analysis**.

Additional Key Words and Phrases: landscape analysis, multi-objective combinatorial optimization, feature importance, algorithm footprints

1 INTRODUCTION

In black-box optimization (BBO), selecting the best algorithm for a given problem is challenging due to the abundance of options. This has led to research in meta-learning for automated algorithm selection, which uses machine learning (ML) models to predict algorithm performance based on problem properties (landscape features) [20, 22].

Extensive research in single-objective optimization (SOO) has led to numerous methods for extracting landscape features for continuous SOO problems, including Exploratory Landscape Analysis (ELA), Topological Landscape Analysis (TLA), and deep learning-based approaches (TransOpt, DeepELA, Doe2Vec), as summarized in a recent survey [4]. These techniques are evaluated for problem classification and automated algorithm selection/configuration. Other studies compare and combine features to improve predictive models and identify truly impactful features, avoiding the introduction of less meaningful ones [5, 35].

Compared to SOO, multi-objective optimization (MOO) is relatively under-explored. In continuous MOO, various techniques have been studied, including cost landscapes [14], gradient field heatmaps [18], local dominance landscapes [13], and landscape plots illustrating optimal trade-offs [34]. For combinatorial MOO, tools such as Pareto Optimal Solution Networks (PLOS-net) [24] and Pareto Local Optima Networks (PLON) [12] have proven valuable for analyzing optimization problem properties. More recently, funnel features [30] based on solution rankings using non-dominated sorting, along with the compressed Pareto Local Optimal Solution Network (C-PLOS-net) [25], have been proposed and studied to predict the performance of established local search and evolutionary algorithms. The main challenge is the lack of evaluation of feature groups to determine whether their combinations can yield improved results. Classical correlation methods or global feature importance often overlook their impact on specific problem landscapes.

Our contribution: We present a customized analysis of the importance of landscape features by tailoring it to specific ρ mnk-landscapes and combinatorial MOO algorithm combination. We consider the recently proposed 10 C-PLOS-net features in addition to the 24 PLOS-net features from [24], and 7 *funnel* metrics from [31], extracted from a set of ρ mnk-landscapes with 2 and 3 objectives and various levels of ruggedness and objective correlation.

The combined feature groups are used to predict the performance of three algorithms: PLS [32], GSEMO [21], and NSGA-II [10]). Our study offers tailored insights into which landscape features are the most important for the performance of specific landscapes, for different MOO algorithms. Additionally, we identify landscape feature combinations that make certain ρ mnk-landscapes particularly hard for some algorithms. The analysis is conducted twice, using two distinct performance metrics: resolution and hypervolume. In summary, unlike previous studies that evaluate global feature importance across all algorithms and different ρ mnk-landscapes, this analysis is tailored to specific ρ mnk-landscapes, algorithm and performance metric combination. This analysis offers deeper insights into the relevance of various feature groups and helps identify features that make certain landscapes particularly challenging for specific algorithms to solve. The results show that the number of interactions is the most influential benchmark parameter affecting the hardness of problem instances for the algorithms. It also shows that combining two feature sets is crucial, as they complement each other and are key for analyzing specific landscapes with different algorithms.

Outline: Section 2 reviews the background on combinatorial MOO and summarizes previous landscape analyses. Section 3 briefly revisits the methodology employed for the analysis. Section 4 outlines the experimental design and data used. In Section 6, we discuss the strengths and limitations of the analysis. Finally, Section 7 concludes the paper and offers directions for future research.

Data and code availability: The data and code used in this study are available at [29].

2 BACKGROUND

Here, we briefly explain combinatorial MOO and the ρ mnk-landscape benchmark suite. Then, we provide an overview of related work on landscape analysis for the benchmark suite.

2.1 Combinatorial multi-objective optimization

Combinatorial MOO involves finding the optimal solution to optimization problems characterized by decision variables that take on discrete values and multiple conflicting objectives [7]. Solving MOO problems requires identifying trade-offs among objectives and determining a set of non-dominated solutions (i.e. Pareto optimal solutions). The collection of Pareto optimal solutions is known as the Pareto set, and its representation in the objective space is referred to as the Pareto front. MOO algorithms aim to identify or provide a good approximation of the Pareto set/front.

2.2 ρ MNK Landscapes

ρ mnk-landscapes [38] is a benchmark suite for combinatorial MOO that consists of multi-objective multi-modal combinatorial problems with objective correlation, extending the the benchmark suite of single-objective and nk-landscapes with independent objectives [1, 17, 19]. The solution space is $X = \{0, 1\}^n$, where n denotes the problem size or the total number of binary decision variables. The objective vector $f = (f_1, \dots, f_i, \dots, f_m)$ maps $f: \{0, 1\}^n \rightarrow [0, 1]^m$, where m is the number of objectives and each objective f_i is maximized. The value $f_i(x)$ for a solution $x = (x_1, \dots, x_j, \dots, x_n)$ is the average of the contributions from each variable x_j . The contribution of x_j depends on its value and the values of $k < n$ other variables, selected uniformly at random from the $(n - 1)$ variables excluding x_j [17]. By adjusting k , the problem landscape transitions from smooth to rugged. In ρ mnk-landscapes, the contribution values follow a multivariate uniform distribution, with the correlation coefficient $\rho > \frac{-1}{m-1}$ controlling the objectives' conflict level [38]. Positive ρ reduces conflict, while negative ρ increases it. Notably, in the ρ mnk-landscapes suite poses varying challenges for multi-objective algorithms [9, 23].

2.3 Landscape analysis

The study of MOO landscapes is relatively limited compared to the extensive literature on SOO landscapes. In the case of combinatorial MOO, several studies address the landscape analysis of MOO combinatorial ρ mnk landscapes.

In [25], the authors extend the concept of the PLOS-net by proposing the C-PLOS-net. Statistical metrics are calculated for both networks as mathematical objects and to predict algorithm performance with a Random

Forest (RF) [3] and algorithm selection via a Decision Tree (DT). Global importance derived from the RF model reveals the relevance of the features as descriptors of diverse problem landscapes. Additionally, the Pearson coefficient and Spearman’s rank showed that most features were significantly correlated to the performance of the analyzed algorithms. A total of 24 features can be computed for the PLOS-net, while the C-PLOS-net allows for the calculation of 10 additional ones. Recently, five new funnel features have been introduced to characterize the presence of funnels in $\rho m n k$ landscapes. These features are derived from solution ranks using non-dominated sorting – and a variation of the graph-based model for MOO landscapes, the C-PLOS-net. The funnel features, combined with the benchmark parameters ρ , m , n , and k , have been utilized to predict algorithm performance using a Random Forest model and to perform algorithm selection via a Decision Tree. The proposed funnel features effectively capture the landscape’s global structure, exhibit strong correlations with the benchmark parameters, and provide valuable insights into the performance of established multi-objective local search and evolutionary algorithms. The mentioned landscape feature groups, the 24 base PLOS-net features, the additional 10 C-PLOS-net, and the five funnel features have been studied individually, but a comprehensive analysis combining them for algorithm performance prediction is still lacking. Additionally, only the global importance of these features has been assessed, although some features may be more relevant for a specific $\rho m n k$ landscape-algorithm combination.

In [22], dominance and (hypervolume) metric-based landscape features have been proposed. A subsequent study [26] demonstrated that these features correlate with the performance of dominance-based MOEAs, but their correlation with the performance of decomposition MOEAs was found to be less significant. This observation motivated the development of decomposition-based landscape features in [8]. The features are used to predict the ρ and k benchmark parameters of different $\rho m n k$ landscapes and to support algorithm selection among three variants of the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [39], leveraging Random Forest models for these tasks. All the proposed sets of features are interesting and complementary. One advantage of C-PLOS-net features is that they are independent of any particular algorithm or performance metric, at least when they are based on a full enumeration of the search space as is the case of the instances studied in this article.

3 METHODOLOGY

We used the recently proposed *benchmarking algorithm footprints* approach [28] to analyze similarities and differences among SOO algorithms in BBO, following these steps:

(1) Meta-Learning for Performance Prediction: This approach involves training a multi-target regression (MTR) model to simultaneously predict the performance of multiple algorithms based on the landscape features of a problem instance. Each problem instance has a single feature representation, capturing its inherent landscape properties. The MTR model leverages the same feature representation to predict the performance of multiple algorithms [40].

(2) Meta-Representation Creation: This step involves applying an explainability method to the trained MTR model to compute *local feature importance*, resulting in feature importance scores for each problem instance in the dataset. Given that MTR models predict the performance of multiple algorithms, the local feature importance scores are computed for each algorithm separately. The local feature importance scores serve as the basis for constructing *meta-representations*, that show which landscape feature combination is important for the performance of the algorithm, capturing the relationship between the landscape properties of the problem instance and the algorithm behavior. Performing this for all algorithms, results in tailored meta-representations to each problem instance-algorithm pair.

(3) Clustering of Meta-Representations: A clustering algorithm is used to automatically group the meta-representations, and identify clusters of problem instance-algorithm pairs where a similar landscape feature

Table 1. Benchmark parameters for generating ρmnk problem instances, with 10 random instances per parameter combination.

description	values
number of variables	$n = 16$
number of interactions	$k \in \{1, 2, 4\}$
number of objectives	$m \in \{2, 3\}$
objectives correlation	$\rho \in \{-0.4, 0.0, 0.4\}$

combination is important, suggesting similar algorithm behavior, and different algorithm behavior between clusters.

(4) Footprint Analysis and Comparison: By examining the distribution of meta-representations associated with an algorithm across the clusters, we define its unique *algorithm footprint*. By benchmarking the footprints across algorithms, we reveal performance similarities, differences, and easy and hard landscapes for the algorithm portfolio as a whole.

(5) Important Feature Identification: Identify which landscape feature combination is important in each cluster, by aggregating the feature importance score over the meta-representations in the cluster. Local feature importance scores can be analyzed at multiple levels: for a single problem instance to compare algorithms or at the cluster level by aggregating their scores, highlighting landscape features that impact algorithms performance.

4 EXPERIMENTAL DESIGN

Benchmark suite: In the ρmnk -landscapes benchmark suite different problem instances can be generated by varying the benchmark parameters ρ , m , n , and k . Table 1 presents the ranges for the benchmark parameters, with a total of 18 combinations. For each benchmark parameter combination, 10 problem instances are randomly generated, resulting in a total of 180 ρmnk problem instances. The parameter choices enable the analysis of landscapes ranging from smooth to rugged, with two or three objectives that may be conflicting, uncorrelated, or correlated.

Landscape features: The landscape characteristics of ρmnk problem instances are described using 41 features: 24 base PLOS-net features, 10 C-PLOS-net features, and seven funnel features. The funnel features include five from [31] and two new ones: *pos_num* (number of Pareto optimal solutions) and *pos_strength* (the incoming weighted degree to Pareto nodes).

Algorithm portfolio: We used three widely used multi-objective algorithms for solving the ρmnk problem instances: PLS, a local search method that maintains an unbounded archive of mutually non-dominated solutions; GSEMO, a global elitist steady-state evolutionary algorithm; and NSGA-II, a dominance-based evolutionary algorithm.

Performance data: Performance data is taken from a prior study [25], with each algorithm run 30 times per problem instance. PLS is terminated upon reaching a Pareto local optimum set [32], while G-SEMO and NSGA-II are stopped after 10,000 evaluations. NSGA-II uses a population size of 100 (further presented on the plots as *nsa_pop*). Algorithm performance was evaluated using *pareto resolution* (*reso*), representing the proportion of identified Pareto optimal solutions, and *relative hypervolume* (*hv*), which measures the hypervolume of the final archive relative to the exact Pareto front, the reference point was set to the origin). Higher values indicate better performance for both metrics.

MTR predictive model: To find a good MTR model, we conduct experiments with three machine learning algorithms: Random Forest (RF) [3], Neural Network (NN) [15], and Multi-Task Elastic Net (MTEN) [41]. The RF and MTEN models are implemented using the *scikit-learn* library [33], while the NN model is built using the *keras* framework [6], in Python. As a data preprocessing step, each feature has been normalized in the range 0 to

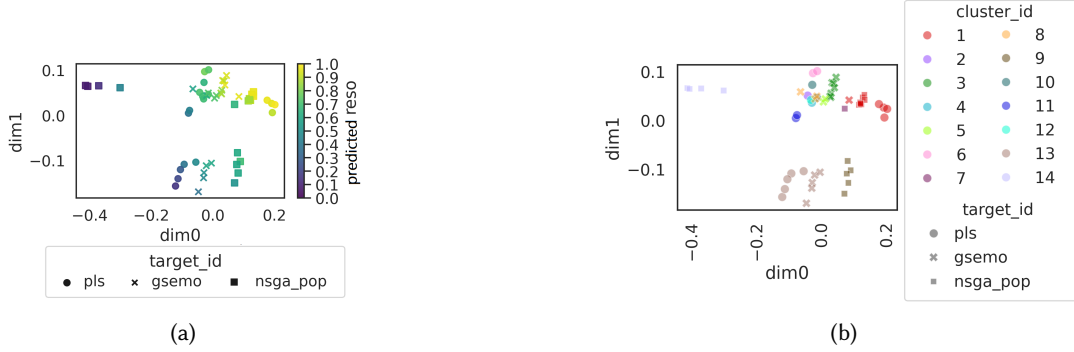


Fig. 1. 2D visualization of ρmnk landscapes using SHAP meta-representation from the test data, showing (a) ground-truth reso performance for three algorithms and (b) clustering results. Axes dim0 and dim1 represent the projected 2D dimensions.

1, to bring the features to the same scale as this can affect the prediction task. The scaling parameters are fitted on the training dataset, and then the test dataset is transformed using the same parameters.

MTR model evaluation: The dataset for training the MTR models includes 180 problem instances, with 10 problem instances generated for each of the 18 benchmark parameter combinations. A dataset split allocates one instance per benchmark parameter combination for testing (18 problem instances) and the remaining 162 problem instances for training. The model performance is evaluated using 9-fold stratified cross-validation on the training set, which allocates one instance per benchmark parameter combination for validation. Models are trained nine times, with eight folds for training and one for validation, and the average validation performance determines the best model. The final model is retrained on the full training set and evaluated on the test set using Mean Absolute Error (MAE) and R-squared (R^2). MAE measures average absolute prediction error, while R^2 quantifies the proportion of variance in the ground-truth labels that is explained by the model's predictions.

MTR model hyper-parameter tuning: Optimal hyperparameters for the MTR models are determined using the Tree-structured Parzen Estimator (TPE) algorithm from the Optuna library [2]. The search, with a budget of 50 trials, begins with 40 Random Search trials to broadly explore the space, followed by 10 TPE trials to focus on promising regions. The hyperparameters yielding the best cross-validation score on the training set are selected.

Feature selection: After hyperparameter tuning, we applied Sequential Forward Feature Selection (SFFS) [11] to potentially further improve MTR model performance. Starting with an empty feature set, SFFS iteratively adds the feature that improves the performance the most. This process continues until all features are added. We applied SFFS to the RF and MTEL models but omitted it for the NN model, as due to high computational cost it is not common practice. The feature subset with the best cross-validation score was selected.

Model Selection: We compared the performance of the MTR models to a baseline which always predicts the average algorithm performance as calculated on the training set. The RF model with hyperparameter tuning and feature selection achieved the lowest cross-validation error and is used for the subsequent analysis. The selected hyperparameters area: using all available features to build the trees, the maximal depth of the trees of seven, 192 trees in the forest, minimum samples to split of three, and minimum samples per leaf of one. The feature selection process identified 23 features as the optimal subset for the RF model.

SHAP local feature importance: Landscape features describe ρmnk landscapes without directly linking to algorithm performance. Using the MTR model with SHAP [37], we calculate local feature importance to show how features influence performance predictions. SHAP values link features to performance via q -dimensional

meta-representations s_1, s_2, \dots, s_q , where q is the number of features. Each ρmnk instance has multiple SHAP meta-representations (e.g., one per algorithm), enabling comparison of feature patterns driving algorithm performance. Calculations use the SHAP library [27] in Python.

Clustering: We use hierarchical clustering [36] to automatically group the SHAP meta-representations of the problem instances. To assess the quality of the clusters, we use the Silhouette coefficient, which ranges from 0 to 1 and evaluates how distinct and compact the clusters are. A higher Silhouette score reflects better-defined clusters. We optimize the hyperparameters of the hierarchical clustering algorithm and choose the configuration that yields the highest Silhouette score. This resulted with the highest silhouette score of 0.63, 14 as an optimal number of clusters, *cosine* as distance metric and *average* linking.

5 RESULTS

We present a visualization of algorithm behavior on the ρmnk problem instances, along with the footprints for each algorithm. Finally, we identify the key landscape features and their combination, tailoring our analysis to each benchmark parameter combination and to the identified performance regions of the footprints.

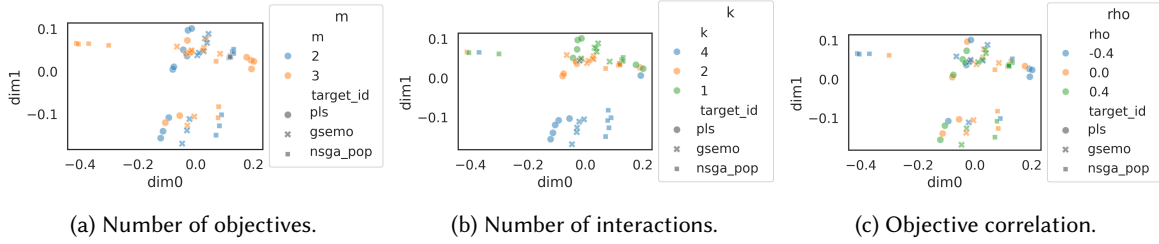


Fig. 2. 2D visualization of ρmnk landscapes using SHAP meta-representation from the test data, showing the benchmark parameters.

5.1 Algorithm footprints

5.1.1 Resolution (*reso*) performance metric. Figure 1 visualizes the meta-representations corresponding to the pairings of the 18 problem instances from the test dataset with the three algorithms. Principal Component Analysis (PCA) [16] is employed as a dimensionality reduction technique to map the high-dimensional meta-representations into two dimensions for visualization. The distinct marker shapes distinguish between meta-representations associated with different algorithms, as detailed in the legend. In Figure 1a the color coding reflects the predicted *reso* performance achieved on the problem instances, where lower values indicate poor performance (dark blue) and higher values indicate good performance (yellow). The *reso* metric measures the proportion of identified Pareto optimal solutions. The plot reveals that meta-representations that correspond to problem instances on which algorithms have poor performance are positioned towards the bottom-left side of the plot, the ones with medium algorithm performance are scattered around the center of the plot, while those corresponding to problem instances on which the algorithms have good performance are located towards the top-right corner. This indicates that based on the meta-representations (i.e. important feature combinations) we can effectively distinguish between different algorithm behaviors.

While the plot reveals roughly three performance groups, we further cluster the meta-representations for automatic grouping. Figure 1b visualizes the clustering of the meta-representations, which resulted into 14 clusters. This suggests the existence of 14 distinct important landscape feature combinations. Combining the two figures, we can see how the position of the clusters corresponds to the predicted *reso* performance. We

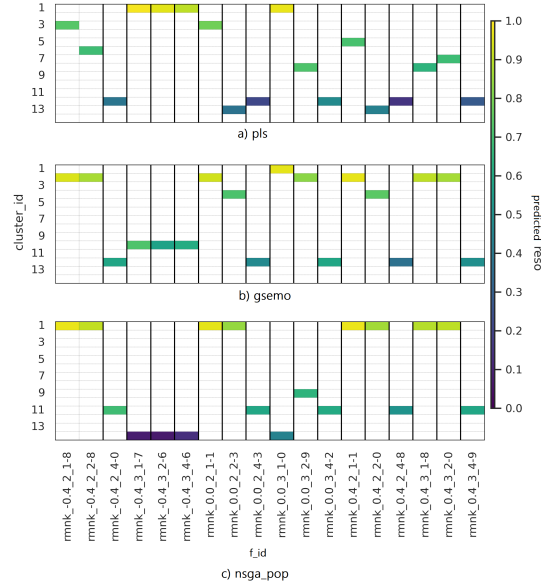


Fig. 3. Algorithm footprints shown as contingency matrices: the y-axis represents 14 performance regions, and the x-axis denotes ρmnk landscapes with varying benchmark parameters. Subfigures show footprints for PLS (top), GSEMO (middle), and NSGA-II (bottom).

enumerated the clusters by average predicted reso performance in the cluster, in decreasing order. The first cluster contains meta-representations with good reso performance on average, while the last 14th cluster contains meta-representations with poor reso performance on average. It is important to note that while some clusters may exhibit similar reso performance (e.g. clusters with IDs one, two, and three all have good performance), different landscape feature combinations are important in those clusters, indicating that both combinations of landscape features can be responsible for predicting similar reso performance.

To examine how the meta-representations correspond to the benchmark parameters used to generate the ρmnk problem instances, the color coding in Figure 2 reflects the benchmark parameters: number of objectives m , number of interactions k , and the objective correlation degree ρ . The results reveal a clear pattern - good reso performance (top-right corner of the visualizations) corresponds to problem instances with fewer interactions ($k = 1$), while bad reso performance corresponds with the increase in the number of interactions ($k = 4$). The other two benchmark parameters do not exhibit a clear pattern in relation to the performance. This suggests that the k benchmark parameter has the most influence on the hardness of the problem instances for the three algorithms.

To formulate the footprints of the three algorithms, we create a contingency matrix that shows how the meta-representations associated with a single algorithm are distributed across the clusters. Figure 3 shows the algorithm footprints, with the rows representing the 14 clusters and the columns representing the ρmnk problem instances generated with different benchmark parameters. The color coding reflects the predicted reso performance. The sub-figures display the footprints for PLS (top), GSEMO (middle), and NSGA-II (bottom). By analyzing each footprint separately, we observe that PLS performs exceptionally well in a specific landscape with low objective correlation ($\rho = -0.4$) and a high number of objectives ($k = 3$). However, as both the objective correlation and the number of interactions increase ($\rho \in \{0.0, 0.4\}$, particularly for $k \in \{2, 3\}$), its performance drastically deteriorates. On the other hand, GSEMO demonstrates good performance in many problem instances

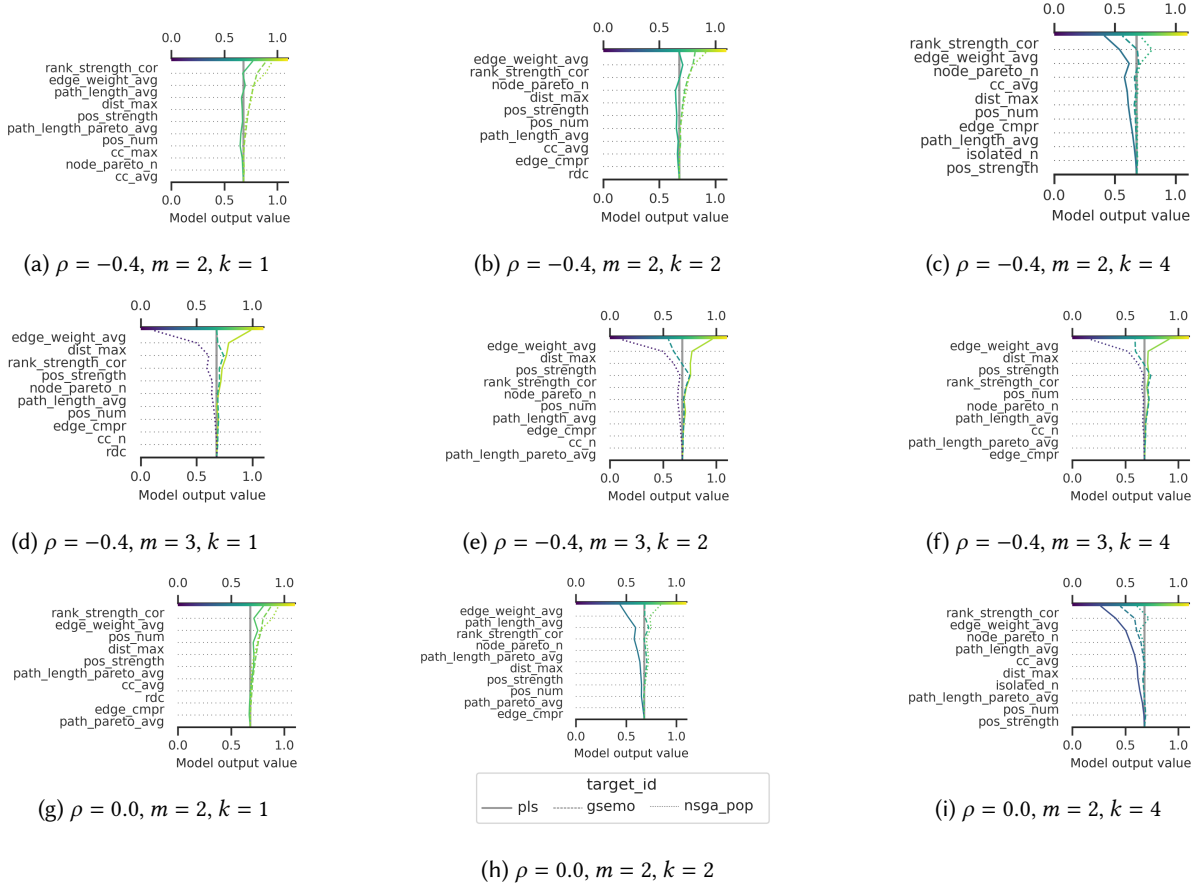


Fig. 4. Key landscape features for ρmnk landscapes are shown, with each plot representing a landscape and three lines for algorithm-feature interactions influencing performance: $\rho \in \{-0.4, 0\}$, $m \in \{2, 3\}$, $k \in \{1, 2, 4\}$. The x-axis shows the predicted $reso$ performance.

and is less sensitive to changes in objective correlation. However, its performance significantly drops when the number of interactions is very high ($k = 4$). Finally, NSGA-II exhibits similar behavior to GSEMO, but seems to be also affected by an increase in the number of objectives, especially when the number of interactions is set to four or for low and medium objective correlations ($\rho \in \{0.0, 0.4\}$).

By comparing the three footprints, we can see to which cluster the same ρmnk problem instance belongs for each of them (thus analyzing column-wise across sub-figures). Focusing on the first three ρmnk problem instances, corresponding to $\rho = -0.4$, $m = 2$, and $k \in \{1, 2, 4\}$, we can observe that NSGA-II efficiently solves these landscapes except in the case of very high number of interactions ($k=4$). A similar pattern can be observed for GSEMO, although driven by a different important landscape feature combination, as the clusters differ between the two algorithms. In contrast, PLS performs worse than both NSGA-II and GSEMO, struggling even for moderate interactions ($k = 2$) and showing significant performance degradation when the number of interactions becomes very high ($k = 4$). Examining the next three columns, corresponding to $\rho = -0.4$, $m = 3$, and $k \in \{1, 2, 4\}$, we

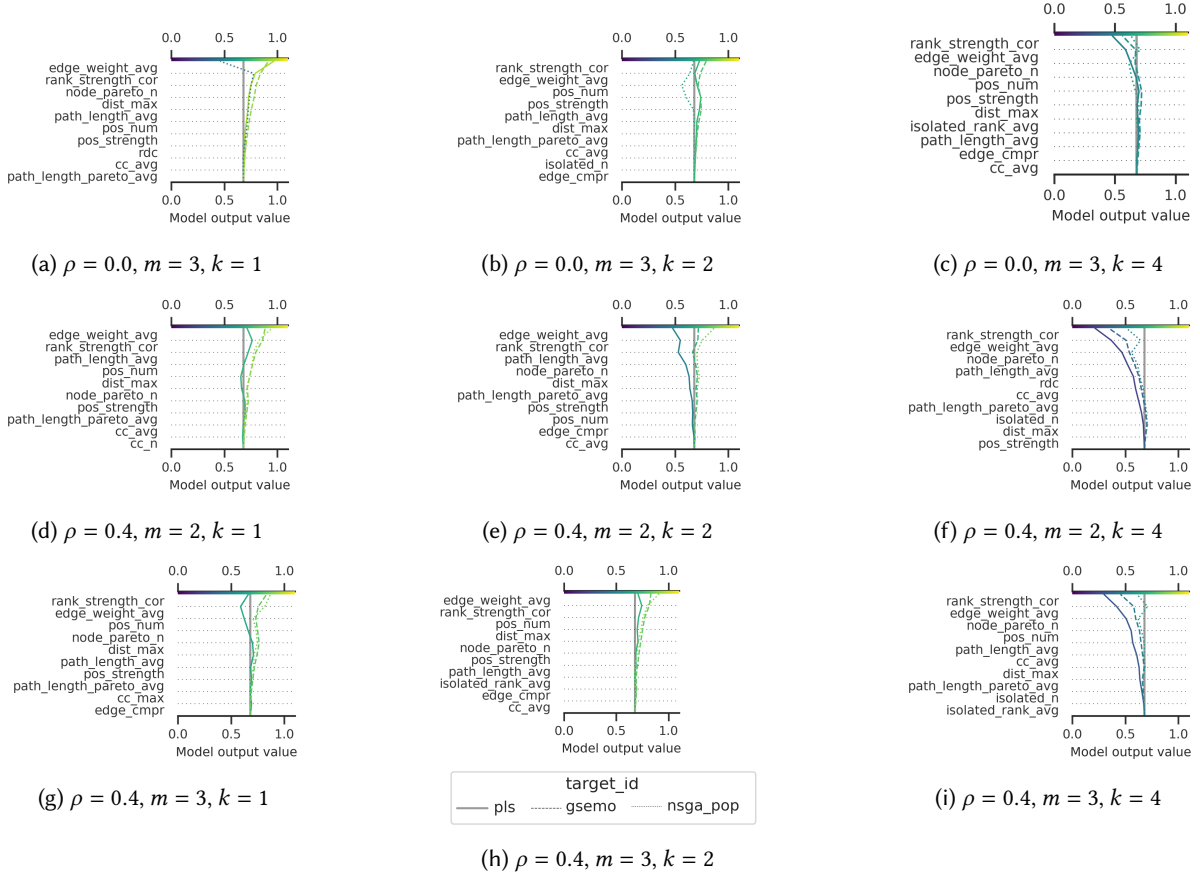


Fig. 5. Key landscape features for ρmnk landscapes are shown, with each plot representing a landscape and three lines for algorithm-feature interactions influencing performance: $\rho \in \{0, 0.4\}$, $m \in \{2, 3\}$, $k \in \{1, 2, 4\}$. The x-axis shows the predicted *reso* performance.

observe notable differences in algorithm performance. PLS excels in solving these landscapes, demonstrating highly effective performance across all cases. GSEMO shows moderate performance, handling these problem instances reasonably well but not matching the effectiveness of PLS. In contrast, NSGA-II performs poorly, struggling significantly in these landscapes. These results highlight the complementarity of the algorithms.

5.1.2 Important landscape feature combinations across different ρmnk problem instances. Figures 4 and 5 showcase the most important landscape feature combination for each of the 18 benchmark parameter combinations. Each sub-figure displays a SHAP *decision plot*, a powerful visualization used to explain the predictions made by the MTR model. In these plots, the y-axis lists the landscape features ranked by their importance, with the most important features positioned at the top. The x-axis corresponds to the model's prediction for *reso*, with the base value (the model's prediction in the absence of any landscape feature information) marked as a vertical line. This base value serves as the reference point, indicating what a baseline model would predict when no landscape features are provided. Additionally, the plot includes segmented lines, where each segment corresponds to a landscape feature. The deviation of the line segment from the base value represents how much the feature

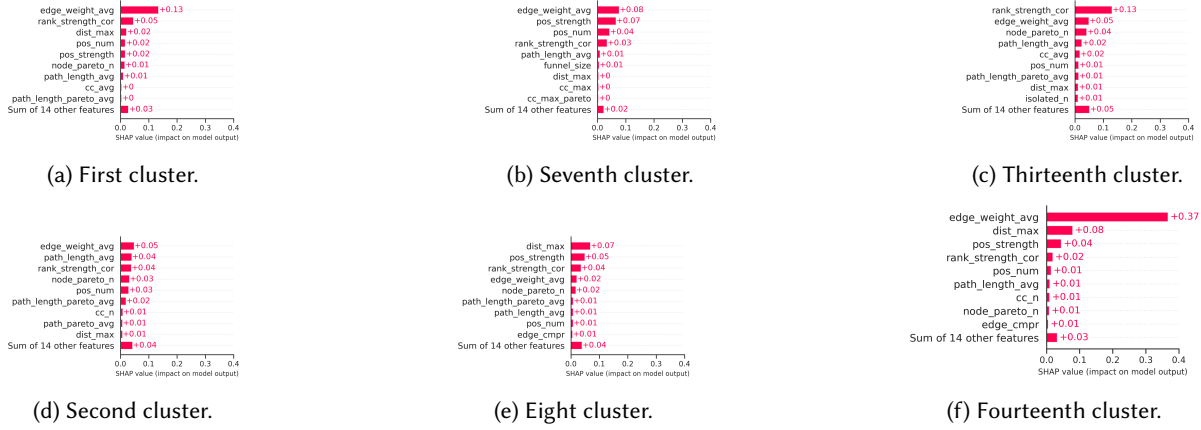


Fig. 6. The key features driving reso performance are shown for each cluster (performance region). Analysis includes the top two best, two medium, and two worst-performing regions.

influences the model's prediction. The three lines trace the path from the base value to the final model predictions for each algorithm separately, allowing for a clear comparison of feature importance across the three algorithms. The line color reflects predicted reso performance, ranging from dark blue (the worst) to yellow (the best). The line style distinguishes between the three algorithms with the solid line corresponding to PLS, the dashed to GSEMO, and the dotted line to NSGA-II, as indicated by the legend at the bottom of the plot.

Focusing on Figure 4b ($\rho = -0.4$, $m = 3$, and $k = 1$), we can see the combination of the landscape features that are causing the bad performance of NSGA-II, the medium performance of GSEMO, and the good performance of PLS. By examining the feature importance, we can conclude that some features, such as `rdc`, `cc_n`, and `edge_cmpr`, play a minimal role in determining the model's predictions, as the line segments remain close to the base value. The other landscape features starting from `pos_num` have a much larger impact on the predictions. This means that they are the characteristics of this landscape that make it particularly challenging for NSGA-II but relatively easy for PLS to navigate, among which `edge_weight_avg` holds the highest importance. We argue that smooth landscapes ($k = 1$) generally have a single funnel, thus local search can effectively follow direct heavily visited (i.e. high edge weight) "up-hill" pathways leading to good solutions. For GSEMO the landscape features from `node_pareto_n` to `rank_strength_cor` contribute to predicting good reso performance. However, `dist_max` and `edge_weight_avg` prevent GSEMO from achieving near-optimal performance (reso close to 1). For the PLS algorithm, similar patterns are observed, but `dist_max` and `edge_weight_avg` become advantageous landscape features, enabling near-perfect performance. Again, this is probably due to the most effective, greedy search that PLS can achieve in relatively smooth landscapes.

Next, from the last column of sub-figures ($k = 4$) the combination of the landscape features that are causing bad performance for all the algorithms become evident. Across all the plots, we consistently observe that the features `rank_strength_cor`, `edge_weight_avg`, and `node_pareto_n` emerge as the most important landscape features. These features influence all three algorithms, albeit to varying extents. This suggests that the PLS algorithm struggles to handle landscapes characterized by these features, whereas NSGA-II is able to achieve better performance to some extent. Due to space constraints, we will not present explanations for all benchmark parameter combinations. However, the figures provide insights into the importance of various landscape features, tailored to specific ρ mnk-landscapes and algorithms.

In this study, we examined landscape features both in isolation and in combination, revealing that the two new proposed funnel features from this study, such as `pos_num` and `pos_strength`, consistently appear as important across nearly all landscapes, while `rdc` is crucial for several. Among the landscapes where `rdc` plays a role, its most significant contribution occurs for $\rho = 0.4$, $m = 2$, and $k = 4$. This feature, alongside others, makes this particular landscape challenging for all algorithms. It is important to note that the landscape features presented here represent a selection of those that contribute to the best predictive MTR model. For instance, other funnel features correlate with `path_length_avg` and `path_pareto_avg`, which are part of the selected set. In the future, if a specific feature portfolio needs to be evaluated, the corresponding footprints can be easily calculated.

5.1.3 Important landscape feature combinations across clusters. Each cluster contains problem instances with similar reso performance (good, medium, or bad), achieved by different algorithms, due to similar reasons (similar most important feature combination). However, when two clusters contain problem instances with similar reso performance, the underlying reasons for achieving that performance differ (distinct most important feature combinations between the clusters). Figure 6 presents the most important feature combination for each cluster. The analysis focuses on a subset of clusters, including two clusters with good algorithm performance (the first and second clusters), two with medium performance (the seventh and eighth clusters), and two with bad performance (the thirteenth and fourteenth clusters). When examining cluster pairs column-wise—clusters share similar reso performance – the most important feature combination is distinct. For example, the importance of the `edge_weight_avg` feature is significantly more pronounced in the first cluster compared to the second. Further, when comparing clusters row-wise—clusters with different reso performance (ranging from good to bad) – the most important feature combinations vary substantially. Features such as `pos_strength`, `pos_num` and `dist_max` become important. Note that we show absolute feature importance, so the same features may matter in both good and bad reso clusters, though their influence (direction) differs as explained earlier.

6 DISCUSSION

We analyzed the hv metric, but the results are omitted due to space (available in our repository). As algorithms show little complementarity on this metric, landscape features have a similar influence, though interaction patterns vary specific to the landscapes. For the hv, the clustering resulted in 14 regions. Key features also differ from those for the reso metric. The features that appear among the most important features are `cc_avg`, `sink_num`, `sink_strength`, `pos_strength`, `dist_pareto_avg`, `dist_max`, `rank_strength_cor`, `node_pareto_n`, and `assort_degree`. Algorithm footprints are based on MTR model predictions, which, despite minor errors, closely approximate ground-truth reso performance. While our method uses an MTR predictive model, increasing the number of targets poses challenges due to the curse of dimensionality, a known issue in ML. Addressing this, deep learning techniques and larger benchmark sets could enhance MTR models to handle more algorithms. The applicability of the approach can be extended to larger-scale multi-objective combinatorial problems or different combinatorial structures, provided that relevant data can be collected.

7 CONCLUSION

By features from the C-PLOS-net model in combination with two newly introduced funnel features, we analyzed benchmark ρ mnk-landscape-landscapes with varying ruggedness and objective correlations. The evaluation of PLS, GSEMO, and NSGA-II using resolution metric revealed key feature combinations that influence algorithm performance. Additionally, the analysis highlights challenges specific to certain landscapes, offering deeper insights into feature importance and its implications for algorithm design and benchmarking in multi-objective optimization. The findings highlight the number of interactions as the most influential benchmarking parameter shaping problem difficulty, providing valuable insights for algorithm selection and optimization strategies. Furthermore, the study revealed that combining two sets of features is essential, as their combination emerges as the most significant for specific landscapes when paired with different algorithms. This indirectly provides

empirical evidence that these feature sets offer complementary information about the characteristics of the landscapes.

ACKNOWLEDGMENTS

We acknowledge the support of the Slovenian Research and Innovation Agency through program grant P2-0098, and project grants No.J2-4460 and No. GC-0001, and young researcher grant No. PR-12897 to AN. This work is also funded by the European Union under Grant Agreement 101187010 (HE ERA Chair AutoLearn-SI).

REFERENCES

- [1] Hernán Aguirre and Kiyoshi Tanaka. 2007. Working principles, behavior, and performance of MOEAs on MNK-landscapes. *European Journal of Operational Research* 181, 3 (2007), 1670–1690.
- [2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. Association for Computing Machinery, New York, NY, USA, 2623–2631.
- [3] Gérard Biau and Erwan Scornet. 2016. A random forest guided tour. *Test* 25 (2016).
- [4] Gjorgjina Cenikj, Ana Nikolikj, Gašper Petelin, Niki van Stein, Carola Doerr, and Tome Eftimov. 2024. A Survey of Meta-features Used for Automated Selection of Algorithms for Black-box Single-objective Continuous Optimization. *arXiv preprint arXiv:2406.06629* (2024).
- [5] Gjorgjina Cenikj, Gašper Petelin, and Tome Eftimov. 2024. A cross-benchmark examination of feature-based algorithm selector generalization in single-objective numerical optimization. *Swarm and Evolutionary Computation* 87 (2024), 101534.
- [6] François Chollet. 2015. Keras. <https://keras.io>.
- [7] Carlos A Coello Coello, Clarisse Dhaenens, and Laetitia Jourdan. 2010. Multi-objective combinatorial optimization: Problematic and context. In *Advances in multi-objective nature inspired computing*. Springer, Berlin, Heidelberg, 1–21.
- [8] Raphaël Cosson, Bilel Derbel, Arnaud Liefoghe, Hernán Aguirre, Kiyoshi Tanaka, and Qingfu Zhang. 2021. Decomposition-based multi-objective landscape features and automated algorithm selection. In *Evolutionary Computation in Combinatorial Optimization: 21st European Conference, EvoCOP 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings* 21. Springer, 34–50.
- [9] Fabio Daolio, Arnaud Liefoghe, Sébastien Verel, Hernán E. Aguirre, and Kiyoshi Tanaka. 2017. Problem Features versus Algorithm Performance on Rugged Multiobjective Combinatorial Fitness Landscapes. *Evolutionary Computation* 25, 4 (2017).
- [10] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.
- [11] 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.
- [12] Jonathan E. Fieldsend and Khulood Alyahya. 2019. Visualising the Landscape of Multi-Objective Problems Using Local Optima Networks. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019*. ACM, Prague, Czech Republic, 1421–1429.
- [13] Jonathan E. Fieldsend, Tinkle Chugh, Richard Allmendinger, and Kaisa Miettinen. 2022. A visualizable test problem generator for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 26, 1 (2022), 1–11.
- [14] Carlos M. Fonseca and Peter J. Fleming. 1996. On the performance assessment and comparison of stochastic multiobjective optimizers. In *Parallel Problem Solving from Nature, PPSN IV*. Springer, Berlin, Heidelberg, 584–593.
- [15] 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.
- [16] Ian T Jolliffe and Jorge Cadima. 2016. Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences* 374, 2065 (2016), 20150202.
- [17] Stuart A. Kauffman. 1993. *The Origins of Order*. Oxford University Press.
- [18] Pascal Kerschke and Christian Grimme. 2017. An Expedition to Multimodal Multi-objective Optimization Landscapes. In *Evolutionary Multi-Criterion Optimization, EMO 2017 (Lecture Notes in Computer Science, Vol. 10173)*. Springer, 329–343.
- [19] Joshua D. Knowles and David Corne. 2007. Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization. In *Evolutionary Multi-Criterion Optimization Conference, EMO 2007 (Lecture Notes in Computer Science, Vol. 4403)*. Springer, Matsushima, Japan, 757–771.
- [20] Ana Kostovska, Anja Jankovic, Diederick Vermetten, Sašo Džeroski, Tome Eftimov, and Carola Doerr. 2023. Comparing algorithm selection approaches on black-box optimization problems. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*. 495–498.
- [21] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. 2004. Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem. *Natural Computing* 3, 1 (2004), 37–51.

- [22] Arnaud Liefooghe, Fabio Daolio, Sébastien Verel, Bilel Derbel, Hernan Aguirre, and Kiyoshi Tanaka. 2019. Landscape-aware performance prediction for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 24, 6 (2019), 1063–1077.
- [23] Arnaud Liefooghe, Fabio Daolio, Sébastien Verel, Bilel Derbel, Hernán Aguirre, and Kiyoshi Tanaka. 2020. Landscape-Aware Performance Prediction for Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation* 24, 6 (2020), 1063–1077.
- [24] Arnaud Liefooghe, Bilel Derbel, Sébastien Verel, Manuel López-Ibáñez, Hernán Aguirre, and Kiyoshi Tanaka. 2018. On Pareto Local Optimal Solutions Networks. In *Parallel Problem Solving from Nature, PPSN XV*. Springer International Publishing, Cham, 232–244.
- [25] Arnaud Liefooghe, Gabriela Ochoa, Sébastien Verel, and Bilel Derbel. 2023. Pareto local optimal solutions networks with compression, enhanced visualization and expressiveness. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 713–721.
- [26] Arnaud Liefooghe, Sébastien Verel, Bilel Derbel, Hernan Aguirre, and Kiyoshi Tanaka. 2020. Dominance, indicator and decomposition based search for multi-objective QAP: landscape analysis and automated algorithm selection. In *International Conference on Parallel Problem Solving from Nature*. Springer, 33–47.
- [27] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4765–4774. <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>
- [28] Ana Nikolikj and Tome Eftimov. 2024. Comparing Solvability Patterns of Algorithms across Diverse Problem Landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. Association for Computing Machinery, New York, NY, USA, 143–146.
- [29] Ana Nikolikj, Gabriela Ochoa, and Tome Eftimov. 2025. Repository. <https://doi.org/10.5281/zenodo.15116349>. <https://doi.org/10.5281/zenodo.15116349>
- [30] Gabriela Ochoa, Arnaud Liefooghe, and Sébastien Verel. 2024. Funnels in multi-objective fitness landscapes. In *International Conference on Parallel Problem Solving from Nature*. Springer, 343–359.
- [31] Gabriela Ochoa, Arnaud Liefooghe, and Sébastien Verel. 2024. Funnels in Multi-objective Fitness Landscapes. In *Parallel Problem Solving from Nature - PPSN XVIII (Lecture Notes in Computer Science, Vol. 15148)*. Springer, 343–359. https://doi.org/10.1007/978-3-031-70055-2_21
- [32] Luís Paquete, Tommaso Schiavinotto, and Thomas Stützle. 2007. On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research* 156, 1 (2007), 83–97.
- [33] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.
- [34] Lennart Schäpermeier, Christian Grimme, and Pascal Kerschke. 2020. One PLOT to Show Them All: Visualization of Efficient Sets in Multi-objective Landscapes. In *Parallel Problem Solving from Nature, PPSN XVI*. Springer International Publishing, Cham, 154–167.
- [35] Moritz Seiler, Urban Škvorc, Carola Doerr, and Heike Trautmann. 2024. Synergies of deep and classical exploratory landscape features for automated algorithm selection. In *International Conference on Learning and Intelligent Optimization*. Springer Nature Switzerland, Cham, 361–376.
- [36] Eric K Tokuda, Cesar H Comin, and Luciano da F Costa. 2022. Revisiting agglomerative clustering. *Physica A: Statistical mechanics and its applications* 585 (2022), 126433.
- [37] Guy Van den Broeck, Anton Lykov, Maximilian Schleich, and Dan Suciu. 2022. On the tractability of SHAP explanations. *Journal of Artificial Intelligence Research* 74 (2022), 851–886.
- [38] Sébastien Verel, Arnaud Liefooghe, Laetitia Jourdan, and Clarisse Dhaenens. 2013. On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives. *European Journal of Operational Research* 227, 2 (2013), 331–342.
- [39] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.
- [40] Xiantong Zhen, Mengyang Yu, Xiaofei He, and Shuo Li. 2017. Multi-target regression via robust low-rank learning. *IEEE transactions on pattern analysis and machine intelligence* 40 (2017), 497–504.
- [41] 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.