

This is the Author-Accepted Version of the paper:

A. Nikolikj, A. Kostovska, G. Cenikj, C. Doerr and T. Eftimov, "Generalization Ability of Feature-Based Performance Prediction Models: A Statistical Analysis Across Benchmarks," *2024 IEEE Congress on Evolutionary Computation (CEC)*, Yokohama, Japan, 2024, pp. 1-8, doi: 10.1109/CEC60901.2024.10611952.

© **2024 IEEE**. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Generalization Ability of Feature-based Performance Prediction Models: A Statistical Analysis across Benchmarks

Ana Nikolikj, Ana Kostovska, Gjorgjina Cenikj, Carola Doerr, and Tome Eftimov

Abstract—This study examines the generalization ability of algorithm performance prediction models across various benchmark suites. Comparing the statistical similarity between the problem collections with the accuracy of performance prediction models that are based on exploratory landscape analysis features, we observe that there is a positive correlation between these two measures. Specifically, when the high-dimensional feature value distributions between training and testing suites lack statistical significance, the model tends to generalize well, in the sense that the testing errors are in the same range as the training errors. Two experiments validate these findings: one involving the standard benchmark suites, the BBOB and CEC collections, and another using five collections of affine combinations of BBOB problem instances.

Index Terms—meta-learning, single-objective optimization, generalization, performance prediction

I. INTRODUCTION

Automated algorithm configuration [1], [2] and selection [3], [4] are gaining a lot of attention in evolutionary computation. In most cases, they are performed by using supervised Machine Learning (ML) predictive models that use the feature representation of a problem instance (i.e., its characteristics) as input data and predict the performance of a specific algorithm (instance) achieved on that problem instance. However, one of the main drawbacks presented in these learning tasks is the low generalization ability of the predictive models. That is, the models tend to fail to provide accurate predictions for problem instances whose feature representation is underrepresented or not presented in the training data. For example, Škvorc et al. [5] showed that a random forest (RF) model trained on the BBOB suite of the COCO environment [6] yields poor results when tested on

the artificially generated problem instances from [7] and vice-versa. Kostovska et al. [8] show that an automated algorithm selector which is based on performance prediction models trained on the BBOB benchmark suite, cannot generalize on problem instances that are part of the Nevergrad’s YAB-BOB [9] benchmark suite.

By using feature representation of problem instances, several studies [10]–[15] perform complementary analyses of different benchmark suites in the feature space. However, all the analyses are descriptive, trying to understand the similarities and differences between the problem instances across different benchmark suites without quantifying the similarities on a benchmark suite level. Nikolikj et al. [16] have explored how well a performance predictive model can adapt based on the benchmark suite coverage. Empirical meta-features for each suite have been created by clustering instances across all suites and examining their similarities. The findings indicated that when two benchmark suites share similar empirical coverage, an ML model trained on one can perform well on the other.

Our contribution: In this study, we investigate the generalization ability of a performance prediction model through a statistical measure assessing the similarity of coverage among benchmark suites. Unlike the previous published empirical approach, which involved condensing high-dimensional benchmark data into a lower-dimensional space using clustering to define meta-representations for each suite, we directly utilize the raw benchmark suite data from the high-dimensional space – representing all instances with meta-features. Employing a statistical test allows us to compare suite coverage distributions in their original high-dimensional space without losing information through conversion to a lower-dimensional space, as done previously. To assess patterns between the feature landscape and performance realms, we trained a predictive model for a specific optimization algorithm on one benchmark suite and evaluated it on another. The results imply that statistical insights from the feature landscape can anticipate how well a model extends to various suites. When the high-dimensional feature landscape distributions of training and testing suites are not statistically significant, the model archives good performance on the testing suite preserving an error within the training error range. These conclusions arise from two experiments: one encompassing typical benchmark suites for algorithm evaluation and another that employed sampling to generate five new artificial benchmark suites from

Ana Nikolikj (Email: ana.nikolikj@ijs.si), Gjorgjina Cenikj (Email: gjorgjina.cenikj@ijs.si) and Tome Eftimov (Email: tome.eftimov@ijs.si) are with Computer Systems Department, Jožef Stefan Institute, 1000 Ljubljana, Slovenia.

Ana Kostovska (Email:ana.kostovska@ijs.si) is with the Department of Knowledge Technologies, Jožef Stefan Institute, 1000 Ljubljana, Slovenia.

Ana Nikolikj, Gjorgjina Cenikj, and Ana Kostovska are also with the Jožef Stefan International Postgraduate School, 1000 Ljubljana, Slovenia.

Carola Doerr (Email: carola.doerr@lip6.fr) is with the Sorbonne Université, CNRS, LIP6, 1000 Paris, France.

The authors acknowledge the support of the Slovenian Research Agency through program grants P2-0098 and P2-0103, project grant J2-4460, and young researcher grant No. PR-12897 to AN, PR-12393 to GC, and PR-09773 to AK, and a bilateral project between Slovenia and France grant No. BI-FR/23-24-PROTEUS-001 (PR-12040).

problem instances.

Outline: Section II presents an overview of complementary analyses performed across different benchmark suites. Section III introduces the workflow used to estimate and evaluate the generalization ability of a predictive model across different benchmark suites. The experimental design is explained in Section IV, followed by a discussion of key results in Section V. Section VI concludes the paper.

Data and code availability: The data and the code involved in this study are available at [17].

II. RELATED WORK

Most of the studies performed in the direction of performance prediction of single-objective black-box optimization algorithms rely on Exploratory Landscape Analysis (ELA) [18] to calculate features that describe the properties of the problem instances. ELA is a set of mathematical and statistical techniques that use a sample of candidate solutions from the problem instance decision space, generated using a certain sampling technique. They can be calculated using the R programming language package called “flacco” [19]. A recent version of the package has also been published in Python [20]. These features have been used in several studies for complementary analysis between the benchmark suites.

Zhang and Halgamuge analyze the similarity of continuous problem instances by representing them with algorithm performance [10]. The results indicate that the problem instances from different problem classes exhibit similarities in performance and that low-dimensional instances could also share performance similarities with their high-dimensional counterparts. Škvorc et al. [11] analyze the complementary of BBOB and CEC benchmark suites by representing the problem instances using ELA features and further visualizing them with the t-distributed stochastic neighbor embedding (tSNE) method [21] in lower dimensions. The results show that the benchmark suites have different distributions over the landscape feature space. Muñoz and Smith Miles [12] use genetic programming to generate new problem instances with controllable characteristics to increase the coverage of the problem landscape. The results demonstrate that the newly generated problem instances are more challenging for the algorithms to solve than the well-known benchmark suites. Eftimov et al. [13] perform a correlation analysis of the projection of the ELA feature representations into the subspace obtained by a singular value decomposition, between the BBOB problem instances and the HappyCat and HGBat problem instances. Cenikj et al. [14] present an approach, SELECTOR, for selecting diverse problem instances based on their ELA feature representation. They evaluate different sampling heuristics, one based on clustering and two based on graph embeddings. The results show that regardless of the choice of sampling heuristic, the approach leads to a reproducible statistical comparison of algorithm performance. Long et al. [15] have provided a detailed analysis of landscape properties and algorithm performance across BBOB problem instances.

Previous studies have explored the complementarity of benchmark suites empirically, without directly linking it to predictive model generalization. A recent study found that when benchmark suites share similar empirical coverage [16], training a model on one suite yields good generalization on another. However, this depends on the meta-representation used. Our study addresses this by examining a statistical measure based on raw problem landscape data.

III. STATISTICAL MEASURE FOR ACCESSING SIMILARITY OF BENCHMARK SUITES

Consider a scenario involving m benchmark suites, each comprising varying numbers of problem instances. From this pool, one of the m suites is selected to serve as the training set for the supervised ML predictive model (\mathcal{M}), while the remaining $m - 1$ suites are utilized for testing purposes. To evaluate the model \mathcal{M} 's generalization ability across diverse benchmark suites used for testing, we outline the following workflow:

- 1) Establishing a unified meta-representation at the individual problem instance level involves characterizing instances across all benchmark suites using a shared set of n meta-features, describing their landscape properties. This approach ensures that all selected problem instances are mapped into the same n -dimensional vector space. With this representation, each benchmark suite can be represented as a matrix $BS_{k \times n}$, where k is the number of instances that are part of a benchmark suite (which can be different for different benchmark suites).
- 2) Once each benchmark suite is represented by its matrix, we can use a statistical test to compare the high-dimensional coverage distributions between two of them (one used for training and the other for testing). To this end, a statistical test for comparing high-dimensional distributions should be utilized. A category of consistent, distribution-free tests applicable to high-dimensional spaces relies on nearest neighbors using the Euclidean distance metric [22], [23]. Székely and Rizzo introduced the multivariate \mathcal{E} test, demonstrating its universal consistency against all alternatives (not necessarily continuous) possessing finite second moments. Notably, the computational complexity of this test remains independent of dimensionality or sample size, making it a formidable contender among nearest-neighbor tests. Findings highlighted in [24] suggest that the multivariate \mathcal{E} test stands out as one of the most robust tests available for analyzing high-dimensional data, which makes it a good choice for our analysis.

Let us assume that two benchmark suites are involved, $P_{k_1 \times n}$ and $Q_{k_2 \times n}$, where p_1, p_2, \dots, p_{k_1} and q_1, q_2, \dots, q_{k_2} are problem instances represented by n landscape meta-features (i.e., vectors in \mathbb{R}^n) that belong to the two benchmark suites respectively. The multivariate \mathcal{E} test statistic between

them is defined as:

$$\begin{aligned} \mathcal{E}_{k_1, k_2} = & \frac{k_1 k_2}{k_1 + k_2} \left(\frac{2}{k_1 k_2} \sum_{i=1}^{k_1} \sum_{m=1}^{k_2} \|p_i - q_m\| \right. \\ & - \frac{1}{k_1^2} \sum_{i=1}^{w_1} \sum_{j=1}^{w_1} \|p_i - p_j\| \\ & \left. - \frac{1}{k_2^2} \sum_{l=1}^{k_2} \sum_{m=1}^{k_2} \|q_l - q_m\| \right). \end{aligned} \quad (1)$$

The initial double sum in the equation above indicates the distance between problem instances from the benchmark suites, while the subsequent double sums delineate the internal distances within each benchmark suite (P and Q). The test statistic follows a degenerate two-sample V-statistic [25] – readers interested in the mathematical details of the test are invited to confer [24].

3) The outcome of this comparison yields a p-value, serving as an indicator to ascertain whether a difference exists or not in the coverage distribution between the two benchmark suites. If there is no statistical difference, there is a high likelihood that a performance predictive model trained on one of the benchmark suites can be also utilized and generalize the results on the other benchmark suite and vice-versa.

IV. EXPERIMENTAL DESIGN

We begin this section by detailing the benchmark suites chosen for conducting two experiments. Additionally, we include details on the performance and problem landscape data and we provide details on training the performance prediction models.

Benchmark suites: As mentioned before, we conduct two experiments where we perform statistical analysis of the generalization ability of the performance prediction models across various benchmark suits.

First experiment: The first experiment statistically assesses the generalization ability of predictive models across four widely-used benchmark suites: BBOB (COCO) [6], CEC 2013 [26], CEC 2014 [27], and CEC 2015 [28]. BBOB comprises 24 problem classes, from which we use five instances each. The CEC 2013, CEC 2014, and CEC 2015 benchmark suites contain 28, 30, and 15 problems, respectively. Finally, for CEC 2013, we ended up with 27 problems. This is because three problems (specifically, the 3rd, 7th, and 20th problems) were excluded due to missing data arising from the landscape feature calculation process, as detailed later in this section. The experiment considers problems with $D = 10$ numerical decision variables.

Second experiment: The second set of experiments uses statistical analyses to assess how predictive models generalize on artificially generated benchmark suites. Demonstrating the impact of a more strategic training data selection, we generate these suites by using instances created as affine recombinations of pairs of BBOB problem instances, as introduced in [29]. The experiment focuses on a fixed problem dimensionality of

$D = 5$. This choice allows us to re-use available performance data.

To ensure a representative and diverse set of problem instances from those generated through affine recombinations, we apply the SELECTOR methodology [14]. This involves converting benchmark problem instances into a graph format, where nodes represent individual problems and an edge is created if the cosine similarity between their meta-representations is 0.9 or higher. The Maximal Independent Set (MIS) algorithm [30] is used to select instances, ensuring diversity by making sure selected instances have a pairwise cosine similarity less than 0.9. Since the MIS algorithm is stochastic, we repeat the process five times, resulting in five benchmark suites: BS1, BS2, BS3, BS4, and BS5. They contain 56, 57, 56, 55, and 53 problem instances respectively, with minimal overlap between instances across the benchmarks. The sole instance of overlap occurs between the first and second benchmark suites and between the third and fifth benchmark suites, each involving a single problem instance.

Performance data: For the first experiment, we analyze performance data from a portfolio of three algorithms, the Covariance Matrix Adaption Evolutionary Strategy (CMA-ES) [31], the Real Space Particle Swarm Optimization (PSO) [32], and of Differential Evolution (DE) [33], respectively. Their implementations are taken from the Nevergrad library [34], with each algorithm being configured to its default hyper-parameter setting. We fix both the budget and the target. The computational budget for executing the algorithms is limited to 100,000 function evaluations. We also set a target precision threshold at 10^{-8} . The algorithm terminates upon either exhausting its allocated budget or when achieving the target precision, defined as the absolute difference $f(x^{\text{best}}) - f^*$ between the quality of the best-found solution x^{best} and that of a global optimum $f^* := \inf_x f(x)$. The experiments are run using the IOHexperimenter [35] environment, for convenience of accessing the BBOB functions and for logging the search trajectories in a standardized way. Due to the stochastic nature of the algorithms, 30 independent runs of each algorithm on each problem instance have been performed. Finally, the median target precision across 30 repetitions has been calculated.

In the second experiment, the same algorithms are used, this time evaluated on the affine functions and with a budget of 10,000 function evaluations. Similar to the first experiment, 25 independent runs have been performed of each algorithm on each problem instance and we calculate the median target precision across 25 repetitions.

Problem landscape data: The landscape characteristics of the problems are represented using publicly available ELA features. Specifically, for the first experiment, we utilize the 64 ELA features available from [36]. These feature values were computed using the Improved Latin Hypercube Sampling (iLHS) method [37], with a sample size of $800D = 8,000$ and repeated 30 times. The median value of each feature across 30 repetitions was calculated and used. The choice of a larger sample size was deliberate to minimize the randomness inherent in the feature extraction process. For the second experi-

TABLE I: Comparative statistical analysis of high-dimensional feature-space distributions between paired benchmark suites, scaling based on the collection listed in the row. Presenting p-values; values $\leq .005$ are marked with an *.

	BBOB	CEC2013	CEC2014	CEC2015
BBOB	/	0.005*	0.005*	0.005*
CEC2013	0.035*	/	0.105	0.005*
CEC2014	0.005*	0.245	/	0.690
CEC2015	0.005*	0.205	0.490	/

ment, we utilized the 14 ELA features available from [29]. The calculation of these feature values was carried out employing Sobol’ sampling, with a sample size of $250 \times D = 1,250$, and repeated 30 times. Same as the first experiment, the median value of each feature across 30 repetitions was calculated and used.

Comparing benchmark suite distribution: The statistical comparisons have been performed using the R programming language. To compare the distributions of high-dimensional data the *multivariate \mathcal{E} test* is used, which is a part of the “energy” package [38].

Predictive models: For each benchmark suite, we train a Random Forest (RF) regression model to predict the algorithm’s performance, measured by the target precision achieved with the 100,000 (first experiment) and 10,000 function evaluations (second experiment), respectively. Instead of predicting the median target precision for each problem instance in the original space, we train the models in log space. This is the value we are predicting with the ML models. We use the default implementation of the RF regressor from the *scikit-learn* package in Python. The performance of each of the trained models is evaluated on the other benchmark suites (that have not been used for training the model) and we report the median absolute error (MDAE) across all problem instances in the test benchmark suite. We analyze the obtained results to examine whether the patterns identified in the coverage matrix are similarly reflected in the performance of the automated algorithm performance prediction model. For this experiment, the feature values are scaled by subtracting the mean and scaling to unit variance, using the *scikit-learn* package. The parameters used for the scaling are learned using the training suite and then applied to the test suite.

V. RESULTS AND DISCUSSION

A. First experiment

Table I displays the p-values obtained from the statistical comparison of feature-space distributions across various pairs of benchmark suites. Here, we only consider the feature space of the problem, disregarding the performance of the algorithms.

Note also that we consider here pairwise comparisons, not multiple ones. The comparison matrix is not symmetric, caused by the scaling procedure explained above (which depends on the training set, i.e., here the set in the row).

From this table, we observe that the feature-space distribution of the **BBOB** benchmark suite exhibits statistical signifi-

TABLE II: The MDAE during training of the RF model within every benchmark suite for the algorithms CMA-ES, PSO, and DE.

Algorithm	BBOB	CEC2013	CEC2014	CEC2015
CMA	0.033	0.261	0.234	0.228
PSO	0.055	0.173	0.223	0.208
DE	0.033	0.231	0.201	0.279

cance when (individually) compared to CEC2013, CEC2014, and CEC2015, respectively, suggesting that a performance predictive model trained on BBOB data will likely yield higher errors than the training error when utilized for predictions on CEC2013, CEC2014, and CEC2015. For the CEC2013 suite, no statistical significance is observed when compared to CEC2014, however, statistical significance has been noted when compared to the BBOB and the CEC2015 suites. These findings suggest that a model trained using CEC2013 will demonstrate good predictive performance when applied to CEC2014. However, the prediction errors are likely to increase when this model is used for predictions on BBOB and CEC2015 benchmark suites. For **CEC2014** and **CEC2015**, please confer the table.

To determine if the statistical patterns observed in the feature landscape space are consistent in the performance space, Fig 1 showcases the MDAE (calculated as the median of the absolute differences between the predicted values and the ground truth value (i.e., log from the median target precision)) of the RF predictive model. This model is trained using one benchmark suite indicated in the rows of the heatmap and tested on the remaining three benchmark suites (columns of the heatmap). This analysis is conducted independently for three algorithms (CMA-ES, DE, and PSO). Additionally, Table II displays the training errors of the RF models for each benchmark suite individually, so we can further analyze if the testing errors are in the same ranges as the training errors.

Here are the outcomes derived from assessing performance predictive models for three algorithms across individual benchmark suites:

BBOB – The models trained on BBOB consistently yield larger errors (compared to the training errors) across all benchmark suites for PSO and CMA-ES, aligning with the anticipated outcomes based on the feature space observations. However, for DE, the errors display variability across the benchmark suites, a trait that might be influenced by the specific behavior of the DE algorithm. **CEC2013** – Training models on CEC2013 result in comparable testing errors across all benchmark suites and algorithms. However, in line with the statistical pattern observed in the feature space indicating smaller errors on CEC2014, it is not clearly visible. **CEC2014** – The models trained using CEC2014 exhibit anticipated larger errors when evaluated on BBOB. Regarding evaluations on CEC2015, all algorithms showcase smaller errors, in line with the statistical patterns observed in the feature space. Specifically, when evaluating on CEC2013, the PSO and DE algorithms reflect the statistical pattern observed in the

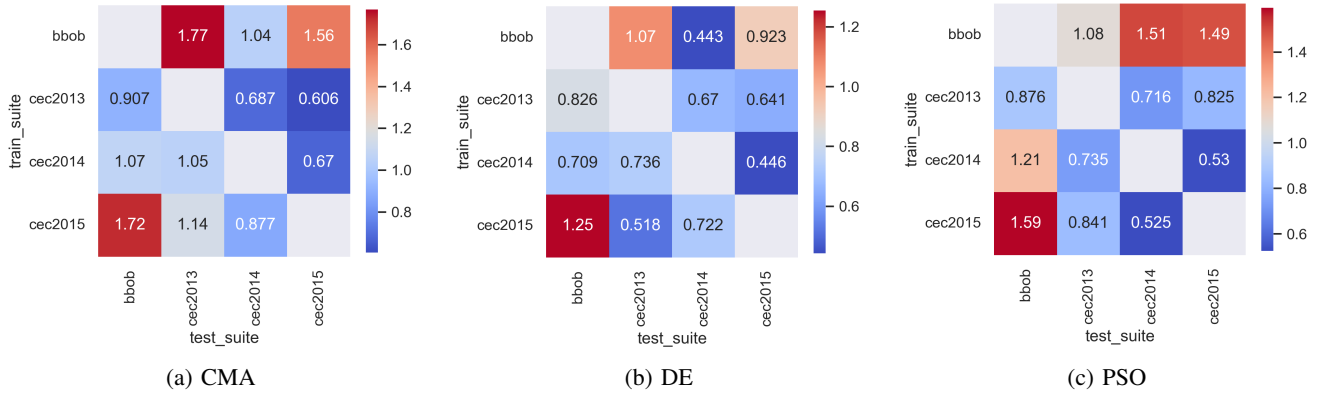


Fig. 1: Heatmap showing the MDAE of an RF model when predicting the performance of a) CMA, b) DE, and c) PSO, on BBOB, CEC2013, CEC2014, CEC2015, and CEC2017. Rows indicate the training benchmark suite and columns indicate the benchmark suite of the model was evaluated on.

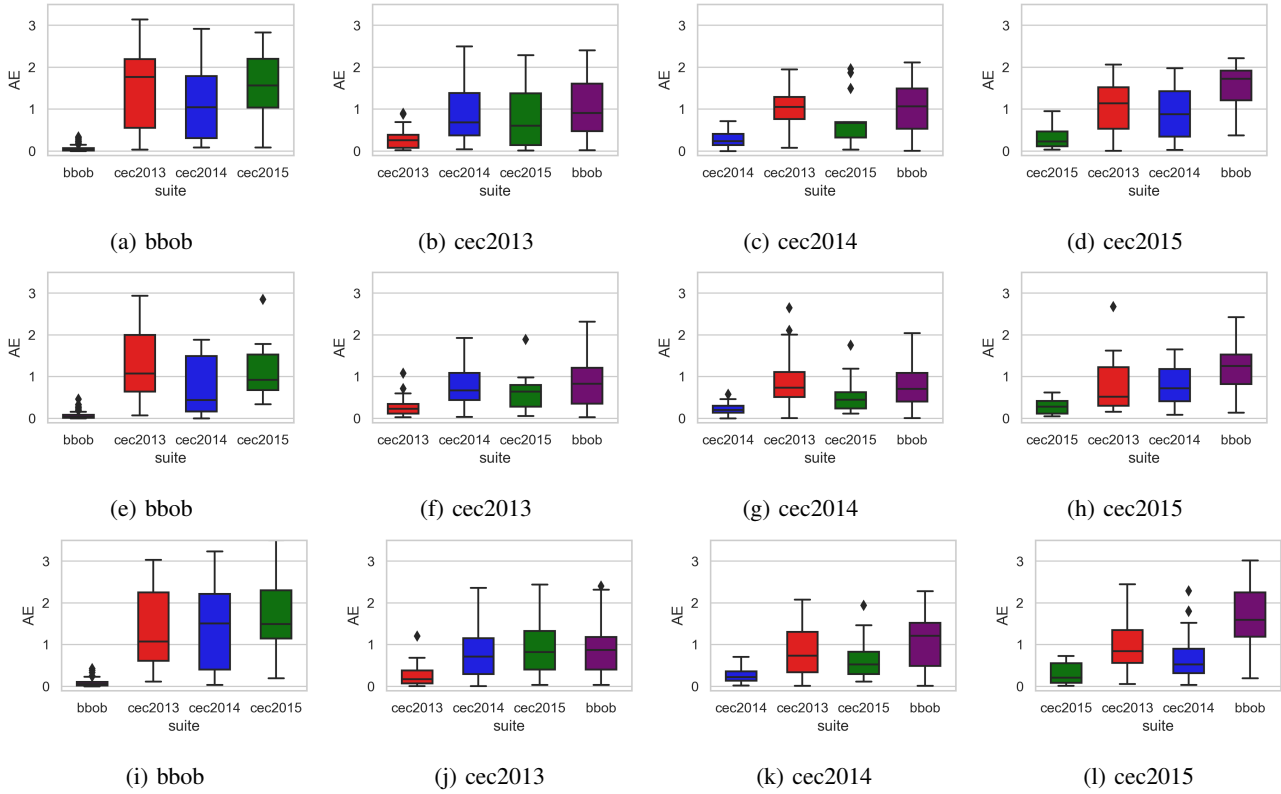


Fig. 2: Box-plots showing the AE (Absolute error) of an RF model when predicting the performance of a-d) CMA, e-h) DE, and i-l) PSO. Subplot titles name the training benchmark suite, with one box plot showing train AEs and others depicting corresponding test AEs.

feature space, whereas CMA exhibits errors more akin to those obtained on BBOB, differing from the feature space pattern. **CEC2015** – Across all three algorithms, we observe a consistency between the outcomes depicted in the landscape feature space and the evaluation of predictive models. Lower errors are evident in CEC2013 and CEC2014, whereas they notably rise in BBOB.

To support our findings, instead of using MDAE (aggregated errors across the entire test suite), we have included box plots displaying absolute errors for individual instances from

a specific set of an RF model predicting the performance of CMA, DE, and PSO. Subplot titles denote the benchmark suite used for model training, with one box plot illustrating training absolute errors and the others depicting corresponding test absolute errors (Fig 2).

The statistical patterns observed in the feature space for BBOB, CEC2014, and CEC2015, generally align with the evaluation of algorithm prediction models. However, this expectation does not hold as strongly for the CEC2013 benchmark suite. Additionally, in examining this result, we assess

the performance distribution of individual algorithms within each benchmark suite. This involves conducting separate pairwise comparisons of performance distributions across diverse benchmark suites for each algorithm. Fig. 3 displays the p-values resulting from comparing an algorithm’s performance distributions by using the two-sample Kolomogorov-Smornov test across pairs of benchmark suites. Each row and column represent benchmark suites used in the pairwise comparison. The heatmaps are in the upper triangle due to the symmetric nature of the comparisons.

If we revisit the CEC2014 case study, the statistical observation in the feature landscape space compared to CEC2013 suggests that these models exhibit generalization ability. However, this is true for the PSO and DE, while it is not in the case for CMA. Upon analyzing the performance distribution comparison, we observe that there is no statistically significant difference between the performance distributions of DE across CEC2014 and CEC2013, the same is true for PSO. However, examining CMA reveals a statistical significance in its performance distributions between CEC2014 and CEC2013. This outcome indicates that despite the statistical similarity found in the feature landscape space of these benchmark suites, such similarity is not evident in the performance space of the CMA algorithm. This outcome suggests that the chosen ELA feature portfolio might lack the capability to detect varied CMA behavior, which is not the case for DE and PSO. In the future, we aim to identify specific meta-features tailored to individual algorithms or their respective families. This could involve conducting feature selection on the ELA features or creating and assessing alternative landscape features [39], [40].

B. Second experiment

Table III illustrates the p-values acquired through the pairwise comparisons between two selected benchmark suites, where one suite is utilized for training and the other for testing the model. The outcomes revealed no statistically significant differences among the pairs, as anticipated. This aligns with our expectations because all chosen benchmark suites were sampled using the same technique – SELECTOR. Based on the results, we anticipate that models trained on one selected benchmark suite will demonstrate good generalization ability when tested on another selected benchmark suite.

TABLE III: Comparative statistical analysis of high-dimensional feature-space distributions between paired benchmark suites artificially sampled from AFFINE problems (presenting p-values).

	BS1	BS2	BS3	BS4	BS5
BS1	/	0.86	0.97	0.56	0.92
BS2	0.90	/	0.47	0.51	1.00
BS3	0.99	0.88	/	0.94	0.95
BS4	0.98	0.89	0.99	/	0.97
BS5	0.93	1.00	1.00	0.98	/

Fig 4 presents the MDAE of an RF model in predicting the performance of CMA, DE, and PSO across benchmark suites selected from the affine problem instances. Rows represent the training benchmark suite, while columns indicate the evaluated

benchmark suite for the model. Based on the findings provided, we can infer that consistent smaller errors are observed among the tested pairs of benchmark suites, mirroring the outcomes showcased in the feature landscape space (i.e., the high-dimensional distributions within the feature landscape space of the benchmark suites show no statistical significance.). Moreover, to illustrate that the testing errors are in similar ranges with the training errors, Table IV displays the training errors of the RF model within each artificially selected benchmark suite from the affine problems for the algorithms Diagonal CMA, PSO, and DE.

TABLE IV: The errors during training of the RF model within every benchmark suite artificially sampled from the affine problems for the algorithms Diagonal CMA, PSO, and DE.

Algorithm	SB1	SB2	SB3	SB4	SB5
CMA	0.0619	0.051	0.0890	0.039	0.046
PSO	0.098	0.076	0.081	0.114	0.095
DE	0.045	0.022	0.046	0.044	0.043

C. Discussion

Since the data used in this study has been taken from another study, we provide a discussion about the results obtained here and the previous results reported. The main difference is the measures used to estimate the generalization ability. In the other study, empirical measures have been introduced. All problem instances from various benchmark suites are aggregated, represented by the same meta-features, and subsequently clustered. To evaluate the similarity among the benchmark suites, a coverage matrix is computed. This matrix quantifies the percentage of problem instances from each suite represented within each cluster. Next, it establishes the benchmark suite meta-representation by utilizing the distribution percentages of instances from each benchmark suite across all clusters. This facilitates the comparison between the meta-representations of two benchmark suites by using similarity measures (e.g., cosine similarity), where one is utilized to train the performance predictive model and the other for testing purposes. Here, instead of using empirical measures, statistical measures are used to estimate the generalization ability. This means that a high-dimensional statistical test is utilized to compare the raw benchmark suite data – all problem instances represented by their meta-features, without sacrificing information by converting it into a lower-dimensional space, as was done in the empirical case.

When empirical measures are employed [16], it becomes evident that all CEC benchmark suites exhibit substantial similarity, displaying minimal differences in the feature space. However, employing innovative statistical measures enables the identification of finer disparities among them, enhancing the accuracy of our expectations regarding model performance errors. Looking ahead, as these measures possess distinct natures and are not directly comparable, employing ensemble techniques can combine their perspectives, leveraging their differing views of the same data.

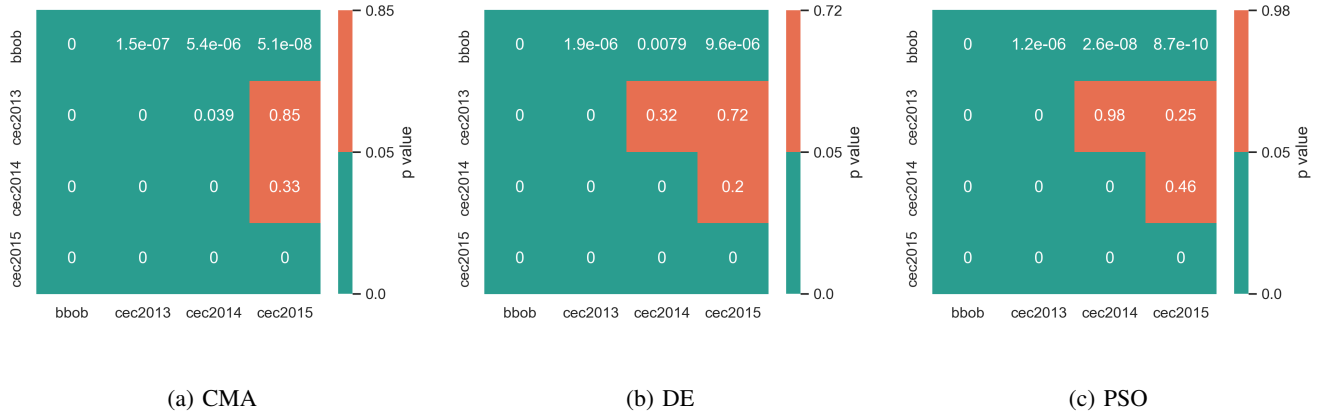


Fig. 3: Heatmap that visualizes the p-values obtained by comparing an algorithm’s performance distributions among pairs of benchmark suites (a) CMA, b) DE c) PSO). Rows and columns depict benchmark suites in paired comparisons. Upper triangle heatmaps show symmetry. A two-sample Kolmogorov-Smirnov test (p -value ≤ 0.05) indicates significant differences in algorithm performance between benchmark suites.

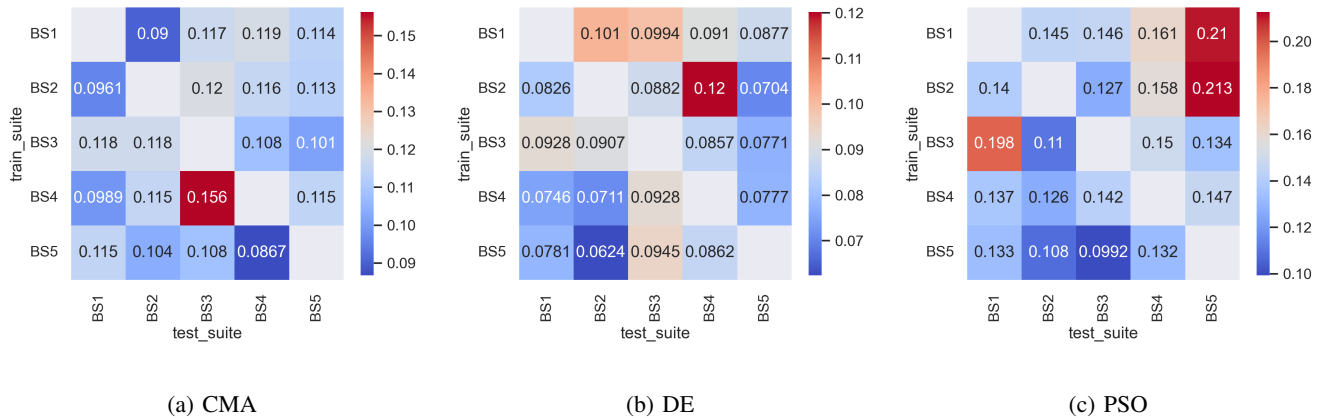


Fig. 4: Heatmap showing the MDAE of an RF model when predicting the performance of a) CMA, b) DE, and c) PSO on the benchmark suites sampled from the affine problems. Rows indicate the training benchmark suite and columns indicate the benchmark suite of the model was evaluated on.

VI. CONCLUSIONS

This study examined how well a performance predictive model can adapt to new scenarios. We used statistical tests to compare suite coverage distributions in their original high-dimensional feature landscape. By training a model on one benchmark suite and testing it on another, we found that statistical similarities in feature landscape patterns can indicate the model’s generalizability. When the distributions between training and testing suites show no significant difference, the model effectively generalizes, maintaining a similar error range.

In our future work, we plan to conduct a comprehensive experiment using a wider range of algorithms, including Nevergrad [34], to see if the insights gained from the feature landscape analysis extend to broader algorithmic families. Our study demonstrated that performance prediction models

built on ELA features effectively generalize across the three tested algorithms. Next, we will explore additional feature landscape meta-features, such as topological features [39] and those derived from deep neural network architectures [40], comparing them with ELA features to enhance predictive accuracy. Finally, we aim to evaluate these measures in an active learning setting, using them to determine if a model is suitable for new instances or if further training and fine-tuning are necessary.

REFERENCES

- [1] R. P. Prager, H. Trautmann, H. Wang, T. H. Bäck, and P. Kerschke, “Per-instance configuration of the modularized CMA-ES by means of classifier chains and exploratory landscape analysis,” in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 996–1003.
- [2] N. Belkhir, J. Dréo, P. Savéant, and M. Schoenauer, “Per instance algorithm configuration of CMA-ES with limited budget,” in *Proc. of*

- Genetic and Evolutionary Computation (GECCO'17)*. ACM, 2017, pp. 681–688. [Online]. Available: <https://doi.org/10.1145/3071178.3071343>
- [3] A. Jankovic and C. Doerr, “Landscape-aware fixed-budget performance regression and algorithm selection for modular CMA-ES variants,” in *GECCO*. ACM, 2020, pp. 841–849.
 - [4] P. Kerschke and H. Trautmann, “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.
 - [5] U. Škvorc, T. Eftimov, and P. Korošec, “Transfer learning analysis of multi-class classification for landscape-aware algorithm selection,” *Mathematics*, vol. 10, no. 3, p. 432, 2022.
 - [6] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff, “Coco: A platform for comparing continuous optimizers in a black-box setting,” *Optimization Methods and Software*, vol. 36, no. 1, pp. 114–144, 2021.
 - [7] 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.
 - [8] 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.
 - [9] P. Bennet, C. Doerr, A. Moreau, J. Rapin, F. Teytaud, and O. Teytaud, “Nevergrad: black-box optimization platform,” *ACM SIGEVOlution*, vol. 14, no. 1, pp. 8–15, 2021.
 - [10] Y.-W. Zhang and S. K. Halgamuge, “Similarity of continuous optimization problems from the algorithm performance perspective,” in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 2949–2957.
 - [11] 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.
 - [12] M. A. Muñoz and K. Smith-Miles, “Generating new space-filling test instances for continuous black-box optimization,” *Evolutionary computation*, vol. 28, no. 3, pp. 379–404, 2020.
 - [13] T. Eftimov, G. Popovski, Q. Renau, P. Korošec, and C. Doerr, “Linear matrix factorization embeddings for single-objective optimization landscapes,” in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 775–782.
 - [14] G. Cenikj, R. D. Lang, A. P. Engelbrecht, C. Doerr, P. Korošec, and T. Eftimov, “Selector: Selecting a representative benchmark suite for reproducible statistical comparison,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. *GECCO '22*. New York, NY, USA: Association for Computing Machinery, 2022, p. 620–629. [Online]. Available: <https://doi.org/10.1145/3512290.3528809>
 - [15] F. X. Long, D. Vermetten, B. van Stein, and A. V. Kononova, “Bbob instance analysis: Landscape properties and algorithm performance across problem instances,” *arXiv preprint arXiv:2211.16318*, 2022.
 - [16] A. Nikolikj, G. Cenikj, G. Ispirova, D. Vermetten, R. D. Lang, A. P. Engelbrecht, C. Doerr, P. Korošec, and T. Eftimov, “Assessing the generalizability of a performance predictive model,” in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, ser. *GECCO '23 Companion*. New York, NY, USA: Association for Computing Machinery, 2023, p. 311–314. [Online]. Available: <https://doi.org/10.1145/3583133.3590617>
 - [17] A. Nikolikj. (2024) Cross-benchmark performance. [Online]. Available: <https://github.com/anicolik/cross-benchmark-generalizability-of-prediction-models.git>
 - [18] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, “Exploratory landscape analysis,” in *GECCO*, 2011, pp. 829–836.
 - [19] 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
 - [20] R. P. Prager and H. Trautmann, “Pflacco: Feature-based landscape analysis of continuous and constrained optimization problems in python,” *Evolutionary Computation*, pp. 1–25, 2023.
 - [21] A. Gisbrecht, A. Schulz, and B. Hammer, “Parametric nonlinear dimensionality reduction using kernel t-sne,” *Neurocomputing*, vol. 147, pp. 71–82, 2015.
 - [22] N. Henze, “A multivariate two-sample test based on the number of nearest neighbor type coincidences,” *The Annals of Statistics*, pp. 772–783, 1988.
 - [23] M. F. Schilling, “Multivariate two-sample tests based on nearest neighbors,” *Journal of the American Statistical Association*, vol. 81, no. 395, pp. 799–806, 1986.
 - [24] G. J. Székely and M. L. Rizzo, “Testing for equal distributions in high dimension,” *InterStat*, vol. 5, pp. 1–6, 2004.
 - [25] A. Leucht and M. H. Neumann, “Dependent wild bootstrap for degenerate u- and v-statistics,” *Journal of Multivariate Analysis*, vol. 117, pp. 257–280, 2013.
 - [26] J. Liang, B. Qu, P. Suganthan, and A. Hernández-Díaz, “Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization,” *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.*, 01 2013.
 - [27] J. Liang, B. Qu, and P. Suganthan, “Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization,” *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.*, 12 2013.
 - [28] J. Liang, B. Qu, P. Suganthan, and Q. Chen, “Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization,” *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.*, 2014.
 - [29] K. Dietrich and O. Mersmann, “Increasing the diversity of benchmark function sets through affine recombination,” 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. 590–602.
 - [30] M. Ghaffari, “An improved distributed algorithm for maximal independent set,” in *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2016, pp. 270–277.
 - [31] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001. [Online]. Available: <https://doi.org/10.1162/106365601750190398>
 - [32] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proc. of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.
 - [33] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
 - [34] J. Rapin and O. Teytaud, “Nevergrad - A gradient-free optimization platform,” <https://GitHub.com/FacebookResearch/Nevergrad>, 2018.
 - [35] J. de Nobel, F. Ye, D. Vermetten, H. Wang, C. Doerr, and T. Bäck, “Tohexperimenter: Benchmarking platform for iterative optimization heuristics,” *CoRR*, vol. abs/2111.04077, 2021. [Online]. Available: <https://arxiv.org/abs/2111.04077>
 - [36] R. D. Lang and A. P. Engelbrecht, “An exploratory landscape analysis-based benchmark suite,” *Algorithms*, vol. 14, no. 3, p. 78, 2021.
 - [37] B. Beachkofski and R. Grandhi, “Improved Distributed Hypercube Sampling,” in *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2002, p. 1274. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2002-1274>
 - [38] M. L. Rizzo and G. J. Székely, *energy: E-Statistics: Multivariate Inference via the Energy of Data*, 2016, r package version 1.7-0. [Online]. Available: <https://CRAN.R-project.org/package=energy>
 - [39] G. Petelin, G. Cenikj, and T. Eftimov, “Tinytla: Topological landscape analysis for optimization problem classification in a limited sample setting,” *Swarm and Evolutionary Computation*, p. 101448, 2023.
 - [40] R. P. Prager, M. V. Seiler, H. Trautmann, and P. Kerschke, “Automated algorithm selection in single-objective continuous optimization: a comparative study of deep learning and landscape analysis methods,” in *International Conference on Parallel Problem Solving from Nature*. Springer, 2022, pp. 3–17.