Cenikj, Gjorgjina, Ana Nikolikj, and Tome Eftimov. "Recent Advances in Meta-features Used for Representing Black-box Single-objective Continuous Optimization." *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 2025.

GECCO 2025 Tutorial:
Recent Advances in Meta-features for Representing Black-box Single-objective Continuous Optimization

Gjorgjina Cenikj, Ana Nikolikj, Tome Eftimov
Computer Systems Department, Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
{gjorgjina.cenikj, ana.nikolikj, tome.eftimov}@ijs.si

http://gecco-2025.sigevo.org/

Instructors

Gjorgjina Cenikj is a young researcher at the Computer Systems Department at the Jožef Stefan Institute in Ljubljana, Slovenia. She is currently pursuing a PhD degree at the Jožef Stefan Postgraduate School, targeting the development of representation learning methodologies for single-objective numerical optimization problems and algorithms, with the goal of improving algorithm selection. She has been a major contributor to several feature-based characterizations of optimization problems TransOpt, DynamoRep, TLA, and TinyTLA. Her research interests include machine learning, representation learning, natural language processing, and graph learning.

Ana Nikolikj is a young researcher at the Computer Systems Department at the Jožef Stefan Institute in Ljubljana, Slovenia. She is working towards her PhD at the Jožef Stefan Postgraduate School, focusing on inventing methodologies to understand the behavior of single-objective numerical optimization algorithms via meta-learning. This is aimed at enhancing the process of algorithm performance prediction and algorithm selection. Her areas of interest encompass machine learning, representation learning, and methods for explainability. During her master thesis, she explored algorithm features based on explainable performance prediction models.

Tome Eftimov is a researcher at the Computer Systems Department at the Jožef Stefan Institute. He is a visiting assistant professor at the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Skopje. He was a postdoctoral research fellow at Stanford University, USA, and a research associate at the University of California, San Francisco. He obtained his PhD in Information and Communication Technologies (2018). His research interests include statistical data analysis, meta-learning, metaheuristics, natural language processing, representation learning, and machine learning. His work related to Deep Statistical Comparison was presented as a tutorial (i.e. IJCCI 2018, IEEE SSCI 2019, GECCO 2020, 2021, 2022, 2024, PPSN 2020, 2022, IEEE CEC 2021, 2022, 2023, AutoML 2022) or as an invited lecture to several international conferences and universities. He is an organizer of several workshops related to AI at high-ranked international conferences. He was a coordinator of a national project “Mr-BEC: Modern approaches for benchmarking in evolutionary computation”, a coordinator of a national project “Representation Learning of Landscape Spaces for Explainable Performance of Stochastic Optimization Algorithms”, and actively participates in European projects. Currently, he is a project coordinator of HE Era-Chair AutoLearn-SI.

Course Agenda

Introduction

Calculating/Learning problem landscape features

Calculating/Learning algorithm features

Calculating/learning high-level problem-algorithm interaction features

Calculating/Learning trajectory-based features

Comparing the meta-features for automated algorithm selection

Take home messages

Motivation

Motivation

- Automated algorithm selection requires the application of machine learning (ML) models to the optimization domain.

- Our work lies at the intersection of optimization and ML

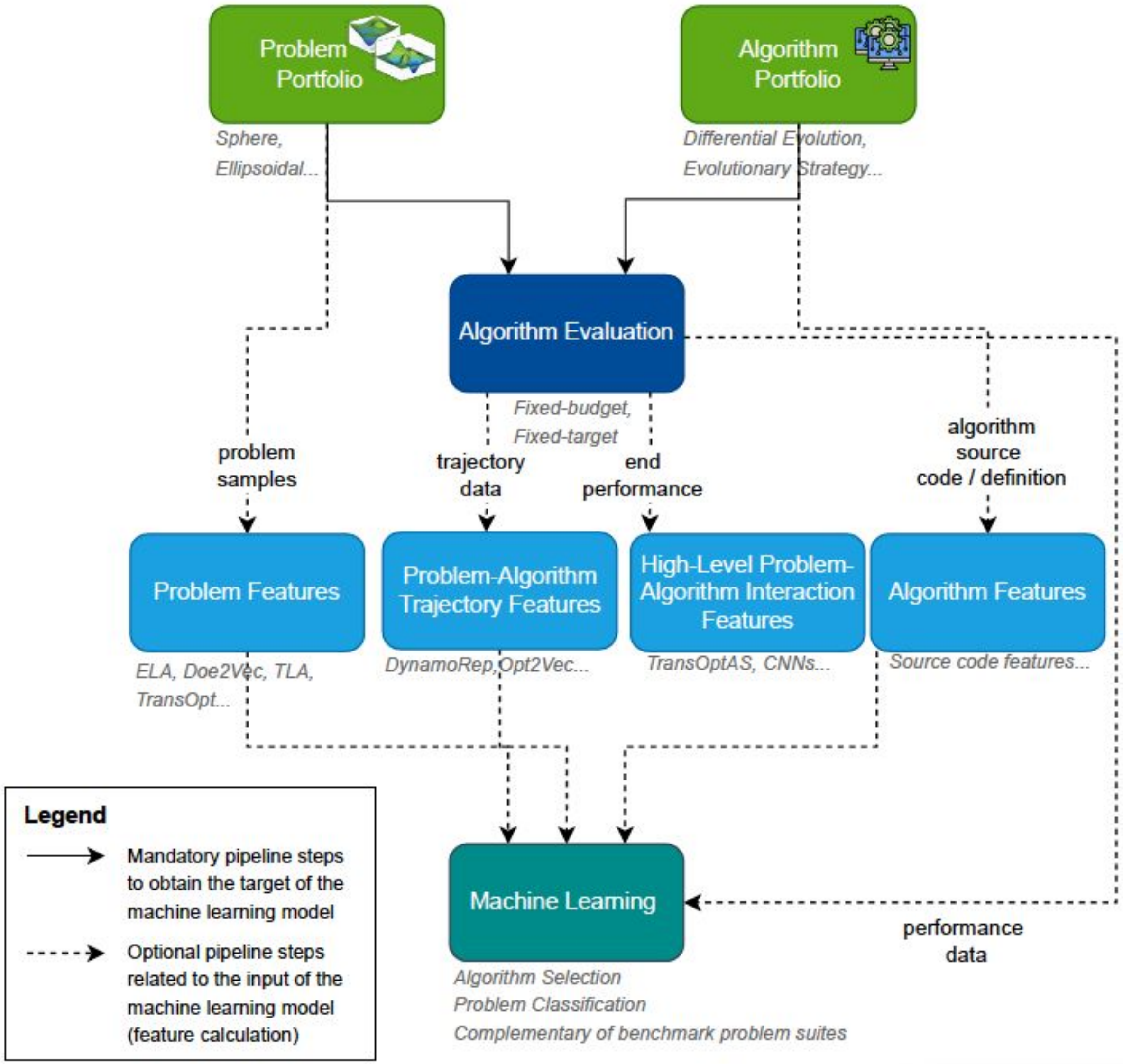
- Most of the GECCO community is familiar with the ELA features, but other feature representations have been proposed in recent years

Algorithm Selection

- Aims to identify the best algorithm (from an existing set of algorithms) to solve a given problem.

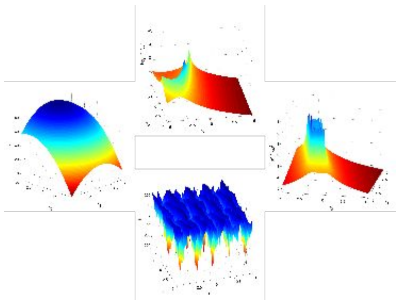
- Leverage algorithm complementarity instead of looking for a single algorithm which works best across all problems.

Algorithm Selection Pipeline



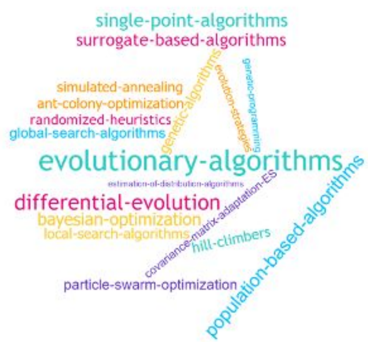
Algorithm Selection in Numerical Black-Box Optimization

Selection of a problem portfolio



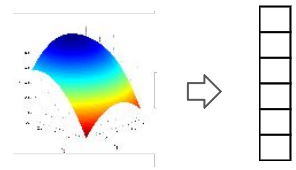
What types of optimization problems will be included in our problem portfolio?

Selection of an algorithm portfolio



Which algorithms will be incorporated into our algorithm portfolio?

Feature Representation



- Problem features
- Algorithm features
- High-level problem-algorithm interaction features
- Problem-algorithm trajectory features

How do we represent optimization problems and algorithms in vector form?

Algorithm selector (AS)

AS approaches:
- (Pairwise-)regression
- (Pairwise-)classification
- …

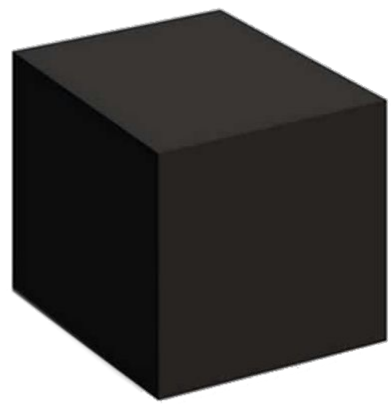
ML methods:
- RandomForest
- XGBoost
- TabPFN
- FTTransformer
- …

No significant difference in performance of different ML models and AS approaches for BBOB!!!

Problem Features

Exploratory Landscape Analysis (ELA)
Topological Landscape Analysis (TLA)
Random Filter Mappings
Deep Learning-Based Features

How do you characterize a black-box function?



Problem Features

- Calculated on the basis of a problem sample

- Candidate solutions are artificially sampled using some sampling technique (Random Sampling, Latin Hypercube Sampling, Sobol Sampling)

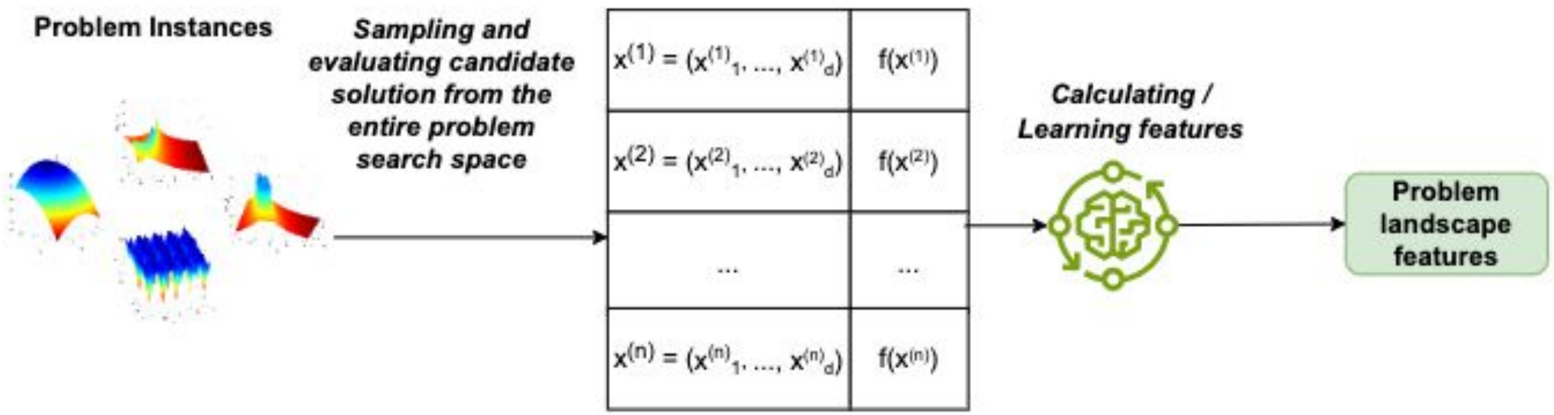
- Each candidate solution is evaluated to obtain its objective function value

DIMENSION + 1

SAMPLE SIZE

$x^1_{1\_1}$	$x^2_{1\_1}$	y_{1_1}
$x^1_{1\_2}$	$x^2_{1\_2}$	y_{1_2}
$x^1_{1\_3}$	$x^2_{1\_3}$	y_{1_3}

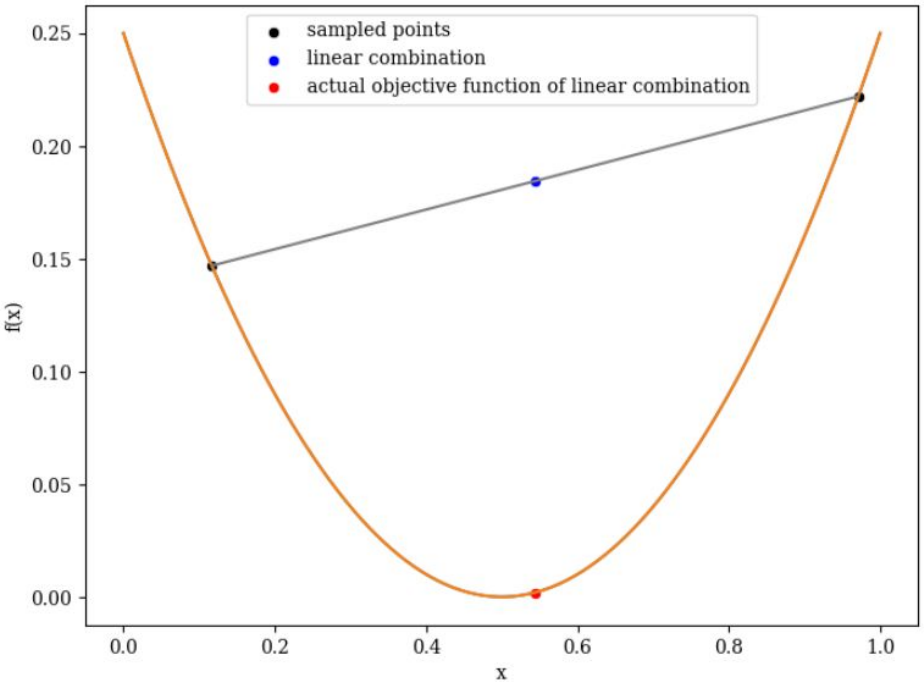
Problem features



Exploratory Landscape Analysis

ELA - Convexity features

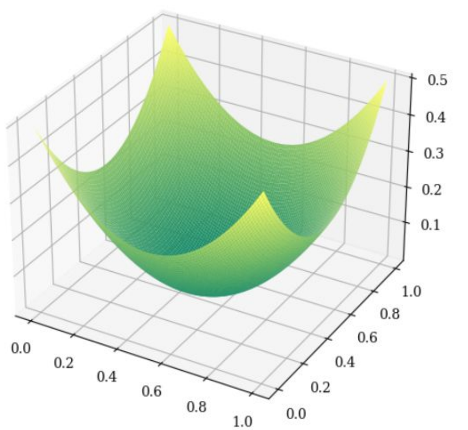
- Based on differences in the objective values of a point which is a linear combination of two randomly sampled points and the convex combination of the objective values of the two sampled points.

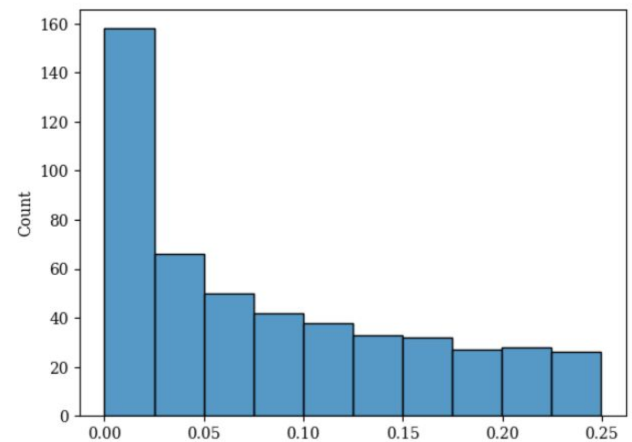


Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., & Rudolph, G. (2011, July 12). Exploratory landscape analysis. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. Presented at the GECCO '11: Genetic and Evolutionary Computation Conference, Dublin Ireland. doi:10.1145/2001576.2001690

ELA - Distribution features

Skewness, kurtosis, and the number of peaks of the distribution of the objective function values, based on a kernel density estimation of the initial design objective values.

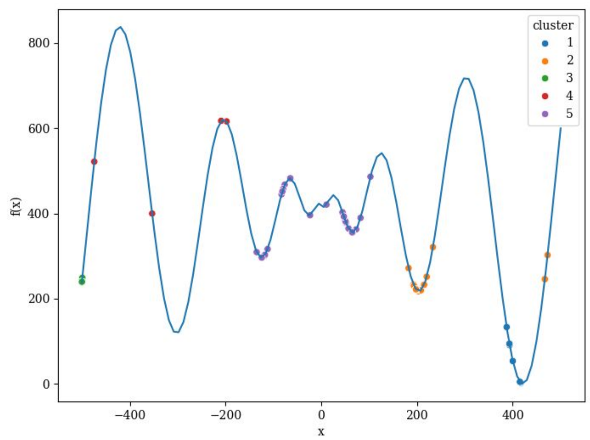




Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., & Rudolph, G. (2011, July 12). Exploratory landscape analysis. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. Presented at the GECCO ’11: Genetic and Evolutionary Computation Conference, Dublin Ireland. doi:10.1145/2001576.2001690

ELA - Local search features

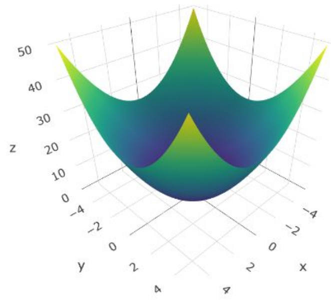
- Local search features - extracted by running a local search algorithm and hierarchically clustering the considered solutions in order to approximate problem properties. For instance, the number of clusters is an indicator of multi-modality, while the cluster sizes are related to the basin sizes around the local optima.



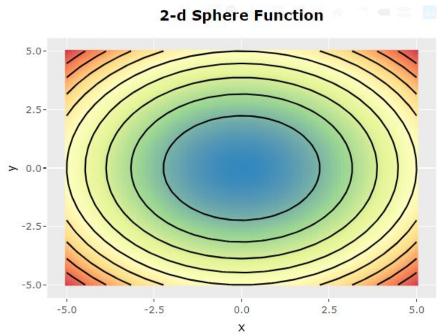
Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., & Rudolph, G. (2011, July 12). Exploratory landscape analysis. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. Presented at the GECCO ’11: Genetic and Evolutionary Computation Conference, Dublin Ireland. doi:10.1145/2001576.2001690

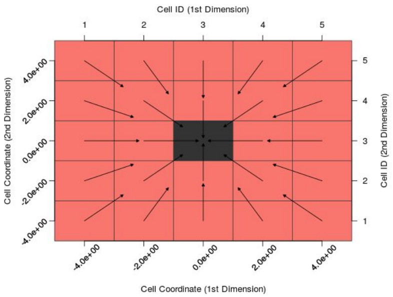
ELA - Cell mapping features

- Discretize the continuous decision space using a predefined number of blocks
- Categorizes cells into attractor, periodic, transient and uncertain cells



Example features:
gcm.mean.pcells = 0.04 (relative number of periodic cells)
gcm.mean.tcells = 0.96 (relative number of transient cells)
gcm.mean.best_attr.prob = 1 (probability of finding the attractor with the best objective value)



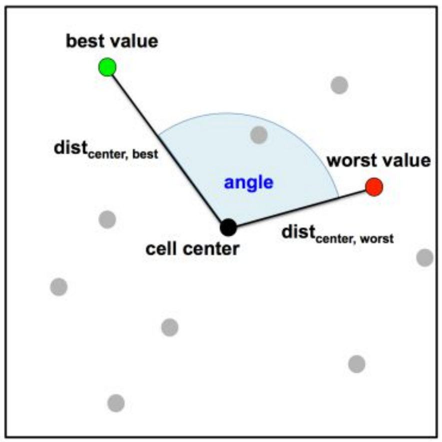


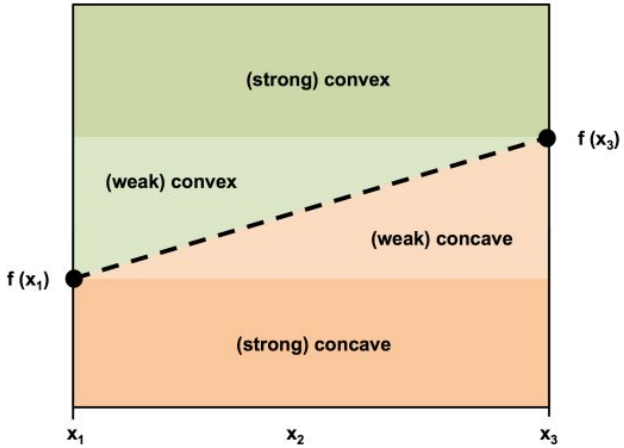
Kerschke, P., Preuss, M., Hernández, C., Schütze, O., Sun, J.-Q., Grimme, C., Rudolph, G., Bischl, B., & Trautmann, H. (2014). Cell Mapping Techniques for Exploratory Landscape Analysis. In Advances in Intelligent Systems and Computing (pp. 115–131). Springer International Publishing. https://doi.org/10.1007/978-3-319-07494-8_9

ELA - Cell mapping features

Angle features: Take into consideration the angle between the vectors connecting the center of each cell to the best and worst value within a cell

Comparing three neighbouring cells allows to draw conclusions on the local convexity





Kerschke, P., Preuss, M., Hernández, C., Schütze, O., Sun, J.-Q., Grimme, C., Rudolph, G., Bischl, B., & Trautmann, H. (2014). Cell Mapping Techniques for Exploratory Landscape Analysis. In Advances in Intelligent Systems and Computing (pp. 115–131). Springer International Publishing. https://doi.org/10.1007/978-3-319-07494-8_9

Other ELA feature groups

- **Levelset features** split samples into two classes based on whether the value of the objective function falls above or below a certain threshold. Linear, quadratic, and mixture discriminant analysis is used to predict whether the objective values fall below or exceed the calculated threshold. The intuition behind this is that multi-modal functions should result in several unconnected sublevel sets for the quantile of lower values, which can only be modeled by the mixture discriminant analysis method. The extracted low-level features are based on the distribution of the resulting misclassification errors of each classifier.
- **Metamodel features** - fit regression models to the sampled data and use the coefficients and accuracy of the model to describe the problem
- **Curvature features** estimate the gradient and Hessians from samples of the function and use their magnitudes to quantify the curvature

ELA feature calculation - Flacco

- Available as a python and R package
- Flacco GUI: https://flacco.shinyapps.io/flacco/

Kerschke, P., & Trautmann, H. (2019). Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-Package Flacco. In Studies in Classification, Data Analysis, and Knowledge Organization (pp. 93–123). Springer International Publishing. https://doi.org/10.1007/978-3-030-25147-5_7

Prager, R. P., & Trautmann, H. (2024). Pflacco: Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems in Python. In Evolutionary Computation (pp. 1–6). MIT Press. https://doi.org/10.1162/evco_a_00341

Topological Landscape Analysis

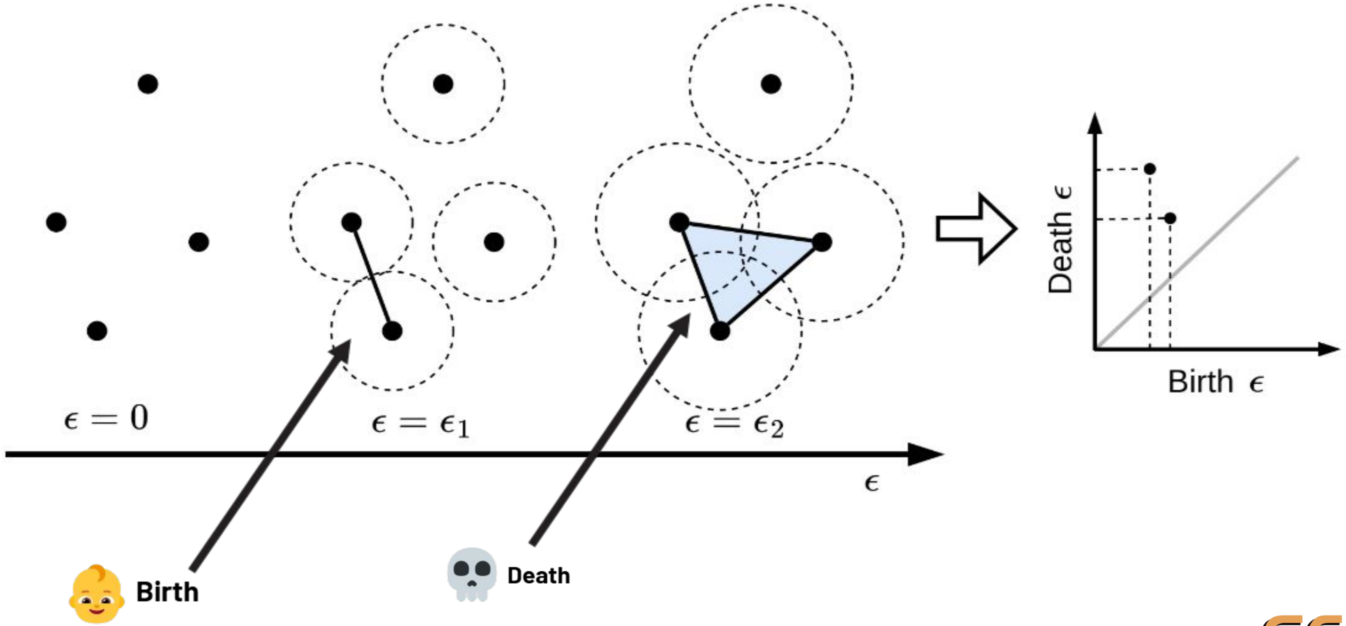
Topological Landscape Analysis (TLA)

- Uses methods from Topological Data Analysis to extract features
- Captures the existence of different topological structures in a point cloud
- Process:
  - Sampling
  - Pairwise calculation of distances between samples
  - Generation of persistence diagram and image

Petelin, G., Cenikj, G., & Eftimov, T. (2024). TinyTLA: Topological landscape analysis for optimization problem classification in a limited sample setting. Swarm and Evolutionary Computation, 84(101448), 101448. doi:10.1016/j.swevo.2023.101448

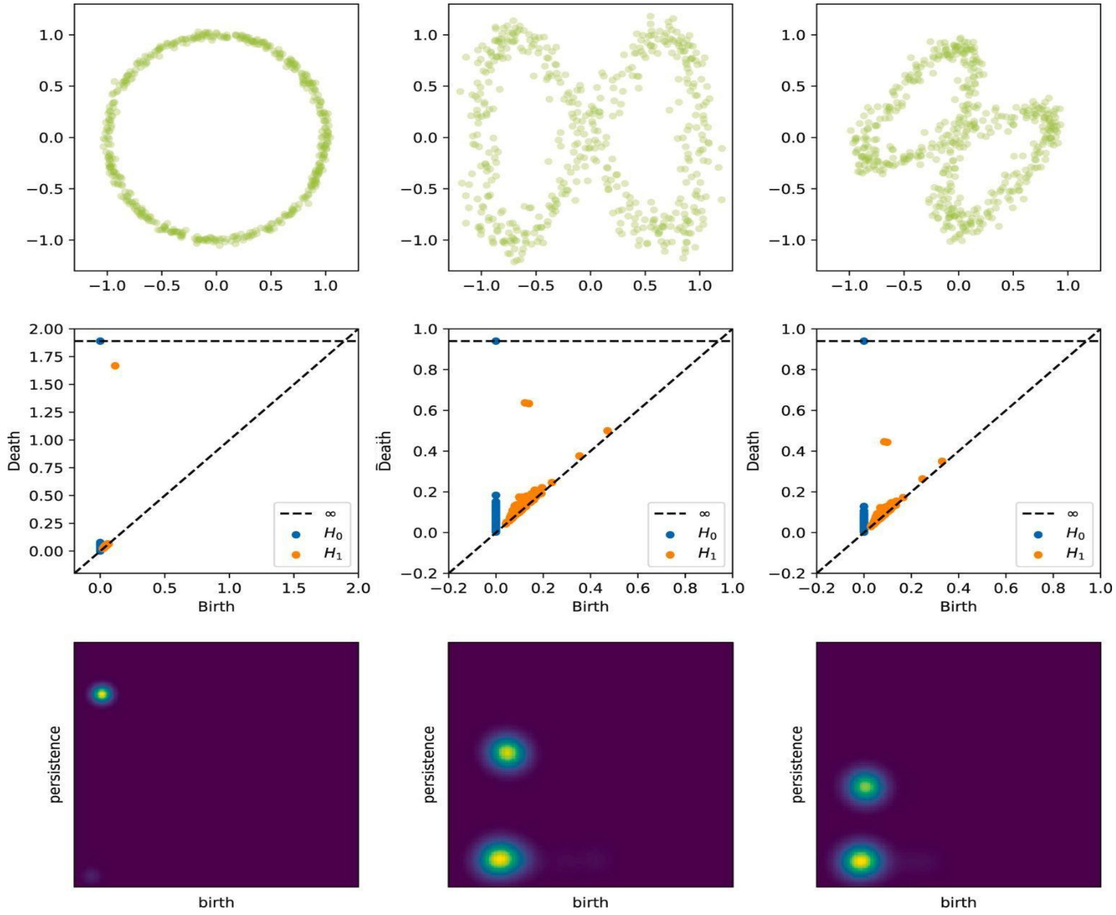
Topological Landscape Analysis

- Captures the existence of different topological structures in a point cloud



Petelin, G., Cenikj, G., & Eftimov, T. (2024). TinyTLA: Topological landscape analysis for optimization problem classification in a limited sample setting. Swarm and Evolutionary Computation, 84(101448), 101448. doi:10.1016/j.swevo.2023.101448

Topological Landscape Analysis



Petelin, G., Cenikj, G., & Eftimov, T. (2024). TinyTLA: Topological landscape analysis for optimization problem classification in a limited sample setting. Swarm and Evolutionary Computation, 84(101448), 101448. doi:10.1016/j.swevo.2023.101448

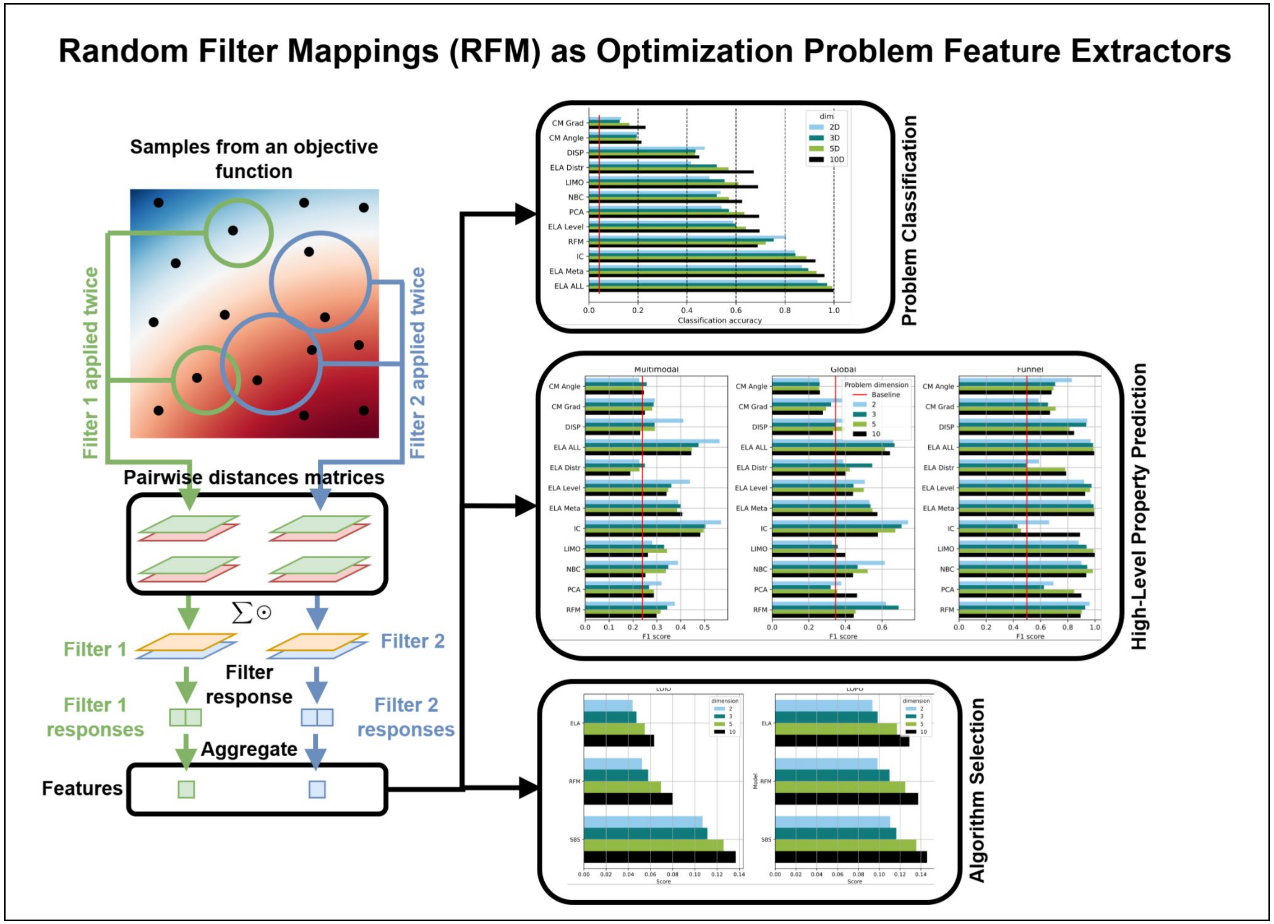
Random Filter Mappings

Random Filter Mappings

- **Objective**: Generate versatile problem features using randomly initialized, domain-specific filters
- **Filter Application**: Apply filters to problem samples to extract key properties
- **Feature Space**: Embed instances in high-dimensional feature space—similar problems cluster together
- **Capability**: Detect complex function traits (e.g., multimodality, global/funnel structures)
- **Use Case**: Classify benchmark instances and guide algorithm selection via meta-models
- **Key Insight**: Features work well for selection when new problems resemble training set
- **Benefit**: A flexible tool for feature extraction across multiple optimization tasks

Petelin, G., & Cenikj, G. (2024). Random Filter Mappings as Optimization Problem Feature Extractors. *IEEE Access*.

Random Filter Mappings



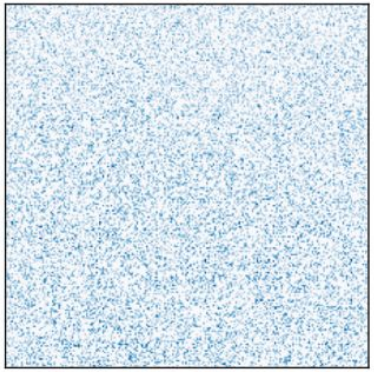
Petelin, G., & Cenikj, G. (2024). Random Filter Mappings as Optimization Problem Feature Extractors. *IEEE Access*.

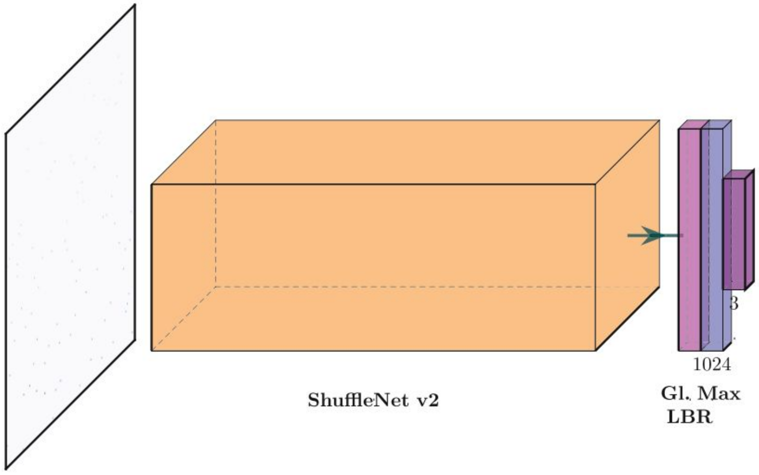
Deep Learning-based Features

Fitness map features
Doe2Vec
TransOptAS
DeepELA

Fitness map features

- Represent problem samples as a fitness map - 2D single-channel image
- Model: CNN (ShuffleNet v2)

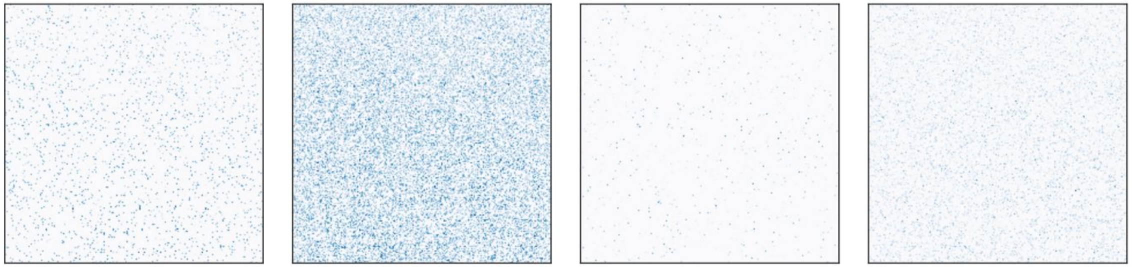




Prager, R. P., Vinzent Seiler, M., Trautmann, H., & Kerschke, P. (2021). Towards Feature-Free Automated Algorithm Selection for Single-Objective Continuous Black-Box Optimization. In 2021 IEEE Symposium Series on Computational Intelligence (SSCI). 2021 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE. https://doi.org/10.1109/ssci50451.2021.9660174

Fitness map features - Extension to higher dimensions

- Adaptation of the fitness map approach for high-dimensional data using dimensionality reduction techniques (PCA/Multi channel)
- Task: Evaluated for the task of predicting high-level features of BBOB problem instances
- Data: BBOB benchmark, 150 instances per problem, D = {2, 3, 5, 10}.
- Weakness: trade-off between information loss for larger dimensions or growing sparsity for smaller one

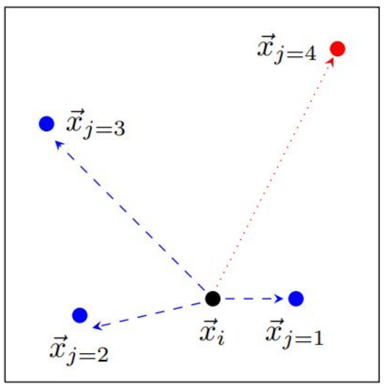


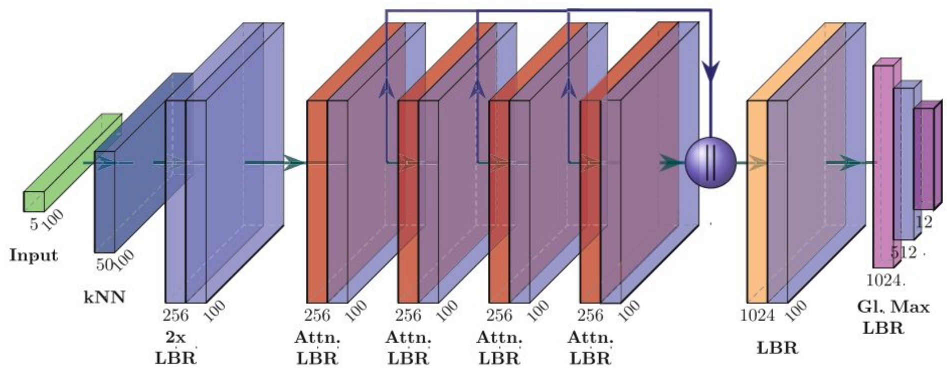
a) $5D$: reduced MC b) $10D$: reduced MC c) $5D$: Conv-rMC d) $10D$: Conv-rMC

Seiler, M. V., Prager, R. P., Kerschke, P., & Trautmann, H. (2022). A collection of deep learning-based feature-free approaches for characterizing single-objective continuous fitness landscapes. Proceedings of the Genetic and Evolutionary Computation Conference, 657–665. Presented at the Boston, Massachusetts. doi:10.1145/3512290.3528834

Fitness map features - Extension to higher dimensions

- Exploration of Point Cloud Transformers
- Modified point cloud transformers to operate on the node of the kNN-graph, embedding each candidate solution into its local neighborhood
- Shown to work for AS, however, not as well as ELA features





Seiler, M. V., Prager, R. P., Kerschke, P., & Trautmann, H. (2022). A collection of deep learning-based feature-free approaches for characterizing single-objective continuous fitness landscapes. Proceedings of the Genetic and Evolutionary Computation Conference, 657–665. Presented at the Boston, Massachusetts. doi:10.1145/3512290.3528834

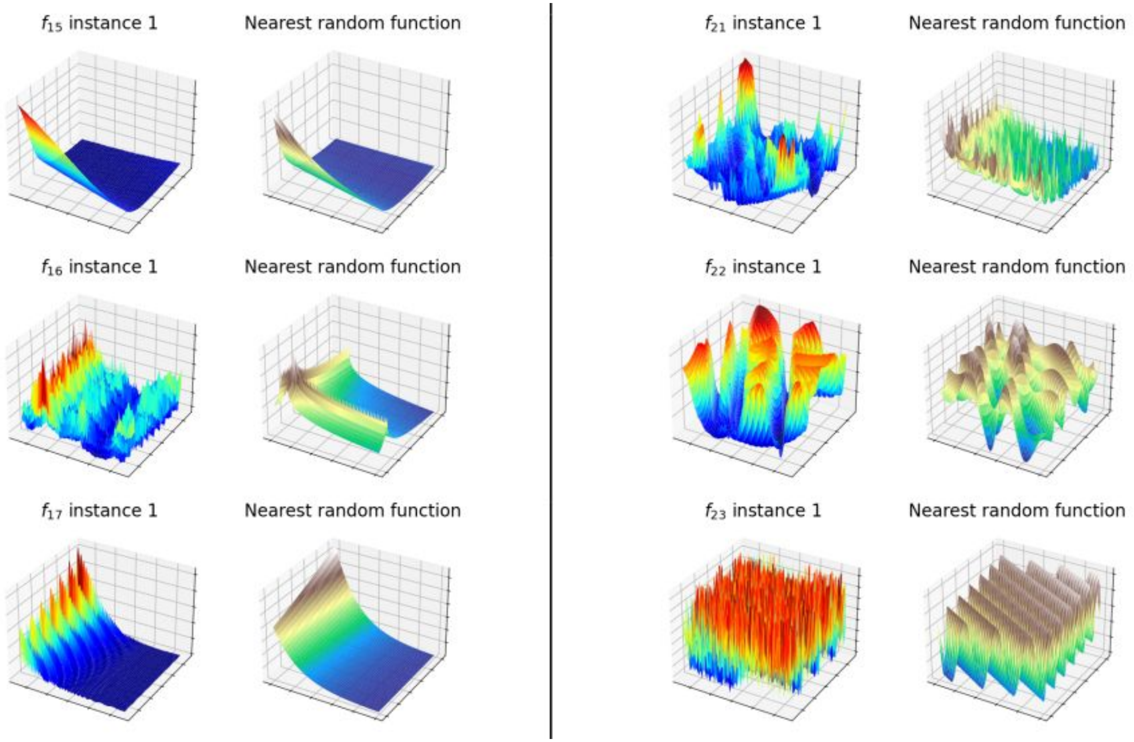
Doe2Vec

- Process:
  - Generate candidate solutions using Latin Hypercube / Sobol sampling
  - Objective solution values are re-scaled within the range of [0,1] and used as input features to train the VAE
- Data: Functions generated using a random function generator
- Task: Predicting high-level properties of BBOB problem instances (multimodality, global structure, funnel structure, etc).

Van Stein, B., Long, F. X., Frenzel, M., Krause, P., Gitterle, M., & Bäck, T. (2023). DoE2Vec: Deep-learning Based Features for Exploratory Landscape Analysis. Proceedings of the Companion Conference on Genetic and Evolutionary Computation, 515–518. Presented at the Lisbon, Portugal. doi:10.1145/3583133.3590609

Doe2Vec

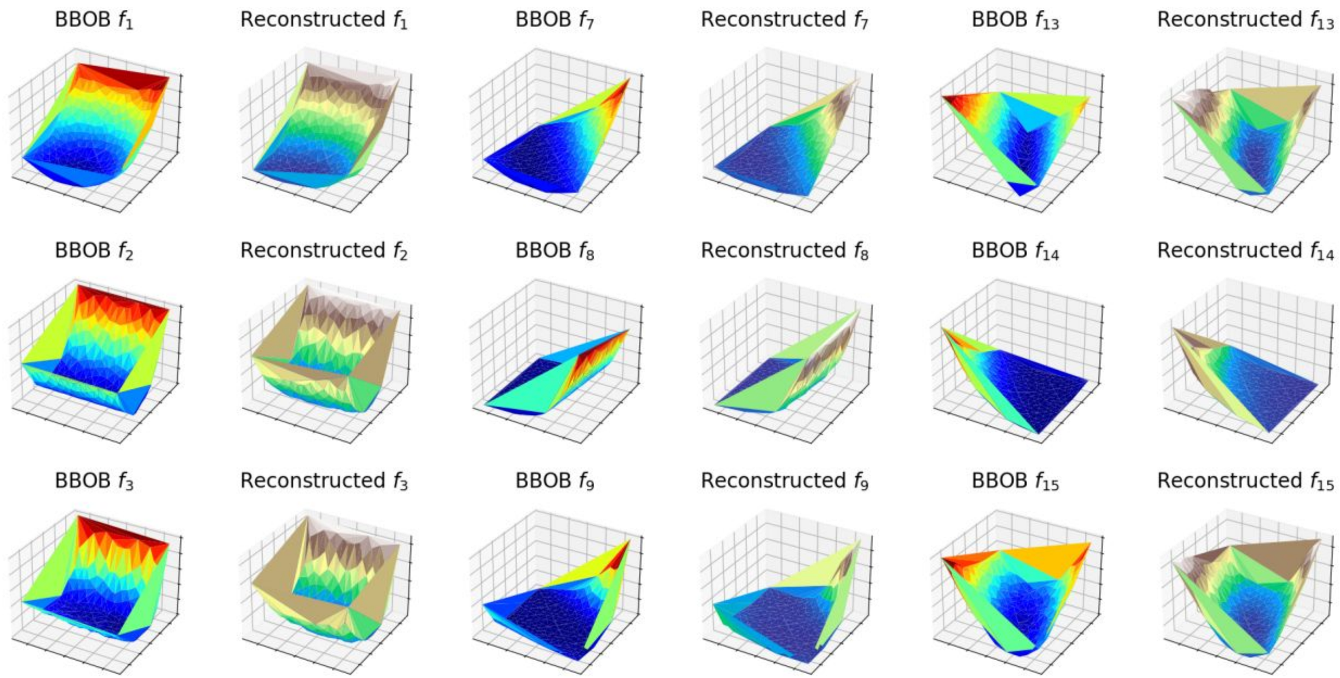
BBOB functions and their most similar random function in terms of Doe2Vec features



Van Stein, B., Long, F. X., Frenzel, M., Krause, P., Gitterle, M., & Bäck, T. (2023). DoE2Vec: Deep-learning Based Features for Exploratory Landscape Analysis. Proceedings of the Companion Conference on Genetic and Evolutionary Computation, 515–518. Presented at the Lisbon, Portugal. doi:10.1145/3583133.3590609

Doe2Vec

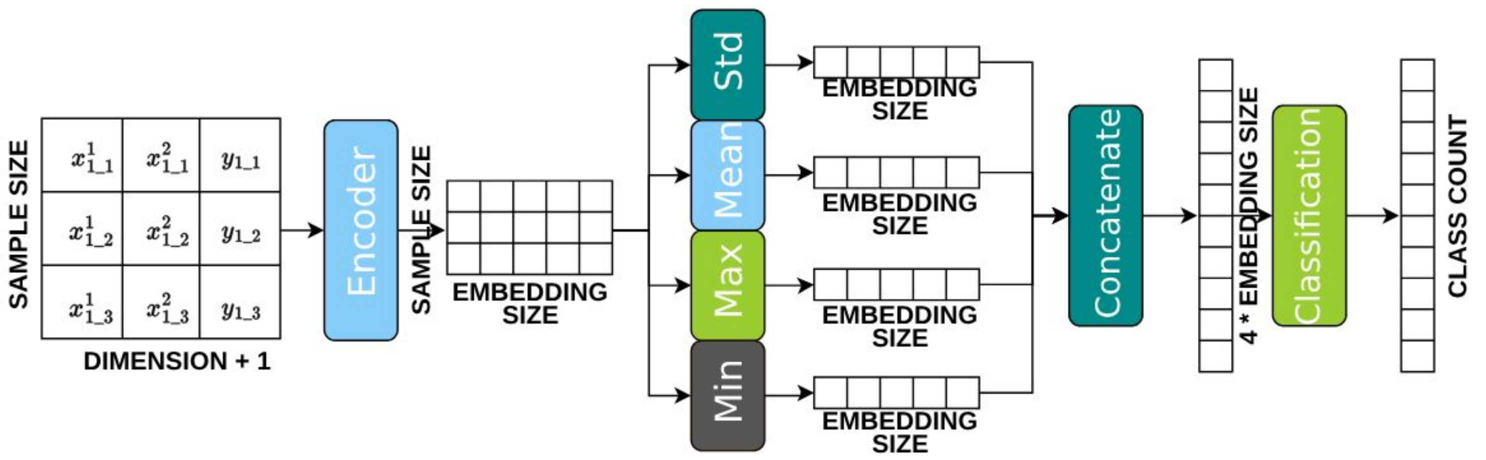
Reconstructions of 2D BBOB function using the Doe2Vec VAE



Van Stein, B., Long, F. X., Frenzel, M., Krause, P., Gitterle, M., & Bäck, T. (2023). DoE2Vec: Deep-learning Based Features for Exploratory Landscape Analysis. Proceedings of the Companion Conference on Genetic and Evolutionary Computation, 515–518. Presented at the Lisbon, Portugal. doi:10.1145/3583133.3590609

TransOpt

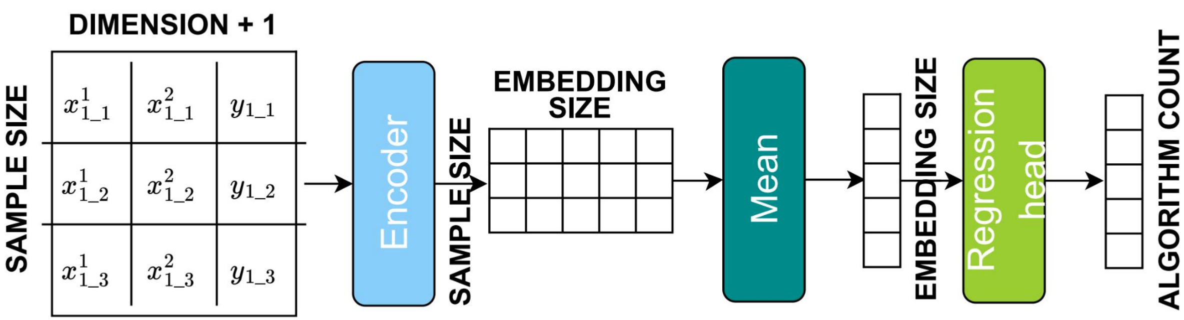
- Process:
  - Generate candidate solutions using Latin Hypercube sampling
  - Train transformer model, which given samples of the optimization function, predicts which of the 24 BBOB problem classes the samples belong to
- Data: BBOB benchmark
- Task: Problem classification



Cenikj, G., Petelin, G., & Eftimov, T. (2023). TransOpt: Transformer-based Representation Learning for Optimization Problem Classification. IEEE Symposium Series on Computational Intelligence (SSCI) https://arxiv.org/pdf/2311.18035

TransOptAS

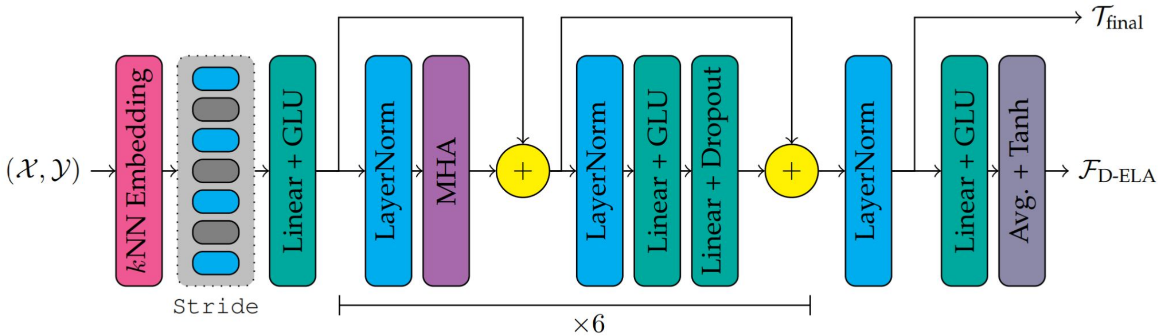
- Process:
  - Generate candidate solutions using Latin Hypercube sampling
  - Train transformer model, which given samples of the optimization function, predicts algorithm performance
- Data: Functions generated using a random function generator
- Task: Algorithm selection



Cenikj, G., Petelin, G., & Eftimov, T. (2024). TransOptAS: Transformer-Based Algorithm Selection for Single-Objective Optimization. In Proceedings of the Genetic and Evolutionary Computation Conference Companion (pp. 403–406). GECCO ’24 Companion: Genetic and Evolutionary Computation Conference Companion. ACM. https://doi.org/10.1145/3638530.3654191

DeepELA

- Process:
  - Generate candidate solutions using Latin Hypercube sampling
  - Self-supervised training of transformer model to produce representations of optimization problems which are invariant to problem transformations
- Data: Functions generated using a random function generator
- Tasks: Predicting high-level properties of BBOB problems; Algorithm selection

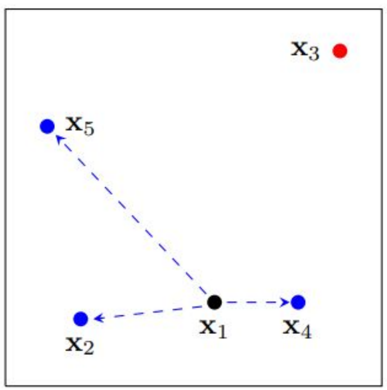


Seiler, M. V., Kerschke, P., & Trautmann, H. (2025). Deep-ela: Deep exploratory landscape analysis with self-supervised pretrained transformers for single-and multi-objective continuous optimization problems. Evolutionary Computation, 1-27.

DeepELA

The input undergoes a k-Nearest-Neighborhood (kNN) embedding with the goal of incorporating the local neighborhood of every $x_i \in X$

A token is every member of $x \in X$ alongside its k nearest neighbors



$$\mathbf{t}_1 = \big(\overbrace{\mathbf{x}_1, \mathbf{y}_1}^{\text{Global Context}}, (\mathbf{x}_4 - \mathbf{x}_1), (\mathbf{y}_4 - \mathbf{y}_1), (\mathbf{x}_2 - \mathbf{x}_1), (\mathbf{y}_2 - \mathbf{y}_1) \ldots\big)^{\mathrm{T}}$$
$$\mathbf{t}_2 = \big(\mathbf{x}_2, \mathbf{y}_2, (\mathbf{x}_1 - \mathbf{x}_2), (\mathbf{y}_1 - \mathbf{y}_2), (\mathbf{x}_4 - \mathbf{x}_2), (\mathbf{y}_4 - \mathbf{y}_2) \ldots\big)^{\mathrm{T}}$$
$$\mathbf{t}_3 = \big(\mathbf{x}_3, \mathbf{y}_3, (\mathbf{x}_4 - \mathbf{x}_3), (\mathbf{y}_4 - \mathbf{y}_3), (\mathbf{x}_5 - \mathbf{x}_3), (\mathbf{y}_5 - \mathbf{y}_3) \ldots\big)^{\mathrm{T}}$$
$$\mathbf{t}_4 = \big(\mathbf{x}_4, \mathbf{y}_4, (\mathbf{x}_1 - \mathbf{x}_4), (\mathbf{y}_1 - \mathbf{y}_4), (\mathbf{x}_2 - \mathbf{x}_4), (\mathbf{y}_2 - \mathbf{y}_4) \ldots\big)^{\mathrm{T}}$$
$$\mathbf{t}_5 = \big(\mathbf{x}_5, \mathbf{y}_5, \underbrace{(\mathbf{x}_2 - \mathbf{x}_5), (\mathbf{y}_2 - \mathbf{y}_5), (\mathbf{x}_1 - \mathbf{x}_5), (\mathbf{y}_1 - \mathbf{y}_5) \ldots}_{\text{Local Context}}\big)^{\mathrm{T}}$$

Find kNN of $\mathbf{x}_i \in X$ and create tokens $\mathbf{t}_i \in T$

Seiler, M. V., Kerschke, P., & Trautmann, H. (2025). Deep-ela: Deep exploratory landscape analysis with self-supervised pretrained transformers for single-and multi-objective continuous optimization problems. Evolutionary Computation, 1-27.

DeepELA

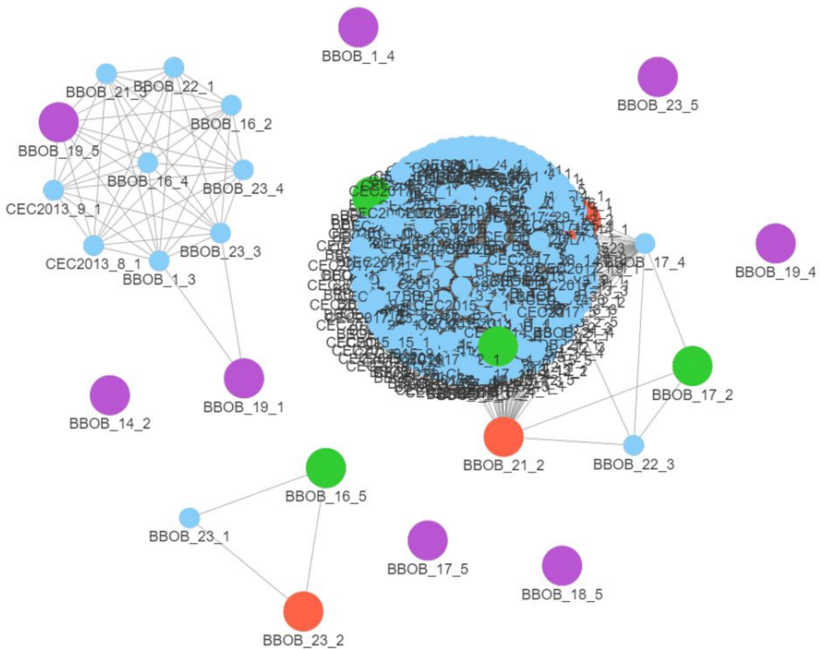
- Student-teacher training strategy with a shared backbone acting as a feature generator
- The training strategy revolves around providing distinct, augmented versions of the same objective instance to both the teacher and student. Here, the teacher generates target projections from which the student gleans insights.
- The loss function aims to maximize the covariance between an instance's online- and target projection and to minimize it between different instances

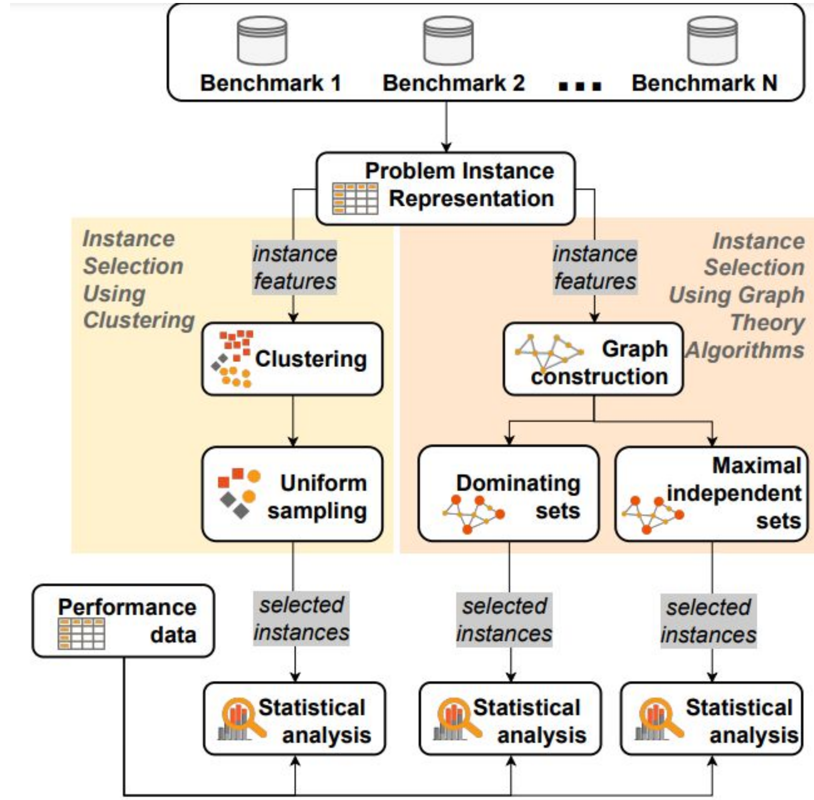
Seiler, M. V., Kerschke, P., & Trautmann, H. (2025). Deep-ela: Deep exploratory landscape analysis with self-supervised pretrained transformers for single-and multi-objective continuous optimization problems. Evolutionary Computation, 1-27.

Application of problem features

Selection of diverse benchmarking problem instances
Per-instance algorithm selection
Explainable algorithm footprint

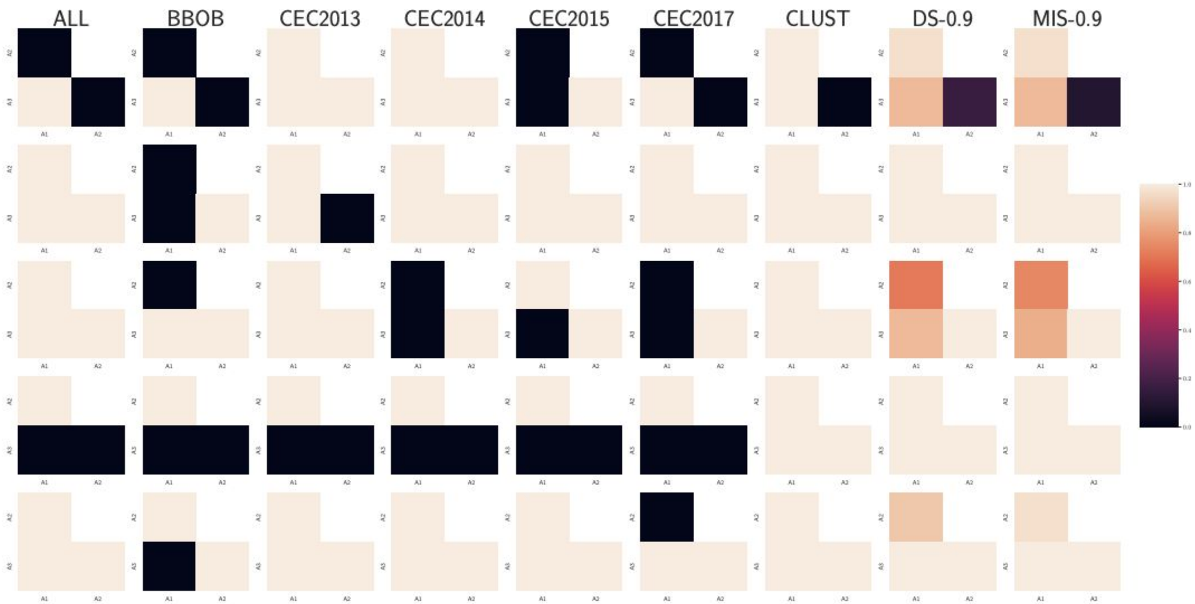
SELECTOR - Selection of diverse benchmark problem instances





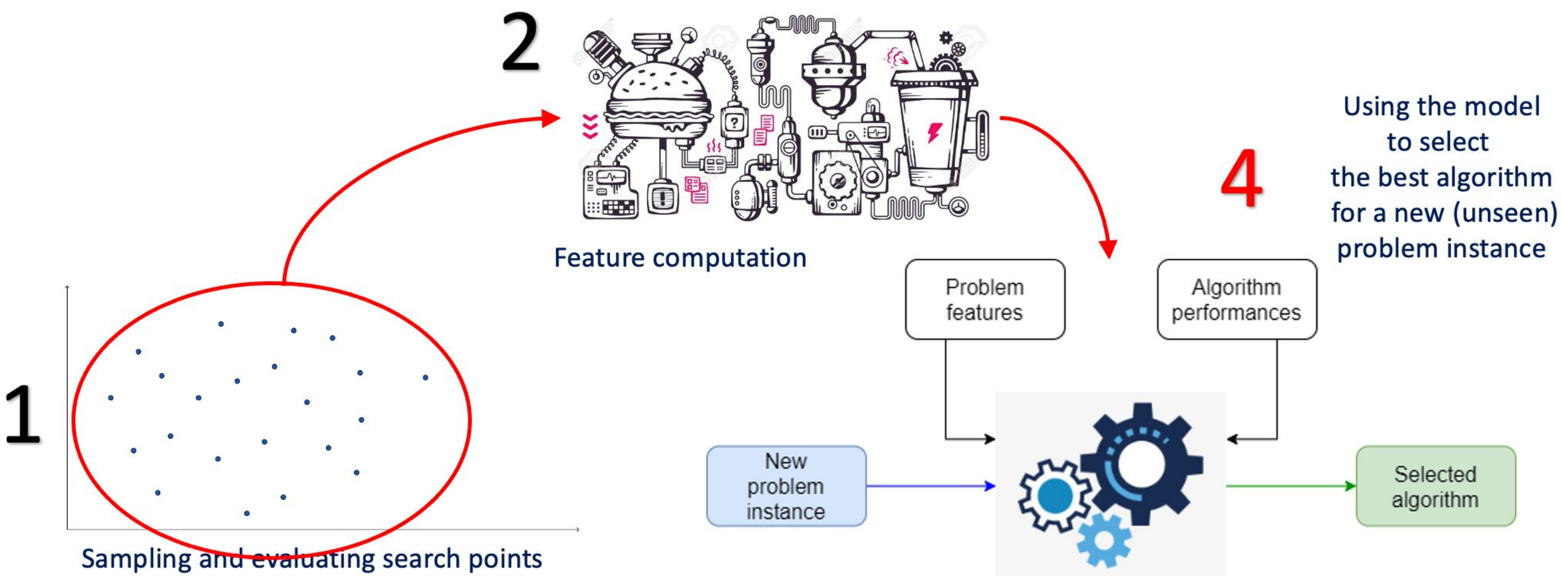
Cenikj, G., Lang, R. D., Engelbrecht, A. P., Doerr, C., Korošec, P., & Eftimov, T. (2022, July). Selector: selecting a representative benchmark suite for reproducible statistical comparison. In *Proceedings of The Genetic and Evolutionary Computation Conference* (pp. 620-629).

Generalization of statistical results

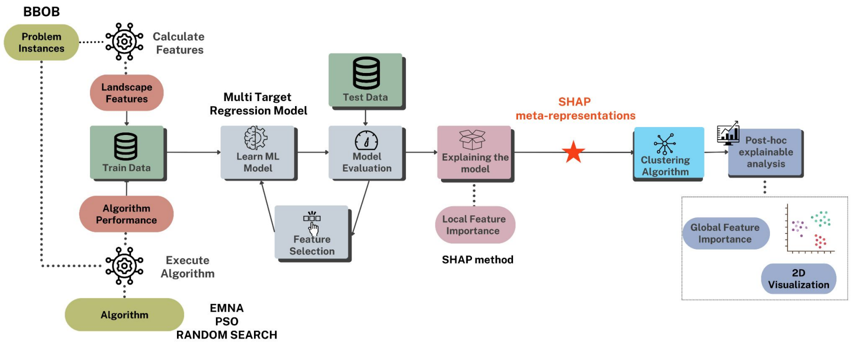


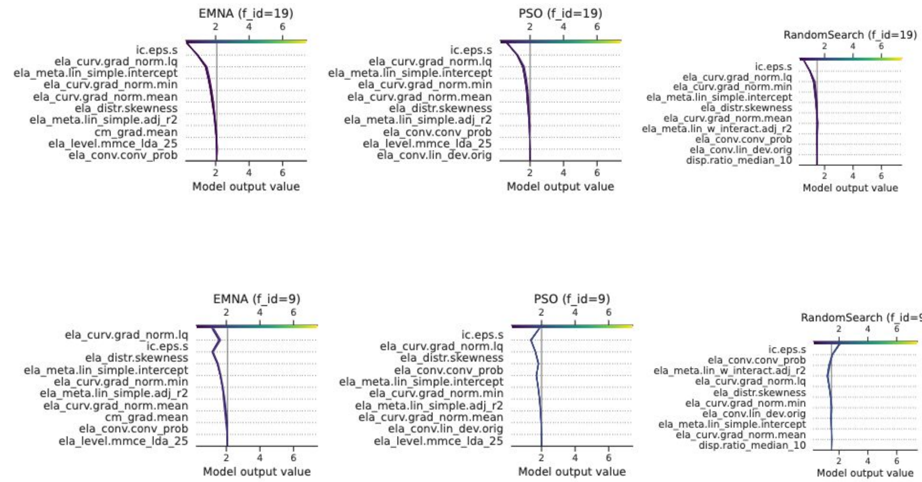
Cenikj, G., Lang, R. D., Engelbrecht, A. P., Doerr, C., Korošec, P., & Eftimov, T. (2022, July). Selector: selecting a representative benchmark suite for reproducible statistical comparison. In *Proceedings of The Genetic and Evolutionary Computation Conference* (pp. 620-629).

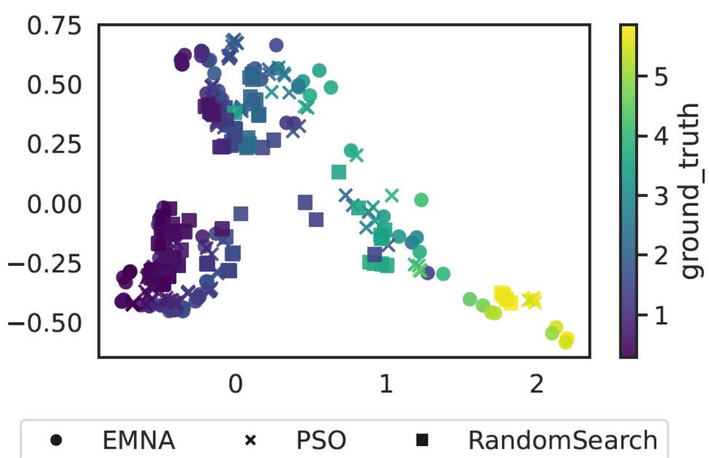
Per-Instance Algorithm Selection



Explainable Algorithm Footprint







Nikolikj, A., Munoz, M. A., & Eftimov, T. (2025). Benchmarking footprints of continuous black-box optimization algorithms: Explainable insights into algorithm success and failure. *Swarm and Evolutionary Computation, 94, 101895.*

Nikolikj, A., & Eftimov, T. (2024, July). Comparing Solvability Patterns of Algorithms across Diverse Problem Landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 143-146).

Nikolikj, A., Džeroski, S., Muñoz, M. A., Doerr, C., Korošec, P., & Eftimov, T. (2023, July). Algorithm Instance Footprint: Separating Easily Solvable and Challenging Problem Instances. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 529-537).

Algorithm Features

Based on source code

Algorithm features based on source code

Extracting algorithm features from source code

Pros: May be used to compare different programing implementation of the algorithms and further investigate which one has better performance

Cons:

- **Parameter Sensitivity**: These features are ineffective for automated algorithm configuration or parameter tuning, as parameter differences are typically evident only during execution, not in the code.

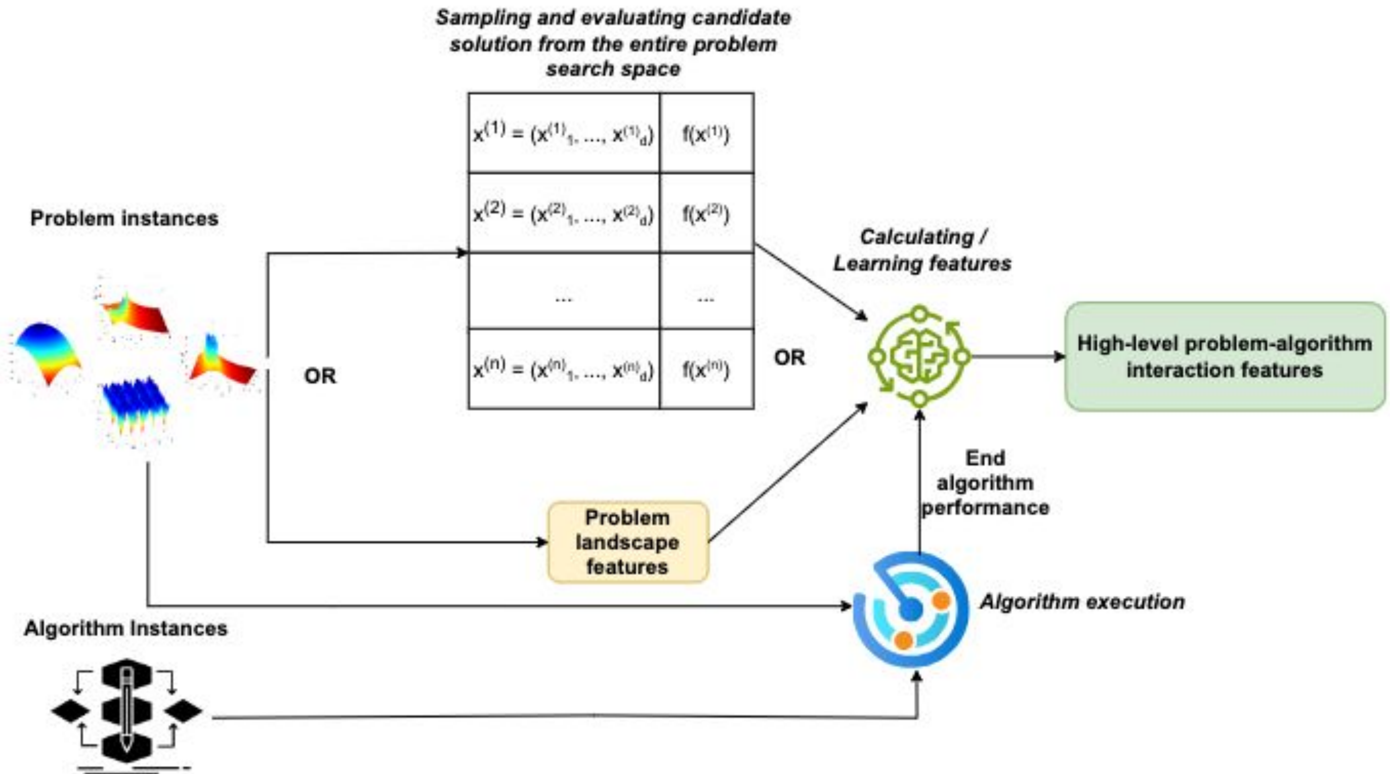
- **Implementation Dependency**: Features extracted from the source code are highly dependent on the programming language and the specific implementation, leading to potential discrepancies even for the same algorithm.

Pulatov, D., Anastacio, M., Kotthoff, L., & Hoos, H. (2022, September). Opening the black box: Automated software analysis for algorithm selection. In *International Conference on Automated Machine Learning* (pp. 6-1). PMLR.

High-Level Problem-Algorithm Interaction Features

Based on performance
Based on Shapley values of performance predictive model
Via Knowledge Graph
Via GNNs
Based on fANOVA
Based on SHAP
FintessMap + CNN
TransOptAS

High-Level Problem-Algorithm Interaction Features



Features based on performance

Calculating Perfromance2vec

- Vector representations consists of performance metric on a set of benchmark problems.

Metrics:

- Simple: Mean or Median across multiple runs
- Complex: Deep Statistical Comparison ranking or …

Pros:

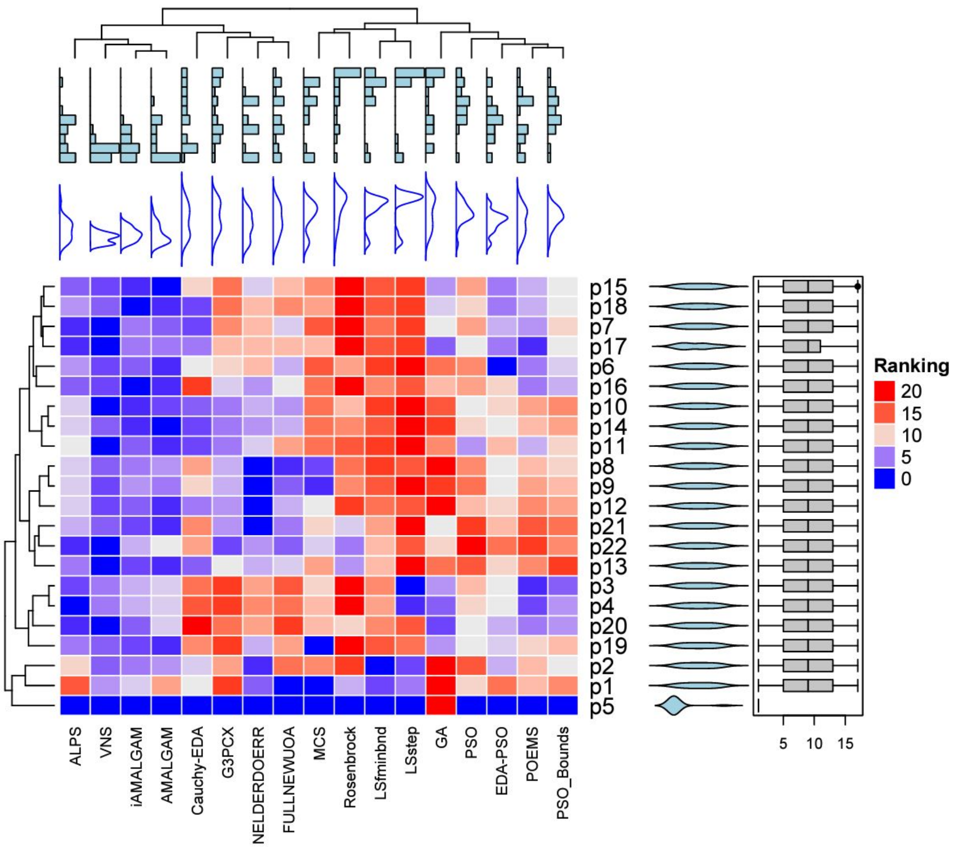
- Facilitates algorithm comparison through performance vectors.

Cons:

- Biased to the selected portfolio of benchmark problems

Eftimov, T., Popovski, G., Kocev, D., & Korošec, P. (2020, July). Performance2vec: a step further in explainable stochastic optimization algorithm performance. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion* (pp. 193-194).

Features based on performance



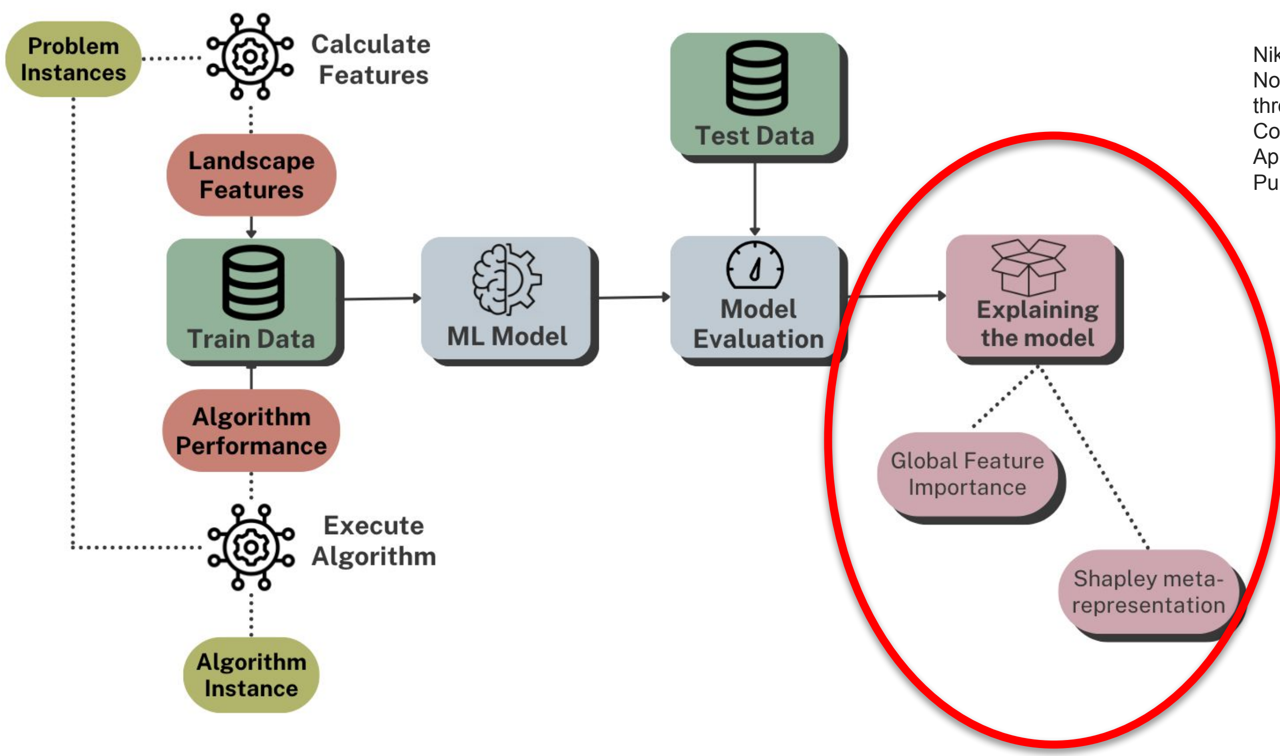
17 algorithms were compared using 22 benchmark problems from BBOB 2009 (dimension 10). Hierarchical clustering was applied to **Performance2Vec** embeddings (columns) and benchmark problem embeddings (rows). The matrix was reorganized to group similar algorithms and problems together. Colors indicate rankings from 1 (best) to 17 (worst). Ranking distributions for each algorithm and problem are shown.

Eftimov, T., Popovski, G., Kocev, D., & Korošec, P. (2020, July). Performance2vec: a step further in explainable stochastic optimization algorithm performance. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion (pp. 193-194).

Features based on Shapley values of performance predictive models

Learning features:

- Derived from the importance of problems features using explainability performance predictive methods.
- SHAP method applied for feature importance.
  - Calculated to determine the contribution of each feature to performance.
  - Global level: Across a set of problem instances.
  - Local level: On individual problem instances.



Nikolikj, A., Lang, R., Korošec, P., & Eftimov, T. (2022, November). Explaining differential evolution performance through problem landscape characteristics. In International Conference on Bioinspired Optimization Methods and Their Applications (pp. 99-113). Cham: Springer International Publishing.

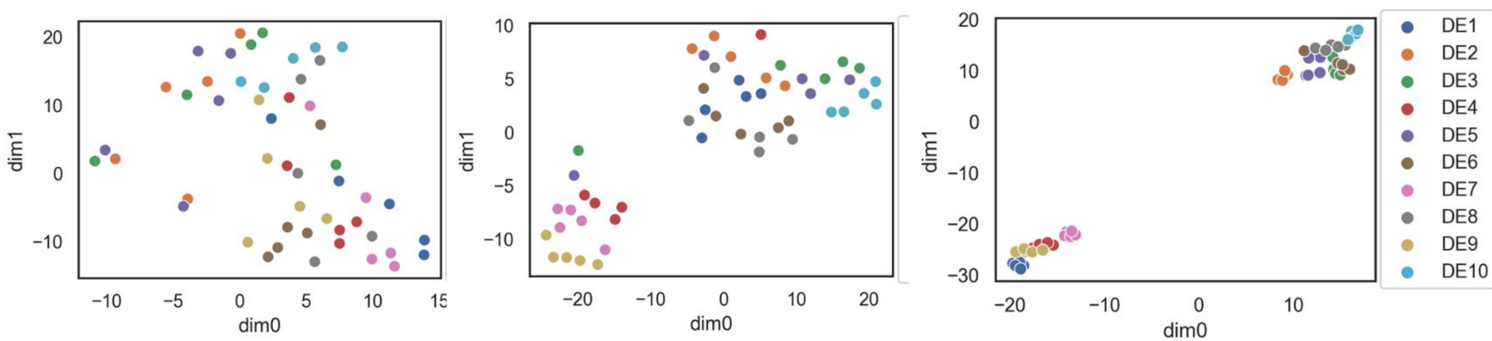
Features based on Shapley values of performance predictive models

Pros:

- Encodes interactions between problem features and algorithm performance.
- Used to find similar algorithm behaviors with the assumption that the predictive models are behave similarly.

Cons:

- Depends on the selected problem features portfolio
- Depends on the selected benchmark problem instances

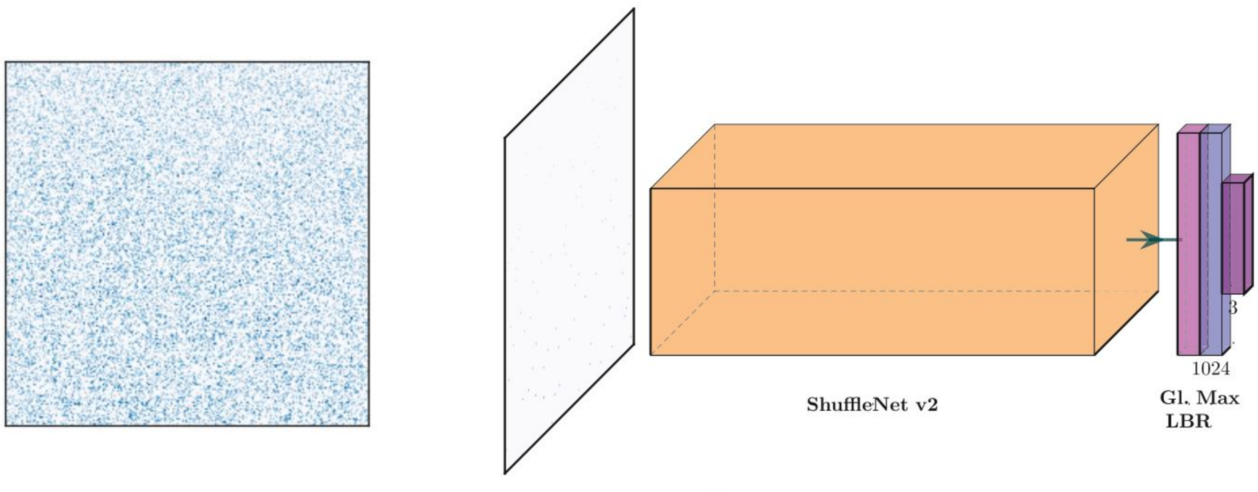


30d 100d 500d

Nikolikj, A., Lang, R., Korošec, P., & Eftimov, T. (2022, November). Explaining differential evolution performance through problem landscape characteristics. In *International Conference on Bioinspired Optimization Methods and Their Applications (pp. 99-113). Cham: Springer International Publishing.*

Fitness map features

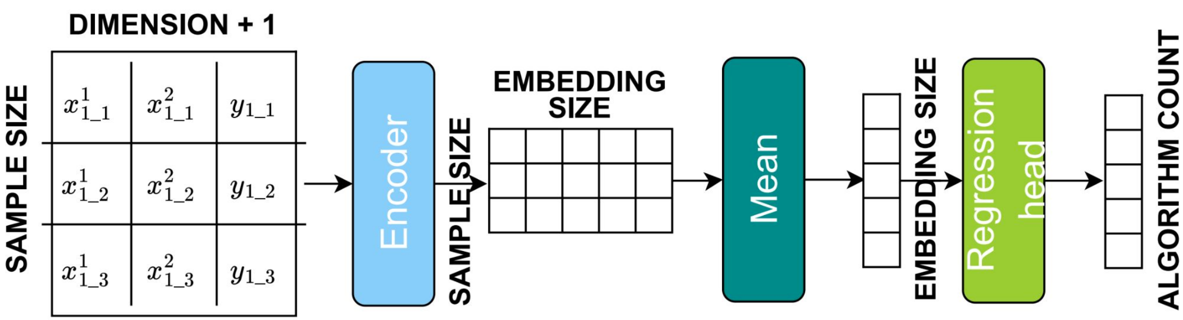
- Represent problem samples as a fitness map - 2D single-channel image
- Model: CNN (ShuffleNet v2)
- Task: algorithm selection of 32 CMAES configurations
- Data: BBOB benchmark, 124 instances per problem



Prager, R. P., Vinzent Seiler, M., Trautmann, H., & Kerschke, P. (2021). Towards Feature-Free Automated Algorithm Selection for Single-Objective Continuous Black-Box Optimization. In 2021 IEEE Symposium Series on Computational Intelligence (SSCI). 2021 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE. https://doi.org/10.1109/ssci50451.2021.9660174

TransOptAS

- Process:
  - Generate candidate solutions using Latin Hypercube sampling
  - Train transformer model, which given samples of the optimization function, predicts algorithm performance
- Data: Functions generated using a random function generator
- Task: Algorithm selection

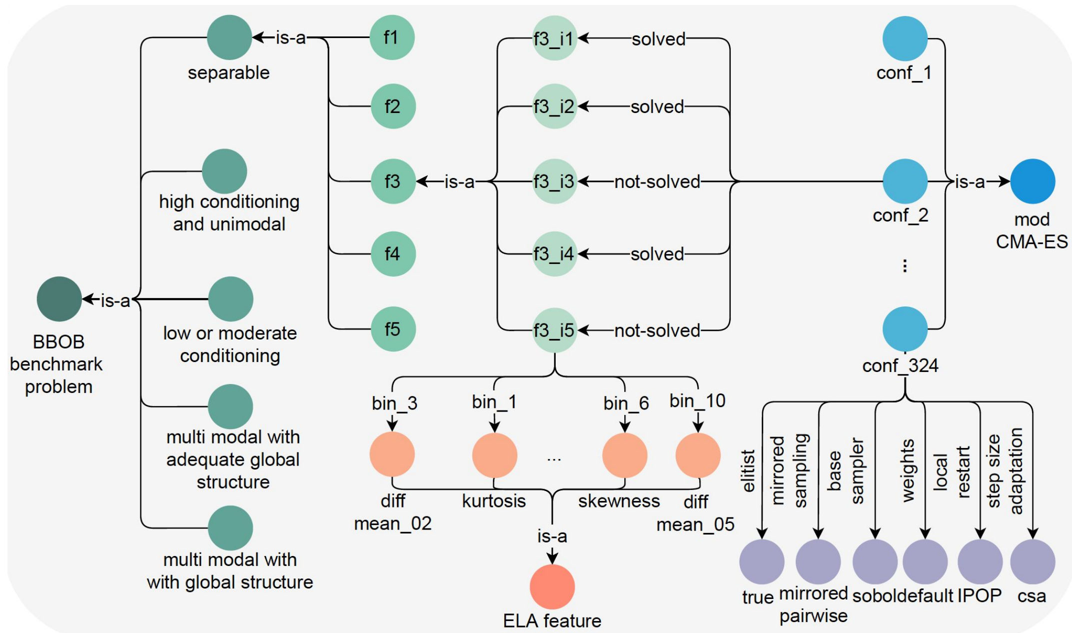


Cenikj, G., Petelin, G., & Eftimov, T. (2024). TransOptAS: Transformer-Based Algorithm Selection for Single-Objective Optimization. In Proceedings of the Genetic and Evolutionary Computation Conference Companion (pp. 403–406). GECCO '24 Companion: Genetic and Evolutionary Computation Conference Companion. ACM. https://doi.org/10.1145/3638530.3654191

Features via Knowledge Graph

Learning Features:

- Leverage interactions with entities in the optimization domain.
- **Knowledge Graph (KG)** methodology:
  - **Nodes Represent**:
    - Problem Instances: Problem class, high-level features, ELA features.
    - Algorithms: Parameters.
  - **Linking Criteria**: Algorithm solves problem instance within a specified error.

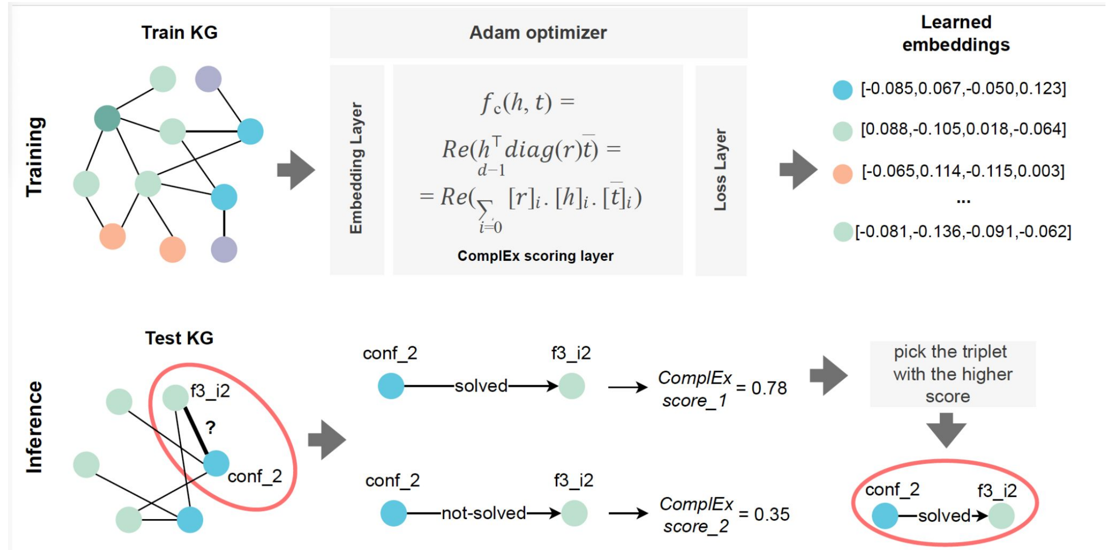


Kostovska, A., Vermetten, D., Džeroski, S., Panov, P., Eftimov, T., & Doerr, C. (2023, April). Using knowledge graphs for performance prediction of modular optimization algorithms. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 253-268). Cham: Springer Nature Switzerland.

Features via Knowledge Graphs

Embedding Representation:

- Use KG embeddings to derive algorithm and problem instance representations.
- Produces **Algorithm Features** or **Problem Instance Features**.
- Problem Instance Features:
  - Distinct from low-level landscape features.
  - Integrate landscape data and algorithm performance interaction.



Pros:

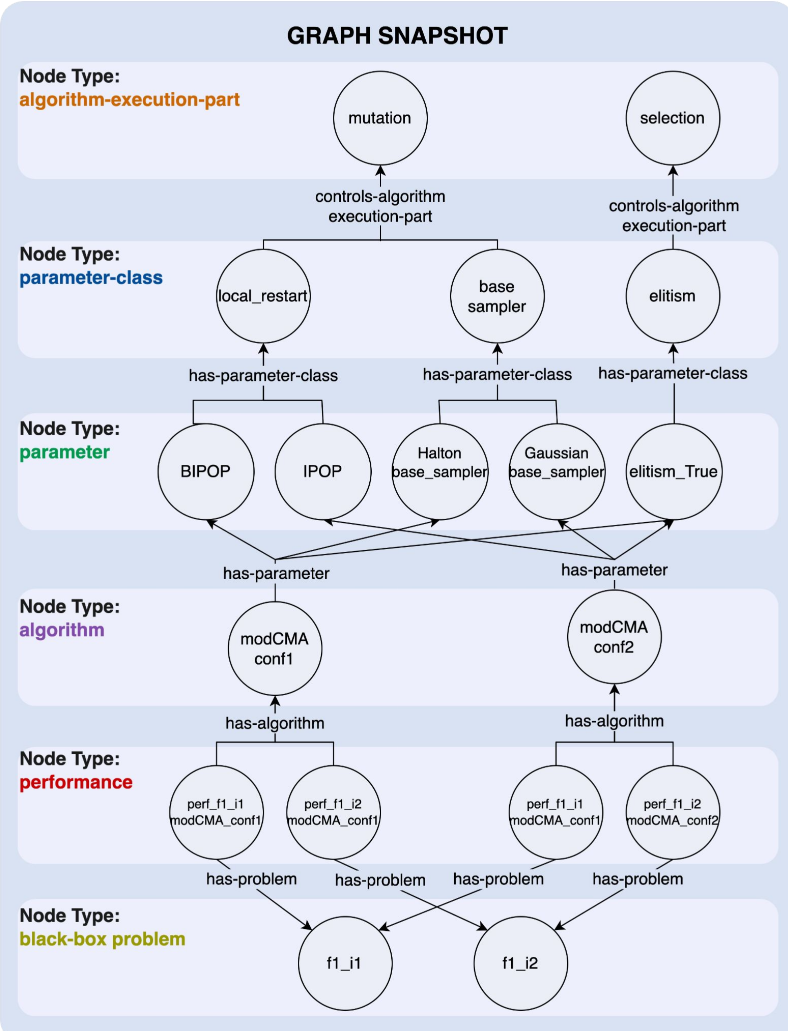
- Encodes interactions between problem features and algorithm performance by also involving the graph neighbourhood.

Cons:

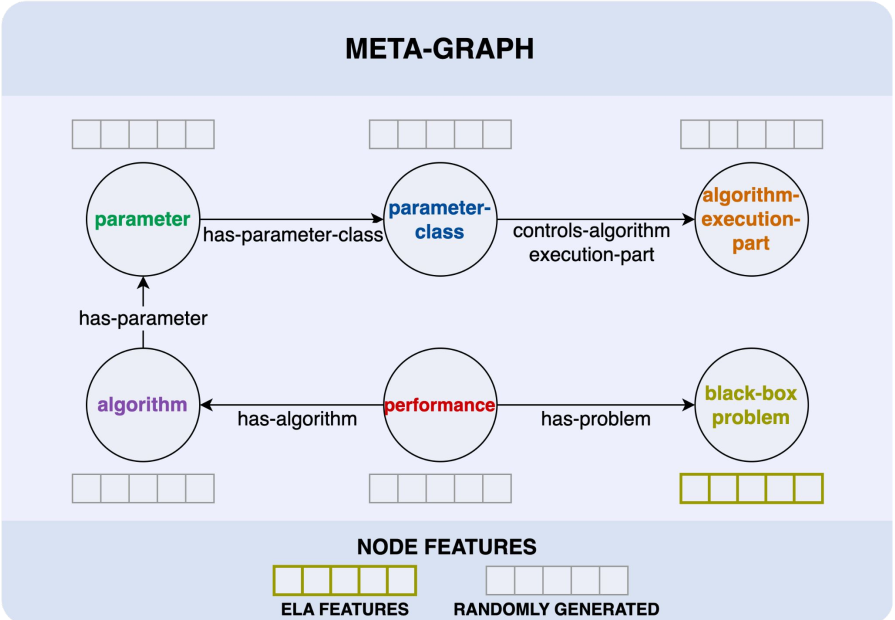
- Depends on the data stored in the KG
- Depends on the KG embedding method

Kostovska, A., Vermetten, D., Džeroski, S., Panov, P., Eftimov, T., & Doerr, C. (2023, April). Using knowledge graphs for performance prediction of modular optimization algorithms. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 253-268). Cham: Springer Nature Switzerland.

Features via GNNs



View of the graph data structure and entity relationships.



The meta-graph for the BBO heterogeneous graph.

Kostovska, A., Doerr, C., Džeroski, S., Panov, P., & Eftimov, T. (2025). Geometric Learning in Black-Box Optimization: A GNN Framework for Algorithm Performance Prediction. *arXiv preprint arXiv:2506.16144*.

Features via GNNs

Embedding Representation:

- The GNN is centered on performance nodes, which aggregate information from both the optimization problem and the algorithm configuration.
- Learning task: node regression
- Message-passing GNNs operate by iteratively aggregating and transforming information from a node's local neighborhood to learn meaningful representations.
- GraphSage and Graph Attention Network (GAT) implementations for heterogeneous graphs.
- Improvement ~ 36% against tree-based model

Pros:

- Encodes interactions between problem features and algorithm performance by also involving the graph neighbourhood.

Cons:

- Depends on the data stored in the graph
- Depends on the GNN method

Features based on fANOVA

Module Interaction Analysis

- Generate problem class-specific datasets:
  - Modules as features; performance as target
- Apply f-ANOVA:
  - Quantify variance contributions from:
  - Individual modules
  - Pairwise interactions
  - Triple interactions
- Results:
  - Feature vectors of module effects
  - Compare across problems to identify similar module-performance patterns

Pros:

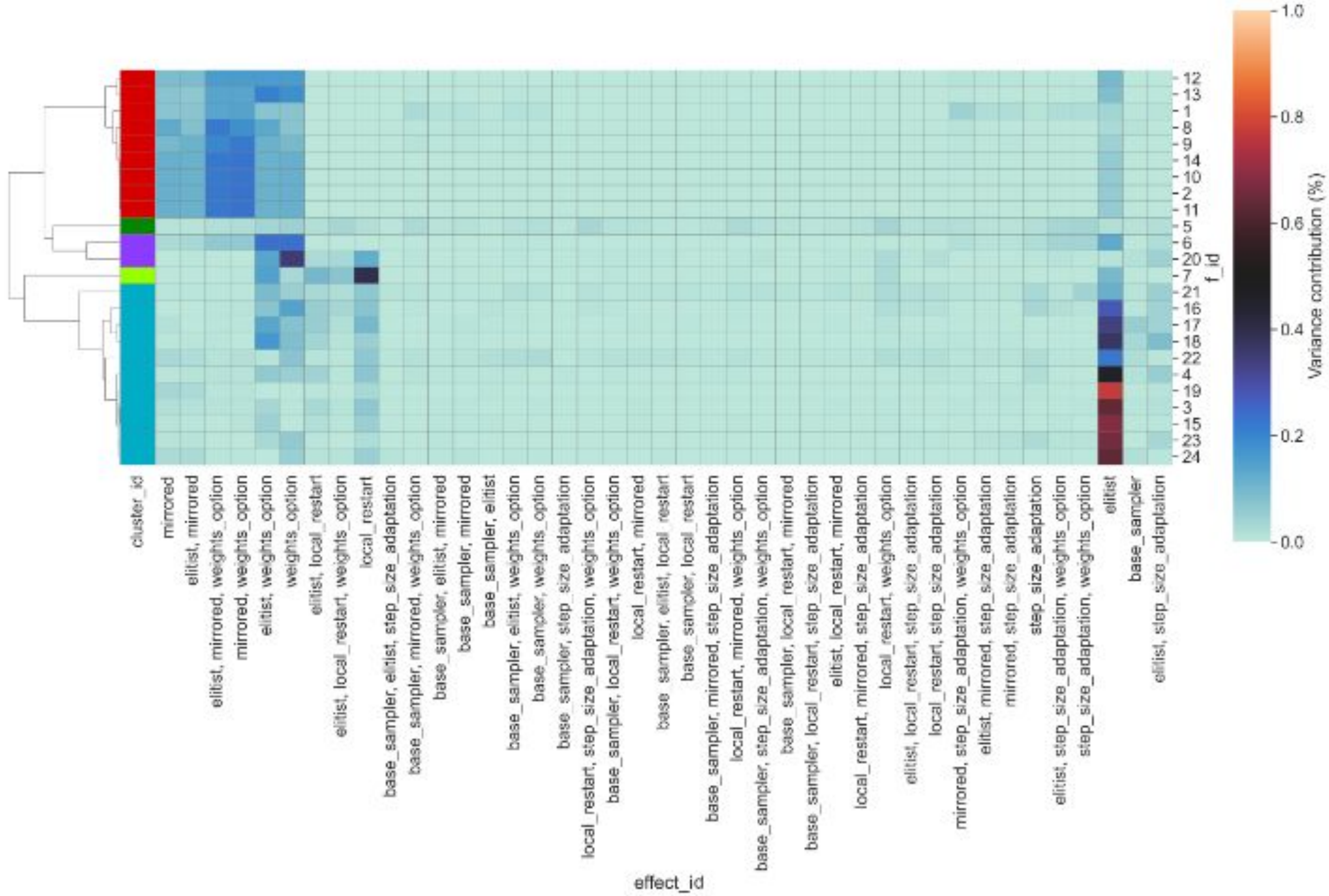
- For each problem, they encode the contributions of modules and their interactions to the final performance of an algorithm.

Cons:

- Depends on the available data from the pool of different configurations

Nikolikj, A., Kostovska, A., Vermetten, D., Doerr, C., & Eftimov, T. (2024, June). Quantifying individual and joint module impact in modular optimization frameworks. In *2024 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-8). IEEE.

Features based on fANOVA



Clustering results of the vector representations of problem classes for low problem dimensionality (d=5)

Features based on SHAP

Module Interaction Analysis

- Estimates contribution of each module to performance
- Similar to f-ANOVA in identifying module importance
- Limitation:
  - Only individual module contributions calculated
  - Higher-order contributions are computationally expensive

Pros:

- For each problem, they encode the contributions of modules and their interactions to the final performance of an algorithm.

Cons:

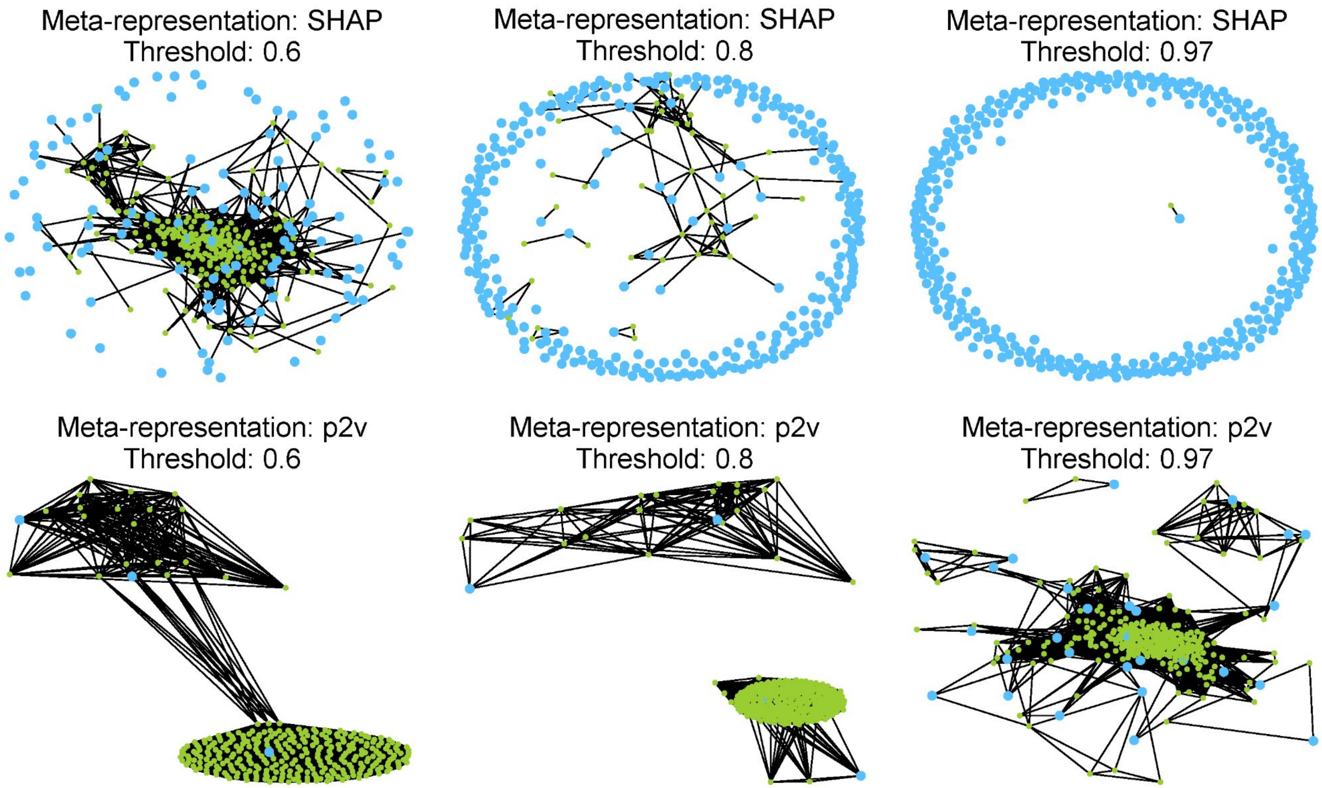
- Depends on the available data from the pool of different configurations

van Stein, N., Vermetten, D., V. Kononova, A., & Bäck, T. (2025). Explainable benchmarking for iterative optimization heuristics. *ACM Transactions on Evolutionary Learning*, *5*(2), 1-30.

Application of high-level problem-algorithm interaction features

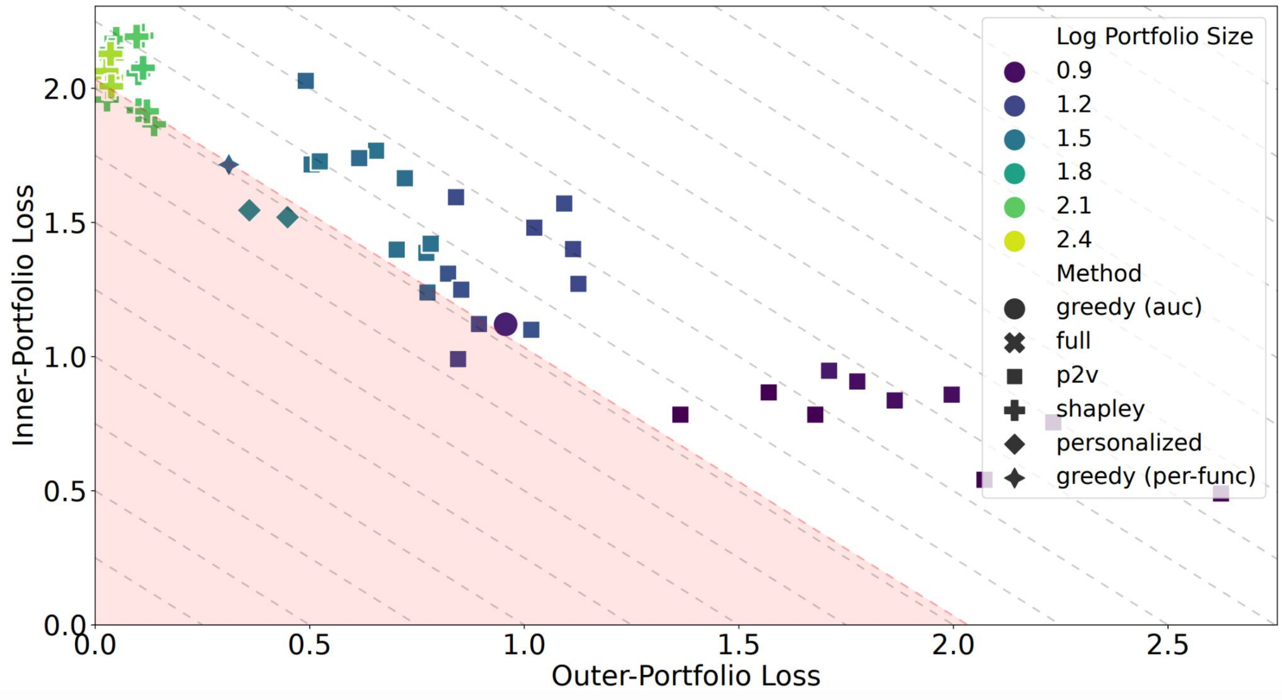
Selection of diverse complementary algorithm portfolio

Selection of complementary algorithm portfolio



Kostovska, A., Cenikj, G., Vermetten, D., Jankovic, A., Nikolikj, A., Skvorc, U., ... & Eftimov, T. (2023, December). PS-AAS: Portfolio Selection for Automated Algorithm Selection in Black-Box Optimization. In *International Conference on Automated Machine Learning* (pp. 11-1). PMLR.

Selection of complementary algorithm portfolio

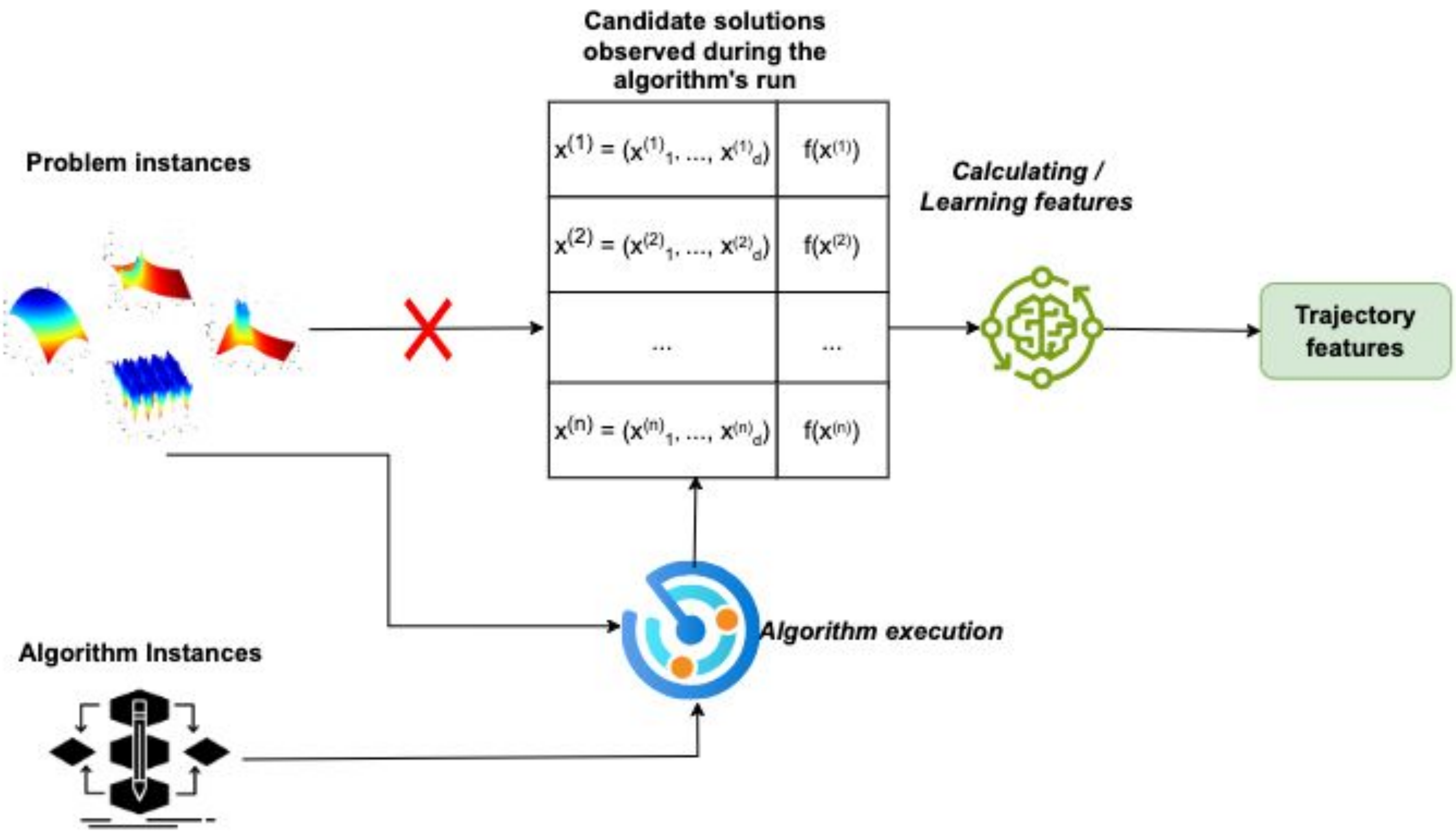


x-axis: the best possible loss of the portfolio = the difference between the portfolio's VBS and the VBS of the full set of 324 algorithms.
y-axis: the loss of the AS = the difference in performance between the algorithm it selects and the VBS of the portfolio it can choose from.

Problem-Algorithm Trajectory Features

Based on internal algorithm parameters
Trajectory-based ELA
DynamoRep
Opt2Vec
Local Optima Networks and variants
Probing trajectories
ClustOpt

Problem-algorithm trajectory features



Trajectory-based features Based on Internal Algorithm Parameters

- **Calculating features:**

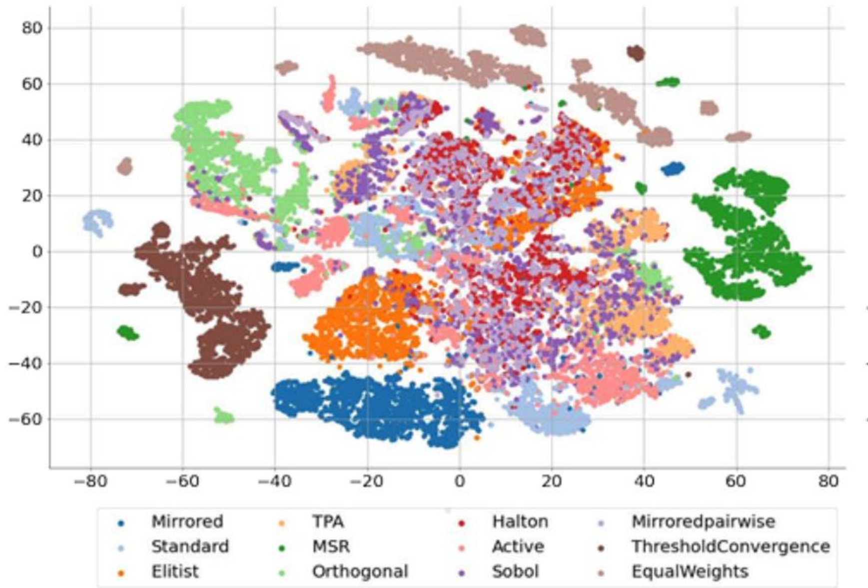
Time-series features extracted from internal parameters that are adjust during the optimization process.

Employed the *tsfresh* library for feature extraction.

- **Application:**

Time-series features helped identify configurations of modular CMA-ES variants.

Step size, Best-so-far value, Evolution path, Conjugate evolution path, Square root of diagonal of covariance matrix eigenvalues



Pros: Capture the behaviour of the algorithm

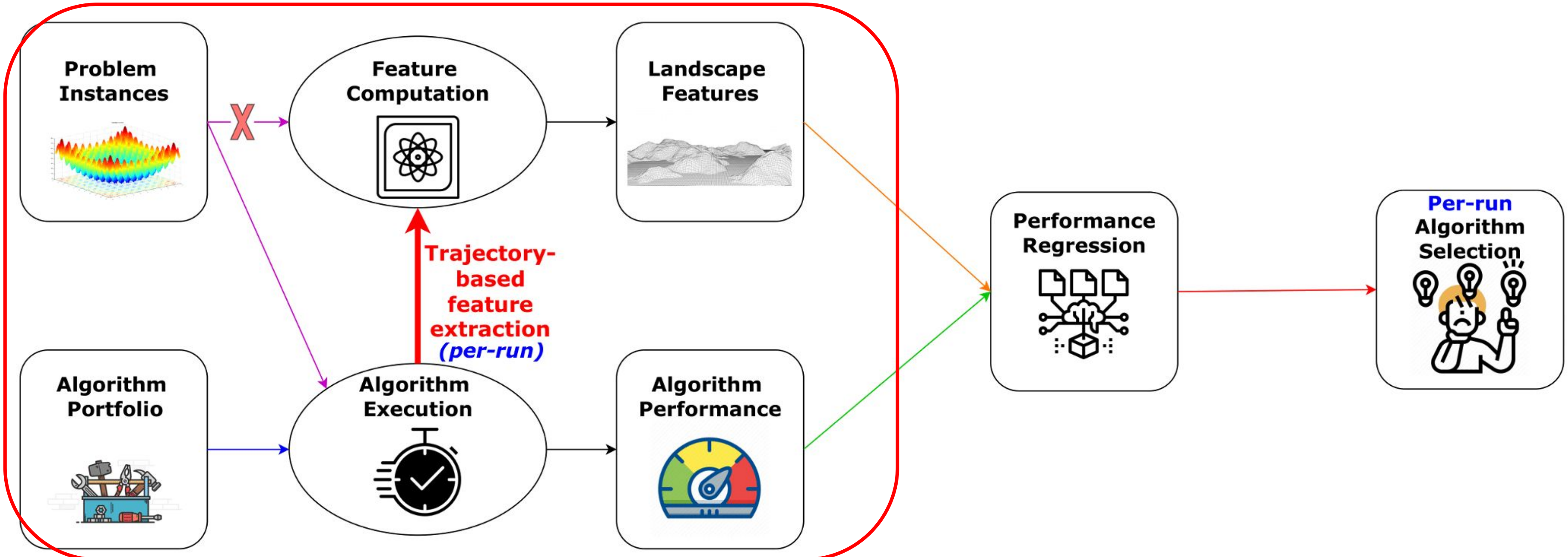
Cons: Lack of comprehensive comparison of different time series features

de Nobel, J., Wang, H., & Baeck, T. (2021, June). Explorative data analysis of time series based algorithm features of CMA-ES variants. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 510-518).

Trajectory-based ELA features

Calculating Features:

- ELA features calculated from populations (candidate solutions and corresponding function values) observed during optimization rather than candidate solutions obtained with a standard sampling techniques.



Jankovic, A., Eftimov, T., & Doerr, C. (2021). Towards feature-based performance regression using trajectory data. In *Applications of Evolutionary Computation: 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 24* (pp. 601-617). Springer International Publishing.

Trajectory-based ELA features

Applications:

- **Fixed-Budget Performance Prediction:** Applied to CMA-ES performance prediction.
- **Per-Run Algorithm Selection:** Used in warm-starting to decide on switching algorithm instances.

Pros:

- Info about the interaction across problem and algorithms **(personalization).**

Cons:

- Does not capture the longitudinal aspect of solutions within algorithm iterations.

Jankovic, A., Eftimov, T., & Doerr, C. (2021). Towards feature-based performance regression using trajectory data. In *Applications of Evolutionary Computation: 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 24* (pp. 601-617). Springer International Publishing.

Iterative-based ELA features

Calculating Features:

- ELA features calculated from a single population (candidate solutions and corresponding function values), one iteration observed during optimization.

Applications:

- Problem and dimension being solved - < 40% accuracy.
- Online algorithm performance improvement prediction - small improvements against a time series baseline model.

Pros:

- Info about the a single timestamp of the optimization process, can easily be combined with ML models that will capture the longitudinality of the search process.

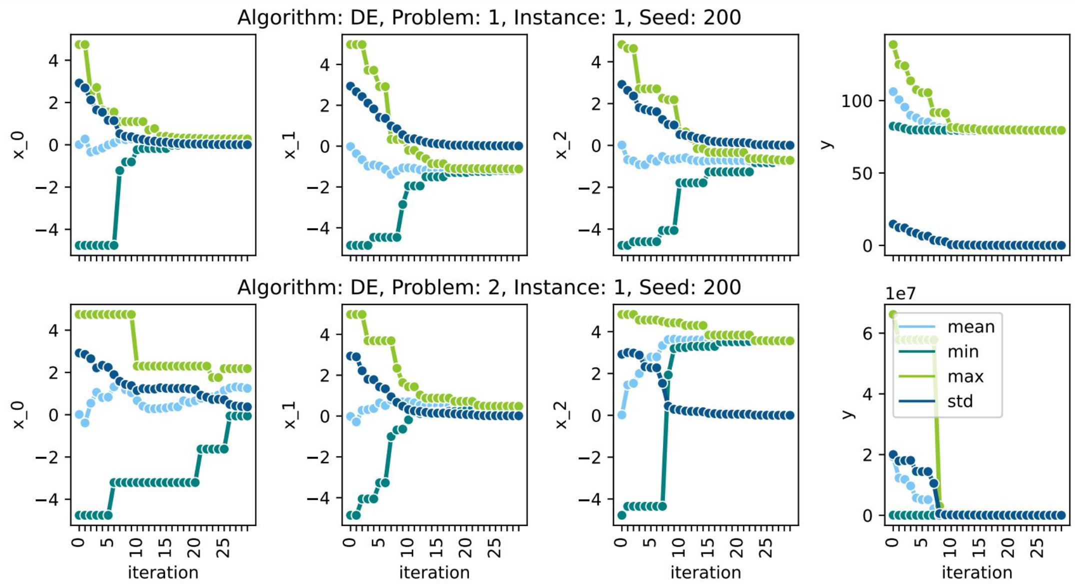
Cons:

- ELA features are sensitive on small sample sizes, which in this case is the dimension of the population.

Korošec, P., & Eftimov, T. (2024). Opt2Vec-a continuous optimization problem representation based on the algorithm's behavior: A case study on problem classification. *Information Sciences*, *680*, 121134.

DynamoRep features

- **Calculating Features:**
    - Constructed by concatenating statistics from each population.
    - Statistics extracted per iteration:
        - Minimum, maximum, mean, and standard deviation.
        - Applied to decision variables and objective function values.
    - For an algorithm with *n* iterations on a problem instance of dimension *d*.
        - Representation size = *4n(d + 1)*.



DynamoRep features generated from the trajectories of one run of the DE algorithm on the first instance of the first two 3d problem classes (sphere and ellipsoidal functions) from the BBOB benchmark suite.

Cenikj, G., Petelin, G., Doerr, C., Korošec, P., & Eftimov, T. (2025). Beyond Landscape Analysis: DynamoRep Features For Capturing Algorithm-Problem Interaction In Single-Objective Continuous Optimization. *Evolutionary Computation*, 1-28.

Cenikj, G., Petelin, G., Doerr, C., Korošec, P., & Eftimov, T. (2023, July). Dynamorep: trajectory-based population dynamics for classification of black-box optimization problems. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 813-821).

DynamoRep features

- **Applications**:
  - **Problem Classification**: Detect the problem class being solved.
  - **Algorithm Classification**: Identify the algorithm solving the problem instance.

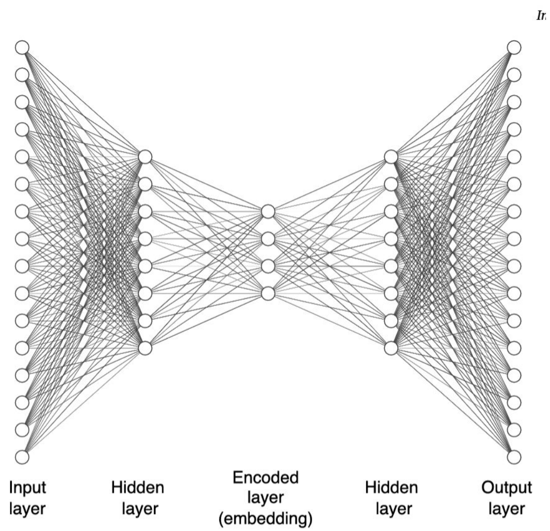
- **Pros**:
  - DynamoRep features are much cheaper to compute compared to state-of-the-art Exploratory Landscape Analysis (ELA) features.
  - Despite lower computational cost, DynamoRep features yield results comparable to those achieved with ELA features, calculated at each iteration of the algorithm's execution.

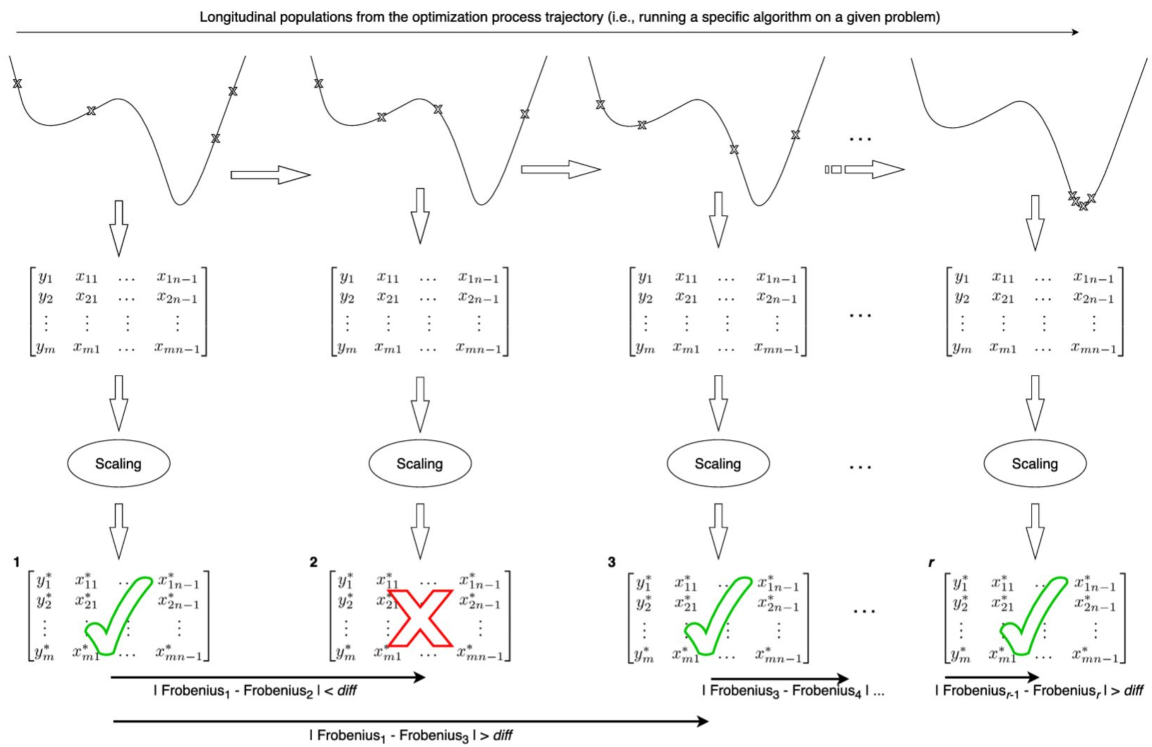
- **Cons**:
  - Limited expressiveness
  - Representation size grows with number of iterations and problem dimension, may require dimensionality reduction as preprocessing step

Opt2Vec features

Learning Features:

- Analyze populations considered by an algorithm in each iteration.
- Scale candidate solutions and objective function values.
- Use autoencoders to embed information from each population (single iteration).





Korošec, P., & Eftimov, T. (2024). Opt2Vec-a continuous optimization problem representation based on the algorithm's behavior: A case study on problem classification. *Information Sciences*, *680*, 121134.

Opt2Vec features

Applications:

- Problem and dimension classification
- Online algorithm performance improvement prediction - improvements against a time series baseline model and iterative ELA.

Pros:

- Capture features specific to parts of the search space explored at a particular iteration.
- Crucial for optimizing dynamic algorithms efficiently.
- First representation that takes into consideration the optimization problem dimension

Cons:

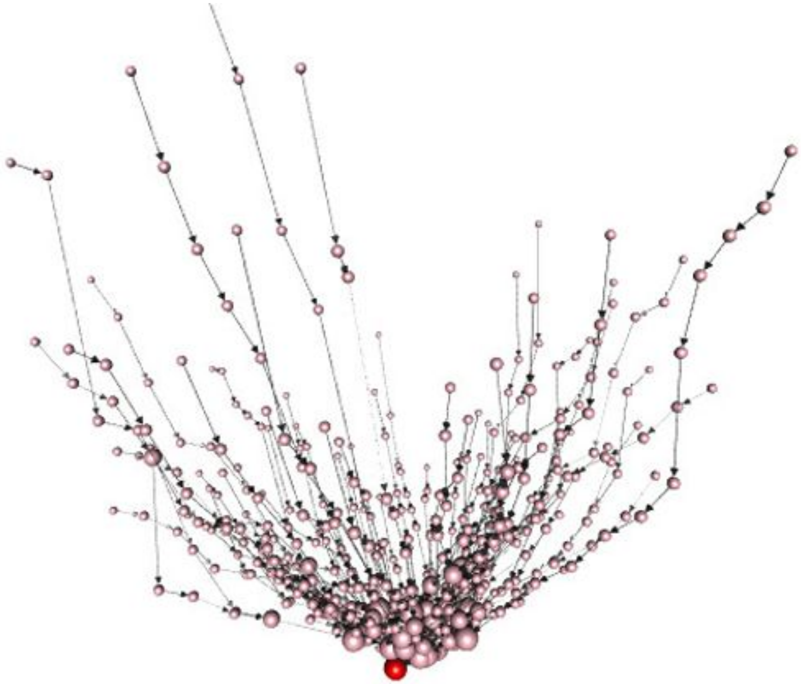
- Depends on the data used to train the autoencoder

Korošec, P., & Eftimov, T. (2024). Opt2Vec-a continuous optimization problem representation based on the algorithm's behavior: A case study on problem classification. *Information Sciences*, *680*, 121134.

Local Optima Networks (LONs) and variants

- **LONs Overview**:
  - Simplified model for discrete fitness landscapes.
  - Nodes represent local optima; edges represent search transitions via exploration operators.
  - Capture the number, distribution, and connectivity patterns of local optima.

- **Variants**:
  - **Monotonic LONs (MLONs)**: Only consider transitions with non-deteriorating fitness.
  - **Compressed MLONs (CMLONs)**: Group nodes with the same fitness in MLONs to account for neutrality.
  - **Search Trajectory Network (STNs)**: Nodes represent different states in the optimization trajectory, not limited to local optima



LON of Rastrigin function

Adair, J., Ochoa, G., & Malan, K. M. (2019, July). Local optima networks for continuous fitness landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 1407-1414).

Local Optima Networks (LONs) and variants

- **Applications**:
  - CMLONs used to visualize and analyze 24 BBOB problem classes across dimensions.
  - Network metrics and dimensionality reduction used to compare problems

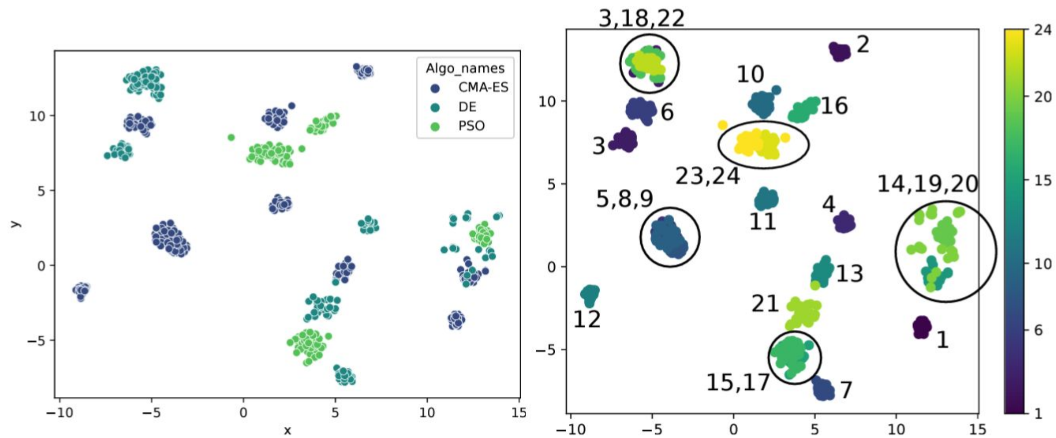
- **Pros**:
  - Nice for visualization purposes

- **Cons**:
  - Costly to compute

Ochoa, G., Malan, K. M., & Blum, C. (2020, April). Search trajectory networks of population-based algorithms in continuous spaces. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 70-85

Probing trajectories

- **Learning features** :
  - Generate short trajectories by running an algorithm on a problem instance.
  - Track current fitness or best-so-far fitness across sequential iterations.
  - Extract time-series features from trajectories using the *tsfresh* library.
  - Or concatenate the tracked values from sequential iterations.



(a) Algorithms (b) BBOB functions

Probing trajectories similarity

Renau, Q., & Hart, E. (2024, March). On the Utility of Probing Trajectories for Algorithm-Selection. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 98-114). Cham: Springer Nature Switzerland.

Probing trajectories

- **Applications**
  - Algorithm selector - comparable to trajectory ELA features
- **Pros:**
  - Potential to be utilized for per-run algorithm selection
- **Cons:**
  - Recently proposed, required more evaluations

ClustOpt

Objective: Compare how different population-based search algorithms explore a problem over time

Applicability: Single run, multiple runs with different seeds, or different algorithms

1. Merge Solutions
Pool every candidate from all runs, iterations, and populations into one set

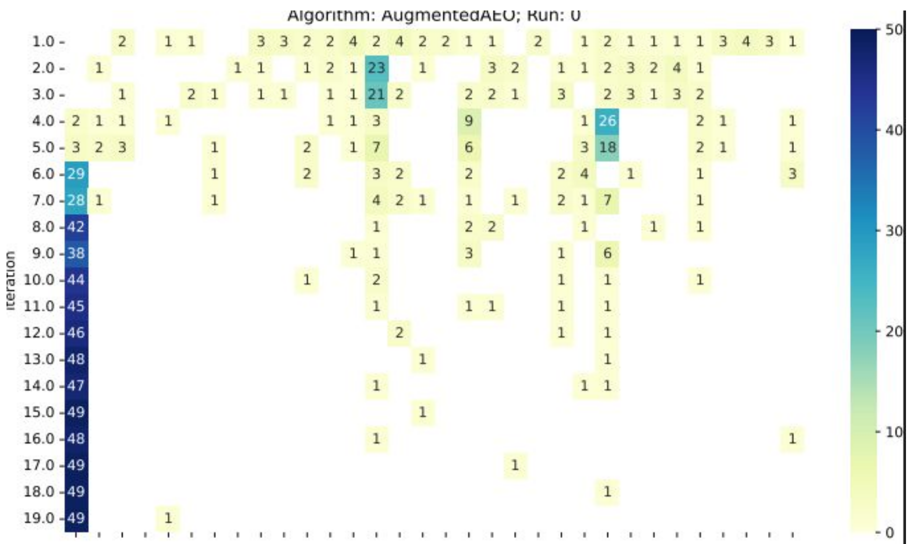
2. Joint Normalization
Rescale each variable of every solution to [0,1] using the combined data

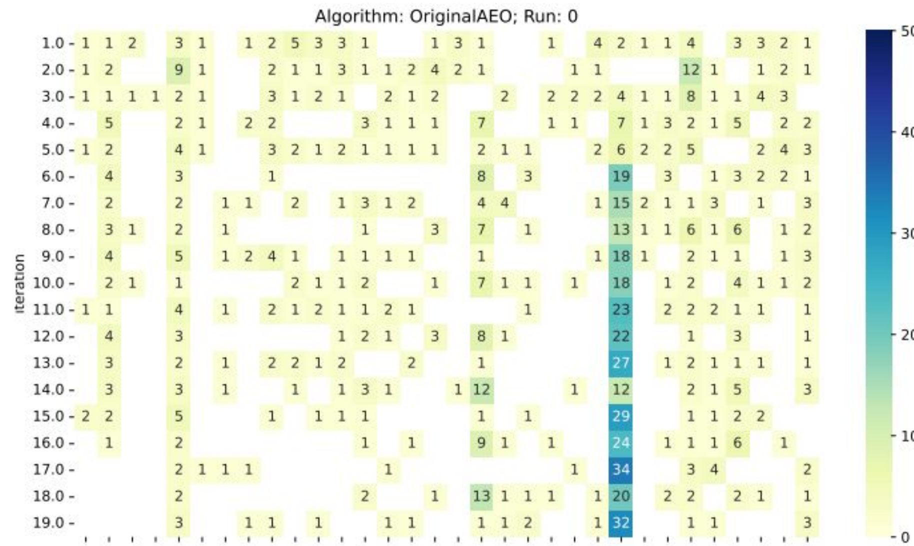
3. Clustering
Use a clustering method to find different search regions visited by the algorithms on a single or set of problems.

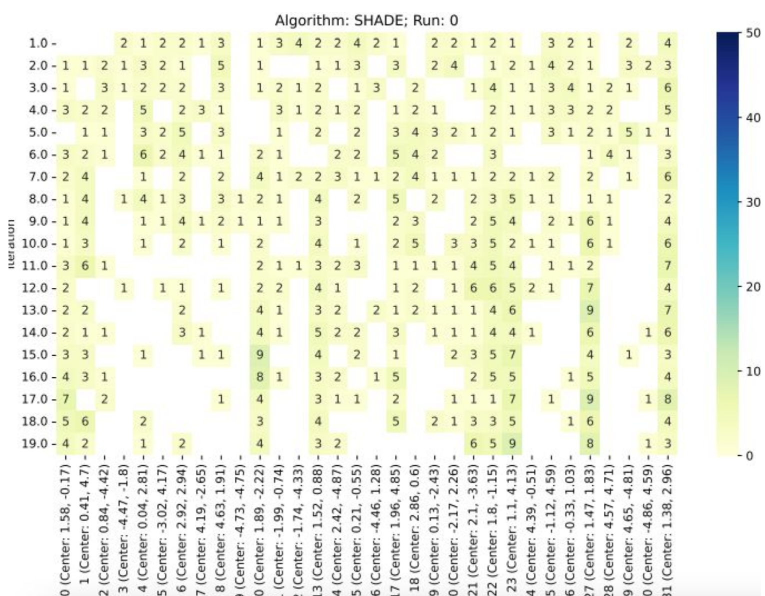
4. Trajectory Encoding
For each run and iteration, count how many solutions fall in each cluster → sequence of count vectors.

Cenikj, G., Petelin, G., & Eftimov, T. (2025, June). ClustOpt: A Clustering-based Approach for Representing and Visualizing the Search Dynamics of Numerical Metaheuristic Optimization Algorithms. In *2025 IEEE Congress on Evolutionary Computation (CEC)*. IEEE.

ClustOpt visualization







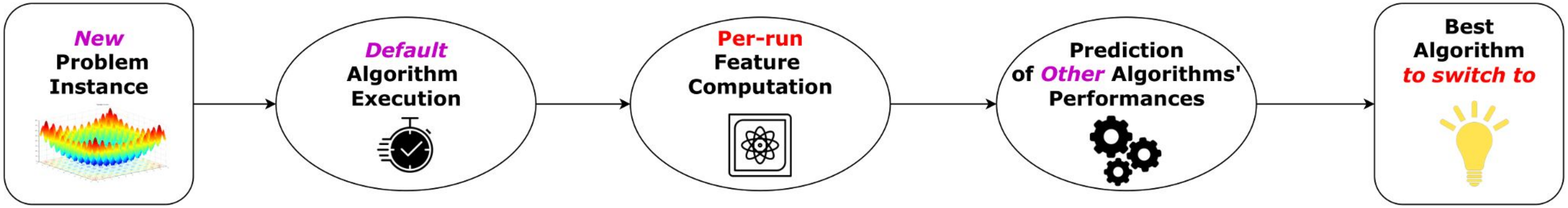
Visualizations of the trajectories of the OriginalAEO, AugmentedAEO and SHADE algorithms on the first instance of the 16th BBOB problem (Weierstrass) in 2 dimensions.

Cenikj, G., Petelin, G., & Eftimov, T. (2025, June). ClustOpt: A Clustering-based Approach for Representing and Visualizing the Search Dynamics of Numerical Metaheuristic Optimization Algorithms. In *2025 IEEE Congress on Evolutionary Computation (CEC)*. IEEE.

Application of problem-algorithm trajectory features

Per-run algorithm selection
Per-run algorithm performance prediction
Dynamic algorithm configuration with reinforcement learning
Representing and visualizing the search dynamics

Per-run algorithm selection



Kostovska, A., Jankovic, A., Vermetten, D., de Nobel, J., Wang, H., Eftimov, T., & Doerr, C. (2022, August). Per-run algorithm selection with warm-starting using trajectory-based features. In *International Conference on Parallel Problem Solving from Nature* (pp. 46-60). Cham: Springer International Publishing.

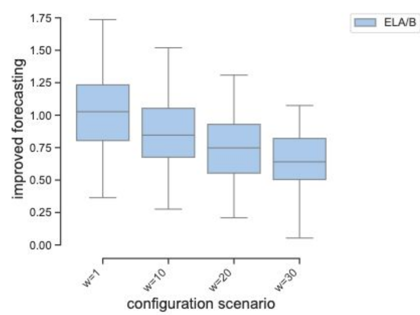
Vermetten, D., Wang, H., Sim, K., & Hart, E. (2023, April). To switch or not to switch: predicting the benefit of switching between algorithms based on trajectory features. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 335-350). Cham: Springer Nature Switzerland.

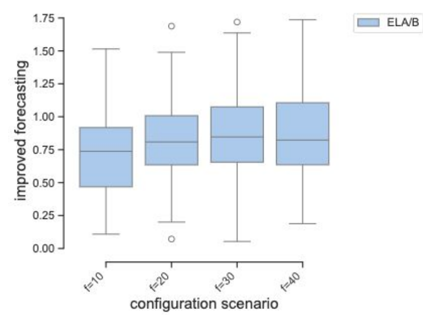
Per-run algorithm performance prediction

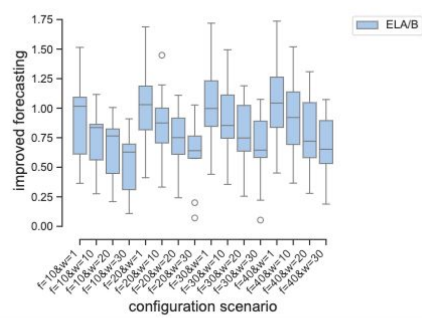
Leveraging iteration information (using candidate solutions observed by the algorithm) to forecast performance improvements using Long Short-Term Memory (LSTM) networks.

Calculate the iterative ELA representations on individual iteration.

Define the LSTM forecasting model with respect to number of sequential iterations and the size of forecasting window.



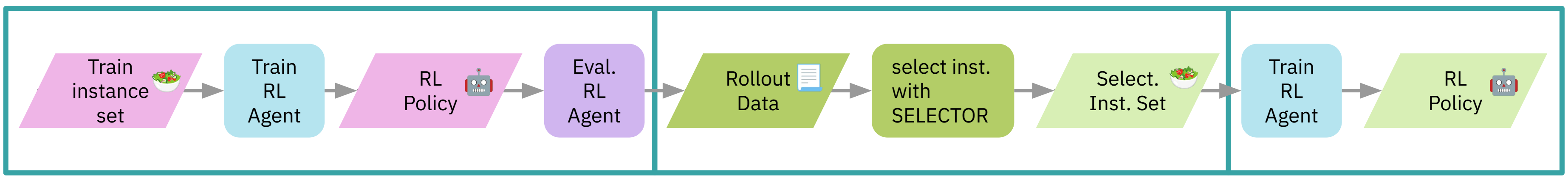




Korošec, P., & Eftimov, T. (2024, July). Per-Run Algorithm Performance Improvement Forecasting Using Exploratory Landscape Analysis Features: A Case Study in Single-Objective Black-Box Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 571-574).

Dynamic algorithm configuration with Reinforcement Learning

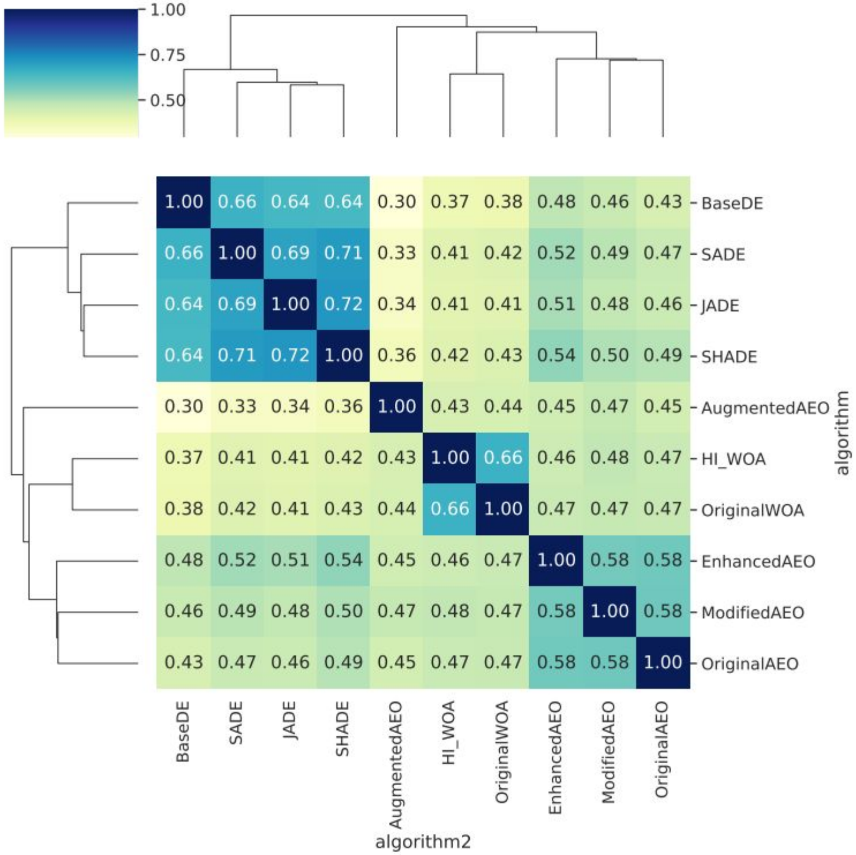
- From full problem instance set to subselection by using trajectory features of the reinforcement agents
- Raw and tsfresh features calculated using actions and rewards
- Better generalization of the RL for DAC on test instances



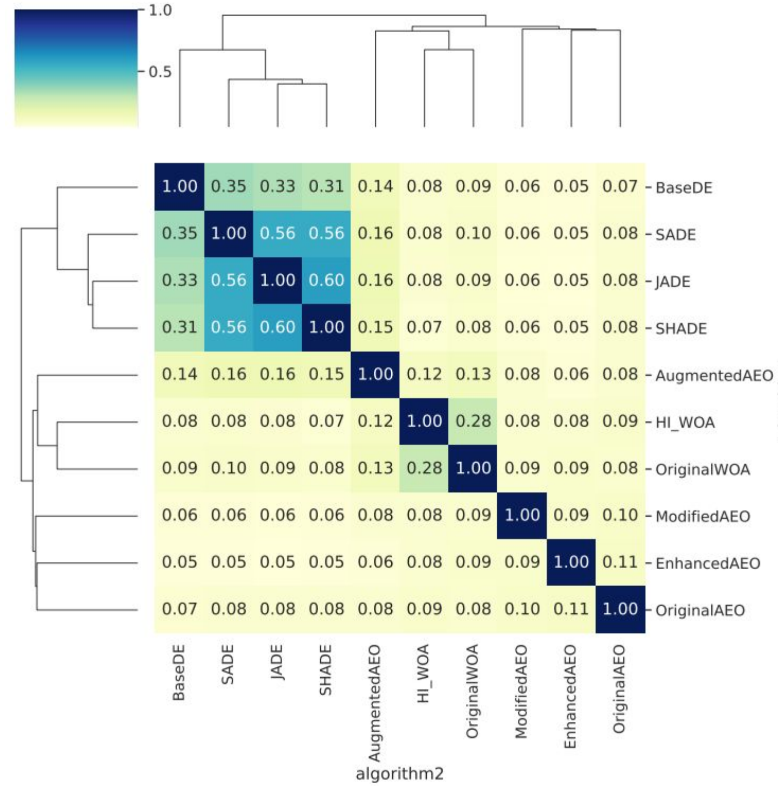
Standard training pipeline | Instance selection | Retraining

Benjamins, C., Cenikj, G., Nikolikj, A., Mohan, A., Eftimov, T., & Lindauer, M. (2024, July). Instance Selection for Dynamic Algorithm Configuration with Reinforcement Learning: Improving Generalization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 563-566).

Representing and visualizing the search dynamics using ClustOpt



2d



10d

Algorithm similarity across problem dimensions

Cenikj, G., Petelin, G., & Eftimov, T. (2025, June). ClustOpt: A Clustering-based Approach for Representing and Visualizing the Search Dynamics of Numerical Metaheuristic Optimization Algorithms. In *2025 IEEE Congress on Evolutionary Computation (CEC)*. IEEE.

Generalization of Algorithm Selection

A cross-benchmark examination of problem features
Have we hit a wall in Algorithm Selection generalization?

A cross-benchmark examination of problem features

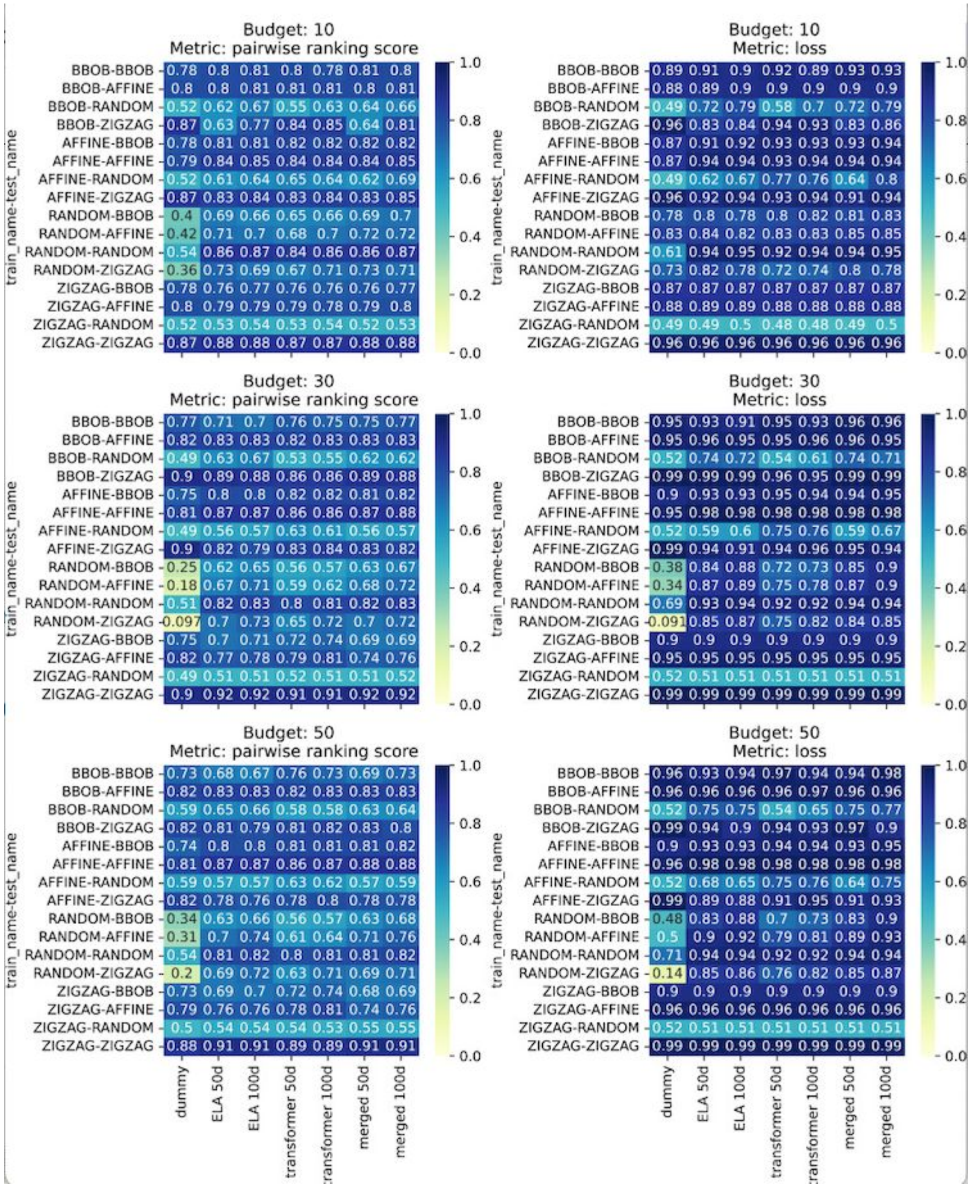
Generalization of an algorithm selector across four different benchmark suites.

Evaluation of TransOpt features for algorithm selection.

Comparison of TransOpt and ELA features.

Cenikj, G., Petelin, G., & Eftimov, T. (2024). A cross-benchmark examination of feature-based algorithm selector generalization in single-objective numerical optimization. *Swarm and Evolutionary Computation*, *87*, 101534.

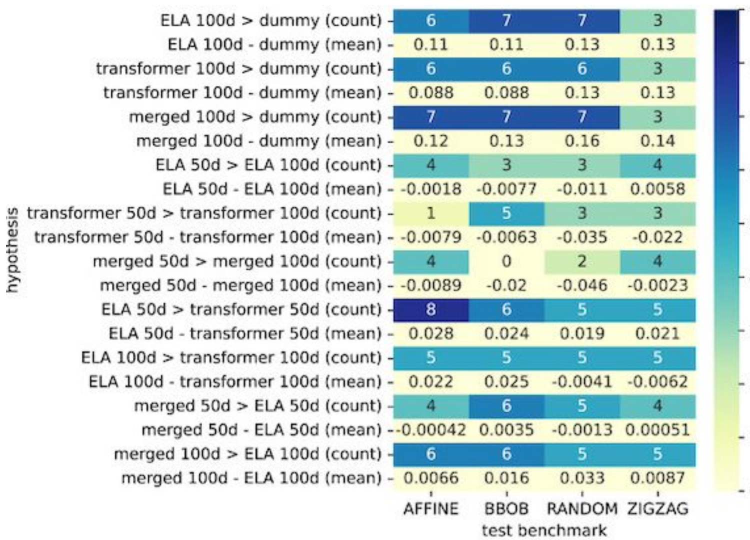
A cross-benchmark examination of problem features



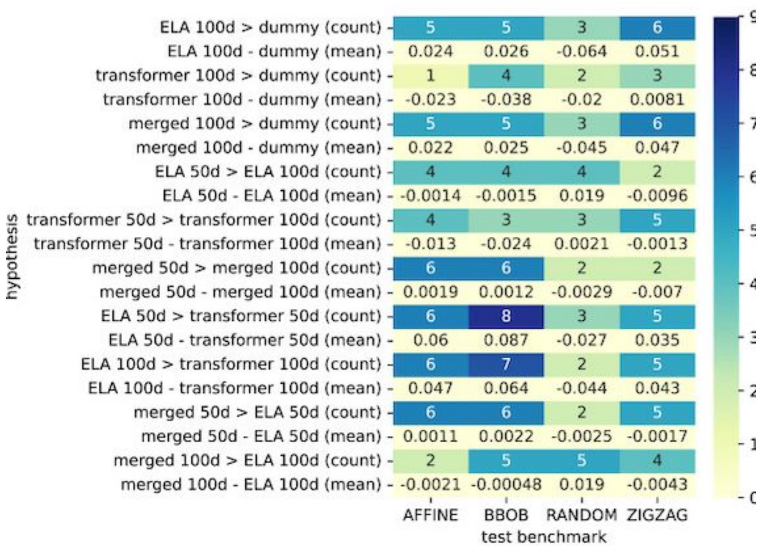
Pairwise ranking score and loss achieved by the models for 3*d* problems.

Cenikj, G., Petelin, G., & Eftimov, T. (2024). A cross-benchmark examination of feature-based algorithm selector generalization in single-objective numerical optimization. *Swarm and Evolutionary Computation*, *87*, 101534.

A cross-benchmark examination of problem features



(a) Summary results for $3d$ problems



(b) Summary results for $10d$ problems

Summary results in terms of the loss metric.

Cenikj, G., Petelin, G., & Eftimov, T. (2024). A cross-benchmark examination of feature-based algorithm selector generalization in single-objective numerical optimization. *Swarm and Evolutionary Computation*, *87*, 101534.

Have we hit a wall in Algorithm Selection generalization?

Benchmarking single-objective numerical optimization problem landscape features.

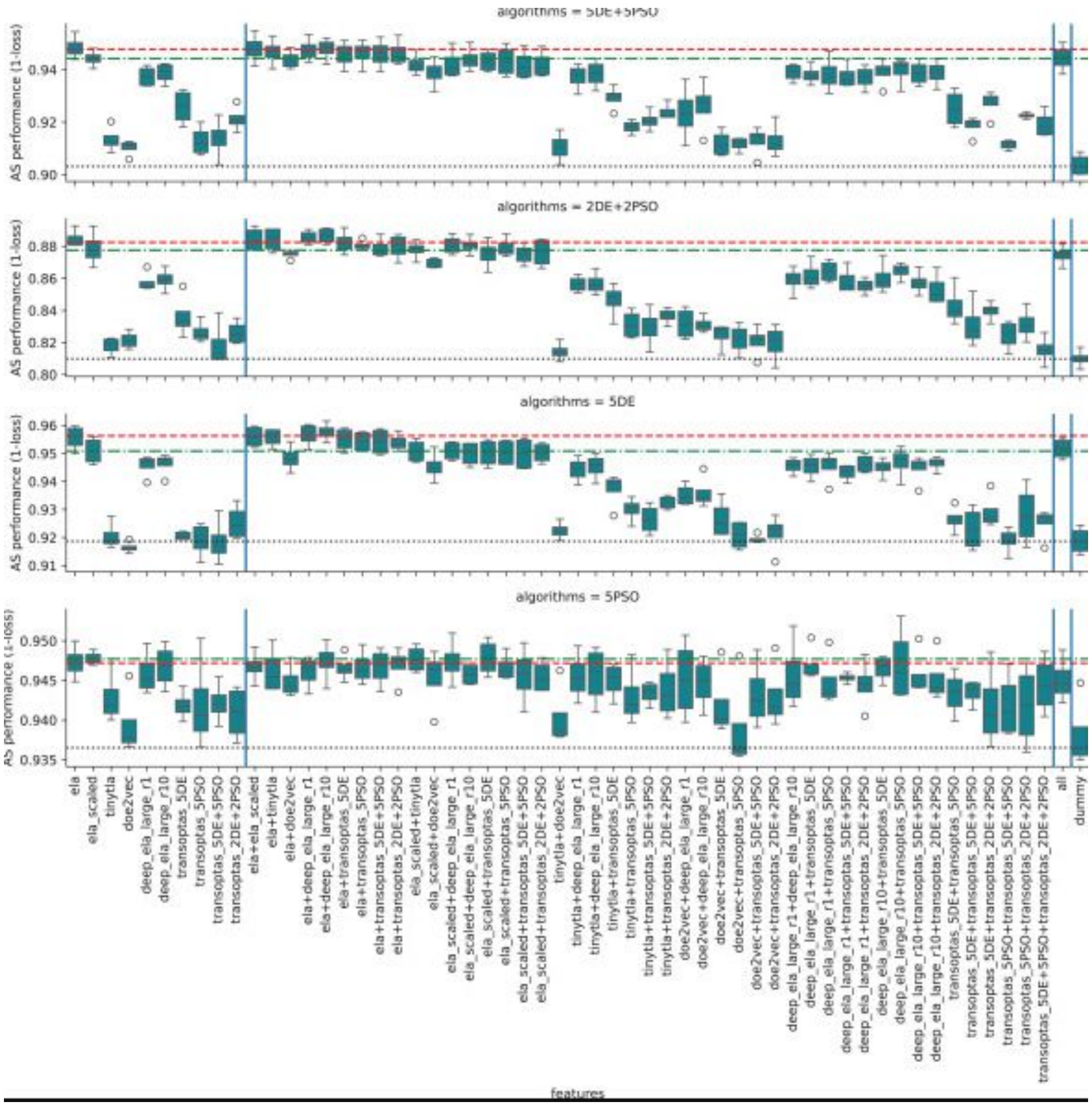
Investigating whether problem landscape features capture algorithm performance.

Analyzing problem landscape feature similarity and complementarity.

Evaluating Algorithm Selection generalizability with different feature groups.

Cenikj, G., Petelin, G., Seiler, M., Cenikj, N., & Eftimov, T. (2025). Landscape Features in Single-Objective Continuous Optimization: Have We Hit a Wall in Algorithm Selection Generalization?. *Swarm and Evolutionary Computation*, *94*, 101894.

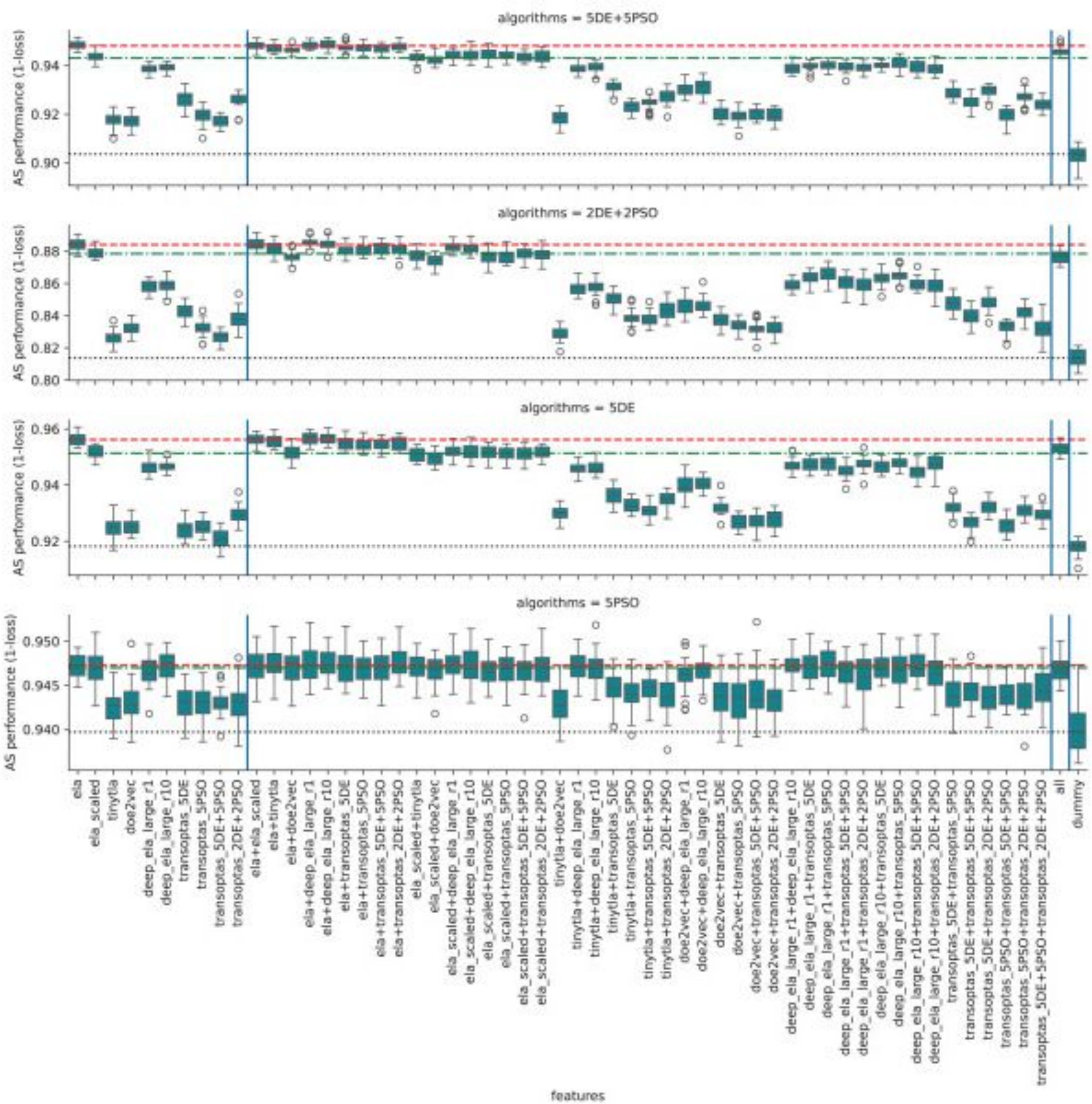
Have we hit a wall in Algorithm Selection generalization?



AS performance obtained for the different feature groups and algorithm portfolios in the instance split evaluation.

Cenikj, G., Petelin, G., Seiler, M., Cenikj, N., & Eftimov, T. (2025). Landscape Features in Single-Objective Continuous Optimization: Have We Hit a Wall in Algorithm Selection Generalization?. *Swarm and Evolutionary Computation*, *94*, 101894.

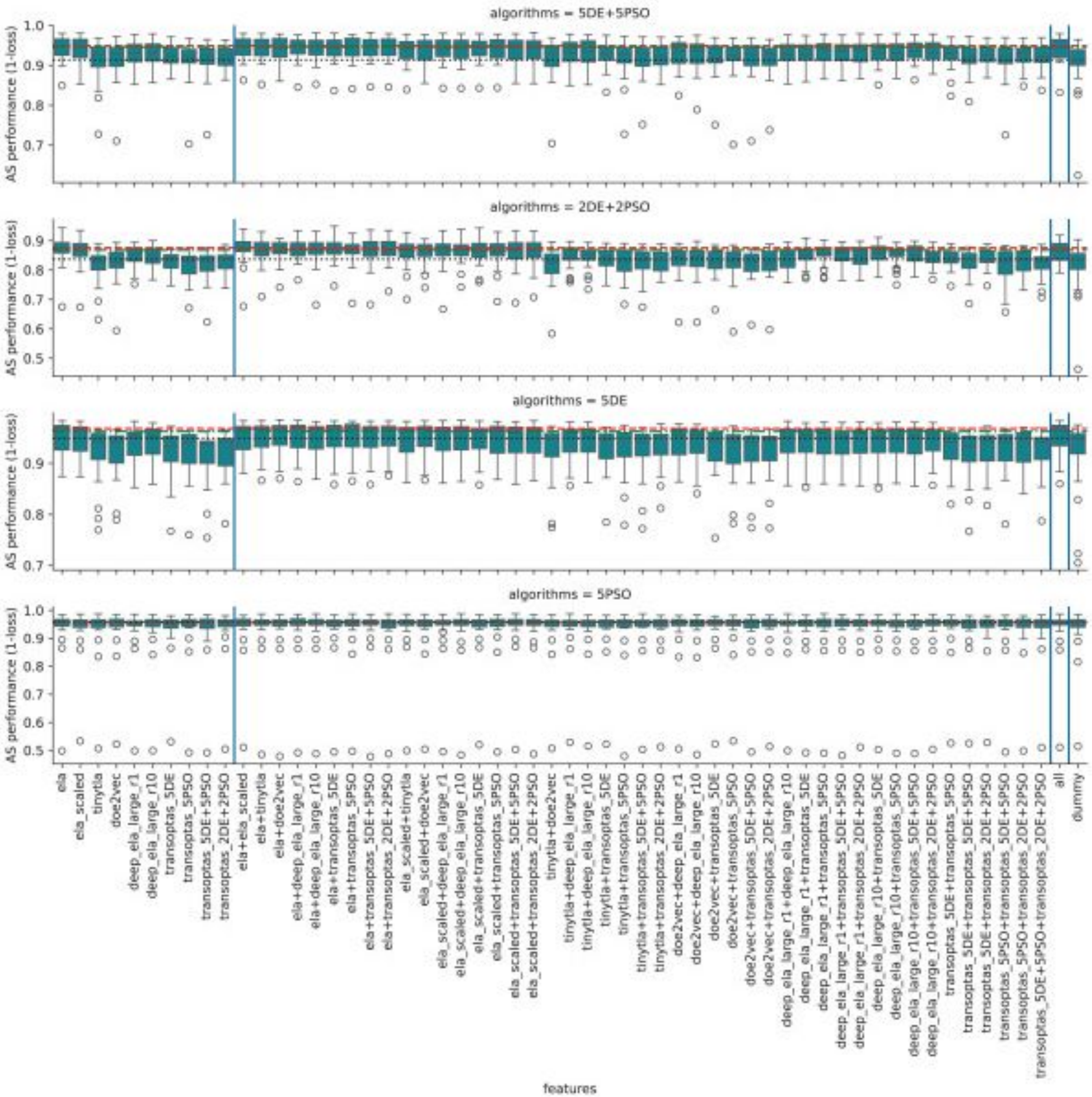
Have we hit a wall in Algorithm Selection generalization?



AS performance obtained for the different feature groups and algorithm portfolios in the random split evaluation.

Cenikj, G., Petelin, G., Seiler, M., Cenikj, N., & Eftimov, T. (2025). Landscape Features in Single-Objective Continuous Optimization: Have We Hit a Wall in Algorithm Selection Generalization?. *Swarm and Evolutionary Computation*, *94*, 101894.

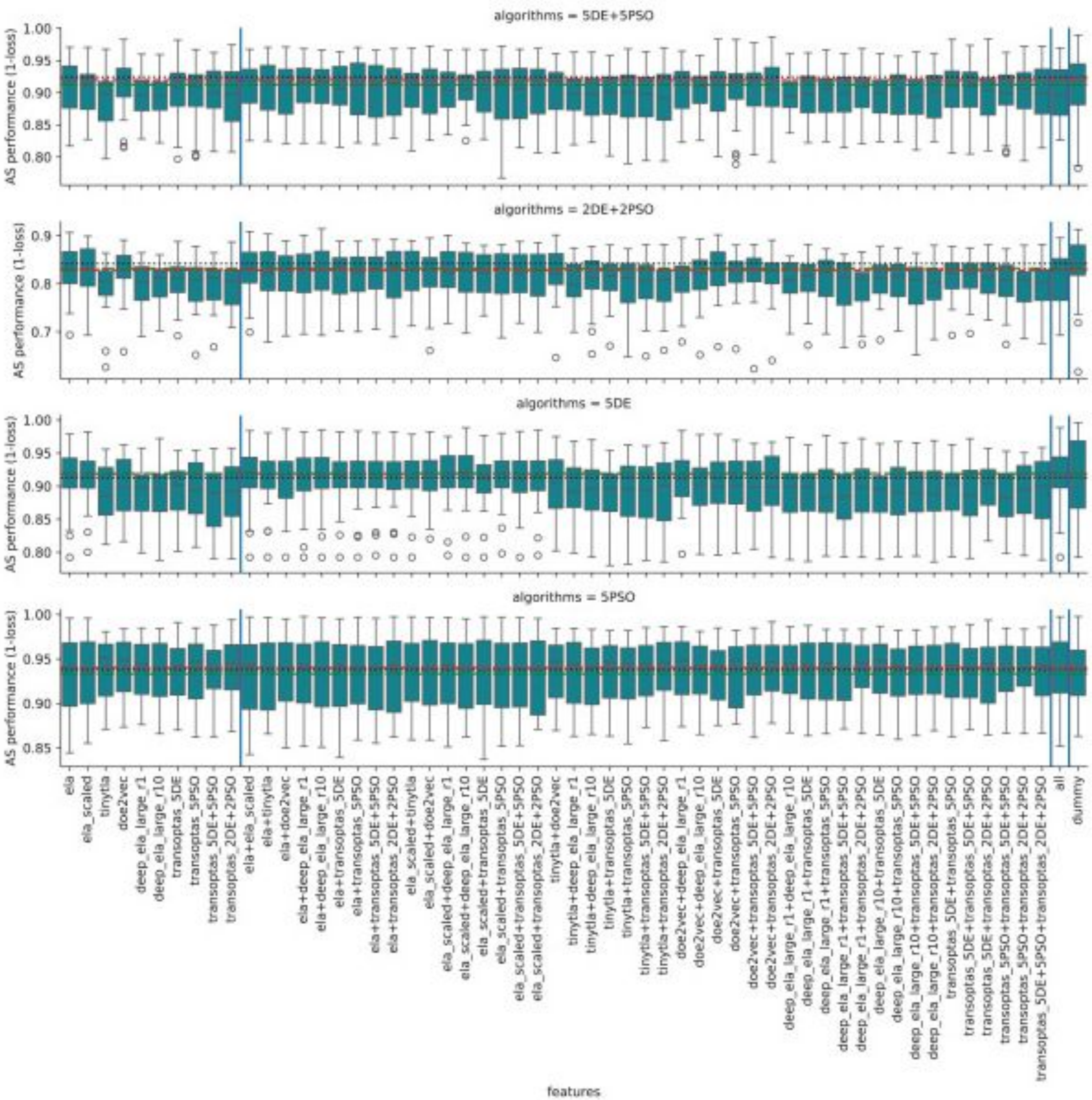
Have we hit a wall in Algorithm Selection generalization?



AS performance obtained for the different feature groups and algorithm portfolios in the problem combination split evaluation.

Cenikj, G., Petelin, G., Seiler, M., Cenikj, N., & Eftimov, T. (2025). Landscape Features in Single-Objective Continuous Optimization: Have We Hit a Wall in Algorithm Selection Generalization?. *Swarm and Evolutionary Computation*, *94*, 101894.

Have we hit a wall in Algorithm Selection generalization?



AS performance obtained for the different feature groups and algorithm portfolios in the problem split evaluation.

Cenikj, G., Petelin, G., Seiler, M., Cenikj, N., & Eftimov, T. (2025). Landscape Features in Single-Objective Continuous Optimization: Have We Hit a Wall in Algorithm Selection Generalization?. *Swarm and Evolutionary Computation*, *94*, 101894.

Landscape of studies grouped based on different features

	Learning tasks				Benchmark suites			Problem generators				
Features	Problem classification	Algorithm selection	Performance prediction	Visualization / Complementarity	BBOB	CEC	Nevergrad	ISA	GP	TR	Affine	GKLS
Problem landscape features												
ELA	[127, 73, 77, 67, 81]	[1, 63, 128, 9, 129, 130, 58, 100, 9, 131]	[55, 128, 132, 57, 54, 56, 133, 134, 135, 129, 136, 36, 108]	[137, 127, 138, 11, 97, 139] [31, 66, 36, 82, 83, 135]	[137, 127, 73, 1, 63, 129, 138, 128, 11, 97, 139, 55, 136, 77, 132, 57, 54, 81, 135, 31, 36, 82, 83, 56, 66, 100, 108, 130, 9, 58, 131]	[136, 138, 11, 132, 133, 134] [83, 82]		[31]	[139]	[97, 66, 9, 100]	[137, 36, 100, 58]	[138]
TLA	[67, 90]	[90]			[67]							
Fitness Map + CNNs	[68]				[68]							
Point Cloud Transformer	[68]				[68]							
DoE2Vec	[95]				[95]					[95]		
TransOpt	[98]				[98]						[100]	
Deep-ELA	[101]	[101]			[101]							
Random Filter Mappings	[104]	[104]			[104]							
Algorithm features												
Source Code		[48]										
High-level problem-algorithm interaction features												
Fitness Map + CNNs		[92]			[92]							
TransOptAS		[105, 100]			[100]					[105, 100]	[100]	
Performance	[49]			[49]	[49]							
Explainable Prediction Models			[132, 57]		[132, 57]							
Internal Algorithm Parameters	[15]	[113]	[15]		[15]							
KG embeddings			[50]		[50]							
GNN embeddings			[51]		[51]							
fANOVA				[52]	[52]							
fANOVA			[108]		[108]							
Trajectory-based features												
Trajectory-ELA	[111]	[112, 113, 114, 140]			[111, 112, 113, 114, 140]		[113]					
DynamoRep	[13]				[13]							
Opt2Vec	[64]					[64]						
Iterative-ELA	[64]											
LON				[116, 118]	[116, 118]							
Probing trajectories		[120]			[120]							
ClustOpt				[121]	[121]							

Cenikj, G., Nikolikj, A., Petelin, G., van Stein, N., Doerr, C., & Eftimov, T. (2024). A Survey of Meta-features Used for Automated Selection of Algorithms for Black-box Single-objective Continuous Optimization. arXiv preprint arXiv:2406.06629.

Open Challenges

Sensitivity to problem transformations, sample size and sampling method
Problem benchmarks
Generalizability

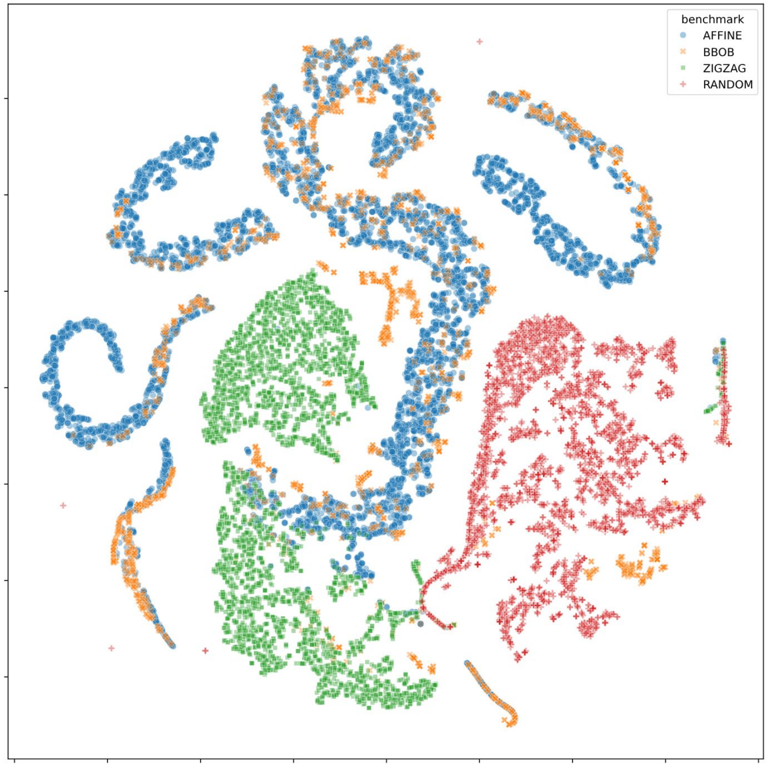
Sensitivity to problem transformations, sample size and sampling method

- Some features are sensitive to transformations of the problem (scaling/shifting)
- Most of the features are sensitive to the size of the sample and the method of sampling the candidate solutions
- Holistic approach looking including different features portfolio

Problem benchmarks

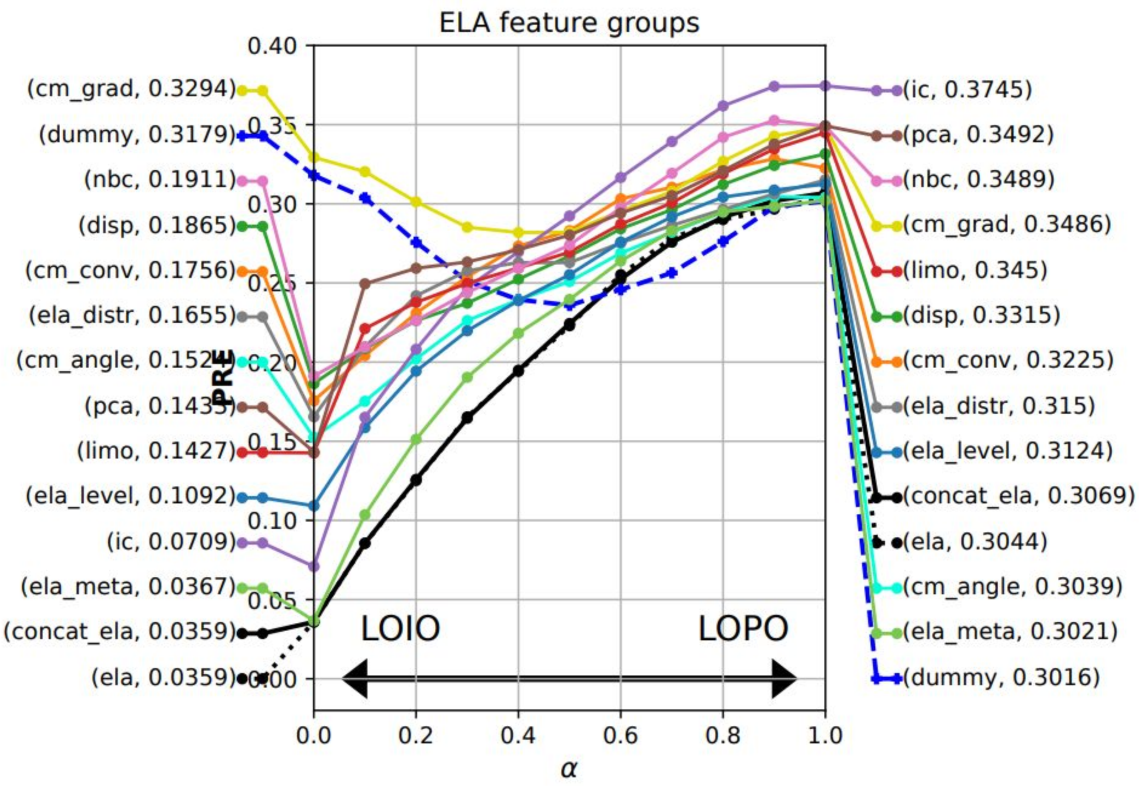
- Lack of problem benchmarks which are representative of real-world problems, and have sufficient diversity and size for training ML models
- The most commonly used BBOB benchmark contains only 24 problems, from which various instances can be generated (low diversity)
- Problem generators are being explored

Problem benchmarks



Cenikj, G., Petelin, G., Eftimov, T. (2024). Impact of Scaling in ELA Feature Calculation on Algorithm Selection Cross-Benchmark Transferability. In *Proceedings of the IEEE Congress on Evolutionary Computation*.

Generalizability



Petelin, G., Cenikj, G. (2024). On Generalization of ELA feature groups. In Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '24 Companion). Association for Computing Machinery, New York, NY, USA, 419–422. https://doi.org/10.1145/3638530.3654124

Key takeaways

Be careful with your experimental setup and evaluation
Be loud and honest about failures
Figure out *why* things are not working

Acknowledgements

Acknowledgements

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 grants No.PR-12393 to GC and No. PR-12897 to AN. This work is also funded by the European Union under Grant Agreement 101187010 (HE ERA Chair AutoLearn-SI).

AUTOLEARN-SI

LEVERAGING BENCHMARKING DATA FOR AUTOMATED MACHINE LEARNING AND OPTIMIZATION

GECCO

Funded by the European Union

References

References are provided through the slides.

AutoLearn-SI

- HE ERA-Chair
- 2.5 million EUR
- Starting date: 01.03.2025
- Scope: AutoML and AutoOPT
- **3 Ph.D. positions** starting at 01.10.2025
- **2 Postdoc Positions** starting at 01.07.2026
- More info: info-autolearnsi@ijs.si

AUTOLEARN-SI

LEVERAGING BENCHMARKING DATA FOR AUTOMATED MACHINE LEARNING AND OPTIMIZATION

Follow us:

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.