Computer Systems
Jožef Stefan Institute

Jožef Stefan Institute
Ljubljana, Slovenia

Recent Advances in Meta-features for Automated Single-Objective Black-Box Optimization



Gjorgjina Cenikj, Ana Nikolikj, Tome Eftimov

AutoML 2024
Paris, France

Importance of BBO in ML

- **Hyperparameter Tuning**
  - It enables efficient search in hyperparameter spaces, crucial for maximizing model performance.
- **Robustness to Uncertainty**
  - Handles noisy or unknown objective functions.
- **Versatility Across Domains**
  - Applicable in various ML tasks like reinforcement learning, neural architecture search, and model selection.
- **Handling High-Dimensional Spaces**
  - Designed to explore large search spaces efficiently, crucial in ML.

Motivation

- Automated algorithm selection requires the application of machine learning (ML) models to the optimization domain, lies at the intersection of optimization and ML
- Similar tasks and challenges as the AutoML community
- Representation learning methodologies and analytical approaches from one domain could be transferred to the other
- Opportunities for mutual learning and collaboration

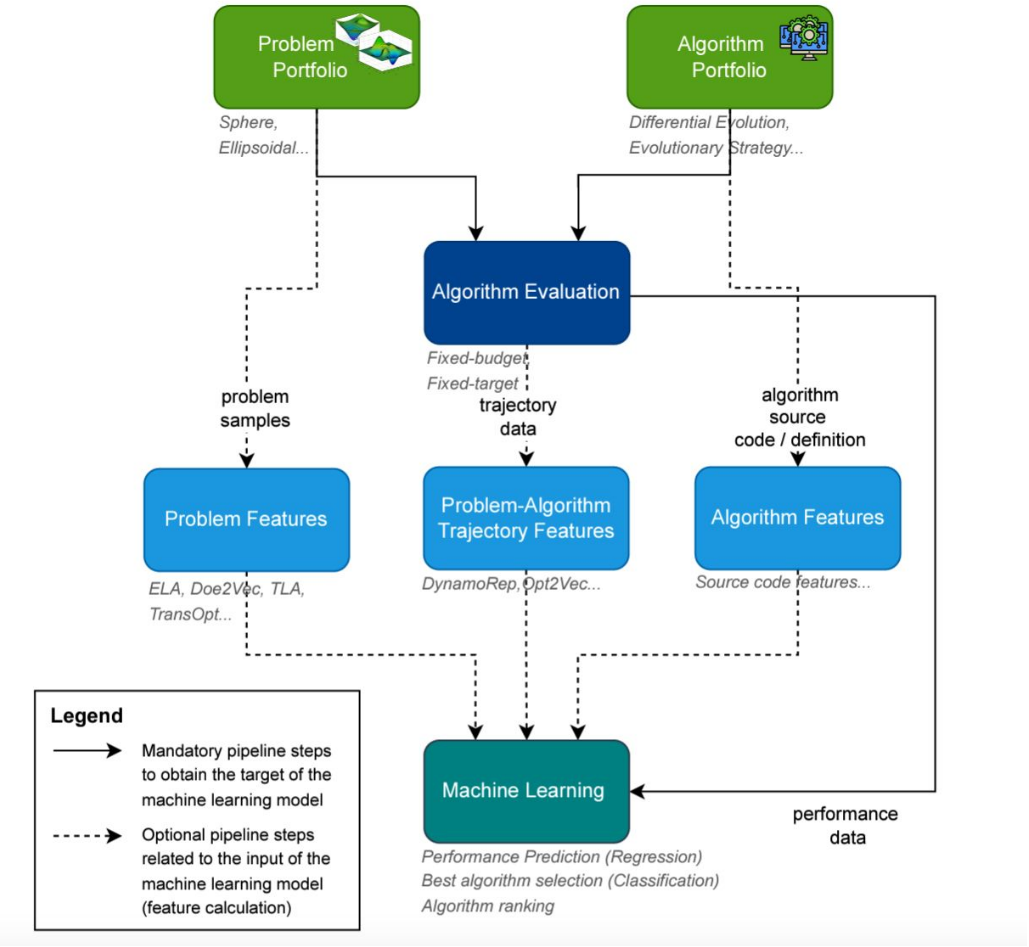
Problem Definition

- An optimization function p, is given by: $p : X \rightarrow Y$, where $X \subseteq R^d$ is the decision and $Y \subseteq R^m$ the objective space. In this context, $x \in X$ is referred to as a candidate solution.
- This tutorial is focused on single-objective optimization (SOO), i.e., m=1.

Algorithm Selection

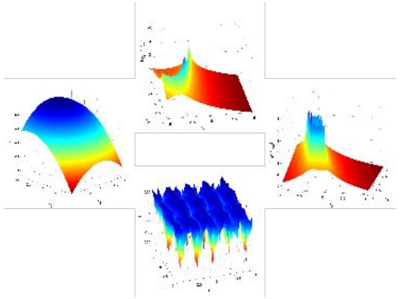
- Aims to identify the best algorithm (from an existing set of algorithms) to solve a given problem
- Leverage algorithm complementarity instead of looking for a single algorithm which works best across all problems

Algorithm Selection Pipeline



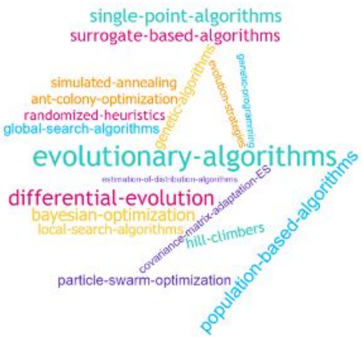
Algorithm Selection in Numerical Black-Box Optimization

Selection of a problem portfolio



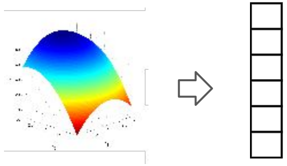
What types of optimization problems will be included in our problem portfolio?

Selection of an algorithm portfolio



Which algorithms will be incorporated into our algorithm portfolio?

Feature Representation



- ❖ Problem features
- ❖ Algorithm features
- ❖ 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!!! [1]

Problem Features

- Exploratory Landscape Analysis
- Topological Landscape Analysis
- Features based on deep learning

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

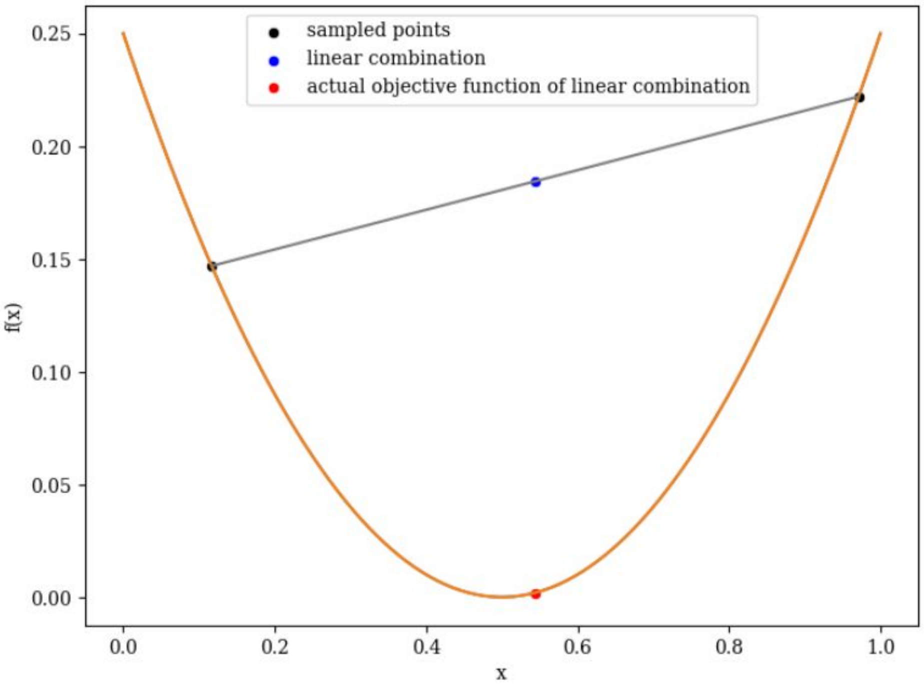
DIMENSION + 1

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}

ELA - Convexity features

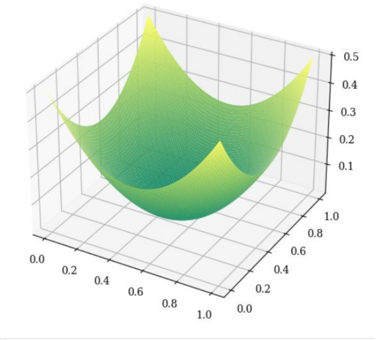
- Based on differences in the objective values of a point which is a linear combination of two randomly sampled points and the convex combination of the objective values of the two sampled points.

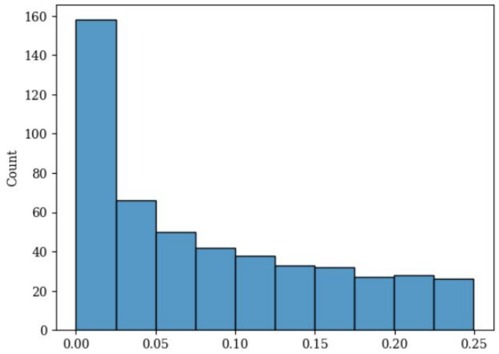


Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., & Rudolph, G. (2011, July 12). Exploratory landscape analysis. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. Presented at the GECCO '11: Genetic and Evolutionary Computation Conference, Dublin Ireland.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

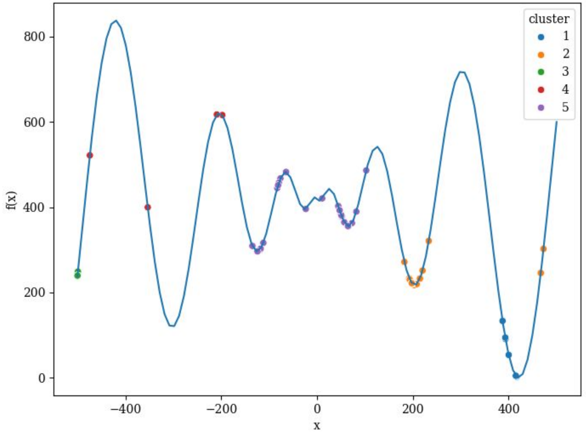




Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., & Rudolph, G. (2011, July 12). Exploratory landscape analysis. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. Presented at the GECCO '11: Genetic and Evolutionary Computation Conference, Dublin Ireland.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

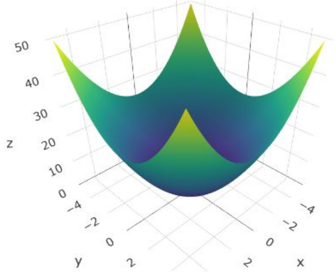
ELA - Local search features

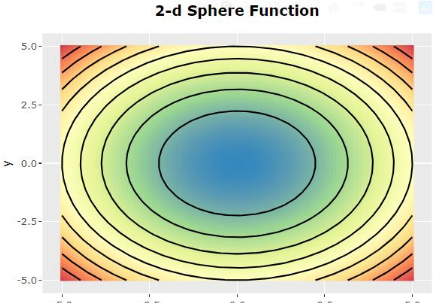
- Local search features - extracted by running a local search algorithm and hierarchically clustering the considered solutions in order to approximate problem properties. For instance, the number of clusters is an indicator of multi-modality, while the cluster sizes are related to the basin sizes around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



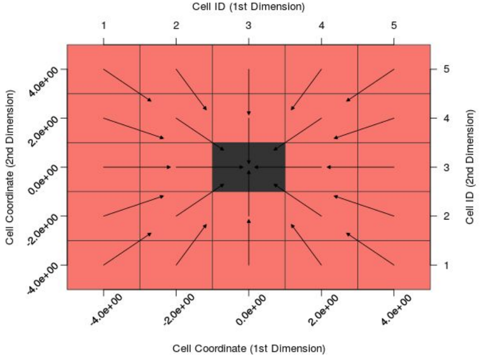
Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., & Rudolph, G. (2011, July 12). Exploratory landscape analysis. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. Presented at the GECCO ’11: Genetic and Evolutionary Computation Conference, Dublin Ireland.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

ELA - Cell mapping features





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



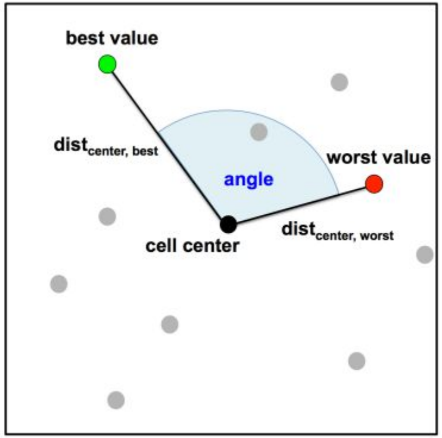
Example features:
gcm.mean.pcells = 0.04 (relative number of periodic cells)
gcm.mean.tcells = 0.96 (relative number of transient cells)
gcm.mean.best_attr.prob = 1 (probability of finding the attractor with the best objective value)

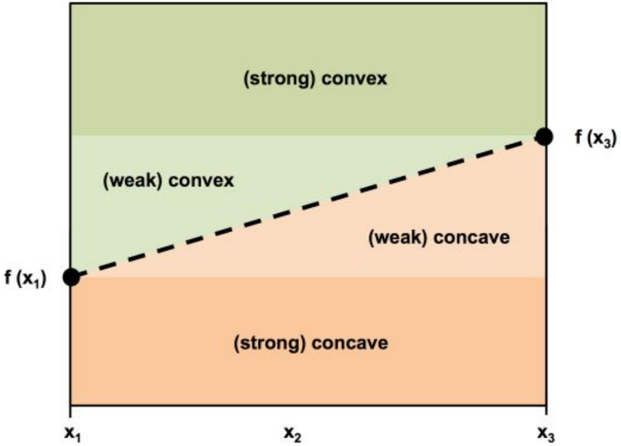
Kerschke, P., Preuss, M., Hernández, C., Schütze, O., Sun, J.-Q., Grimme, C., Rudolph, G., Bischl, B., & Trautmann, H. (2014). Cell Mapping Techniques for Exploratory Landscape Analysis. In Advances in Intelligent Systems and Computing (pp. 115–131). Springer International Publishing. https://doi.org/10.1007/978-3-319-07494-8_9

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

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 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

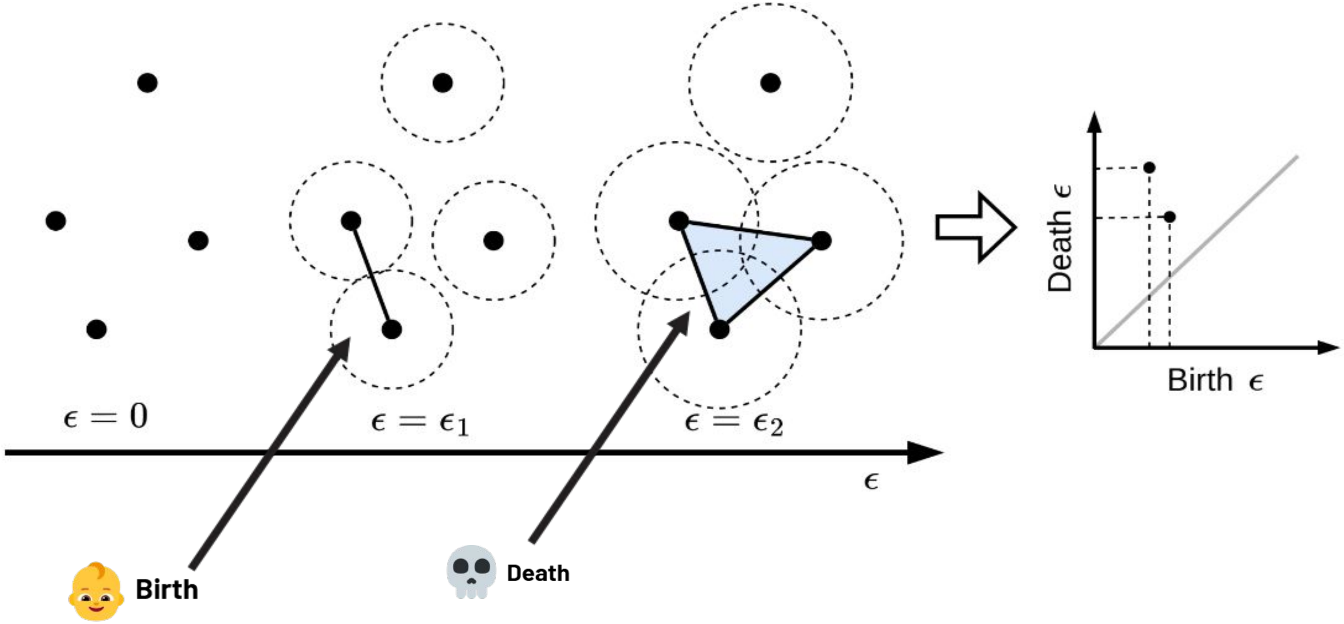
Topological Landscape Analysis

- 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

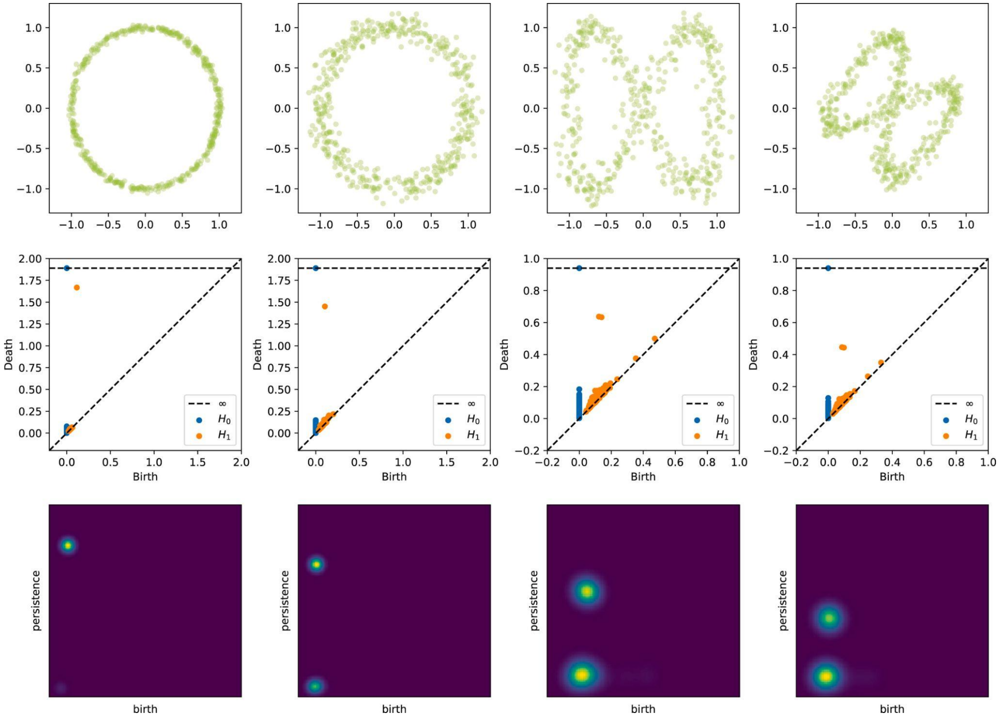
Topological Landscape Analysis

- 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

Fitness map features

- Process:
  - Generate candidate solutions using Latin Hypercube sampling
  - Calculate fitness map as a 2D image with a single channel in the [0,1] range
  - Normalize objective solution values
  - Map candidate solutions to a Cartesian plane based on decision variables and objective values.
- Model: CNN (ShuffleNet v2)
- Task: algorithm selection of 32 CMAES configurations
- Data: BBOB benchmark, 124 instances per problem
- Weakness: Potential information loss when different candidate solutions are mapped to the same pixel.

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

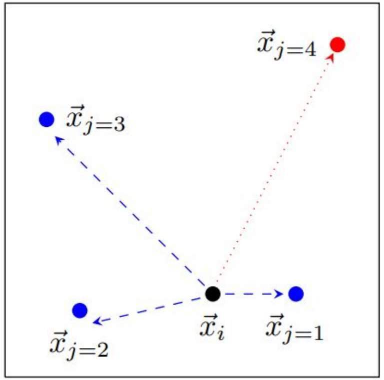
Fitness map features - Extension to higher dimensions

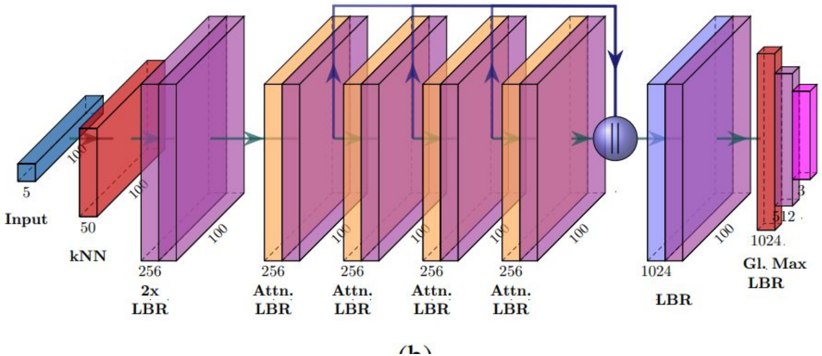
- Adaptation of the fitness map approach for high-dimensional data using dimensionality reduction techniques
- Task: Evaluated for the task of predicting high-level features of BBOB problem instances (multimodality, global structure, funnel structure, etc).
- Data: BBOB benchmark, 150 instances per problem,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

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.





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

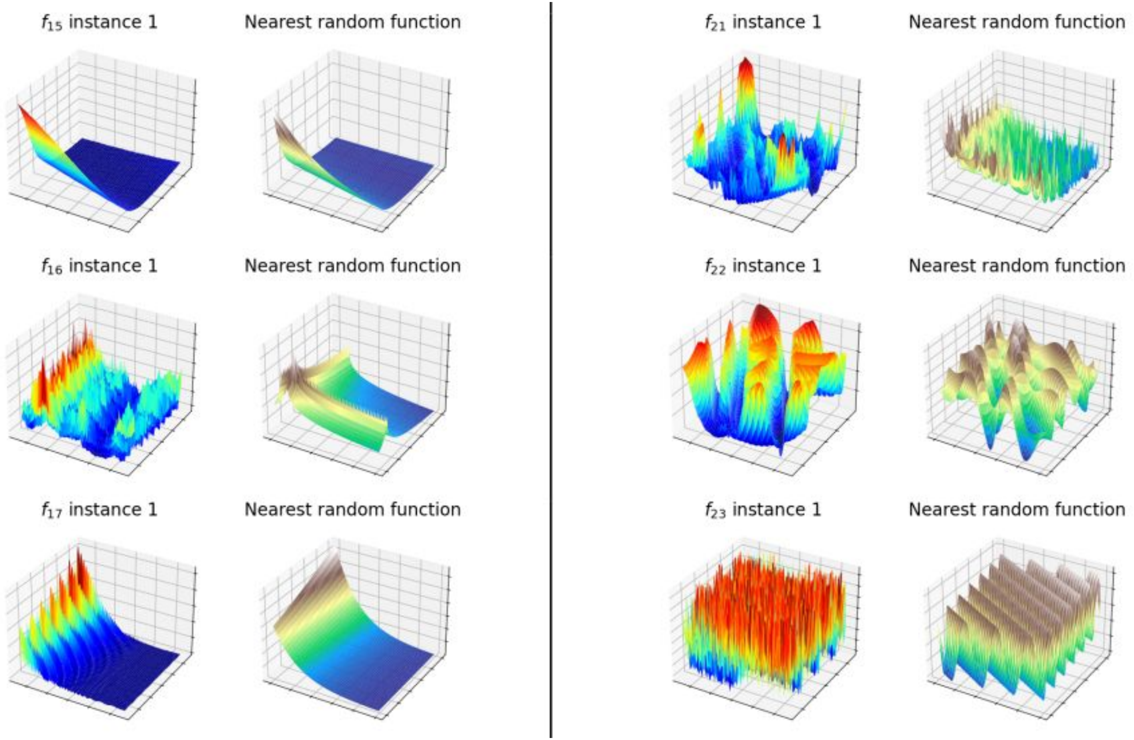
Doe2Vec

- 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

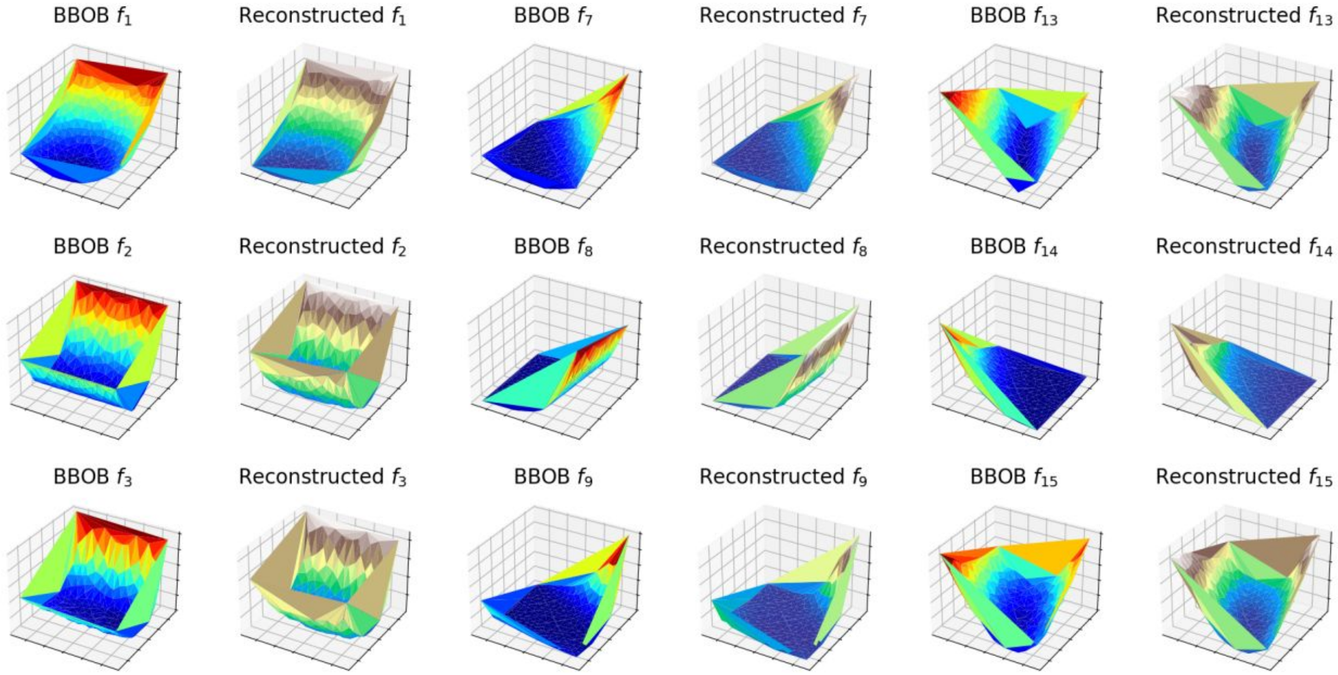
BBOB functions and their most similar random function in terms of Doe2Vec features



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

Doe2Vec

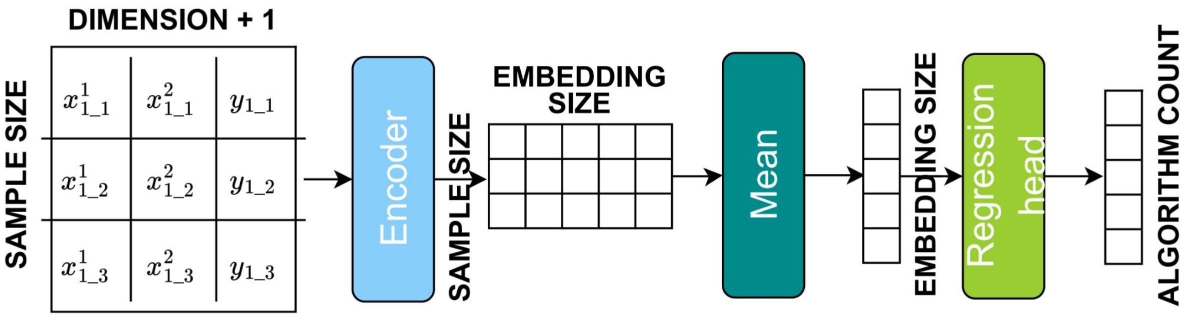
Classification results: macro F1

d	Task	AE-24	AE-32	VAE-24	VAE-32	ELA	PCA*	rMC*	Transformer*	ELA-VAE-32
2	multimodal	0.875	0.849	0.877	0.856	0.984	0.994	0.971	0.991	**0.991**
	global struct.	0.903	0.904	0.902	0.889	0.983	0.992	0.965	0.991	**0.998**
	funnel	0.985	0.974	0.956	0.978	**1.000**	0.999	0.995	1.000	**1.000**
5	multimodal	0.908	0.903	0.880	0.889	0.963	0.897	0.947	0.991	**0.998**
	global struct.	0.838	0.828	0.810	0.793	**1.000**	0.807	0.859	0.978	**1.000**
	funnel	**1.000**	**1.000**	0.996	0.991	**1.000**	0.990	0.989	1.000	**1.000**
10	multimodal	0.877	0.813	0.844	0.838	**1.000**	0.839	0.952	0.974	**1.000**
	global struct.	0.794	0.737	0.783	0.745	0.902	0.774	0.911	0.963	**0.991**
	funnel	0.998	0.993	0.997	0.993	0.972	0.977	0.991	**1.000**	0.997
20	multimodal	0.726	0.722	0.700	0.694	0.970	-	-	-	**0.991**
	global struct.	0.689	0.621	0.606	0.626	0.972	-	-	-	**0.997**
	funnel	0.993	0.982	0.985	0.982	**1.000**	-	-	-	**1.000**

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

TransOptAS

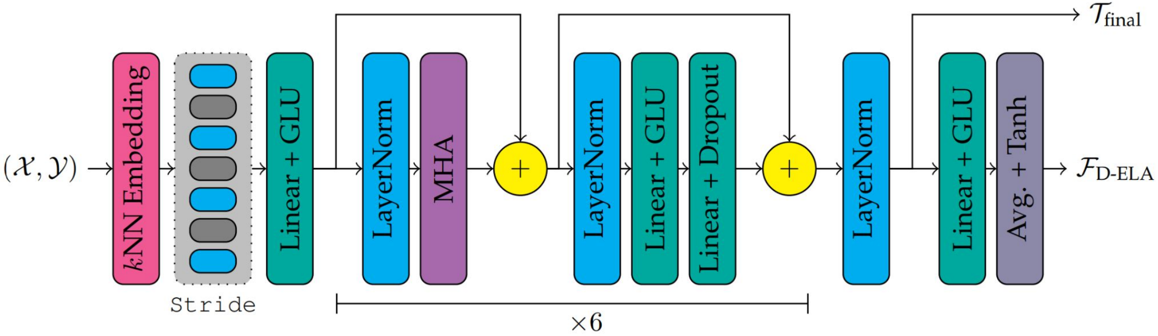
- Process:
  - Generate candidate solutions using Latin Hypercube sampling
  - Train transformer model, which given samples of the optimization function, predicts algorithm performance
- Data: Functions generated using a random function generator
- Task: Algorithm selection



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

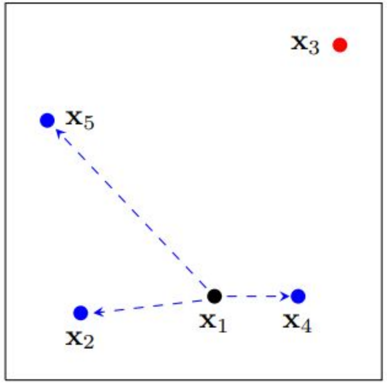


Seiler M.V., Kerschke P., Trautmann H. 2024. Deep-ELA: Deep Exploratory Landscape Analysis with Self-Supervised Pretrained Transformers for Single- and Multi-Objective Continuous Optimization Problems https://arxiv.org/abs/2401.01192

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



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

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

Seiler M.V., Kerschke P., Trautmann H. 2024. Deep-ELA: Deep Exploratory Landscape Analysis with Self-Supervised Pretrained Transformers for Single- and Multi-Objective Continuous Optimization Problems https://arxiv.org/abs/2401.01192

DeepELA

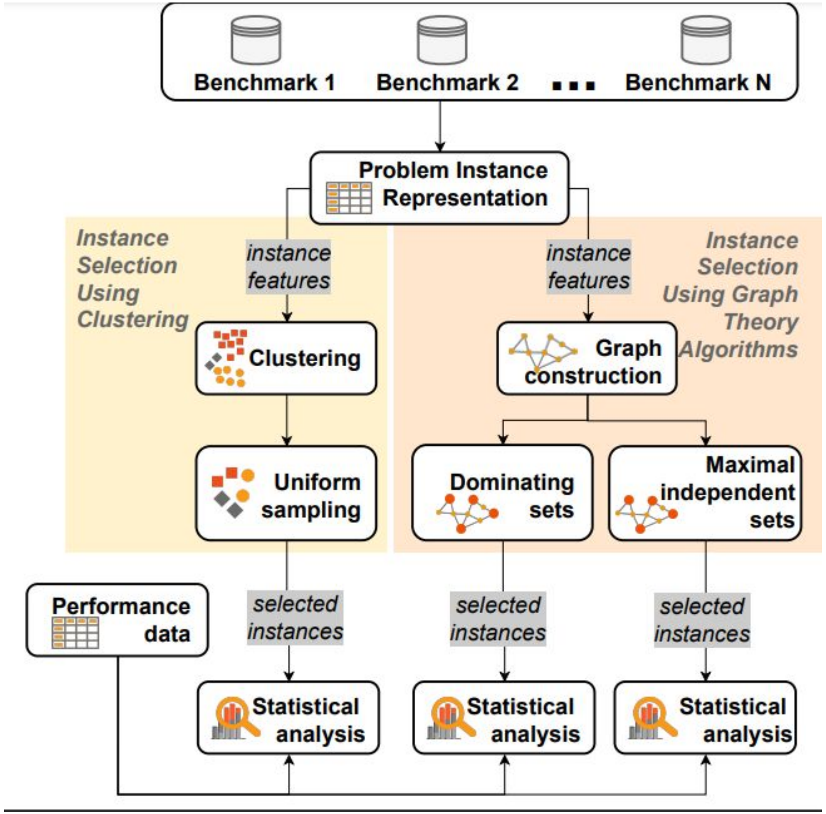
- Student-teacher training strategy with a shared backbone acting as a feature generator
- The training strategy revolves around providing distinct, augmented versions of the same objective instance to both the teacher and student. Here, the teacher generates target projections from which the student 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. 2024. Deep-ELA: Deep Exploratory Landscape Analysis with Self-Supervised Pretrained Transformers for Single- and Multi-Objective Continuous Optimization Problems https://arxiv.org/abs/2401.01192

Application: Selection of diverse benchmark problem instances

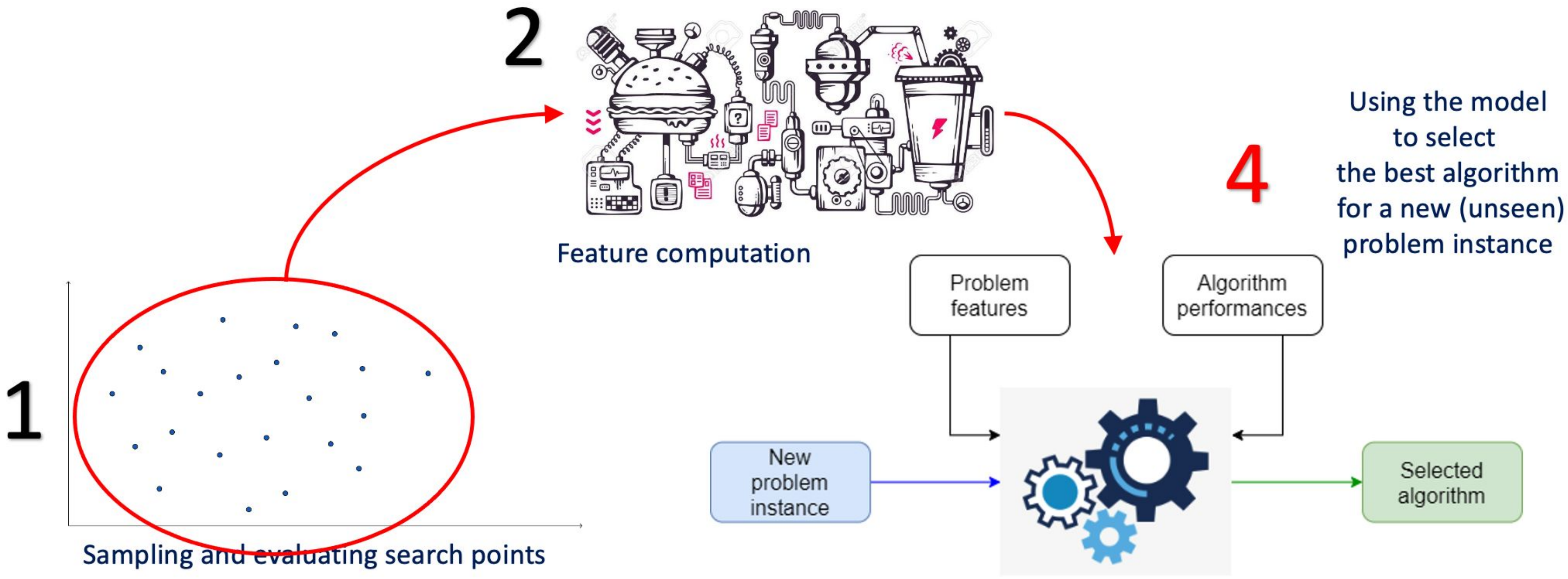
- Instance selection using clustering
- Instance selection using graph theory algorithms
  - Dominating sets
  - Maximal independent sets

- Improved benchmark datasets for training ML models and performing statistical analysis

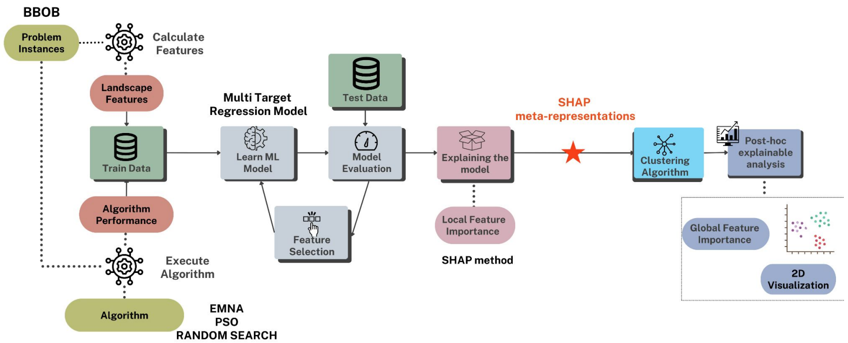


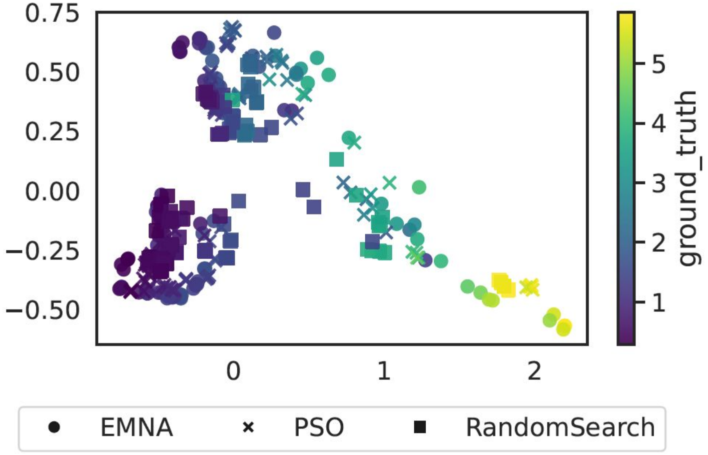
Cenikj, G., Lang, R. D., Engelbrecht, A. P., Doerr, C., Korošec, P., & Eftimov, T. (2022, July). Selector: selecting a representative benchmark suite for reproducible statistical comparison. In *Proceedings of The Genetic and Evolutionary Computation Conference* (pp. 620-629).

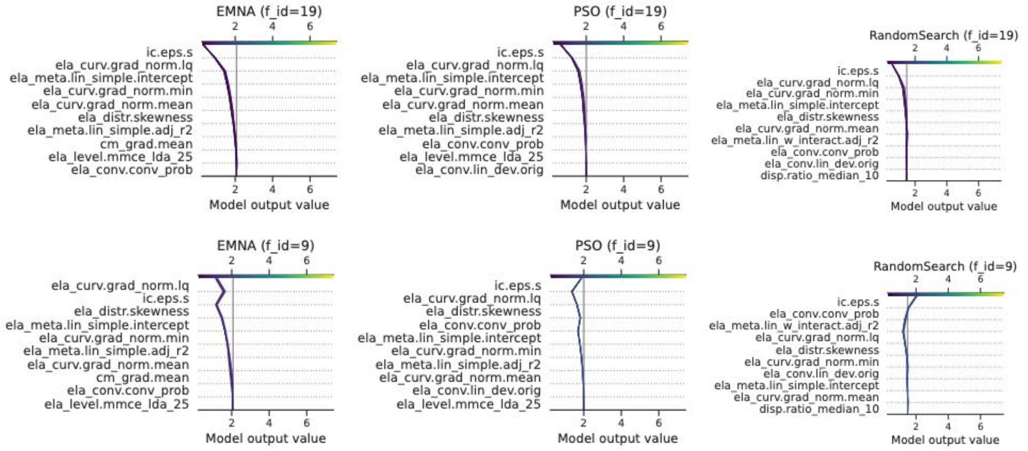
Application: Per-Instance Algorithm Selection



Application: Explainable Algorithm Footprint







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
- Based on performance
- Based on Shapley values of performance predictive model
- Via Knowledge Graph

Algorithm features based on source code

Extracting algorithm features from source code

Props: 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.

Algorithm 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 …

Props:

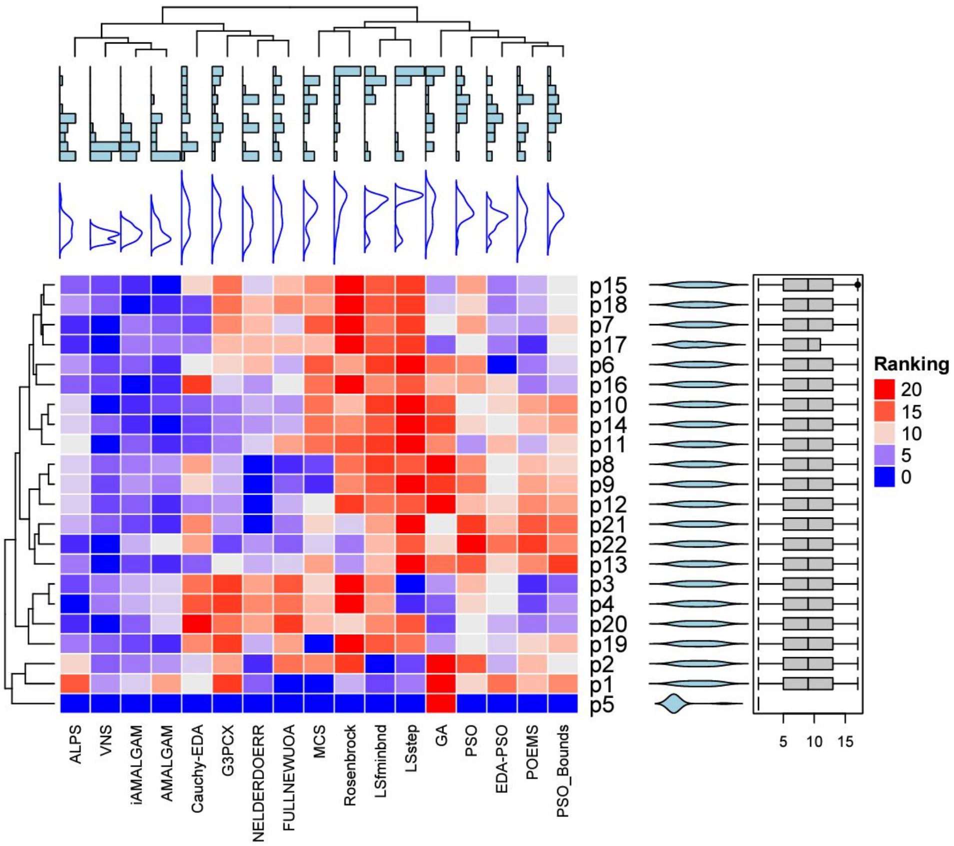
- Facilitates algorithm comparison through performance vectors.

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).

Algorithm features based on performance



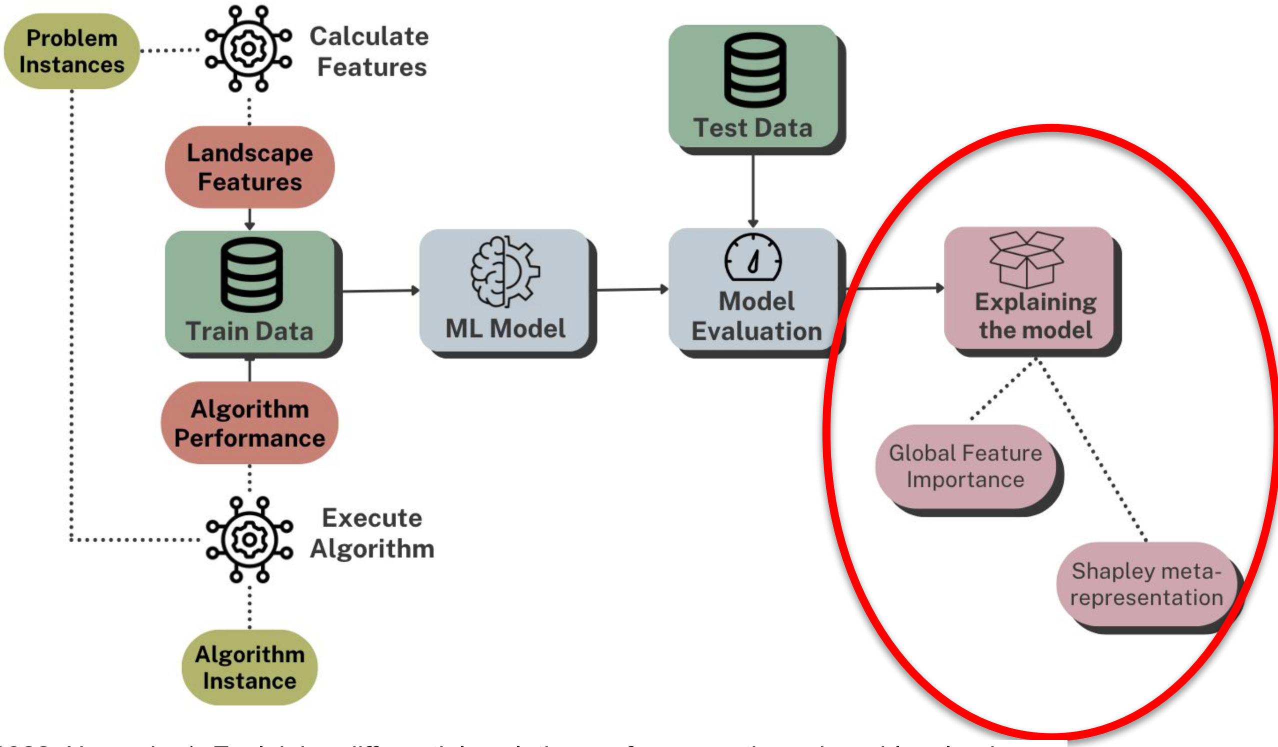
17 algorithms were compared using 22 benchmark problems from BBOB 2009 (dimension 10). Hierarchical clustering was applied to **Performance2Vec** embeddings (columns) and benchmark problem embeddings (rows). The matrix was reorganized to group similar algorithms and problems together. Colors indicate 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).

Algorithm 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.

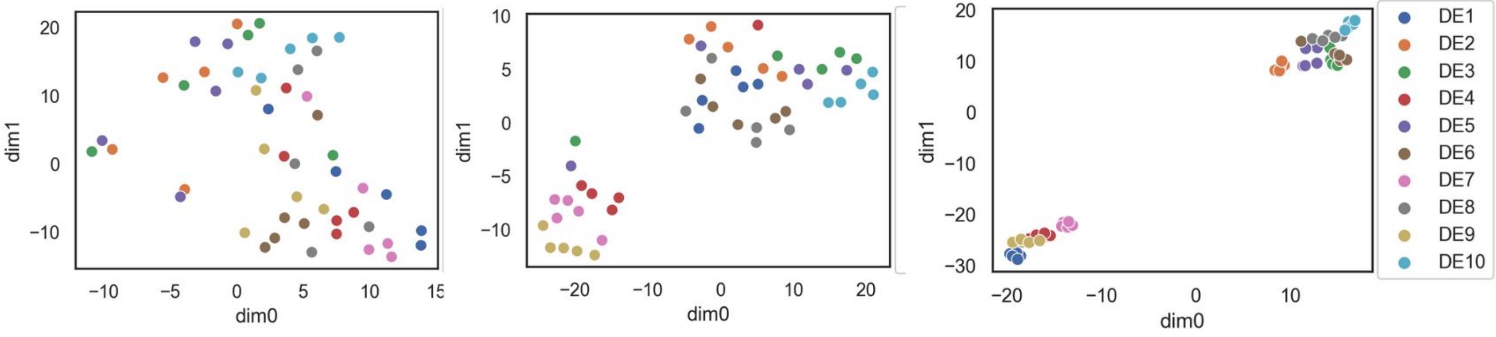
Algorithm features based on Shapley values of performance predictive models

Props:

- 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

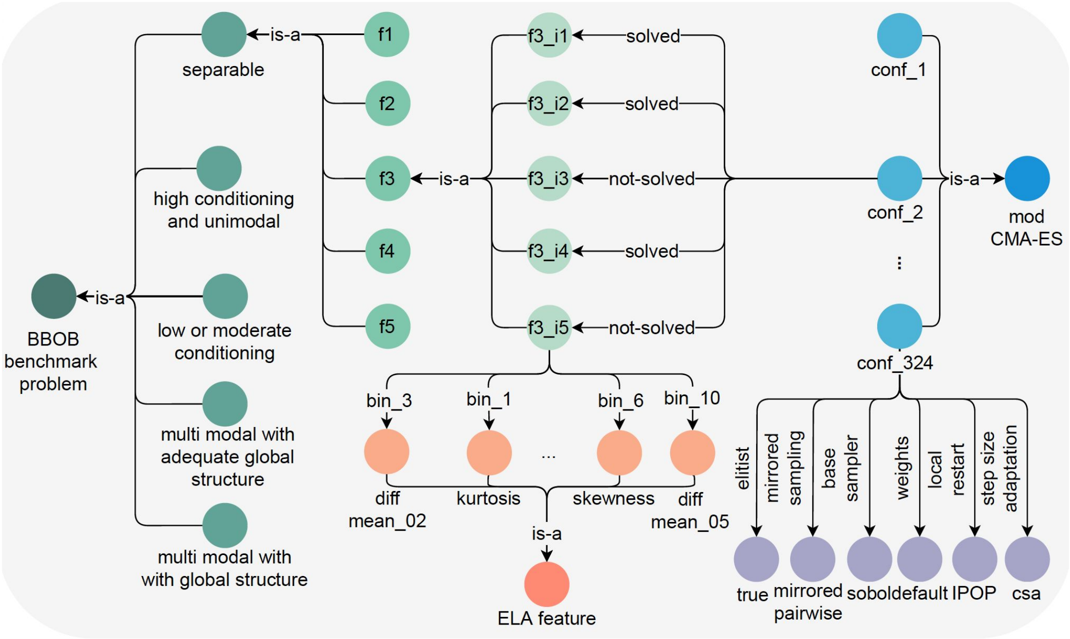
Nikolikj, A., Lang, R., Korošec, P., & Eftimov, T. (2022, November). Explaining differential evolution performance through problem landscape characteristics. In *International Conference on Bioinspired Optimization Methods and Their Applications* (pp. 99-113). Cham: Springer International Publishing.

Algorithm 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.

OPTION KG

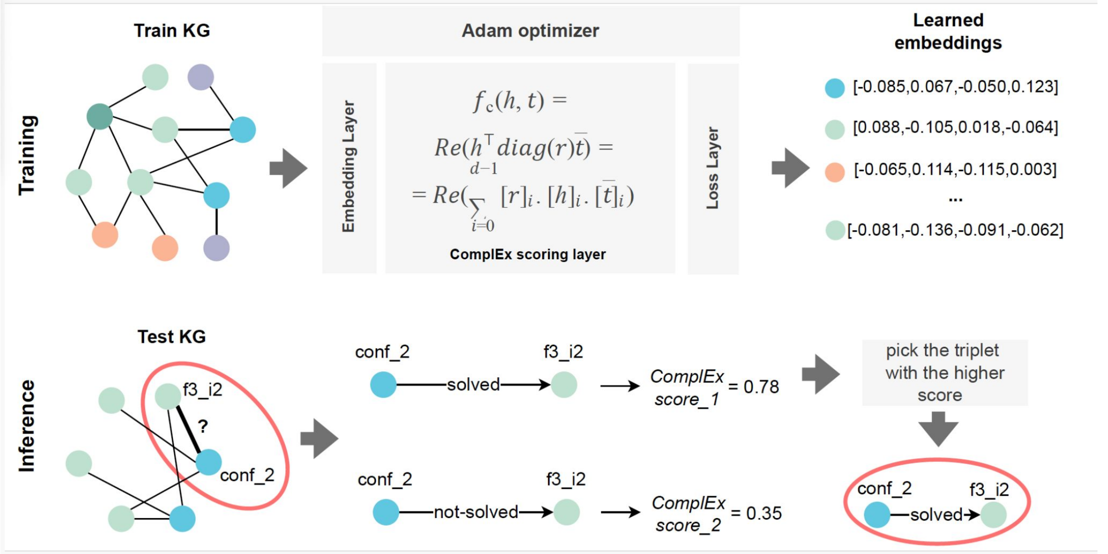


Kostovska, A., Vermetten, D., Džeroski, S., Panov, P., Eftimov, T., & Doerr, C. (2023, April). Using knowledge graphs for performance prediction of modular optimization algorithms. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 253-268). Cham: Springer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

Algorithm 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.



Props:

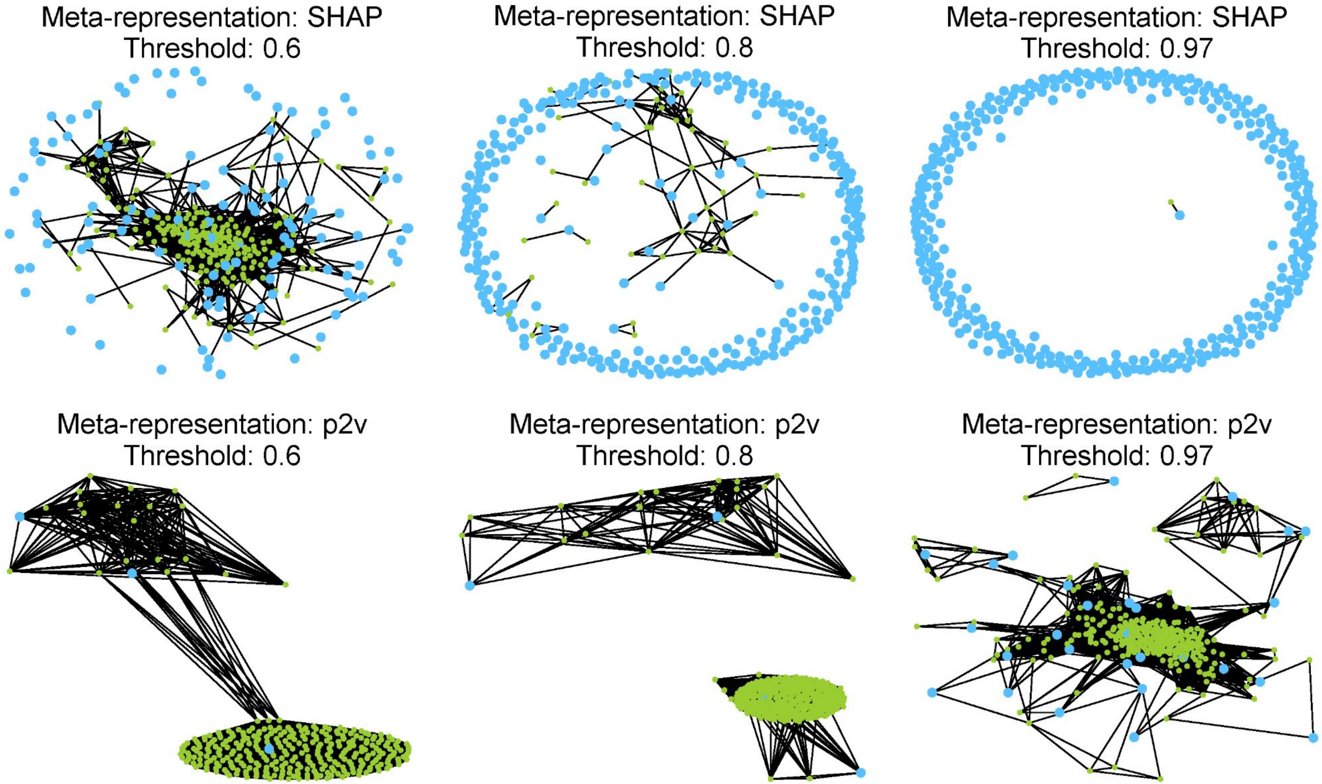
- Encodes interactions between problem features and algorithm performance by also involving the graph neighbourhood.

Cons:

- Depends on the data stored in the KG

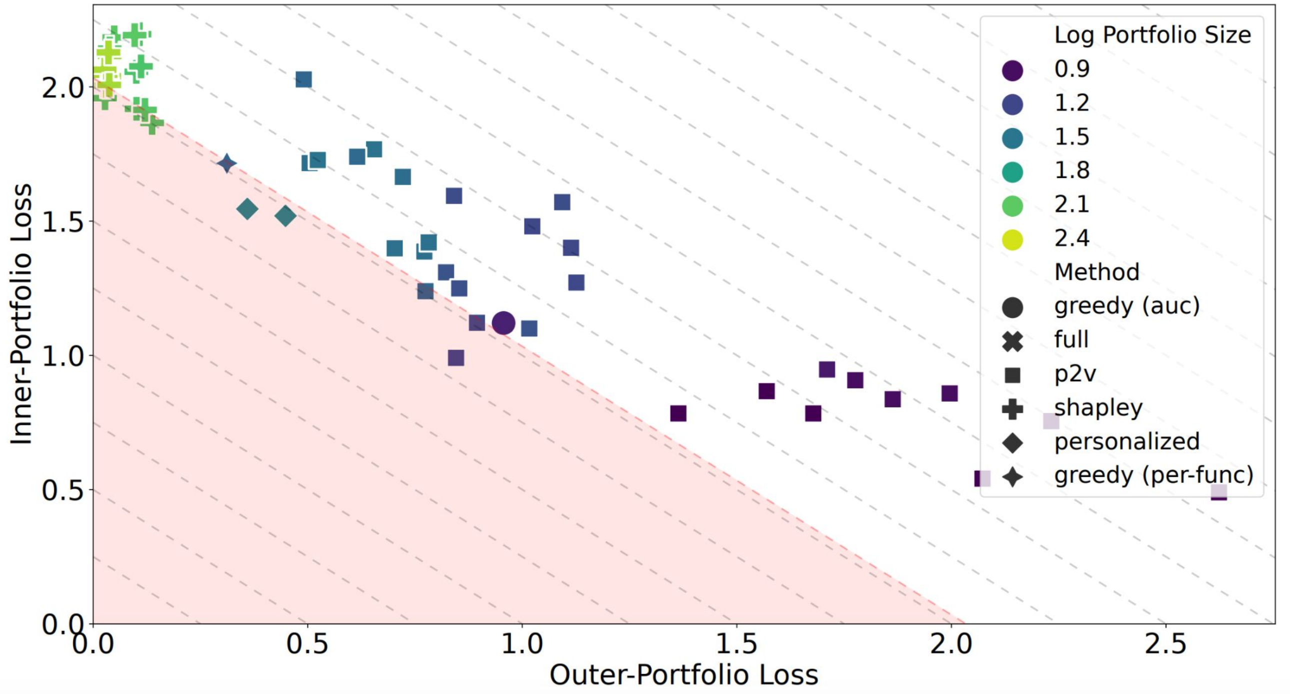
Kostovska, A., Vermetten, D., Džeroski, S., Panov, P., Eftimov, T., & Doerr, C. (2023, April). Using knowledge graphs for performance prediction of modular optimization algorithms. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 253-268). Cham: Springer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

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.

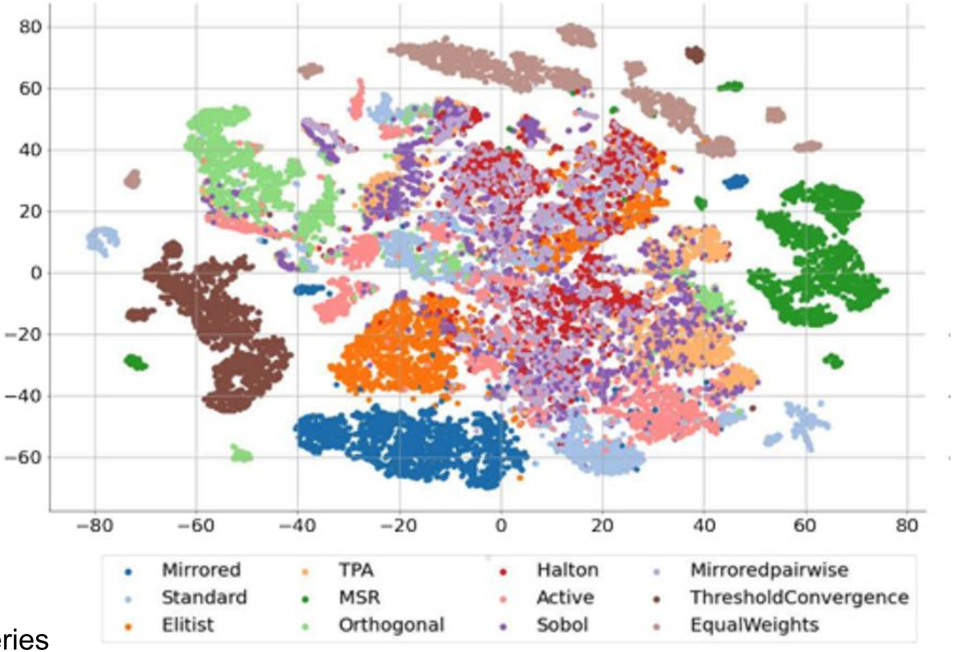
Problem-Algorithm Trajectory Features

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

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



Props: Capture the behaviour of the algorithm

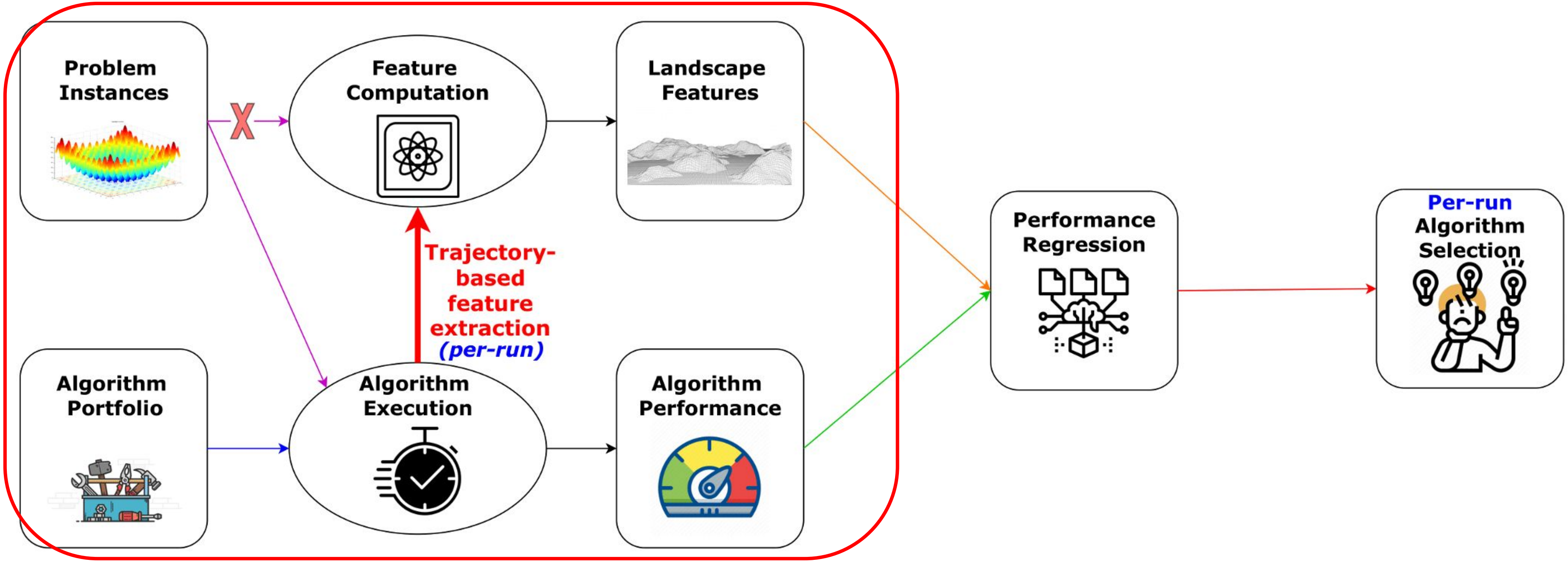
Cons: Lack of comprehensive comparison of different time series features

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.

Props:

- 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.

Props:

- 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.

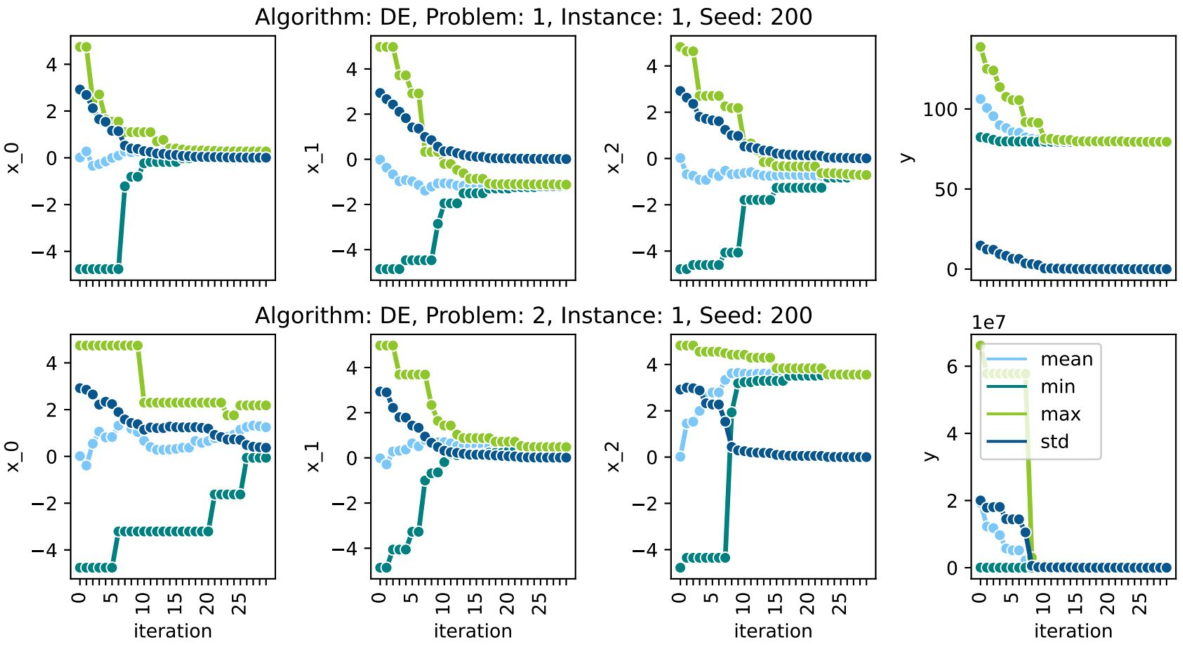
Cons:

- 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.

DynamoRep features

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

- **Props**:
  - 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.

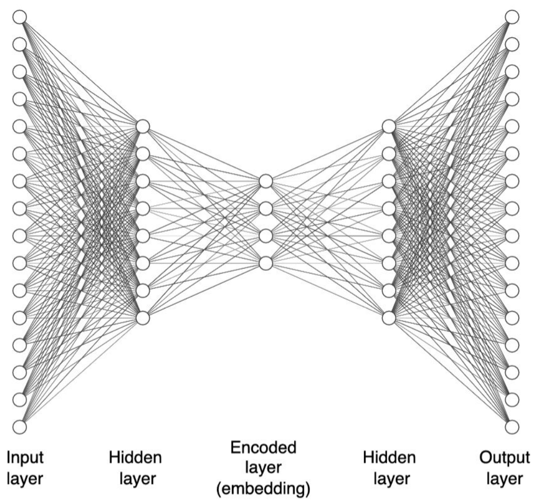
- **Cons**:
  - Limited expressiveness
  - Representation size grows with number of iterations and problem dimension, may require dimensionality reduction as preprocessing step

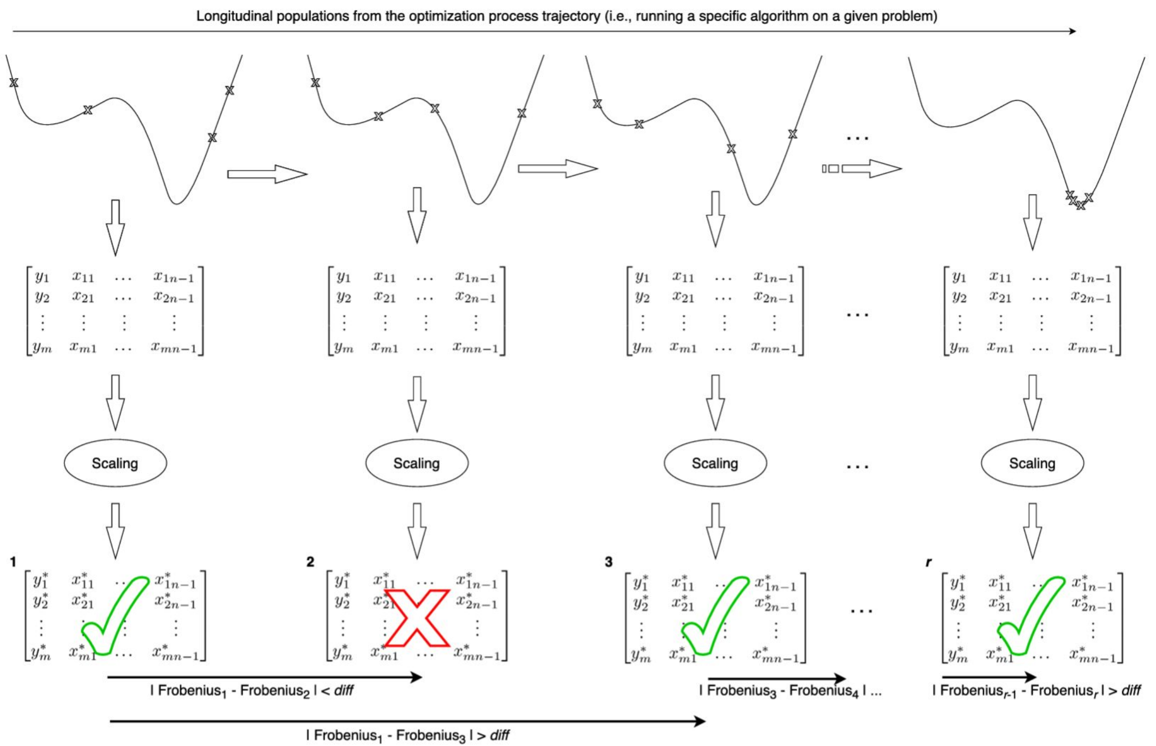
Cenikj, G., Petelin, G., Doerr, C., Korošec, P., & Eftimov, T. (2023, July). Dynamorep: trajectory-based population dynamics for classification of black-box optimization problems. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 813-821).

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.

Props:

- 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:

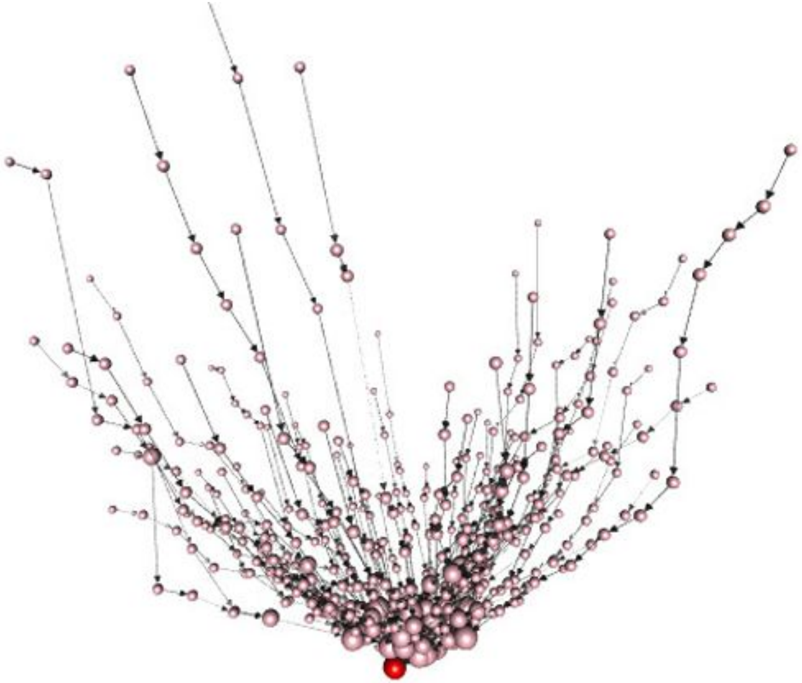
- Depends on the data used to train the autoencoder

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).

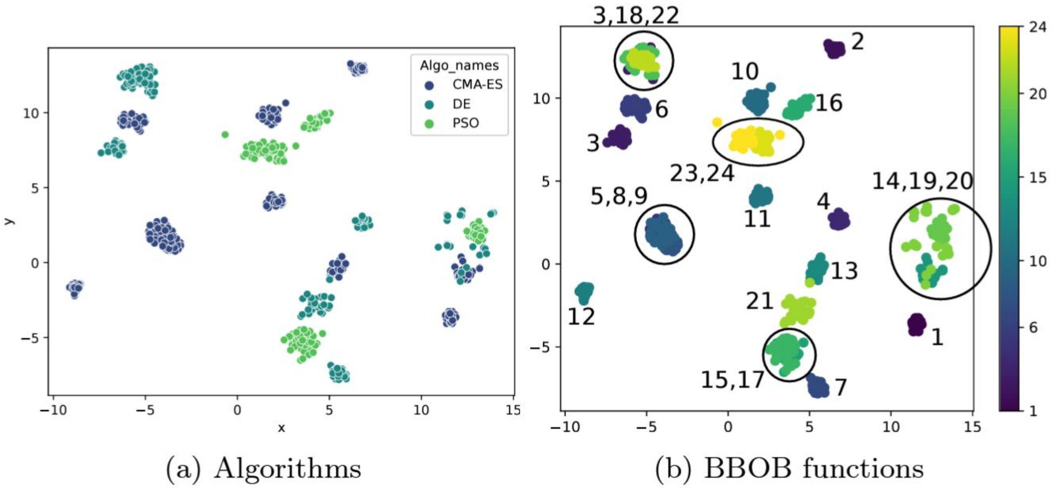
Local Optima Networks (LONs) and variants

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



Probing trajectories similarity

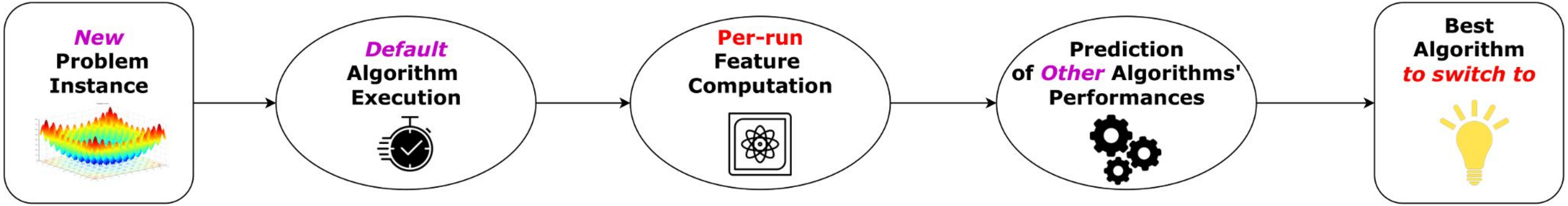
Renau, Q., & Hart, E. (2024, March). On the Utility of Probing Trajectories for Algorithm-Selection. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 98-114). Cham: Springer Nature Switzerland.

Probing trajectories

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

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.

Application: Per-run Algorithm Selection

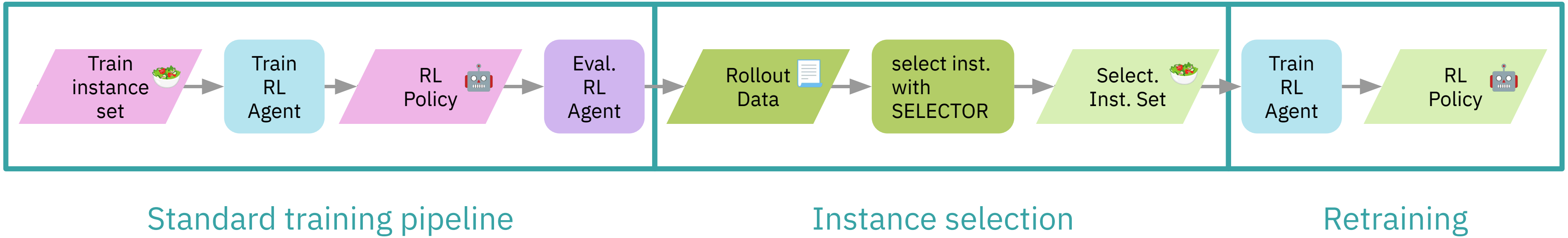


Kostovska, A., Jankovic, A., Vermetten, D., de Nobel, J., Wang, H., Eftimov, T., & Doerr, C. (2022, August). Per-run algorithm selection with warm-starting using trajectory-based features. In *International Conference on Parallel Problem Solving from Nature* (pp. 46-60). Cham: Springer International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

Application: Trajectory features for DAC 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



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).

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	[97], [43], [47], [37], [50]	[1], [36], [98], [27], [99], [96], [95], [69], [27]	[29], [98], [100], [31], [28], [30], [101], [102], [103], [99], [104], [13], [105]	[106], [97], [107], [48], [66], [108] [9], [51], [13], [52], [53], [103]	[106], [97], [43], [1], [36], [99], [107], [98], [48], [66], [108], [29], [104], [47], [100], [31], [28], [50], [103], [9], [13], [52], [53], [30], [51], [69], [105], [96], [27]	[104], [107], [48], [100], [101], [102] [53], [52]		[9]	[108]	[66], [51], [27], [69]	[106], [13], [69], [95]	[107]
TLA	[37], [59]	[59]			[37]							
Fitness Map + CNNs	[38]	[61]			[61], [38]							
Point Cloud Transformer	[38]				[38]							
DoE2Vec	[64]				[64]					[64]		
TransOpt	[67]	[70], [69]			[67], [69]					[70], [69]	[69]	
Deep-ELA	[71]	[71]			[71]							
Algorithm features												
Source Code		[24]										
Performance	[25]			[25]	[25]							
Explainable Prediction Models			[100], [31]		[100], [31]							
Internal Algorithm Parameters	[75]	[79]	[75]		[75]							
KG embeddings			[26]		[26]							
Trajectory-based features												
Trajectory-ELA	[77]	[78], [79], [80], [109]			[77], [78], [79], [80], [109]		[79]					
DynamoRep	[82]				[82]							
Opt2Vec	[83]					[83]						
Iterative-ELA	[83]											
LON				[85], [86]	[85], [86]							
Probing trajectories		[88]			[88]							

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

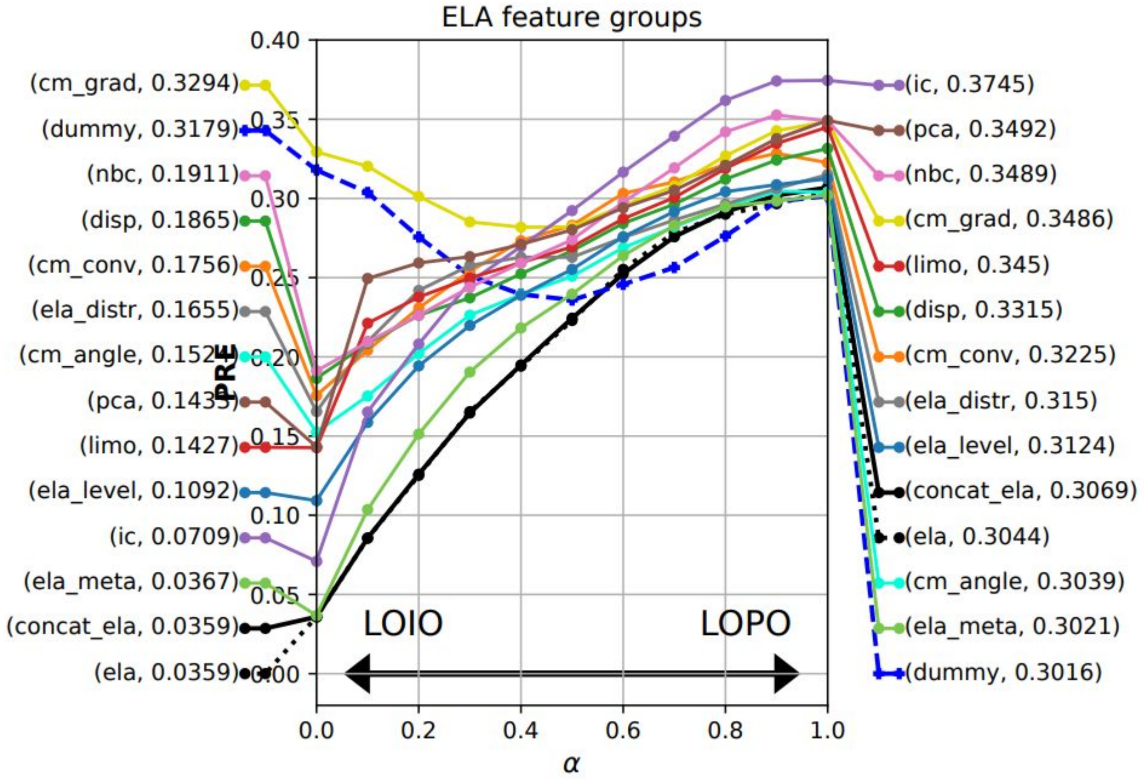
Open Challenges: Sensitivity

- 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

Open Challenges: Problem Benchmarks

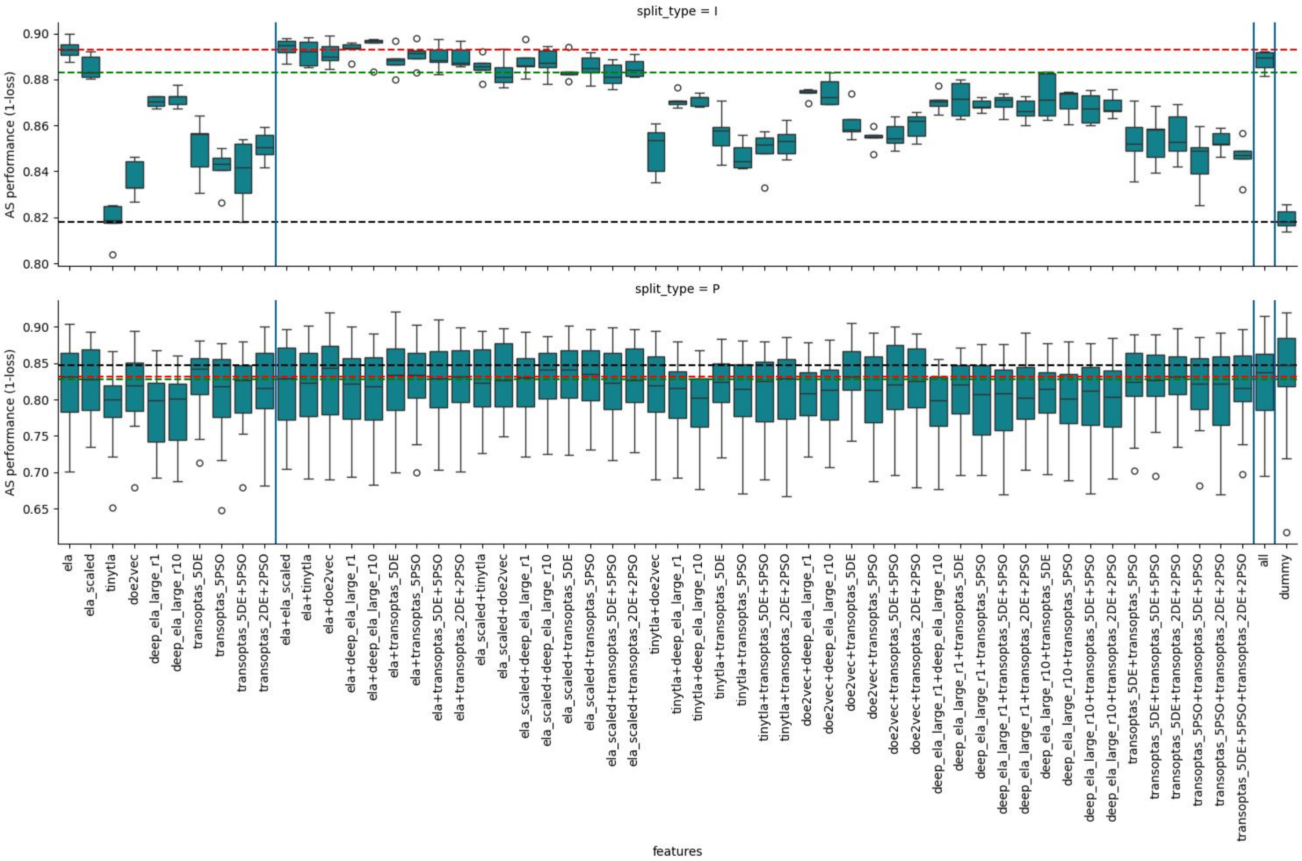
- Lack of problem benchmarks which are representative of real-world problems, and have sufficient diversity and size for training ML models
- The most commonly used BBOB benchmark contains only 24 problems, from which various instances can be generated (low diversity)
- Problem generators are being explored

Open Challenges: 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

Open Challenges: Generalizability



Questions