On the Landscapes of Combinatorial Optimization Problems

Towards the Inferrence of Structural Similarity

Sun, May 28th, 2023

Mingyu Huang

Glasgow College, UESTC

2510648h@student.gla.ac.uk

1 # Introduction

BBOPs, Fitness Landscapes, LONs and their Structural Similarity

2 # Preliminary Work

A Case Study on Number Patitioning Problem

3 # Current Work

Towards the Study of a Wider Range of Problems

4 # Applications

AutoML, Software Engineering

5 # Future Work

Software Package, Survery and Community Development

Part I

Introduction

BBOPs, Fitness Landscapes, LONs

Configurable System

Configurable systems with adjustable features are frequently encountered in real-world engineering.

- Software Systems: e.g., customizable software compilers, software programs. There are various configurable options which allow users to tailor them to adapt to specific scenario.
- Machine Learning Models: usually invovle various adjustable hyper-parameters to be tunned to yield the optimal performance on a given task.
- Operational Research, Finance, etc.

In such systems, each configuration is associated with a performance indicator, or, in evolutionary biology, may be refered to "fitness".

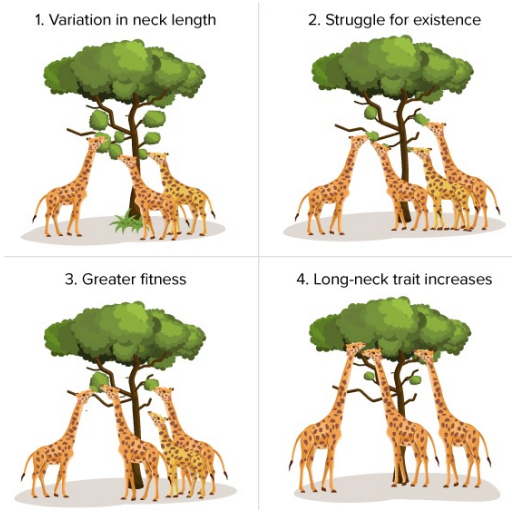


Fig. 1: Evolution of long-neck giraffes: Example of Darwin's Theory of Evolution by Natural Selection.

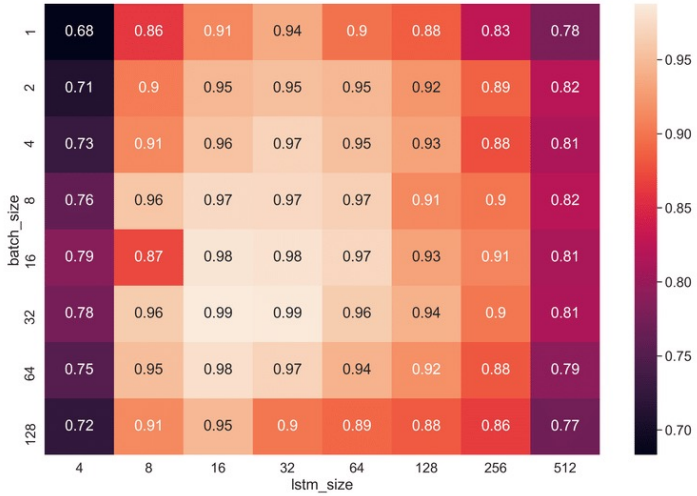


Fig. 2: Gridsearch heatmap of two hyper-parameters.

Configurable System and BBOP

Black-box Nature of Configurable System

The task of finding an optimal configuration $\mathbf{x}$ to maximize (or minimize) certain performance indicator $f(\mathbf{x})$, where $f : \mathbb{R}^n \to \mathbb{R}$, could be formulated as a black-box optimization problem (BBOP).

- Such systems are often treated as black-box, since we do not have knowledge about the explicit or authentic form of the objective function f.
- Thus the performance ("fitness") of a certain configuration is unknown until experimental observation, which could be expensive to conduct.

The Need for In-Depth Understanding of Configurable Systems

- The famous "No-Free-Lunch" theorem: there is no single algorithm that could always yield superior performance than other for all optimization tasks.
- In order to facilitate the design, selection and configuration of appropriate BBOP solvers, in-depth understanding of the characteristics of the underlying BBOP is essential.
- However, this is far from trivial…

[2] D.H. Wolpert and W.G. Macready, "No Free Lunch Theorems for Optimization", TEVC, 1997

Fitness Landscapes

1. Fitness Landscapes

- The concept of fitness landscape dates back to 1932 when Wright pioneered this in the field of evolutionary biology. A formal definition was then given by Stadler as a set: $f : (\mathcal{X}, \mathcal{N}, f)$
  - $\mathcal{X}$: The search space which contains all possible configurations
  - $\mathcal{N}$: The neighbourhood structure definition
  - $f : \mathcal{X} \to \mathbf{R}$, the fitness function

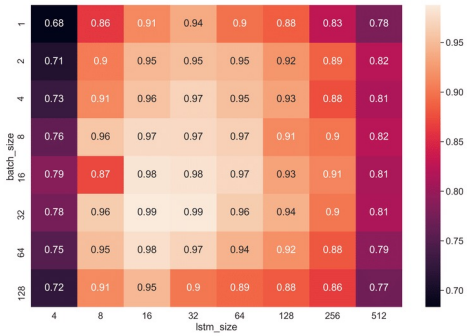


Fig. 2: A gridsearch heatmap with 2 hyper-parameters.

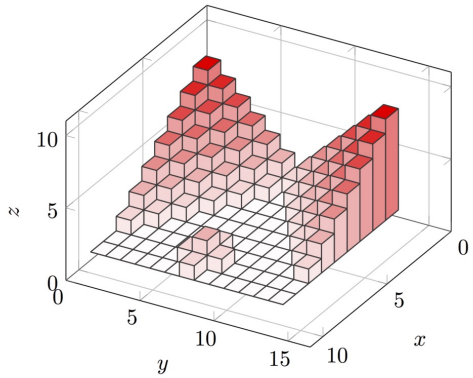


Fig. 3: A simple discrete landscape with 2 components.

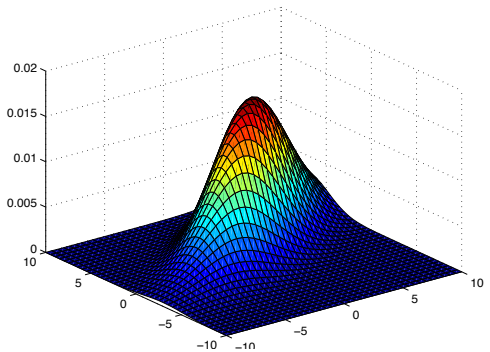


Fig. 4: A uni-model, continuous landscape.

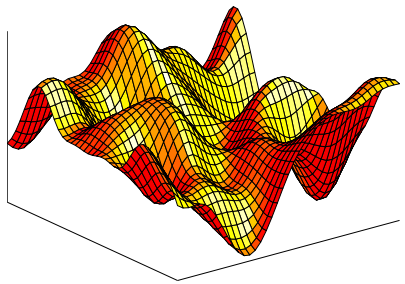


Fig. 5: A more rugged fitness lanscape.

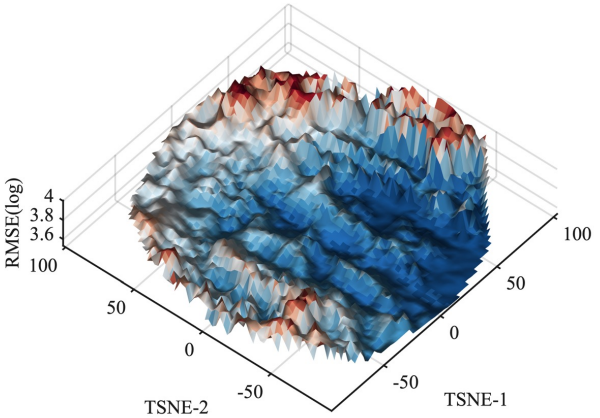


Fig. 6: XGBoost HPO landscape

[1] S. Wright, "The Roles of Mutations, Inbreeding, Crossbreeding and Selection in Evolution", Proc. of the 11th International Conference of Genetics. 1932
[2] P.F. Stadler, "Biological Evolution and Statistical Physics", Springer, 2002

Structures of Fitness Landscapes

A solver's behavior could be considered as a "walk" along the fitness landscape of a problem.

- Therefore, the performance of a BBOP solver on a given problem is highly dependent on the structural characteristics of the underlying fitness landscape.

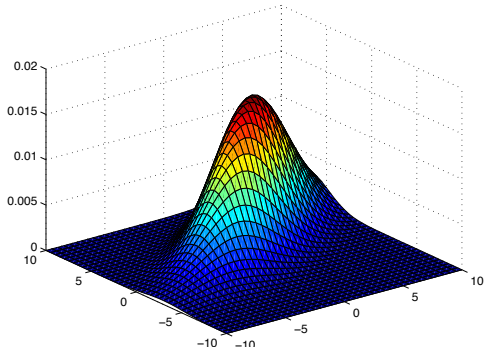


Fig. 4: A uni-model, continuous landscape.

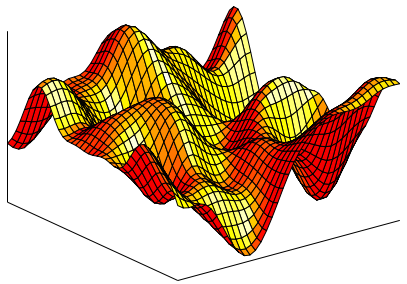


Fig. 5: A more rugged fitness lanscape.

Since the 1990s, numerious statistical metrics have been designed to capture different aspects of landscape characteristcs.

- e.g., ruggedness, neutrality, variable interaction, basin of attraction, etc.

[1] Katherine M. Malan et al., "A Survey of Techniques for Characterising Fitness Landscapes and Some Possible Ways Forward", Info. Sci., 2013
[2] Katherine M. Malan, "A Survey of Advances in Landscape Analysis for Optimisation", Algorithms, 2021
[3] Feng Zou et al, "A Survey of Landscape Analysis for Optimisation", Neurocomputing, 2022

Part II

Preliminary Work

A Case Study on Number Partitioning Problem

Introduction & Motivation

Since a solver's behavior could be considered as a "walk" along the fitness landscape of a problem. We would then wondering do fitness landscapes of different BBOPs share some similarities?

- This question is of great significance, since it would allow us to leverage the idea of analogy in problem solving.
- i.e. if we already know that a BBOP solver is effective for one problem instance, we would then expect it will also be useful for solving another problem whose fitness landscape essentially share structural similarity with each other,
- In other words, our knowledge on a familiar and well-studied problem instance could be transferred to solve a novel and unfamiliar one.

- *Retrieval*: select a set of candidate analogous instances $\mathbf{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$, where the corresponding fitness landscapes are $\mathbf{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_k\}$.
- *Judgement*: quanlitatively or quantitatively determine the structural similarity $\mathbf{Sim}(\mathcal{L}_i, \mathcal{L}_j)$ between the fitness landscapes $(\mathcal{L}_i, \mathcal{L}_j)$ for each pair of instances $(\mathcal{P}_i, \mathcal{P}_j) \in \mathcal{P}^2$ based on a similarity function $\mathbf{Sim} : \mathcal{L} \times \mathcal{L} \to [0, 1]$.
- *Inference*: apply knowledge from the most analogous instance $\hat{\mathcal{P}}_0$ to handle $\mathcal{P}_0$, where $\mathbf{Sim}(\hat{\mathcal{P}}, \mathcal{P}) \geq \max_{\mathcal{P}' \in \mathcal{P}} \mathbf{Sim}(\mathcal{P}', \mathcal{P}_0)$ for $\forall \mathcal{P}' \in \mathcal{P}$. Here, the specific knowledge applied would depend on tasks at hand.

$$\mathbf{S} = \begin{array}{c} \begin{matrix} P_0 & P_1 & P_2 & P_3 \end{matrix} \\ \begin{bmatrix} - & \times & \times & \times \\ 0.9 & - & \times & \times \\ 0.7 & 0.8 & - & \times \\ 0.4 & 0.5 & 0.6 & - \end{bmatrix} \end{array} \begin{matrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{matrix}$$

Fig. 6: A similarity matrix indicating the structural similarity across fitness landscapes of different problem instances.

[1] M. Huang and K. Li, "Exploring Structural Similarity in Fitness Landscapes via Graph Data Mining: A Case Study on Number Partitioning Problems", IJCAI'23, 2023
[2] D, Gentner, "Structure-mapping: A theoretical framework for analogy ", Cognitive Science, 1983

Introduction & Motivation

As the first step towards this direction, we take the number partitioning problem (NPP) as a case study.

In particular, we investigate NPP instances at different dimensions (i.e., different number of items to partition.

> **RQ1:** For NPP, can we investigate any potential structural similarity across landscapes of different dimensions through the mining of statistical features?

> **RQ2:** For NPP, can we investigate any potential structural similarity across landscapes of different dimensions through landscape visualizations?

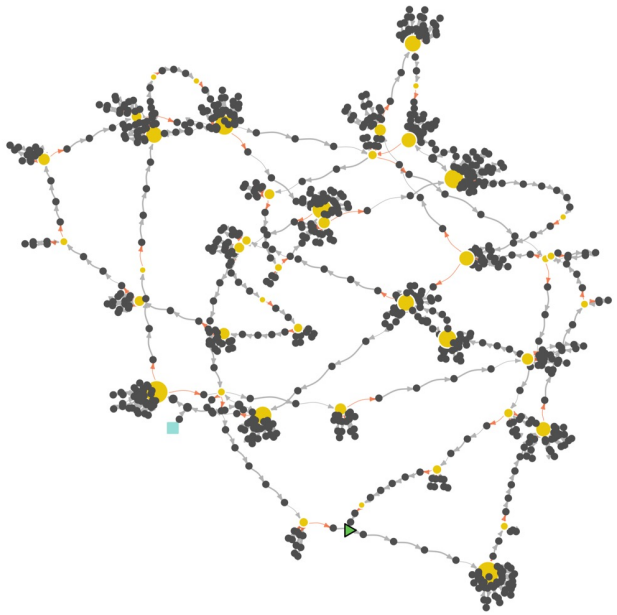
> **RQ3:** For NPP, if such structural similarity exists, can we quantitatively investigate it across landscapes of different dimensions?

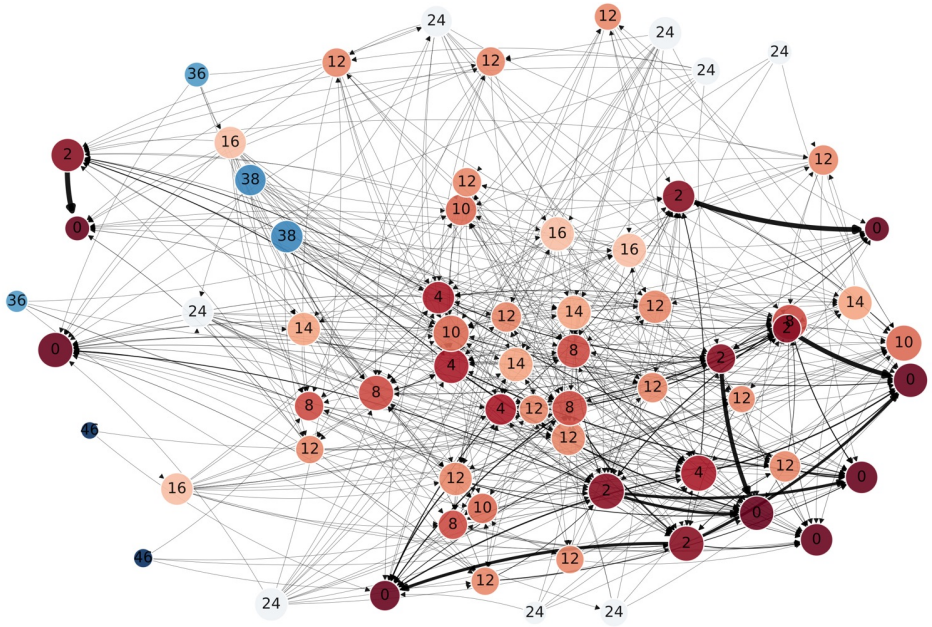
> **RQ4:** For NPP, if such quantitative determination of structural similarity is possible, how effective is it in explaining the performance difference of meta-heuristics between different problem instances?

Tools and Methods

Local Optima Network

- Nodes: local optima in the fitness landscape.
- Edges: transitions between local optima.
- Advantages:
  - Complex network metrics, e.g., density, clustering coefficient, centrality, assortativity, etc.
  - Network visualization methods.
  - Network representation learning methods.





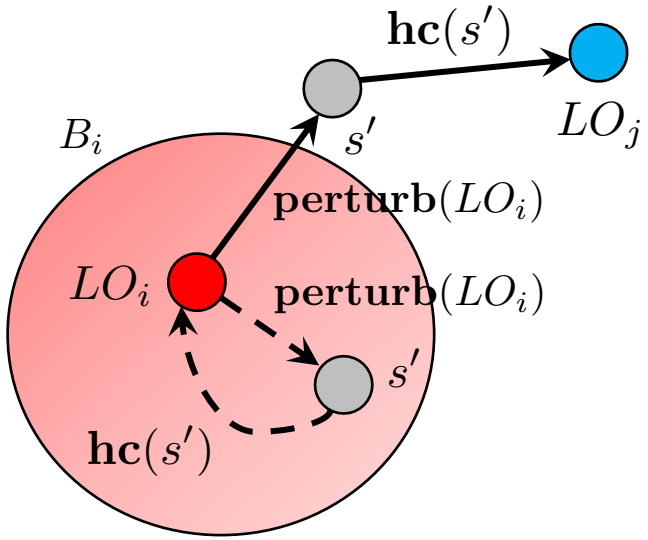


Fig. 7: Examples of LONs.

Analysis Framework

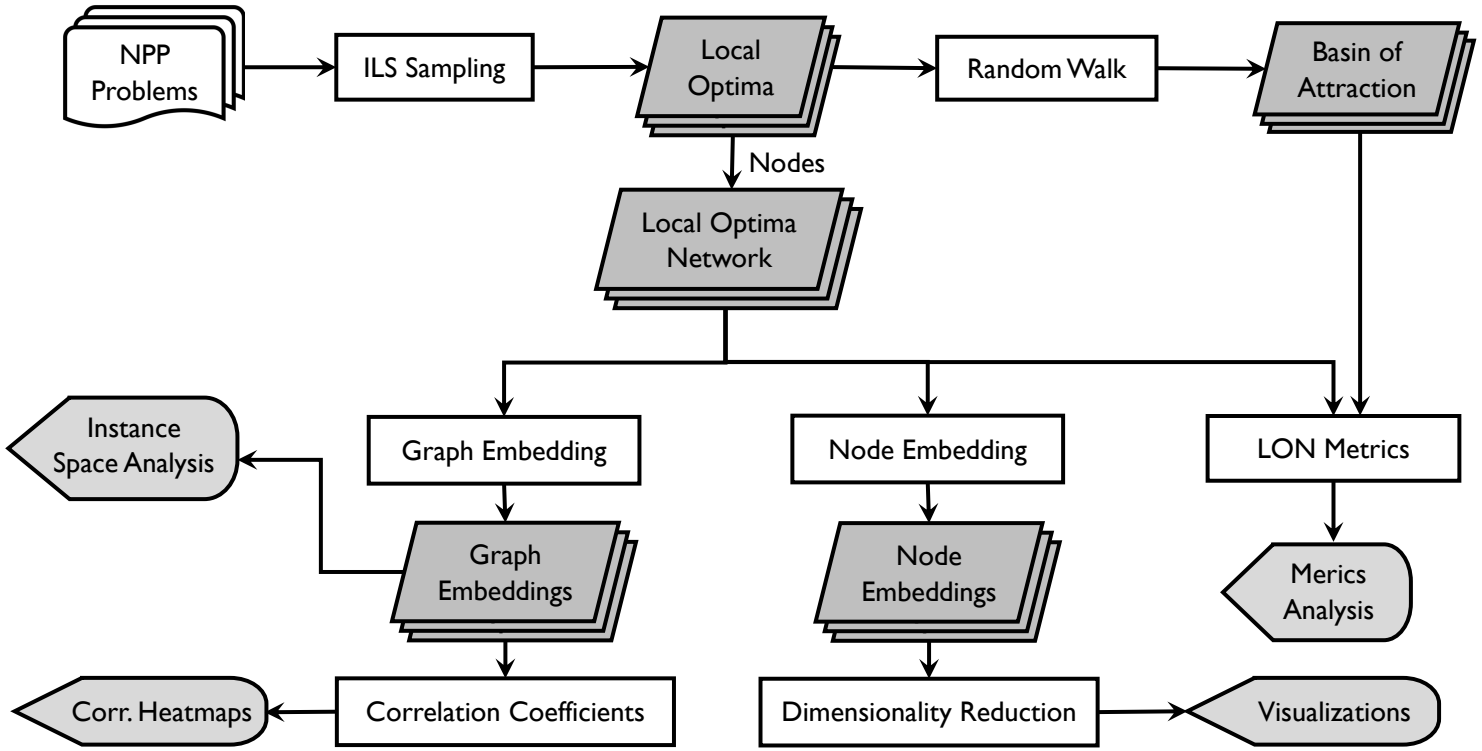


Fig. 8: A high-level overview of the analysis pipeline.

1. Network Metrics

TABLE II
HIGH-LEVEL FEATURES

Symbol	Description
B_{size}†	Average size of basin of attraction.
N_{climb}†	Average number of hill-climb steps taken to reach a *lo*.
N_{pert}†	Average number of perturbations taken to find an improving move from a *lo*.
N_{funnel}†	Number of funnels in the landscape.
N_{node}∗	Number of nodes in LON.
N_{edge}∗	Number of edges in LON.
$dens$∗	Network density.
cc_{avg}∗	Average clustering coefficient.
ast_{deg}∗	Degree assortativity coefficent.
ast_{fit}∗	Fitness assortativity coefficent.
L_{avg}∗	Mean average path lengths to global optima.
L_{min}∗	Mean minimum path lengths to global optima.
CDD⋆	Cummulative degree distribution trajectory of a LON
RCC⋆	Rich club coefficent trajectory of a LON

† denotes landscape features.
∗ denotes network metrics calculated for each LON.
⋆ denotes trajectories calculated for each LON

TABLE I
LOW-LEVEL FEATURES

Symbol	Description
b_{size}†	Size of basin of attraction of a *lo*, approximated using the sampling strategy proposed in [51].
n_{climb}†	Number of hill-climb steps taken to reach a *lo*, recorded in ILS.
n_{pert}†	Number of perturbations taken to find an improving move from a *lo*, recorded during ILS.
$freq$†	Frequency of visits to a *lo* during ILS.
deg∗	Degree of a node, i.e., number of neighbours of a node.
deg_{in}∗	Incoming degree, i.e., number of *lo*s that could transit to the target one via perturbation followed by hill-climbing.
deg_{out}∗	Outgoing degree, i.e., number of *lo*s that could be reached from a source *lo* via perturbation followed by hill-climbing.
c_{betw}∗	*Betweeness centrality*, which is a measure of how often a node lies on the shortest path between other nodes in a network. Nodes with high betweenness centrality can have a large impact on the flow of information through the network.
c_{ev}∗	*Eigenvector centrality*, this is a measure of a node's importance in a network based on the importance of the nodes it is connected to. Nodes with high eigenvector centrality are connected to other nodes that are also important in the network.
c_{close}∗	*Closeness centrality*, a measure of a node's importance in a network based on the number and quality of incoming links to the node. Originally developed for ranking web pages in search engines, PageRank centrality can be applied to other types of networks as well.
c_{pg}∗	*PageRank centrality*, this is a measure of a node's importance in a network based on the number and quality of incoming links to the node. Originally developed for ranking web pages in search engines, PageRank centrality can be applied to other types of networks as well.
cc∗	*Clustering coefficient*. It quantifies the degree to which nodes in a network tend to cluster together.
avg_{deg}∗	Average degree of the neighbours of a *lo*.
avg_{fit}∗	Average fitness of the neighbours of a *lo*.
l_{min}∗	Mininum length to accessible global optima.
l_{avg}∗	Average length to accessible global optima.

† denotes landscape features associated with each *lo*.
∗ denotes network metrics calculated for each node.

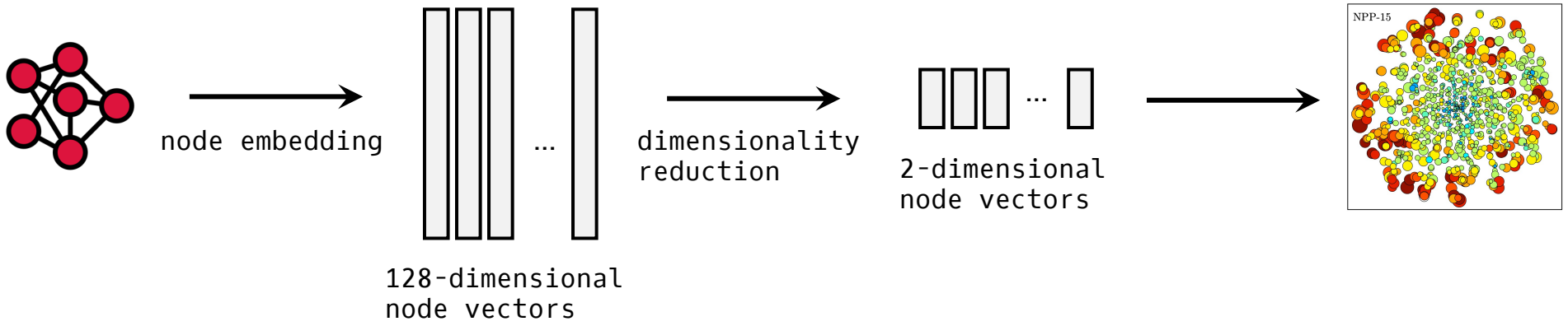
Tools and Methods

2. Graph Visualization Methods

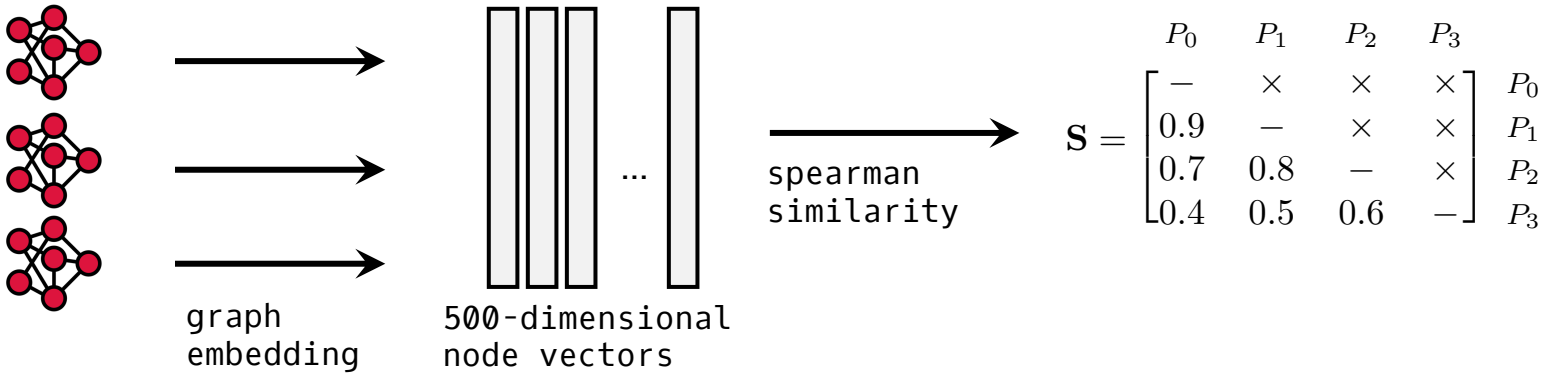
We leverage node embedding and dimensionality reduction technique to generate low-dimensional visualizations of the LONs.

This could preserve essential topological structures of the underlying landscape.

In practice, we believe it is superior than other existing landscape visualization attempts.



3. LON Similarity Calculation



Results and Analysis

1. Structural Similarity Study via Statistical Analysis

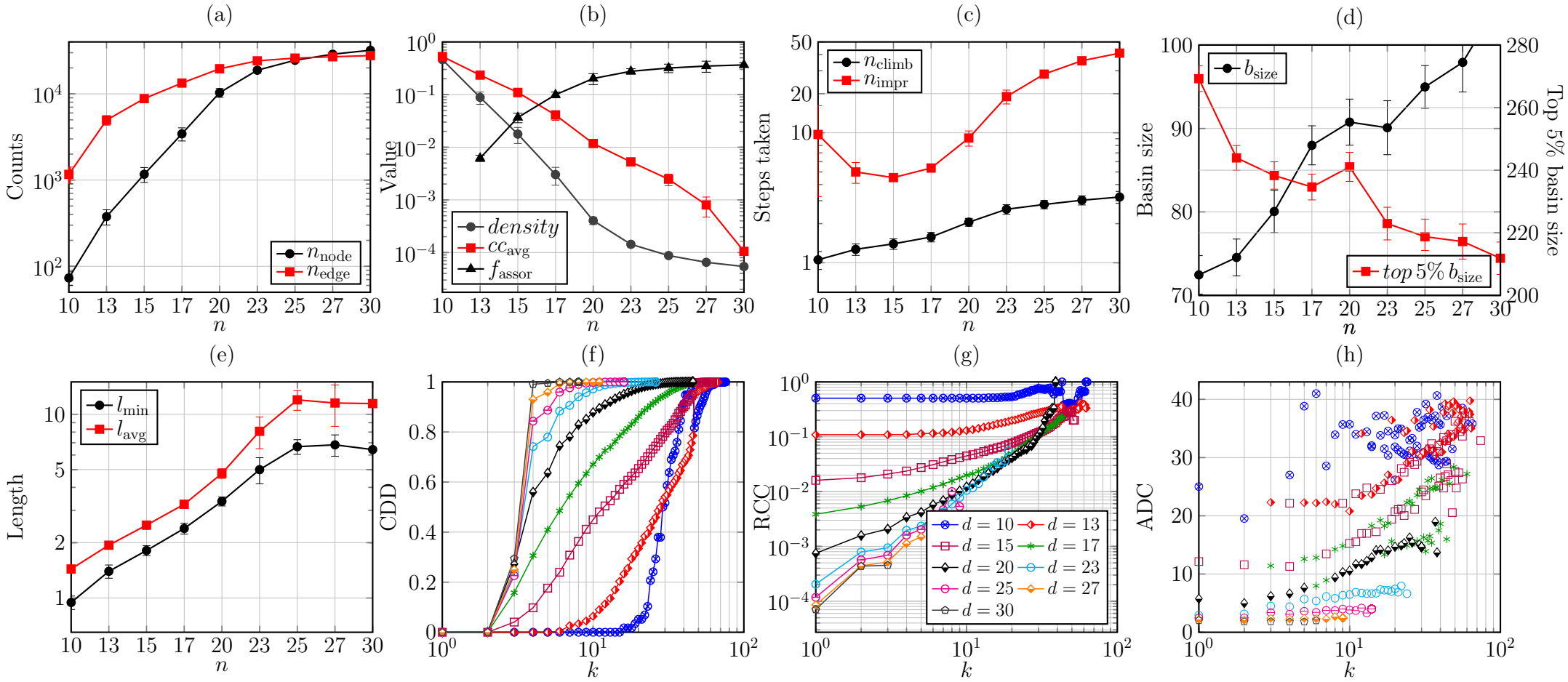
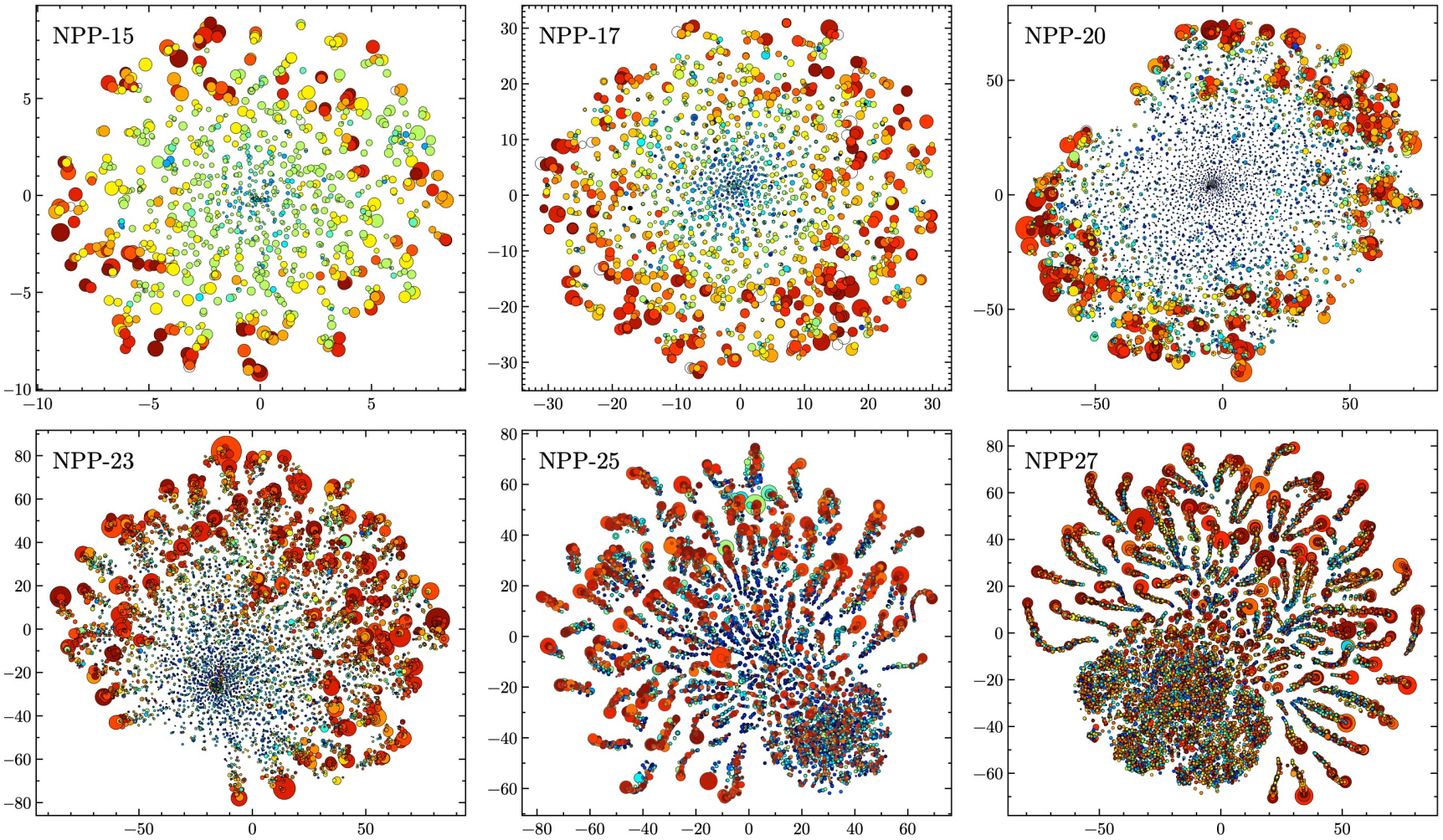


Fig: Trajectories of statistical features of LONs / landscapes with the increase of dimensionality.

Results and Analysis

2. Structural Similarity Study via Visual Analysis



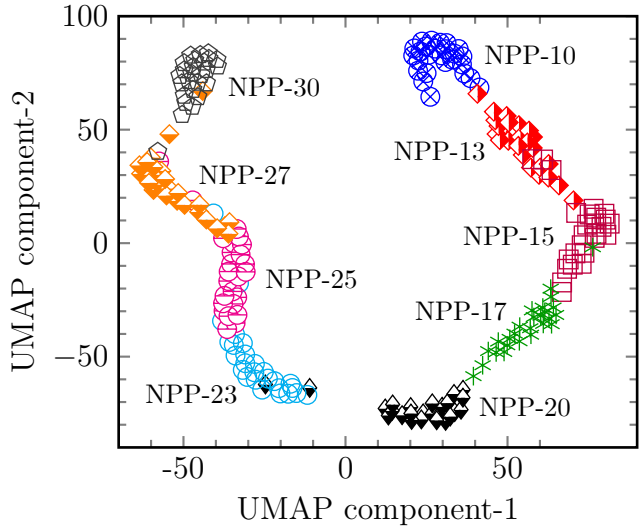


Fig: 2D Instance space analysis of the studied problem instances.

Results and Analysis

3. Structural Similarity Study via Quantatitive Analysis

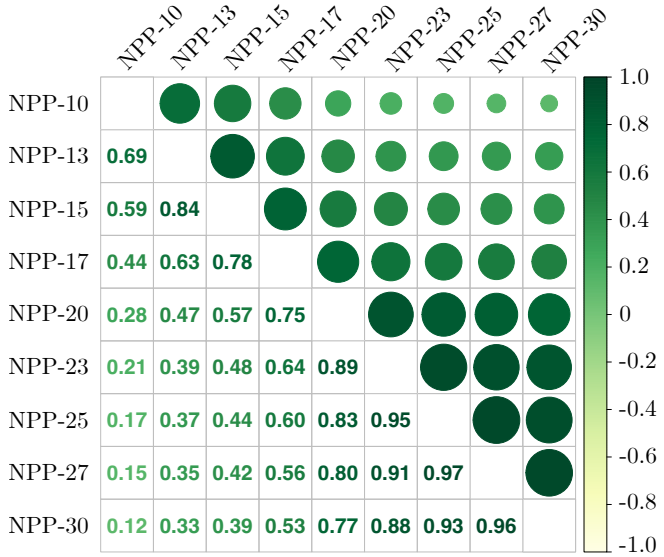
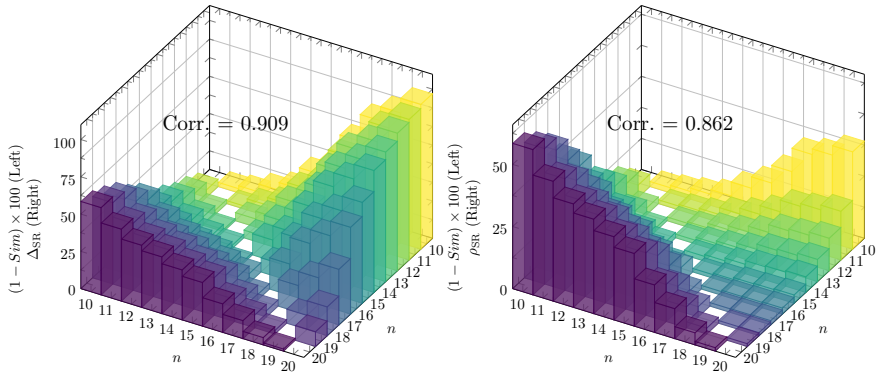


Fig: Similarity matrix of the studied problem instances. Each point is the average across 30 random instances.

4. Verification of the Effectiveness of the Measured Similarity



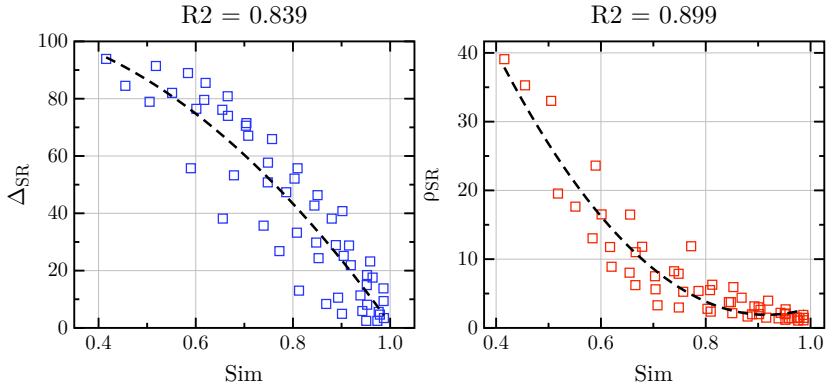


Fig: The scatter plot along with quadratic regression line between the calculated pairwise similarity and measured performance difference of SA.

Part III

Current Work

A More General Study Study with Comprehensive Framework

Introduction

Based on our previous work with NPP of different dimensions as the case study, we now proceed to conduct the investigation on a broader range of probems.

- ❑ Traveling Salesman Problem (TSP)
- ❑ Maximum-Satisfiability Problem (Max-Sat)
- ❑ Knapsack Problem (KP)
- ❑ NPP

Research Questions

> **RQ1:** For each class of problem, can we investigate any protential structural similarity in fitness landscapes across instances of different dimensions?

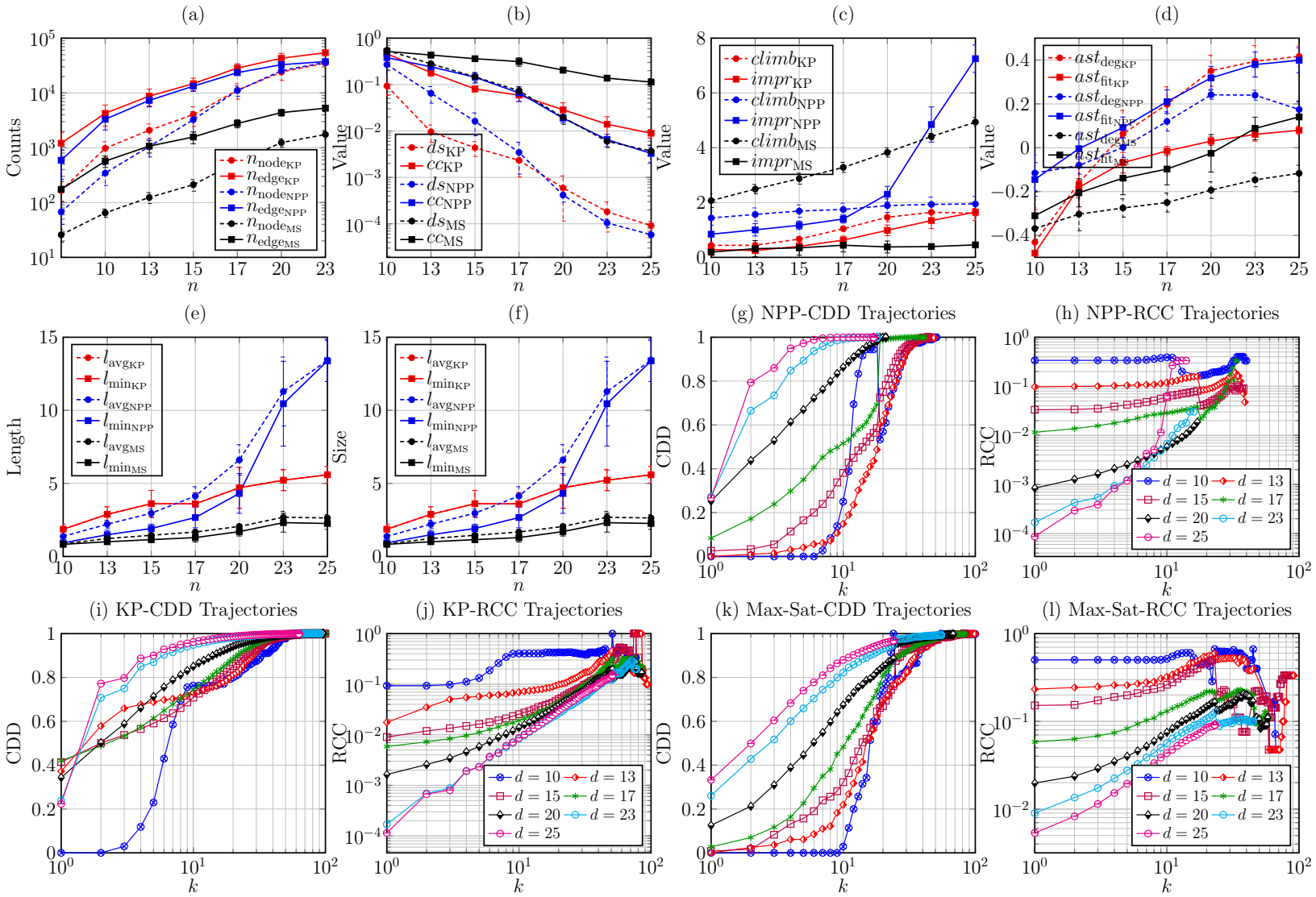
> **RQ2:** For each class of problem, can we investigate any protential structural similarity in fitness landscapes across instances of different sub-classes?

> **RQ3:** Can we investigate any protential structural similarity in fitness landscapes across instances that belong to different classes of problems?

> **RQ4:** How effective is the measured similarity in explaining the performance of meta-heuristics between different problem instances?

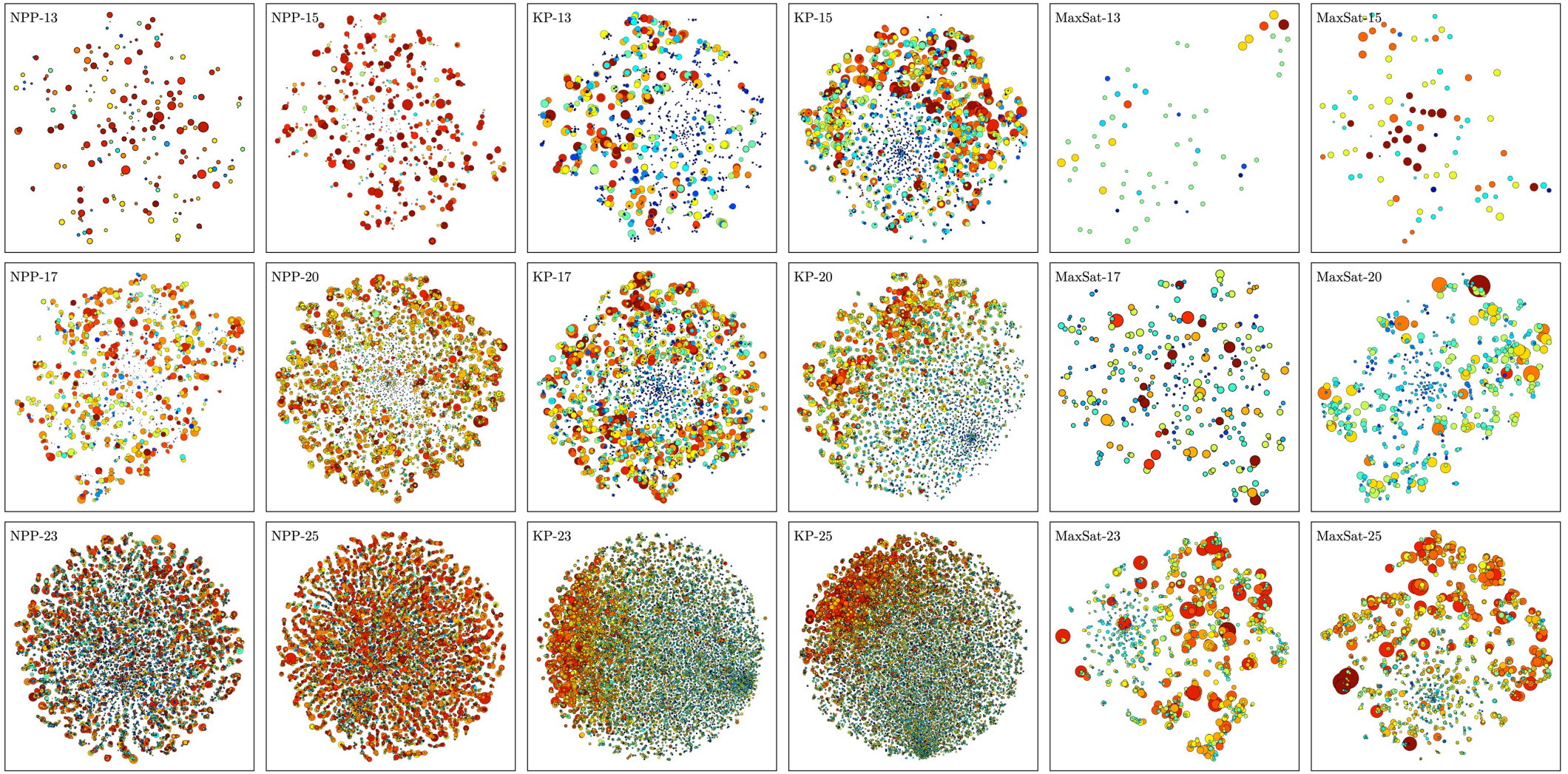
Proposed Analysis Methods

1. Structural Similarity Study via Statistical Analysis



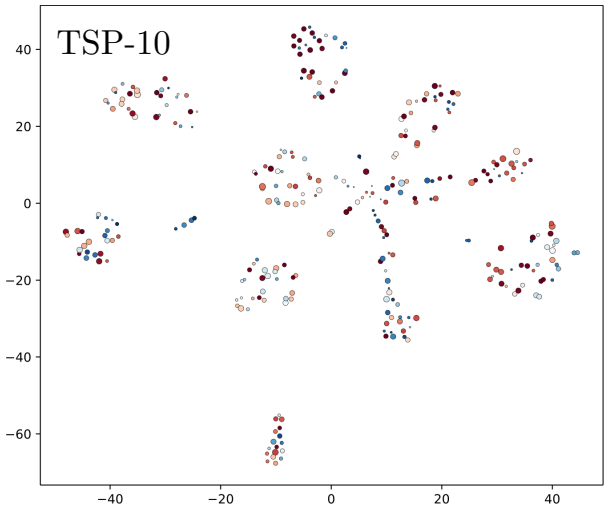
Proposed Analysis Methods

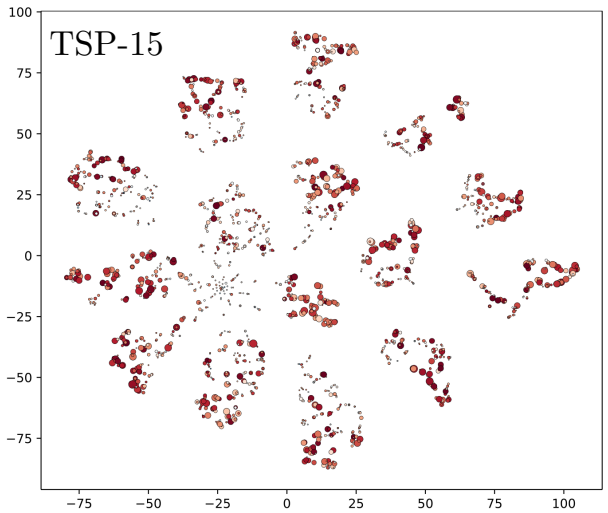
1. Structural Similarity Study via Visual Analysis

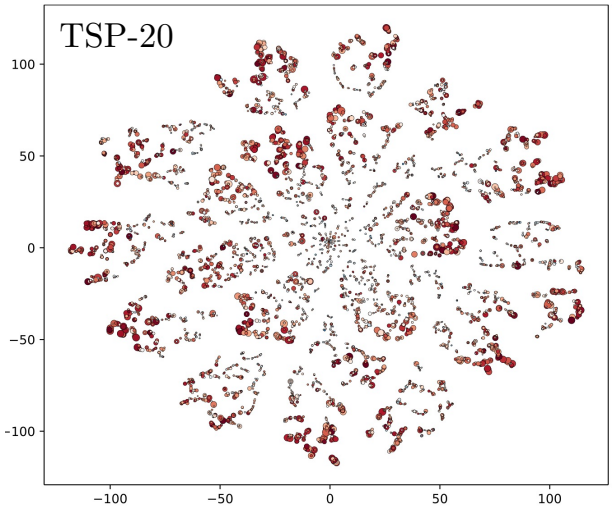


Results and Analysis

2. Structural Similarity Study via Visual Analysis







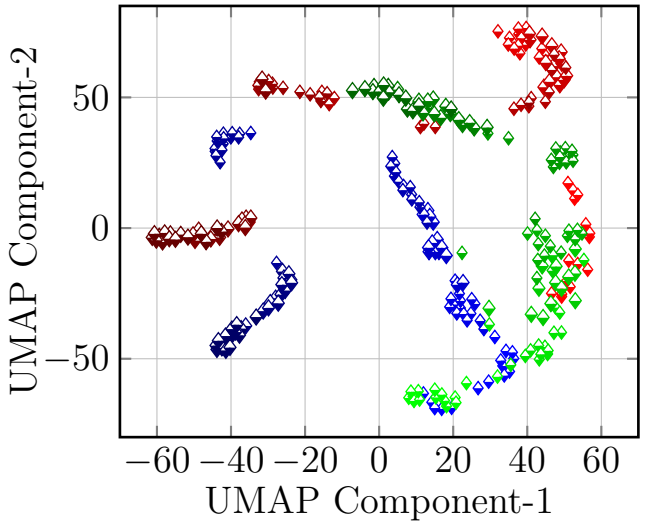
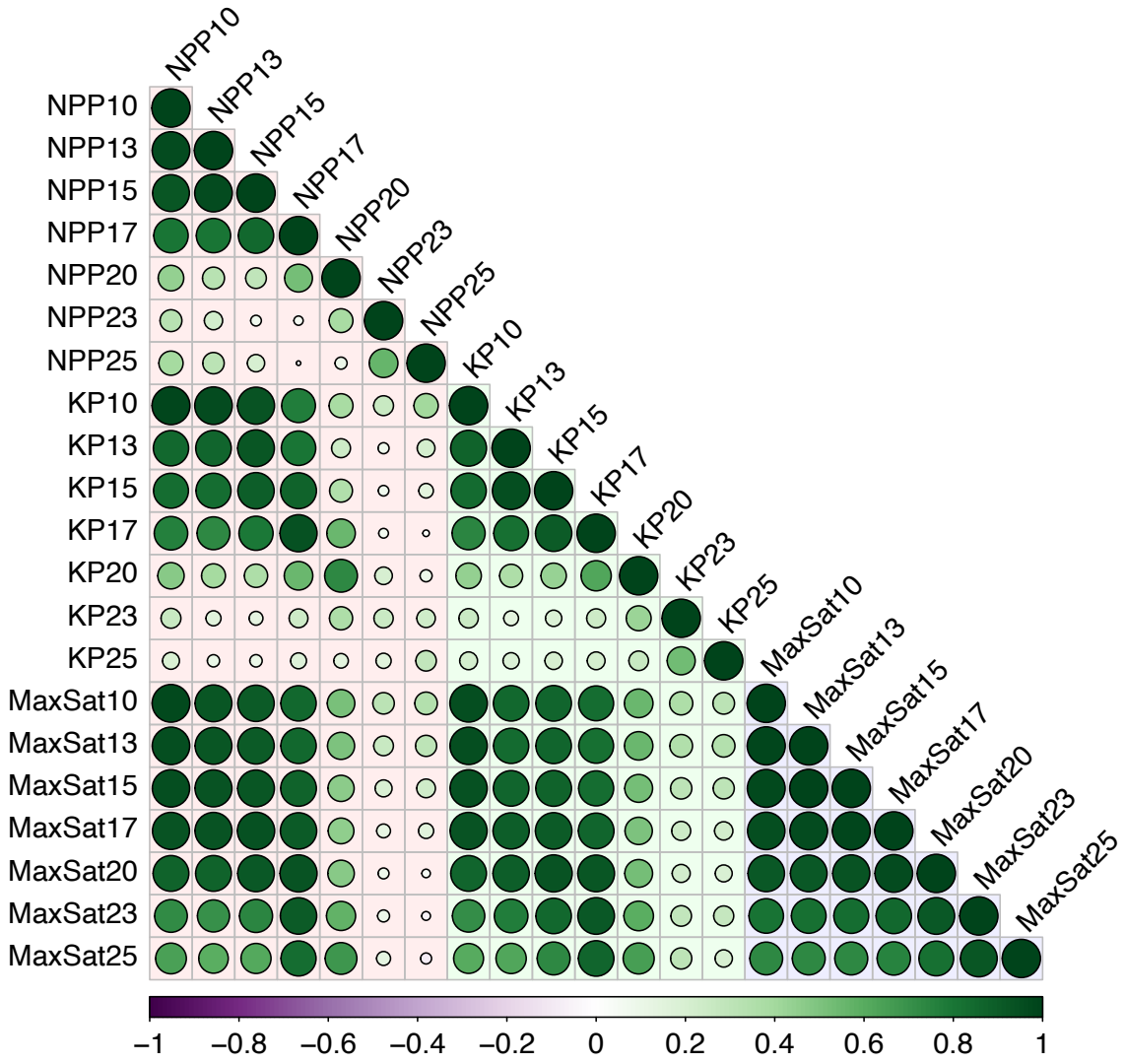


Fig: 2D Instance space analysis of the studied problem instances.
blue: NPP
red: KP
green: Max-Sat

Results and Analysis

3 & 4. Quantitative Analysis and Verification



		NPP			KP			Max-Sat		
		Combined	Embedd	Descri	Combined	Embedd	Descri	Combined	Embedd	Descri
Interactive	Ratio	0.097	0.097	0.071	0.135	0.128	0.119	0.546	**0.554**	0.387
	Diff	**0.910**	0.900	0.767	0.831	0.826	0.600	0.148	0.133	0.324
Class A	Ratio	0.804	0.779	0.925	0.816	0.792	**0.939**	0.900	0.900	0.921
	Diff	**0.913**	0.890	0.760	0.817	0.819	0.792	0.830	0.830	0.839
Class B	Ratio	0.890	0.874	0.929	0.527	0.514	0.886	0.943	**0.944**	0.819
	Diff	**0.909**	0.895	0.812	0.804	0.795	0.635	0.734	0.730	0.813
Class C	Ratio	0.857	0.838	0.943	0.788	0.760	0.935	0.944	**0.947**	0.899
	Diff	**0.883**	0.861	0.814	0.832	0.814	0.823	0.517	0.513	0.587

Fig: The spearman correlation between calculated similarity and measured performan (in ERT) across different sub-classes of problems

		Combined	Embedd	Descri
Interactive	Ratio	0.229	**0.234**	0.003
	Diff	**0.774**	0.762	0.646
NPP	Ratio	0.803	0.773	**0.919**
	Diff	**0.904**	0.873	0.804
KP	Ratio	0.826	0.801	**0.922**
	Diff	**0.831**	0.819	0.792
MaxSat	Ratio	0.817	0.788	**0.904**
	Diff	**0.925**	0.901	0.734

Fig: The spearman correlation between calculated similarity and measured performan (in ERT) across different classes of problems

Part IV

Applications

AutoML and Software Engineering

AutoML Landscapes

Fitness Landscapes of XGBoost

- Fitness landscapes on training data and test data.
- Fitness landscapes on different datasets.

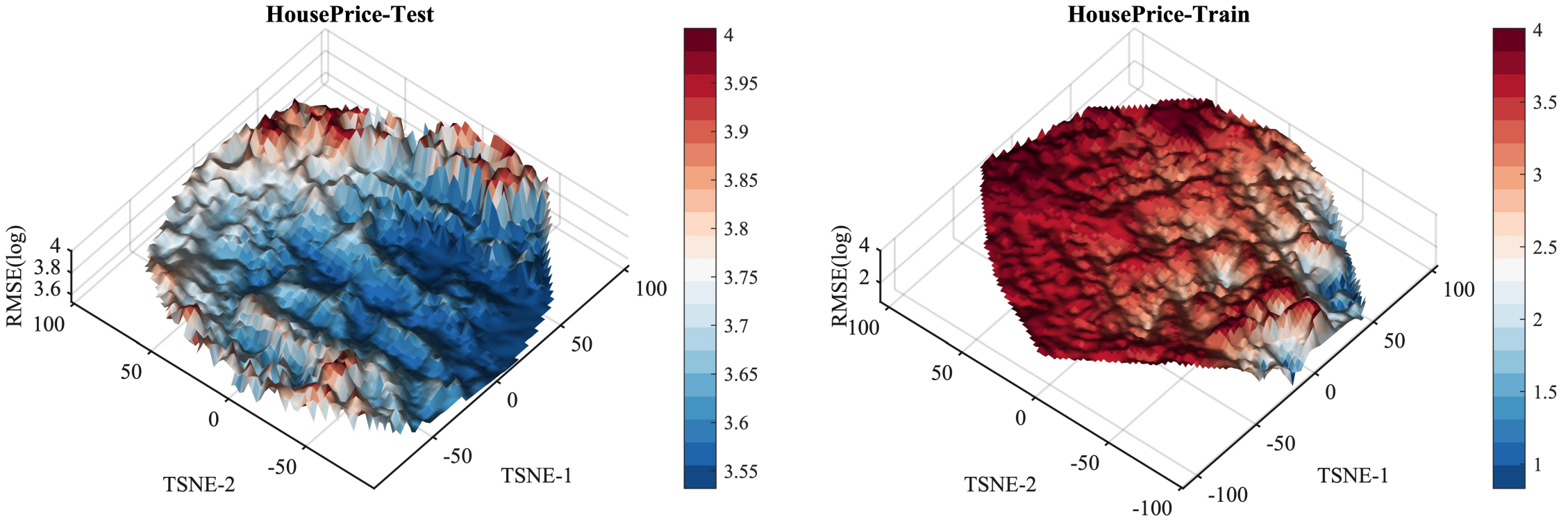


Fig: A comparison of the HPO landscape of XGBoost on the training data and the test data of my house price dataset. This dataset contains 73 features and more than 1 million house profiles. 50,000 random samples are used to conduct the experiment. A total of 19,250 configurations were measured.

AutoML Landscapes

Fitness Landscapes of XGBoost

- Fitness landscapes on training data and test data.
- Fitness landscapes on different datasets.

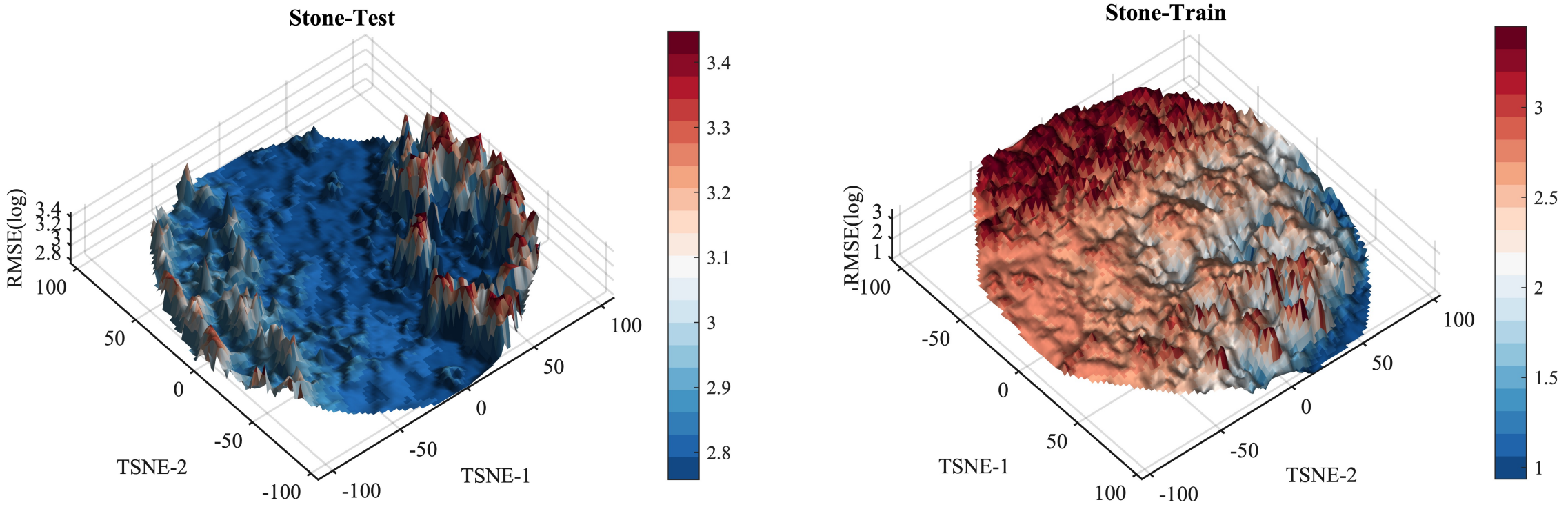


Fig: A comparison of the HPO landscape of XGBoost on the training data and the test data of Kaggle gemstone dataset. This dataset contains 11 features and about 200,000 stone profiles. 50,000 random samples are used to conduct the experiment. A total of 19,250 configurations were measured.

Part V

Future Work

Software Package, Survey and Community Development

GBFLAT Software Package

1. Main Workflow

Problem Instances → ILS Sampling → LON Construction → Analysis, Manipulation and Visualization

2. Core Modules

Problems

- NumberPartitioning
- TravelingSalesman
- MaximumSatisfiability
- Knapsack
- Solution

ILS

- IteratedLocalSearch
- hill_climb_local_searcher
- filp_neighbor_explorer
- two_opt_neighbor_explorer
- two_bit_flip_perturber
- double_bridge_perturber

LON

- LocalOptimaNetwork
  - read_ils
  - describe
  - draw
  - draw_embedding
  - save_lon

Enumeration

Algorithms

GBFLAT Software Package

3. Core Features

- **Interactive Programming:** The core modules of GBFLAT are specially designed to be used in Jupyer Notebook enviroment to allow for interactive data analysis.
- **Easy Parallelization:** ILS search and LON construction process natively support CPU parallelization.
- **Flexible Data Format:** The main LON object is available in both pandas dataframe and NetworkX graph form, making it extremely suitable for conducting data mining, manipulation and visualization.
- **Highly Modularized:** The main working pipeline are consisted of various sub-components which are highly customizable, allowing

```python
# create a problem instance
instance = NumberPartitioning(n = 10, k = 0.7, seed = 1)

# create an ils_searcher object
ils_searcher = IteratedLocalSearch(n_runs = 1000,
                                   max_iters = 100,
                                   local_searcher = hill_climb_local_searcher,
                                   neighbour_explorer = flip_neighbour_explorer,
                                   perturbator = two_bit_flip_pertubator)

# perform ILS sampling using multiple CPU threads
ils_searcher.search(instance = instance,
                    path = "ils_data/",
                    n_jobs = 8)

# create LON object using ILS data
lon = LocaloptimaNetwork()
lon.read_ils(problem_name = "NPP",
             n = 10,
             k = 0.7,
             seed = 1,
             nb_runs = 1000,
             nb_iters = 100,
             weighted = True,
             directed = True)

# get LON data
graph = LON.graph
data = LON.data

# data manipulation example
print(data.sort_values(by = "fitness"))
# graph manipulation example
sol = [1, 0, 0, 1, 1, 1, 0, 0, 1, 1]
print(nx.neighbors(graph, sol))

# draw LON as graph
lon.draw_lon()
# draw LON in 2D via node embedding and dimensionality reduction
lon.draw_embedding(model = HOPE(), reducer = TSNE())
# get statistical description of LON
lon.describe()

# save LON
lon.save_lon(name = "npp_n10_k0.7_seed1")
```

Survey Paper

Fig. X: A wordclouds plot of the author names appeared the collected literature.

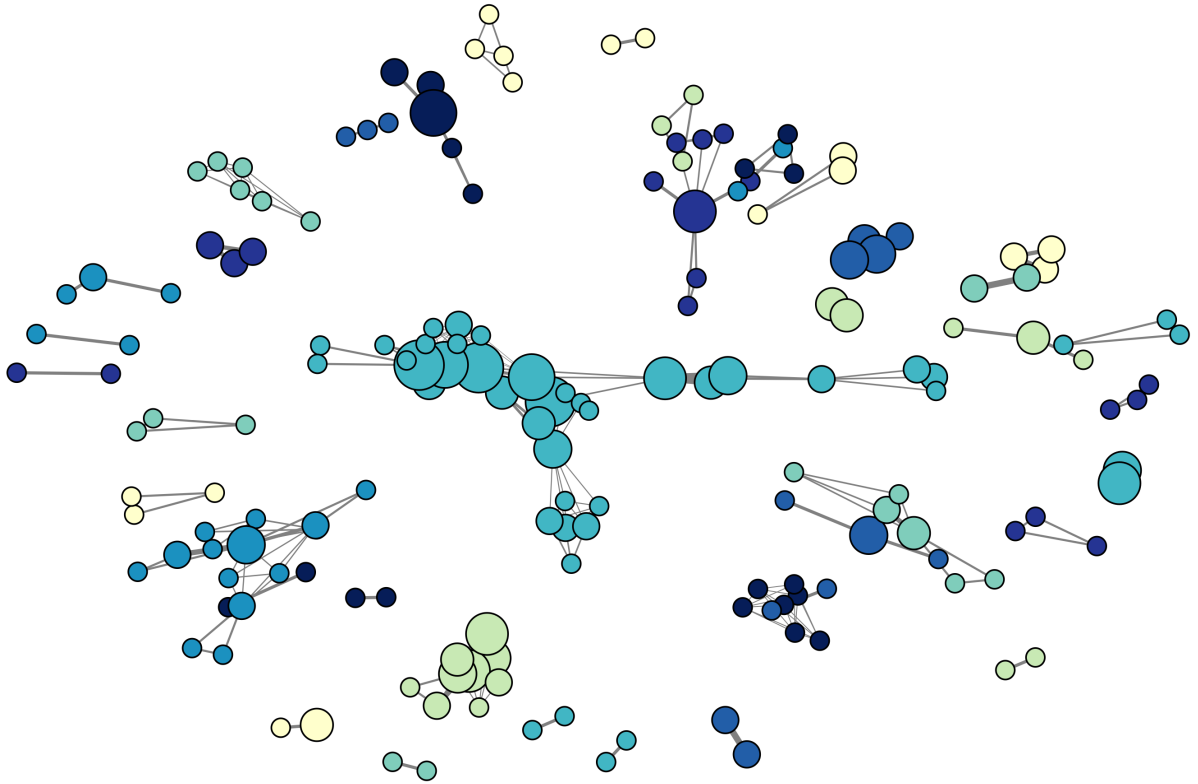


Fig. X: The collaboration network of the authors. Bigger nodes indicate higher publication frequency in our survey, and colors denotes different connected components. Edge width indicates the strength of collaboration.

COLALab
Computational Optimization for Learning & Adaptive Systems

Landscape Analysis Community @COLA

Website Development

- Our Work
  - Publications
  - Analysis Methods
  - Case Studies
    - Combinatorial Optimization
    - AutoML
    - Software Engineering
- Resources
  - Paper List
  - Projects and Websites
- GBFLAT
  - Framework
  - Installation
  - Get Started
  - Documentation