Exploring Structural Similarity in Fitness Landscapes via Graph Data Mining: A Case Study on Number Partitioning Problems (Appendix)

A Iterated Local Search

ILS is a well-known heuristic for combinatorial optimization problems, and has been served as a part of the standard pipeline for developing LONs from combinatorial landscapes. The pseudo-code describing the ILS sampling process used in this paper is shown in Algorithm 1. Generally, the algorithm starts by randomly initializing a n-dimensional binary vector $\mathbf{x}$ as the initial solution, and then local search is applied to $\mathbf{x}$ until a local optimum $\mathbf{x}^{\ell}$ is found. In this paper, a *first-improvement hill climbing* is used as the local search strategy, and the *neighborhood* $\mathcal{N}(\mathbf{x})$ of a solution $\mathbf{x}$ is defined over one-mutant neighbors, i.e, the set of all configurations that differ from $\mathbf{x}$ in exactly one position. Afterwards, a *perturbation*, which in our case, a two-bit-flip operation is applied on $\mathbf{x}^{\ell}$ to escape to a new solution $\mathbf{x}'$, and the algorithm then starts climbing from $\mathbf{x}'$ until a new local optima is reached. The perturbation process is repeated until termination condition is reached.

B Features and Metrics

In this section, we delineate the features we studied in Section 4.1 as well as correlation and performance metrics we used in this study.

B.1 Network metrics

Density Network density (denoted as $density$) is a measure of the proportion of the possible edges that actually exist in the network. Given a network with n nodes, the maximum number of edges is $n(n-1)/2$. If the actual number of edges is m, then the density of such a network could be calculated as:

$$density = \frac{m}{n(n-1)/2}, \tag{1}$$

Clustering coefficient It is a metric that measures the proportion of paths of $l = 2$ in the network that are closed. This value could provide a sense of the extent to which pairs of nodes with a common neighbor are also themselves neighbors. At local neighborhood level, the clustering coefficient for a single node could be defined as:

$$c_u = \frac{2T(u)}{deg(u)(deg(u)-1)} \tag{2}$$

where $T(u)$ is the number of triangles through node u and $deg(u)$ is the degree of u. The clustering coefficient for the whole graph could be obtained by taking averaging through all nodes:

$$C = \frac{1}{n}\sum_{v\in\mathcal{V}} c_v \tag{3}$$

Assortativity coefficient Given a directed graph, the assortativity evaluates the Pearson correlation coefficient of degree between pairs of linked nodes and it is calculated as:

$$AC = \frac{\sum_{jk} jk(e_{jk} - q_j^{in} q_k^{out})}{\sigma_{in}\sigma_{out}} \tag{4}$$

where e_{jk} refers to the fraction of edges that connect vertices of degree j and k and q_k is the distribution of the remaining degree and is calculated as:

$$q_k = \frac{(k+1)p_{k+1}}{\sum_k kp_k} \tag{5}$$

where p_k refers to the degree distribution, i.e., the probability that a randomly chosen vertex will have degree k. A positive AC value indicates a correlation between nodes of similar degree, while a negative value indicates the opposite side, i.e., a correlation between nodes of different degree.

Cumulative degree distribution (CDD) It measures the fraction of vertices with degree smaller than k ($k \geqslant 1$) and it is calculated as:

$$CDD = P(X \leqslant k), \tag{6}$$

where $P(k) = n_k/n$ is the degree distribution or the proportion of the vertices with degree k, n denotes the total number of vertices in a graph and n_k is the number of vertices with degree k.

Rich club coefficient (RCC) RCC measures the extent to which well-connected vertices also connect to each other. Networks which have a relatively high RCC value are said to demonstrate the rich-club effect and will have many connections between vertices of high degree. Mathematically, the RCC metric is calculated as:

$$RCC = \frac{2E_{>k}}{N_{>k}(N_{>k}-1)}, \tag{7}$$

where $N_{>k}$ is the number of vertices with degree larger than or equal to k, and $E_{>k}$ is the number of edges among those vertices.

Algorithm 1: ILS for sampling local optima

```
Input: Search space $\mathcal{X}$, fitness function $f$
Output: $\mathcal{V}$, $\mathcal{E}$
$\mathcal{V} \leftarrow \emptyset, \mathcal{E} \leftarrow \emptyset$;
Generate an initial candidate $\mathbf{x} \in \mathcal{X}$ by random sampling;
$\mathbf{x}^{\ell} \leftarrow$ localSearch($\mathbf{x}$);
$\mathcal{V} \leftarrow \mathcal{V} \cup \{\mathbf{x}^{\ell}\}$;
$i \leftarrow 0$;
while $i \leqslant K$ do
    $\mathbf{x}' \leftarrow$ perurbation($\mathbf{x}^{\ell}$);
    $\mathbf{x}^{\ell'} \leftarrow$ localSearch($\mathbf{x}'$);
    if $f(\mathbf{x}^{\ell'}) \leqslant f(\mathbf{x}^{\ell})$ then
        $f(\mathbf{x}^{\ell}) \leftarrow f(\mathbf{x}^{\ell'})$;
        $\mathcal{V} \leftarrow \mathcal{V} \cup \{\mathbf{x}^{\ell}\}$;
        Construct an edge between $\langle \mathbf{x}^{\ell}, \mathbf{x}^{\ell'} \rangle$ and add it to $\mathcal{E}$;
        $i \leftarrow 0$;
    $i \leftarrow i + 1$;
return $\mathcal{V}$, $\mathcal{E}$
```

Average Degree Connectivity (ADC). ADC is the average nearest neighbor degree of nodes with degree k, which could be calculated by:

$$ADC = \frac{1}{|\mathbf{N}(i)|} \sum_{j \in \mathbf{N}(i)} k_j, \tag{8}$$

where $\mathbf{N}(i)$ is the the neighborhood of node i, and k_j is the degree of node j

Centrality Centrality measures the importance of a node in the network, and a lot of metrics have been proposed towards this end.

1. *Degree Centrality*: Degree centrality is the simplest method of measuring node importance, which simply adopts node degree as importance indicators.
2. *Eigenvector Centrality*: As an extended version of degree centrality, eigenvector centrality considers the connections to more influential nodes contribute more to the centrality score than connections to less important nodes. Formally, eigenvector centrality of a node could be calculated as:

$$x_i = \lambda^{-1} \sum_{x_j \in \mathcal{V}} x_j \tag{9}$$

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x} \tag{10}$$

   where λ is a constant, $\mathbf{A}$ is the adjacency matrix, $\mathbf{x}$ is the vector with elements equal to the centrality scores x_i.

3. *PageRank centrality*: PageRank centrality is proposed by Google founders Larry Page and Sergei Brin, which is a variant of eigenvector centrality and was developed for ranking web pages. In PageRank centrality, nodes connected to an influential node only shares part of its scores, and thus prevents the situation that any number of nodes connected to an important node is also assigned with high scores. It is defined as:

$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j^{out}} + \beta \tag{11}$$

   where α is a free parameter, and β is usually set to 1, k_j^{out} is the out degree of node j.

4. *Betweenness centrality*: Betweenness centrality measures the number of shortest paths a node lies on and hence reflect the importance of the node from a information flow perspective. It could be expressed by:

$$x_u = \sum_{i,j \in \mathcal{V}} n_{ij}^{u} \tag{12}$$

   where $n_{ij}^{u} = 1$ if node u lies on the shortest path from node i to j.

5. *Closeness centrality*: Closeness centrality is also based on shortest paths. Suppose d_{ij} is the shortest distance from node i to node j, then the mean shortest distance from i to every node in the network is:

$$\ell^i = \frac{1}{n} \sum_j d_{ij} \tag{13}$$

   And the corresponding closeness centrality x_i is defined as the inverse of the mean shortest distance:

$$x_i = \frac{1}{\ell^i} = \frac{n}{\sum d_{ij}} \tag{14}$$

B.2 Local optima features

Steps taken to reach The number of hill climb steps taken to reach each local optimum, which is able to reflect the efforts required upon reaching it. This could be recorded during ILS search.

Perturbation taken to improve The number of perturbations taken to find a superior local optimum. More specifically, for each local optima $\mathbf{x}^{\ell}$, a new solution $\mathbf{x}^{'}$ could be generated via a two-flip perturbation. Such a move is could an improving move if $f(\mathbf{LocalSearch}(\mathbf{x}^{'})) < f(\mathbf{x}^{\ell})$. This could be recorded during ILS search.

Length to Global Optimum We consider both the average and minimum length from a local optimum to the accessible global optimum(optima). Though conventionally, this could be defined on Hamming distance of the corresponding solutions, this work, instead, determines this length using LON.

Basin size The size of the basin of attraction $\mathcal{B}$ of a local optimum is defined as the cardinality of the basin set as $|\mathcal{B}|$, while $\mathcal{B}$ in is sampled using a specific sampling strategy. For each local optimum x_0, we conduct a random walk starting from it, where at each step i, a one-bit-flip random mutation is applied on previous solution x_{i-1} and hence results in a new solution x_i. x_i is in the basin of local optimum x_0 if it could converge to x_0 after a hill climbing process. The random walk continues until a solution x_j which is not in the basin of x_0 (i.e., x_j converges to a local optimum different from x_0 after hill climbing) is found. This process is repeated 100 times for each local optimum and we collect all results to constitute the basin set of each local optimum.

B.3 Correlation metrics

Pearson Correlation It is used to measure the statistical association between two continuous variables. Its value ranges from -1 to 1. In particular, a larger PCC value indicates a stronger correlation. Given two random variables $\mathbf{x}^1$ and $\mathbf{x}^2$, their PCC is calculated as:

$$PCC(\mathbf{x}^1, \mathbf{x}^2) = \frac{\mathtt{cov}(\mathbf{x}^1, \mathbf{x}^2)}{\sigma(\mathbf{x}^1)\sigma(\mathbf{x}^2)} \tag{15}$$

where $\mathtt{cov}(\cdot,\cdot)$ evaluates the covariance and $\sigma(\cdot)$ represents the standard deviation.

Spearman Correlation It is a non-parametric measure of rank correlation which assesses how well the relationship between two variables can be described using a monotonic function. The Spearman correlation coefficient is defined as the Pearson correlation coefficient between the rank variables.

$$r_s = PCC(\mathrm{R}(\mathbf{x}^1), \mathrm{R}(\mathbf{x}^1)) = \frac{\mathtt{cov}(\mathrm{R}(\mathbf{x}^1), \mathrm{R}(\mathbf{x}^2))}{\sigma[\mathrm{R}(\mathbf{x}^1)]\sigma[\mathrm{R}(\mathbf{x}^2)]} \tag{16}$$

where $\mathrm{R}(\mathbf{x}^1)$ and $\mathrm{R}(\mathbf{x}^2)$ are rank of $\mathbf{x}^1$ and $\mathbf{x}^2$ respectively.

Kendall Correlation This is a measure of rank correlation, i.e., the similarity of the orderings of the data when ranked by each of the quantities. A pair of observations (x_i, y_i) and (x_j, y_j) (where $i < j$) are said to be concordant if both $x_i > x_j$ and $y_i > y_j$ are satisfied, or, if both $x_i < x_j$ and $y_i < y_j$ are satisfied. In other cases, these two observations are said to be discordant. The Kendall τ correlation between random variables X and Y, where values x_i,y_i in the set $(x_1, y_1), (x_2, y_2) ... (x_n, y_n)$ are unique, is defined as:

$$\tau = \frac{N_{concor} - N_{discor}}{\frac{n(n-1)}{2}} \tag{17}$$

where N_{concor} is the number of concordant pairs, and N_{discor} is the number of discordant ones.

B.4 Performance metrics

$\mathbf{R^2}$ **score** It measures the proportion of the variance in the dependent variable that is predictable from the independent variable.

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(\hat{Y}_i - Y_i)^2}{\sum_{i=1}^{m}(\bar{Y}_i - Y_i)^2} \tag{18}$$

C Graph Representation Learning Techniques

C.1 Hope node embedding

HOPE could generate node features that is able to capture asymmetric high-order proximity in directed networks. For undirected networks, the transitivity is symmetric, but it is asymmetric in directed networks. In order to preserve the asymmetric transitivity, HOPE learns two vertex embedding vectors $U^s, U^t \in \mathbf{R}^{|V| \times d}$, which is called source and target embedding vectors, respectively. After constructing the high-order proximity matrix S from four proximity measures, i.e., Katz Index, Rooted PageRank, Common Neighbors and AdamicAdar. HOPE learns vertex embeddings by solving the following matrix factorization problem:

$$\min_{U_s, U_t} \|S - U^s U^{t^T}\|_F^2 \tag{19}$$

C.2 Feather graph embedding

Feather graph embedding adopts characteristic functions of node features with random walk weights to generate features for each node neighborhood. Assume an unweighted and undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, for each node $v \in \mathcal{V}$, we describe a node feature as a random variable X and specify feature vector $\mathbf{x}_v$, where $\mathbf{x}_v \in \mathbf{R}^{|\mathbf{V}|}$. Given the source node u and target node w, where $u, w \in \mathbf{V}$ and $\sum_{w \in \mathbf{V}} P(w|u) = 1$ and evaluation point $\theta \in \mathbf{R}$ we define real and imaginary part of the $r-scale$ random walk weighted characteristic function for node u as:

$$Re(E(e^{i\theta X}|G, u, r)) = \sum_{w \in \mathbf{V}} \hat{\mathbf{A}}^r_{u,w} cos(\theta \mathbf{x}_w) \tag{20}$$

$$Im(E(e^{i\theta X}|G, u, r)) = \sum_{w \in \mathbf{V}} \hat{\mathbf{A}}^r_{u,w} sin(\theta \mathbf{x}_w) \tag{21}$$

where $\hat{\mathbf{A}}^r_{u,w} = (\mathbf{D}^{-1}\mathbf{A})^r$. Then, the combination of $Re(\bullet)$ and $Im(\bullet)$ could serve as a feature representation for node u. Thereafter, mean pooling is used to develop graph-level features based on node-level features.

D UMAP Dimensionality Reduction

The UMAP is a dimensionality reduction technique that is based on three assumptions:

- Data are uniformly distributed on an existing manifold.
- Topological structure of the manifold should be preserved.
- Manifold is locally connected.

Generally, UMAP comprises two stages, including learning a manifold structure in a high-dimensional space and finding the relative representation in the low-dimensional space. In the first phase, the initial step is to find the nearest neighbors for all datapoints, using the nearest-neighbor-descent algorithm. Then, UMAP constructs a graph by connecting the neighbors identified previously; it should be noticed that the data are uniformly distributed across the manifold, so the space between datapoints varies according to regions where data are denser or sparse. According to this assumption, it is possible to introduce the concept of ‘edge weights’: from each point, the distance with respect to the nearest neighbors is computed, so the edge weights between datapoints are computed, but there exists a problem of disagreeing edges.

E Simulated Annealing

Simulated annealing (SA), analogical to the cooling process of metals and glass, is one of the earliest heuristics that has the capability to overcome local optima. This is achieved by allowing moves that lead to probably less fit solutions compared to current ones and thus increase the diversity of the exploration and enabling the algorithm to escape from local optima. The probability of performing such a move will be reduced as the search process precedes. The pseudo-code describing the SA process is shown in Algorithm 2.

Algorithm 2: Simulated Annealing

```
Input: Maximum Number of Iterations $K$; Initial Temperature $T_0$;
Initialize $\mathbf{x} \in \mathcal{X}$;
$i \leftarrow 0$;
while $i \leqslant K$ do
    choose $\mathbf{x}_{i+1} \in \mathcal{N}(\mathbf{x}_i)$;
    if $f(\mathbf{x}_i) \leqslant f(\mathbf{x}_{i+1})$ then
        $\mathbf{x}_i \leftarrow \mathbf{x}_{i+1}$;
    else if $\exp(\frac{f(\mathbf{x}_{i+1})-f(\mathbf{x}_i)}{T_i}) \geqslant rand(0,1)$ then
        $\mathbf{x}_i \leftarrow \mathbf{x}_{i+1}$;
    $i \leftarrow i + 1$;
    $T_{i+1} \leftarrow T_0 \times 0.8^{i/300}$;
return $\mathbf{x}_{i+1}$
```

The algorithm starts by randomly initializing a solution $\mathbf{x}$ from the search space $\mathcal{X}$. An initial temperature T_0 is also specified and will be reduced during the simulation according to a certain strategy, which in this case, the temperature T_i at i-th iteration is given by $T_0 \times 0.8^{i/300}$. Then, for each iteration, a new solution $\mathbf{x}'$ will be drawn from the neighborhood $\mathcal{N}(\mathbf{x})$. $\mathbf{x}'$ will be directly accepted if $f(s') < f(s)$. Otherwise, if $f(s') \geqslant f(s)$, $\mathbf{x}'$ will replace $\mathbf{x}$ with a probability:

$$\exp{(-\frac{f(\mathbf{x}') - f(\mathbf{x})}{T})} \tag{22}$$

Since the temperature T will be reduced as the process goes on, this probability will becomes smaller accordingly. This would allow the algorithm to converge to a high-quality solution at the end. In addition, such probability is also dependent on the difference $f(\mathbf{x}') - f(\mathbf{x})$. The higher this difference, the lower will be the probability to accept a move between the two solutions.

F Supplementary Experiments

In this section, we present the analytical results using alternative metrics/methods. Specifically, in Figure 1-4, we presented the results based on three different correlation metrics, namely Pearson, Spearman and Kendall. Thereafter, in Figure 5, we repeated the final regression experiment using linear regression. Most of the results are consistent with the ones we reported in the main text and could lead to similar conclusions.

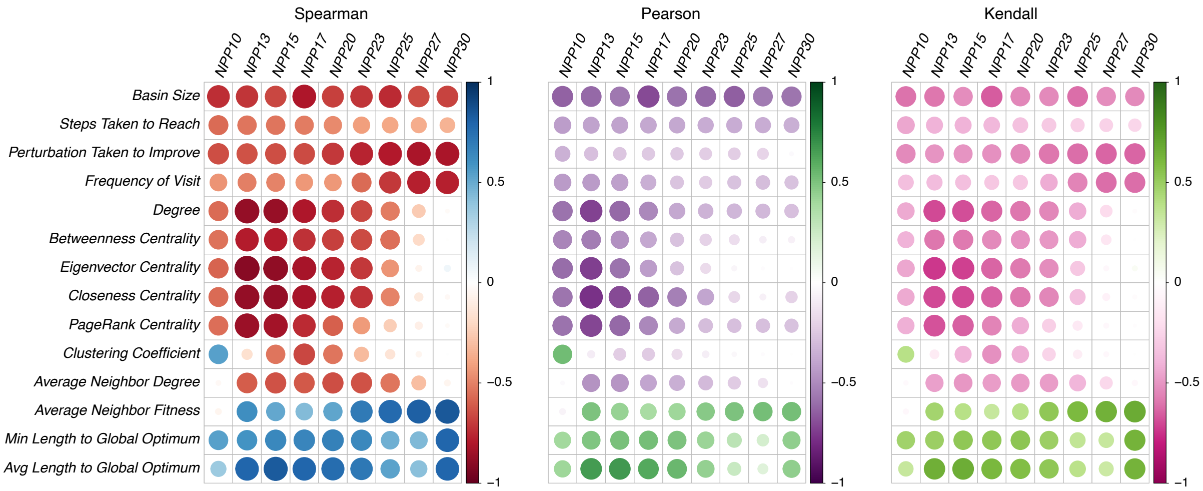

Figure 1: Comparisons of heatmaps of correlations between fitness values and selected features using different correlation measures.

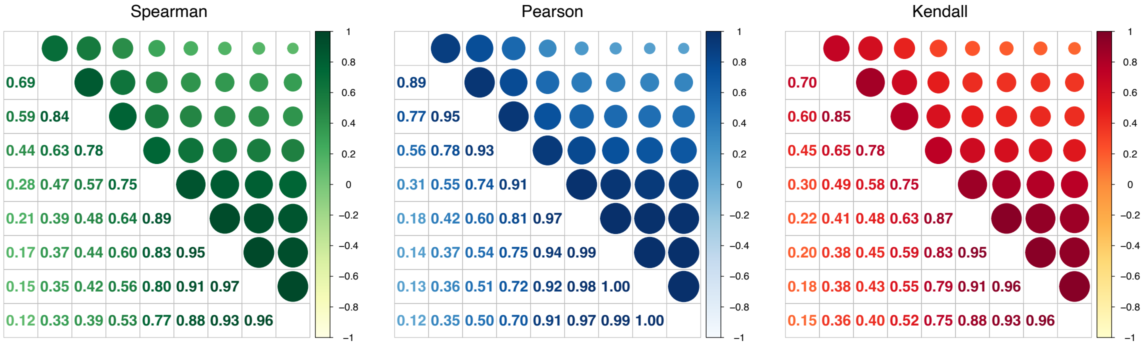

Figure 2: Comparisons of heatmaps of Sim metrics obtained for NPP instances across all studied dimensions using different correlation measures..

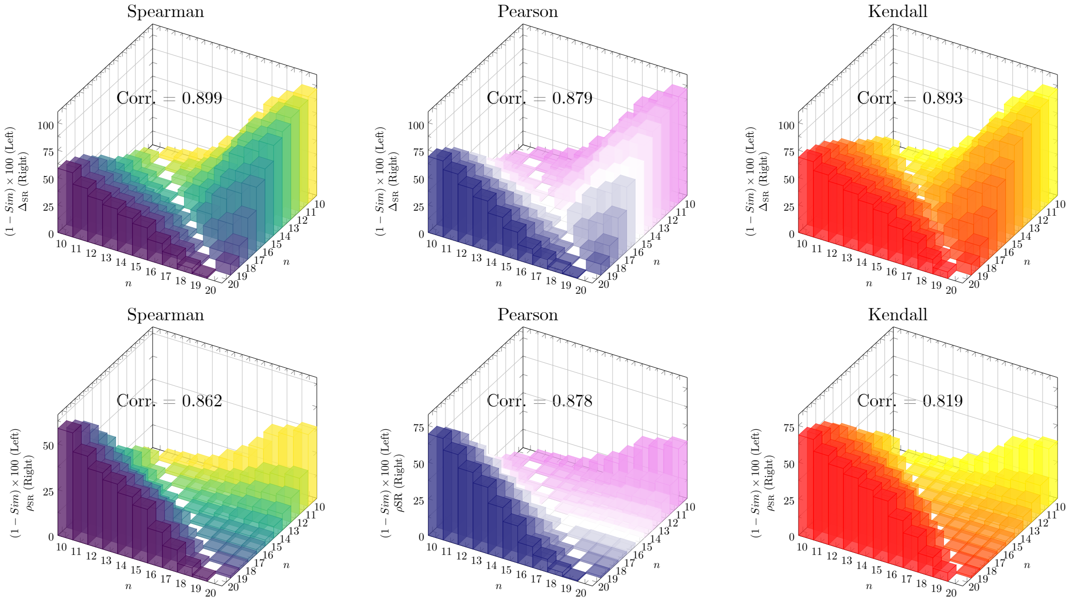

Figure 3: 3D bar charts of the Sim calculated using different correlation measures versus Δ_{SR} and ρ_{SR} across different dimensions.

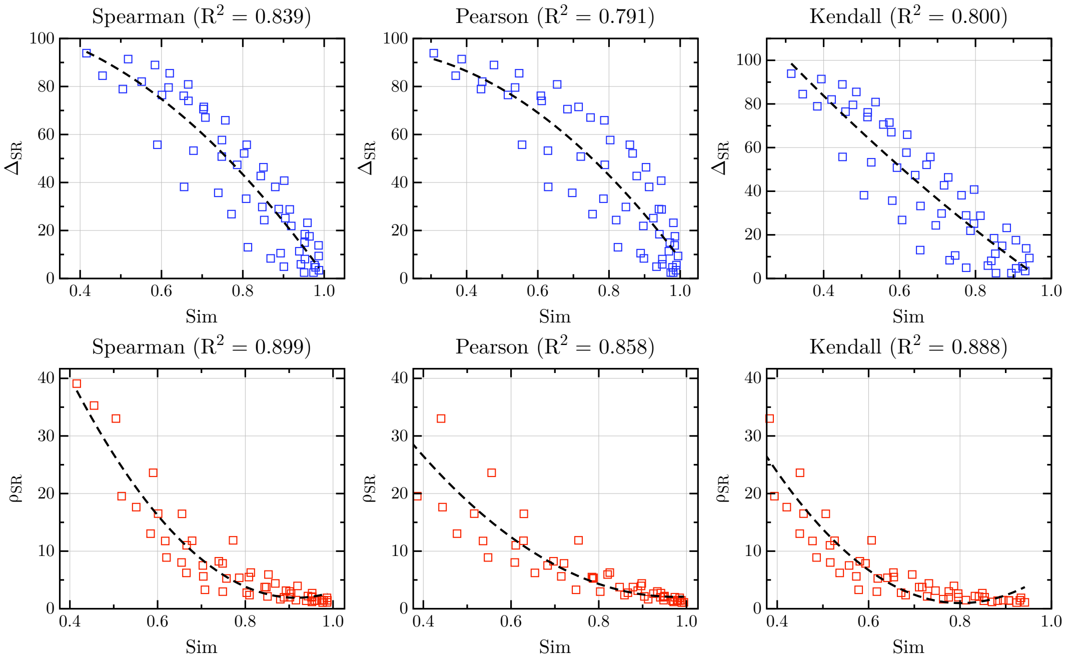

Figure 4: Quadratic regression analysis of Δ_{SR} and ρ_{SR} versus Sim calculated using different correlation measures.

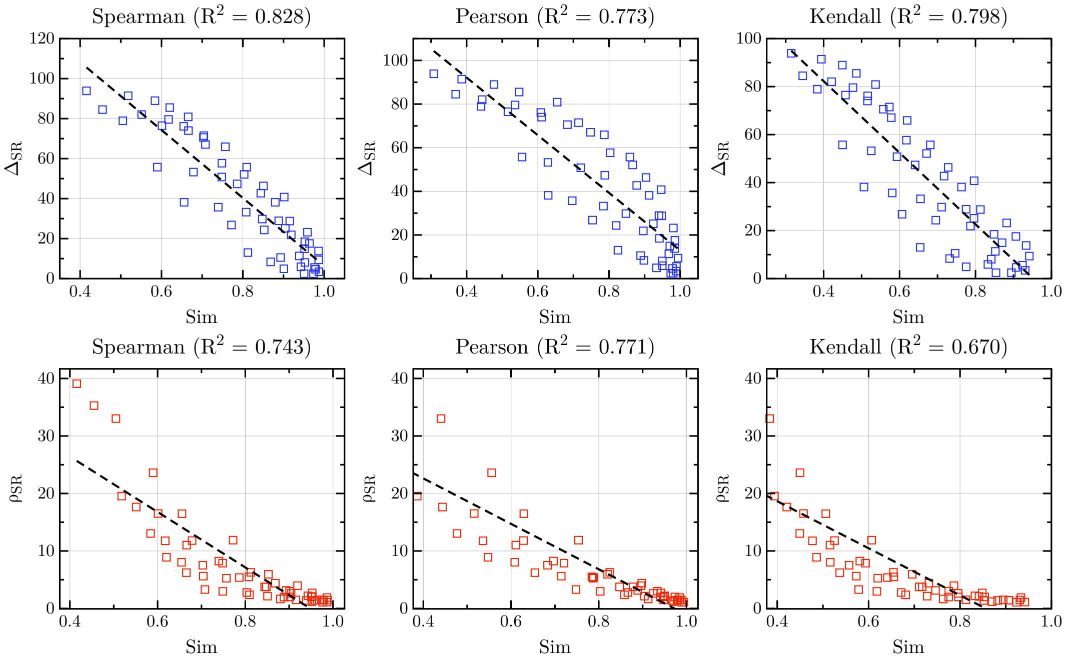

Figure 5: Linear regression analysis of Δ_{SR} and ρ_{SR} versus Sim calculated using different correlation measures.