



HAL
open science

Landscape features and automated algorithm selection for multi-objective interpolated continuous optimisation problems

Arnaud Liefoghe, Sébastien Verel, Benjamin Lacroix, Alexandru-Ciprian Zăvoianu, John Mccall

► To cite this version:

Arnaud Liefoghe, Sébastien Verel, Benjamin Lacroix, Alexandru-Ciprian Zăvoianu, John Mccall. Landscape features and automated algorithm selection for multi-objective interpolated continuous optimisation problems. GECCO 2021 - The Genetic and Evolutionary Computation Conference, Jul 2021, Lille / Virtual, France. pp.421-429, 10.1145/3449639.3459353 . hal-03325676

HAL Id: hal-03325676

<https://hal.science/hal-03325676v1>

Submitted on 25 Aug 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Landscape Features and Automated Algorithm Selection for Multi-objective Interpolated Continuous Optimisation Problems

Arnaud Liefoghe
Univ. Lille, CNRS, Inria, Centrale Lille,
UMR 9189 CRISTAL
F-59000 Lille, France
arnaud.liefoghe@univ-lille.fr

Sébastien Verel
Univ. Littoral Côte d'Opale
UR 4491 LISIC
F-62100 Calais, France
verel@univ-littoral.fr

Benjamin Lacroix
School of Computing
Robert Gordon University
Aberdeen, Scotland, UK
b.m.e.lacroix@rgu.ac.uk

Alexandru-Ciprian Zăvoianu
School of Computing
Robert Gordon University
Aberdeen, Scotland, UK
c.zavoianu@rgu.ac.uk

John McCall
School of Computing
Robert Gordon University
Aberdeen, Scotland, UK
j.mccall@rgu.ac.uk

ABSTRACT

In this paper, we demonstrate the application of features from landscape analysis, initially proposed for multi-objective combinatorial optimisation, to a benchmark set of 1 200 randomly-generated multi-objective interpolated continuous optimisation problems (MO-ICOPs). We also explore the benefits of evaluating the considered landscape features on the basis of a fixed-size sampling of the search space. This allows fine control over cost when aiming for an efficient application of feature-based automated performance prediction and algorithm selection. While previous work shows that the parameters used to generate MO-ICOPs are able to discriminate the convergence behaviour of four state-of-the-art multi-objective evolutionary algorithms, our experiments reveal that the proposed (black-box) landscape features used as predictors deliver a similar accuracy when combined with a classification model. In addition, we analyse the relative importance of each feature for performance prediction and algorithm selection.

CCS CONCEPTS

• **Computing methodologies** → **Continuous space search**; • **Theory of computation** → **Evolutionary algorithms**; • **Applied computing** → **Multi-criterion optimization and decision-making**.

ACM Reference Format:

Arnaud Liefoghe, Sébastien Verel, Benjamin Lacroix, Alexandru-Ciprian Zăvoianu, and John McCall. 2021. Landscape Features and Automated Algorithm Selection for Multi-objective Interpolated Continuous Optimisation Problems. In *2021 Genetic and Evolutionary Computation Conference (GECCO '21)*, July 10–14, 2021, Lille, France. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3449639.3459353>

1 INTRODUCTION

Although the idea of feature-based performance prediction and algorithm selection is not new [27], research interest and effort has intensified over the past few years [2, 14, 30]. With the need to provide a better understanding of problem structure and complexity, various works have provided the community with some interesting sets of quantifiable features evaluating different aspects of the landscape, such as ruggedness or multimodality [16, 23, 24]. These landscape features not only provide a means for analysing the behaviour of search algorithms, but also provide an opportunity to select a priori the most appropriate algorithm for unseen problems. Such a methodology relies on two complementary components: (i) a set of landscape features that characterises the space sufficiently so as to discriminate the performance of given algorithms, and (ii) a comprehensive set of benchmark problems to supply a solid knowledge base for training and testing predictive models.

In this paper, we combine two recent works aimed at providing those two components in the context of continuous multi-objective optimisation problems (MOOPs). The first work [21] provides a comprehensive set of landscape features for multi-objective combinatorial optimisation. These features characterise different facets of difficulty, based on local measures computed over an affordable sample of solutions. In fact, while there exists a substantial literature on the landscape of single-objective continuous optimisation problems, features for continuous MOOPs are lacking [14, 22]. We notice, however, that features from single-objective optimisation have been used to characterise MOOPs [15], although single-objective features certainly do not capture intrinsic properties of multi-objective landscapes, such as interaction among objectives. Here, we propose to transfer the multi-objective landscape features developed in [21] from problems with a combinatorial search space to problems with a continuous search space. In particular, the random and adaptive walks used for sampling are replaced by a fixed-cost latin hypercube sampling. This allows cost efficient application (in terms of evaluations) to automated algorithm selection and performance prediction. The second work this paper builds upon outlines a method for generating a diversified benchmark set of continuous MOOPs [34]. The generator uses interpolated continuous optimisation problems to produce MOOPs with various characteristics and

structures. In total, 1 200 bi-objective problems were investigated on four well-established multi-objective evolutionary algorithms and discriminated the performance between those algorithms. Interestingly, the parameters used to generate the problems have some predictive power on algorithm performance. We evaluate our proposed multi-objective landscape features on these continuous problems, and provide a comprehensive analysis of the relationship between landscape features and algorithm performance. In particular, we analyse the correlation between the absolute performance of each algorithm in terms of hypervolume and the considered features. We then use the features as predictors for performance prediction and algorithm selection on random forest regression and classification models, respectively. We also analyse the predictive power of the proposed landscape features against the parameters used by the random multi-objective interpolated continuous optimisation problem generator from [34].

The paper is organised as follows. Section 2 recalls the construction of multi-objective interpolated continuous optimisation problems. Section 3 gives the experimental setup of our analysis. Section 4 introduces a number of features for characterising continuous multi-objective landscapes, and correlates them with the performance of multi-objective evolutionary algorithms. Section 5 analyses an automated algorithm selection approach using the proposed landscape features as predictors. The last section concludes the paper and discusses further research.

2 MULTI-OBJECTIVE INTERPOLATED CONTINUOUS OPTIMISATION PROBLEMS

2.1 Multi-objective Optimisation

Let us consider an objective function vector $F: X \mapsto Z$ to be minimised, such that $X \subseteq \mathbf{R}^d$ is the variable space and $Z \subseteq \mathbf{R}^m$ is the objective space. Each solution $x \in X$ maps to an objective vector $z \in Z$ such that $z = F(x)$. Given two objective vectors $z, z' \in Z$, z is dominated by z' iff for all $i \in \{1, \dots, m\}$ $z'_i \leq z_i$, and there exists a $j \in \{1, \dots, m\}$ such that $z'_j < z_j$. Similarly, given two solutions $x, x' \in X$, x is dominated by x' iff $F(x)$ is dominated by $F(x')$. An objective vector $z^* \in Z$ is non-dominated if there does not exist any $z \in Z$ such that z^* is dominated by z . A solution $x^* \in X$ is Pareto optimal, or non-dominated, if $F(x)$ is non-dominated. The set of all Pareto optimal solutions is the Pareto set (PS); its mapping in the objective space is the Pareto front (PF). One of the main challenges in multi-objective optimisation is to identify the PS, or a good approximation of it for large-size and complex problems. A number of multi-objective evolutionary algorithms (MOEAs) have been designed to this end since the late eighties [6, 7].

2.2 Problem Definition

Multi-objective interpolated continuous optimisation problems (MO-ICOPs) have been recently introduced in [34] as a new class of benchmark problems with tunable landscapes. MO-ICOPs are defined by an objective function vector that combines m single-objective ICOPs generated using a common set of seeds. A MO-ICOP can be formulated as follows:

$$\text{minimise } F_{(S, U_1, \dots, U_m)}(x) = (f_{S, U_1}(x), \dots, f_{S, U_m}(x))^T, \quad (1)$$

where each $f_{S, U_i}(x)$ represents a single-objective ICOP defined by:

- **A set of seeds** $S \subset X$: a (typically finite) set of distinct candidate solutions with an assigned objective value. Elements of S and their assigned objective values U_i will define the entire optimisation problem via interpolation.
- **An interpolation function** ($f_{S, U_i}: X \mapsto \mathbf{R}$): in this paper, we apply the inverse distance weighting method, originally defined by Shepard for spatial analysis [28]. Assuming the seed set S contains N seeds, labelled $S = \{s_1, \dots, s_N\}$, with the assigned objective values $U_i = \{u_{i,1}, \dots, u_{i,N}\}$, we define for any candidate solution $x \in X$:

$$f_{S, U_i}(x) = \begin{cases} \frac{\sum_{j=1}^N \frac{u_{i,j}}{e(x, s_j)^k}}{\sum_{j=1}^N \frac{1}{e(x, s_j)^k}} & \text{if } e(x, s_j) \neq 0 \text{ for all } j \\ u_{i,j} & \text{if } e(x, s_j) = 0 \text{ for some } j \end{cases} \quad (2)$$

where $e(x, y): X \times X \mapsto \mathbf{R}$ is the Euclidean distance, and k is a positive real number called the *power parameter*. Higher values of k increase the relative influence of nearby seeds on the interpolated value.

3 EXPERIMENTAL SETUP

3.1 Benchmark Dataset

We consider the set of bi-objective problems proposed in [34]. For each MO-ICOP, the set of seeds S is randomly generated in the variable space $X = [-5, 5]^d$. The objective vector of each seed is first assigned to a subset of non-dominated seeds S_{nd} , ensuring that no seed in S_{nd} dominates another seed from S_{nd} . Objective values are then assigned to the remaining dominated seeds S_d , ensuring that each seed in S_d does not dominate any seed in S_{nd} .

Using this method, 50 problems were generated in 4 problem dimensions $d \in \{5, 10, 20, 30\}$ and 6 values for $k \in \{1, 2, 3, 4, 5, 6\}$, creating a total of 1 200 problems covering a wide range of MO-ICOPs. We emphasise that, although the problem dimension is typically given, the seeds and the power of interpolation are *unknown* for unseen (black-box) problems.

3.2 Considered Algorithms

By analysing the outputs of the experiments reported in [34]¹, the present work focusses on the comparative performance of four different MOEAs parameterised using recommended settings from the literature. In [34], the authors state that these MOEAs were chosen because they exemplify different well-proven multi-objective optimisation strategies. Thus, NSGA-II [8] pioneered alongside SPEA2 [38] is an effective evolutionary approach grounded in Pareto-based elitism (as a primary criterion) and objective space distancing (as a tie-breaker). GDE3 [19] also uses this strategy, but replaces the simulated binary crossover (SBX) and polynomial mutation (PM) variation operators [7] popularised by NSGA-II with a differential evolution (DE) strategy: DE/r and 1/bin. MOEA/D-DE-DRA [36] is a state-of-the-art solver that also integrates a DE strategy, but its main characteristic is that it applies a weighting-based aggregation of the objectives to decompose the original MOOP into multiple single-objective sub-problems that are solved concurrently during a single optimisation run. Finally, DECMO2++ [35]

¹Accessible at: https://github.com/czavoianu/PPSN_2020

was designed to achieve a fast convergence across a wide range of benchmark MOOPs by actively pivoting between three search strategies implemented via two actively co-evolved sub-populations (based on SPEA2 and GDE3) and a largely passive decomposition-based archive. In the case of NSGA-II, GDE3 and DECMO2++, a population / archive size of 200 was used. For MOEA/D-DE-DRA, the population size was set to 300, the standard setting for MOOPs with two objectives.

3.3 Budgets and Search Performance

A total computational budget of 50 000 evaluations was allowed for each optimisation run. As 100 independent runs were carried out for each solver on each MO-ICOP, the numerical experiments entailed a total of 480 000 independent optimisation runs across the 1 200 MO-ICOPs from the benchmark set. Search performance is assessed after 10 000, 20 000 and 50 000 calls to the evaluation function, using a normalised version of the hypervolume indicator [37] that quantifies the area of the objective space the MOEA population / archive dominates when compared against the true PF for the problem under consideration. A normalised hypervolume of 1 actually means that the MOEA hit the PF.

When considering end-of-the-run optimisation results across the 1 200 benchmark problems, MOEA/D-DE-DRA achieved the best (i.e., highest) average hypervolume on 545 problems ($\approx 45.4\%$), NSGA-II won on 509 problems ($\approx 42.4\%$), while GDE3 and DECMO2++ scored 125 and 21 *wins*, respectively.

4 FEATURE-BASED LANDSCAPE ANALYSIS

In this section, we define a number of landscape features for characterising MO-ICOPs. In particular, we draw inspiration from the features introduced in [21] for *combinatorial* multi-objective landscapes. We start by describing them, and how we adapt them to *continuous* multi-objective landscapes. Then, we show how they individually correlate with the performance of the considered MOEAs. Finally, we measure their relative importance when predicting the performance of each MOEA for solving MO-ICOPs.

4.1 Features for Continuous Multi-objective Landscapes

The considered landscape features are based on different measures computed over a *sample* of solutions extracted from the problem under consideration. In order to gather information from a given continuous multi-objective landscape, we first sample a reasonable number of n solutions from the search space. The sample P is constructed according to a *random latin hypercube design* [5]. Sampled points are then evaluated by means of the objective function vector F , and the n evaluated solutions constitute the total budget devoted to the computation of landscape features. For each solution $x \in P$, we additionally define its d closest solutions from the sample other than x as its *neighbours*, d being the problem dimension. The closeness is measured as the Euclidean distance in the variable space. Based on the sample of size n and the definition of neighbourhood, we introduce four classes of landscape features below. In addition, we consider five problem-dependent features coming from the construction and mathematical formulation of MO-ICOPs, as discussed in Sect. 3.1. These features will be used as

reference in the coming analysis on the predictive performance of landscape features in a black-box context. The 49 landscape features and 5 problem-dependent features are summarised in Tab. 1.

4.1.1 Global Landscape Features. The first class of landscape features is *not* based on the neighbourhood, and renders global properties from the continuous multi-objective landscape. This class ranges from the (Spearman) correlation among the objective values measured on the sample, to the average and maximum distance among sampled solutions in the variable space and in the objective space. We also measure the proportion of non-dominated solutions among the sample, the proportion of supported points [10] therein, the hypervolume covered by these non-dominated solutions, the average and maximum distance among non-dominated solutions in the variable space, as well as the (Spearman) correlation between the distance among them in the variable space vs. in the objective space – denoted as *fitness-distance-correlation* in [18]. Finally, we divide the sample into different ranks following the principles of non-dominated sorting used, e.g., in NSGA-II [8], and we measure the average and maximum rank [1, 11] as well as the entropy of the distribution of the number of solutions per rank.

4.1.2 Features Characterising the Landscape Multimodality. In terms of multimodality, we define a single-objective local optimum (slo) in the sample as a solution with no improving neighbour for a given objective. Similarly, we define a Pareto local optimum (plo) as a sampled solution for which there does not exist any dominating neighbour [26]. We measure the proportions of slo and plo in the sample, and the average and maximum distances among them in the variable space. We also measure how many (globally) non-dominated points in the sample there are, on average, per plo.

In single-objective landscape analysis, an *adaptive walk* is often performed to estimate the multimodality: assuming isotropy in the landscape, the longer the length of an adaptive walk, the larger the basins size, hence the lower the number of local optima [13]. We here perform a simple multi-objective adaptive walk, where the neighbourhood is explored from the closest to the furthest neighbour w.r.t. the current solution, and where the first dominating neighbour is accepted at each step [32]. The length (`length_aws`) corresponds to the number of steps performed until no further improvement is possible, and the walk falls into a plo. We also record the number of solutions explored along the walk (`eval_aws`). Multiple adaptive walks are typically performed from different starting points to improve feature estimation. We report below average values from 30 independent adaptive walks. It is important to note that each adaptive walk is performed over the pre-computed sample, and does not require any additional evaluation.

4.1.3 Features Characterising the Landscape Evolvability. Let us now quantify the expected improvement that can be achieved from a solution's neighbourhood as measures of *evolvability* [29] for multi-objective optimisation. For each solution from the sample, we measure the proportions of dominating (`sup`), dominated (`inf`), and incomparable (`inc`) neighbours. We also measure the proportion of locally non-dominated neighbours (`lnd`), as well as the proportion of locally supported solutions therein (`lsupp`). In addition, we compute the average distance from each solution to its neighbours in the variable space and in the objective space (`dist_x` and `dist_f`,

Table 1: Considered problem-dependent and landscape features for continuous multi-objective optimisation.

	name	description
problem-dependent	d	number of variables
	k	power of interpolation
	seed_n	proportional number of seeds $ S $
	nd_seed_n	proportion of non-dominated seeds $ S_{nd} $
	dom_seed_n	proportion of dominated seeds $ S_d $
global	f_cor	correlation among objective values
	dist_x_avg	average distance among solutions in the variable space
	dist_x_max	maximum distance among solutions in the variable space
	dist_f_avg	average distance among solutions in the objective space
	dist_f_max	maximum distance among solutions in the objective space
	nd_n	proportion of non-dominated solutions
	supp_n	proportion of supported non-dominated solutions
	hv	hypervolume of non-dominated solutions
	dist_x_nd_avg	average distance among non-dominated solutions in the variable space
	dist_x_nd_max	maximum distance among non-dominated solutions in the variable space
	fdc	fitness-distance correlation among non-dominated solutions
	rank_avg	average rank w.r.t. non-dominated sorting
	rank_max	maximum rank w.r.t. non-dominated sorting
rank_ent	entropy of the number of solutions per rank w.r.t. non-dominated sorting	
multimodality	slo_n	proportion of single-objective local optima per objective
	slo_dist_avg	average distance among single-objective local optima in the variable space
	slo_dist_max	maximum distance among single-objective local optima in the variable space
	plo_n	proportion of Pareto local optima
	plo_dist_avg	average distance among Pareto local optima in the variable space
	plo_dist_max	maximum distance among Pareto local optima in the variable space
	nd_per_plo	proportion of non-dominated solutions per Pareto local optimum (i.e. nd_n / plo_n)
	length_aws	average length of adaptive walks
eval_aws	average number of calls to the evaluation function performed by adaptive walks	
evolvability	sup_avg_neig	average proportion of dominating neighbours
	inf_avg_neig	average proportion of dominated neighbours
	inc_avg_neig	average proportion of incomparable neighbours
	lnd_avg_neig	average proportion of locally non-dominated neighbours
	lsupp_avg_neig	average proportion of supported locally non-dominated neighbours
	dist_x_avg_neig	average distance from neighbours in the variable space
	dist_f_avg_neig	average distance from neighbours in the objective space
	dist_f_dist_x_avg_neig	ratio of the average distance from neighbours in the objective and variable spaces ($dist_f_avg_neig / dist_x_avg_neig$)
	diff_f_avg_neig	average difference per objective with neighbours
	diff_f_dist_x_avg_neig	ratio of the average diff. per objective with neighbours and the dist. in the variable space ($diff_f_avg_neig / dist_x_avg_neig$)
	hv_avg_neig	average (single) solution's hypervolume
hvd_avg_neig	average (single) solution's hypervolume difference with neighbours	
nhv_avg_neig	average hypervolume from the whole neighbourhood	
ruggedness	sup_cor_neig	neighbour's correlation of the proportion of dominating neighbours
	inf_cor_neig	neighbour's correlation of the proportion of dominated neighbours
	inc_cor_neig	neighbour's correlation of the proportion of incomparable neighbours
	lnd_cor_neig	neighbour's correlation of the proportion of locally non-dominated neighbours
	lsupp_cor_neig	neighbour's correlation of the supported locally non-dominated neighbours
	dist_x_cor_neig	neighbour's correlation of the average distance from neighbours in the variable space
	dist_f_cor_neig	neighbour's correlation of the average distance from neighbours in the objective space
	dist_f_dist_x_cor_neig	neighbour's correlation of ratio of the average distance from neighbours in the objective and variable spaces
	diff_f_cor_neig	neighbour's correlation of the average difference per objective from neighbours
	diff_f_dist_x_cor_neig	neighbour's correlation of the ratio of the avg. diff. per obj. from neighbours and the dist. in the variable space
	hv_cor_neig	neighbour's correlation of the average (single) solution's hypervolume
	hvd_cor_neig	neighbour's correlation of the average (single) solution's hypervolume difference with neighbours
	nhv_cor_neig	neighbour's correlation of the average hypervolume from the whole neighbourhood

respectively), the ratio of both ($dist_f_dist_x$), as well as the average differences from each solution to its neighbours per objective ($diff_f$), together with the ratio over their distance in the variable space ($diff_f_dist_x$). At last, we compute the hypervolume covered by each solution (hv), the average difference from the hypervolumes of its neighbours (hvd), and the hypervolume covered by all its neighbours (nhv). For each measure described above, we simply compute the average value over the whole sample to construct the corresponding feature quantifying the landscape evolvability.

4.1.4 Features Characterising the Landscape Ruggedness. The correlation among the information and quality from neighbouring solutions characterises the *ruggedness* of the landscape [25, 33]: the larger the correlation, the smoother the landscape. Before being averaged to compute the features for evolvability, every measure presented in the previous paragraph is assigned to each solution from the sample. Based on this information, and for each of the thirteen measures, we compute the (Spearman) correlation coefficient over each pair of neighbours in the sample. We end up with thirteen additional features characterising the ruggedness of continuous multi-objective landscapes.

Let us recall that all 49 landscape features considered in our analysis are summarised in Tab. 1, below the 5 problem-dependent features. For more details on landscape features, and to better understand their motivations, the reader is referred to [21]. They are all computed over the same fixed-size sampling of size n . We thus stress that the overall cost of landscape features in terms of calls to the evaluation function is exactly n . We analyse the correlation of these features with the performance of MOEAs below.

4.2 Features vs. Search Performance

In the following sections, we start by considering a sampling of size $n = 200 \cdot d$ solutions, where d is the number of variables for the problem under consideration. Fig. 1 reports the Spearman correlation coefficient of each feature with the expected performance of the four considered MOEAs. Due to space restriction, we focus on a large search budget, where each MOEA stops after 50 000 calls to the evaluation function. We recall that search performance is measured in terms of normalised hypervolume, so that a higher value means a better performance. As such, a positive correlation means a favourable effect of the feature on search performance, whereas a negative correlation means the opposite.

We first remark that the correlations are similar for the four algorithms, although MOEA/D-DE-DRA seems to slightly stand out. The features most correlated with favourable algorithm performance are `dist_f_avg` and `dist_f_max`. They measure the average and maximum distance in the objective space among the sampled solutions. In a separate analysis, we noticed that those features are highly correlated with the k parameter of the interpolation function. Indeed, low k values tend to produce landscapes concentrated around the average objective values of the seeds with sharp pits and peaks around the seeds [34]. This creates a large cluster around the expected values in the objective space. This increases the difficulty for MOEAs to reach the PF [34] and, as reflected in Fig. 1, MOEA/D-DE-DRA seems less affected by this effect.

Surprisingly, our coarse-grained analysis reveals that features describing the landscape multimodality (e.g., `slo_n` and `plo_n`) are positively correlated with search performance; that is, the hypervolume tends to be better when there are more local optima. This seems to corroborate recent results from continuous multi-objective optimisation [12]. We believe that a more fine-grained analysis on this specific class of features would be beneficial to better understand the impact of multimodality on MOEAs.

4.3 Importance of Features for Search Performance

We now investigate the *combined* effect of the considered features on search performance. For this purpose, we train a random forest regression model [3] that predicts the expected (relative) hypervolume based on input features, separately for each algorithm. We perform 50 independent trainings on the benchmark dataset using the `randomForest` R package [20] with default parameters, and we report average values below. The coefficients of determination (R^2) of the models on training data are 0.88, 0.89, 0.86, and 0.88 respectively, for NSGA-II, GDE3, MOEA/D-DE-DRA, and DECIMO2++. This means that more than 85% of the variance in the hypervolume of each algorithm between MO-ICOPs is explained by the features.

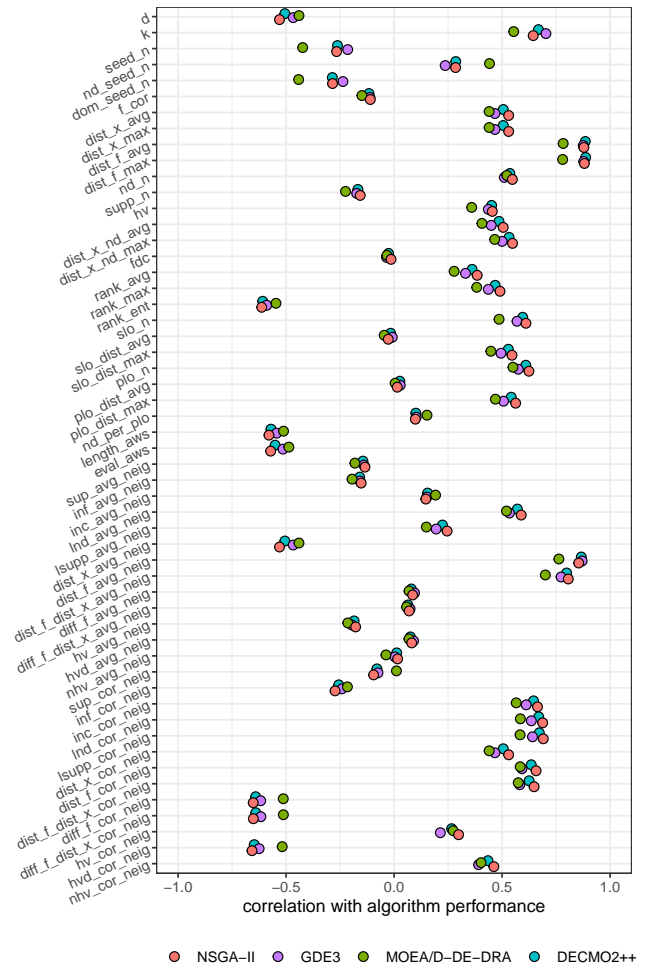


Figure 1: Correlation between the features and the performance of algorithms under a search budget of 50 000.

In addition to prediction accuracy, random forest models yield the relative importance of each feature for prediction. A measure of importance is the so-called mean decrease of prediction accuracy after each split on a given predictor [3]: the larger the decrease, the more important the predictor. The (average) relative importance of features is reported in Fig. 2. For readability, only the 12 most important features are depicted for each algorithm, sorted in decreasing order of importance, from top to bottom. Although the subset of important features is similar for all MOEAs, their relative ranking is slightly different. Two notable features that consistently appear on top of the ranking for all algorithms are the average solution’s hypervolume (`hv_avg_neig`) and the average neighbourhood’s hypervolume (`nhv_avg_neig`), both characterising the landscape evolvability. The fact that algorithm performance is measured in terms of hypervolume partly explains the importance of these features, and the hypervolume measured over the entire sample (`hv`) also appears to be important, although to a lesser extent. As already pointed out in the correlation analysis reported above, the distance between sampled solutions in the objective space (`dist_f_avg` and `dist_f_max`) is also

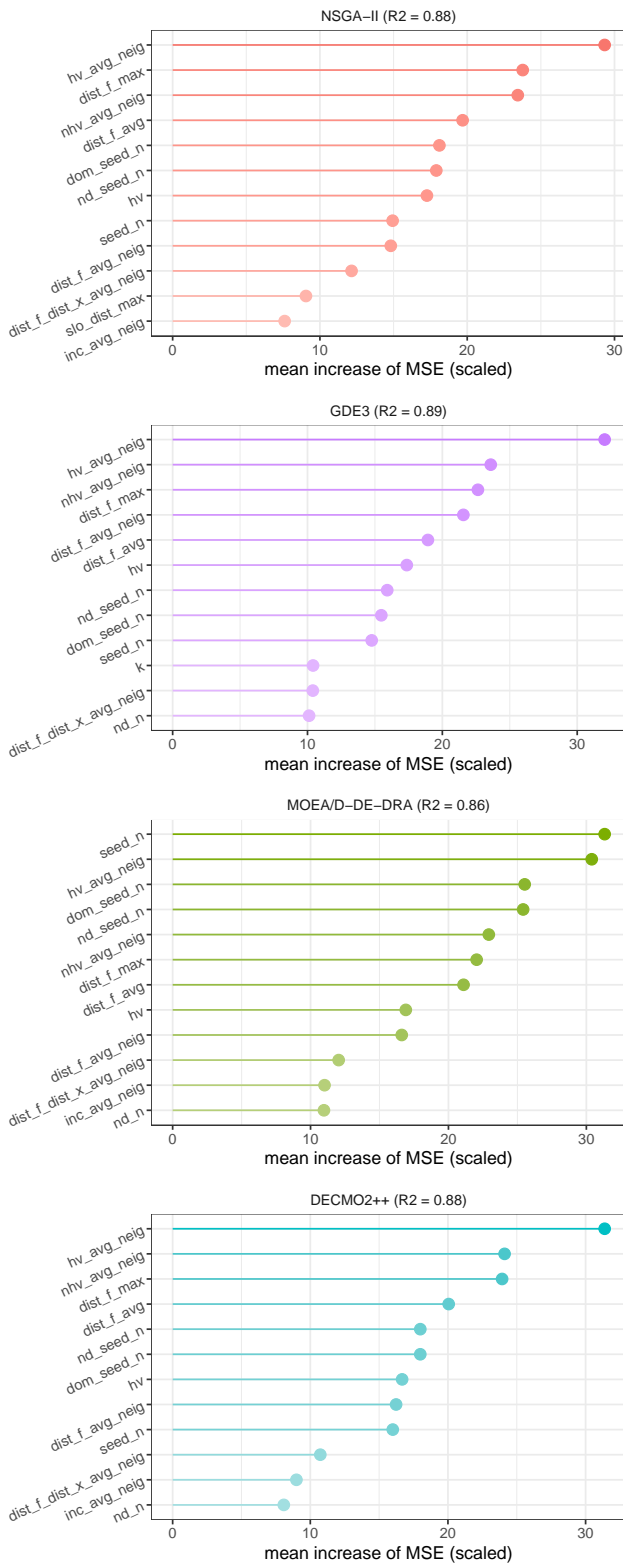


Figure 2: Relative importance of features for predicting the performance of algorithms under a search budget of 50 000.

found to be important, together with different distance-based features characterising evolvability (notably `dist_f_avg_neig` and `dist_f_dist_x_avg_neig`). Finally, the problem-dependent features related to the seeds (`seed_n`, `nd_seed_n`, `dom_seed_n`) seem to significantly impact the performance of all algorithms, and particularly MOEA/D-DE-DRA. Surprisingly, the problem dimension (`d`) is found to be less important than the features mentioned above. It is also worth noticing that there is no feature related to multimodality, nor to ruggedness, other than at the 11th and 12th position for NSGA-II. We will see, however, that these features are important to discriminate between different algorithms.

5 FEATURE-BASED ALGORITHM SELECTION

We conclude our study by analysing the selective power of the proposed landscape features. In contrast with the previous section, we train a random forest *classification* model to predict which of NSGA-II, GDE3, MOEA/D-DE-DRA, or DECMO2++ performs better, on average, for a given problem.

5.1 Importance of Features for MOEA Selection

We now focus on the importance of features for discriminating between the four considered MOEAs. The classification errors observed on the training data are 16.46%, 14.92%, and 22.78%, respectively, for search budgets of 10 000, 20 000 and 50 000 evaluations. This means that the classification model accurately selects the right algorithm in more than 75% of the cases. Features' importance for algorithm selection is depicted in Fig. 3. Whatever the search budget, the maximum distance between sampled solutions in the objective space (`dist_f_max`) appears to be the most important feature to discriminate between different algorithms. For smaller search budgets (10 000 and 20 000), the average distance from a solution's neighbourhood (`dist_f_avg_neig`) and the problem-dependent feature that tunes the power of the interpolation function (`k`) rank next, followed by the average distance among sampled solutions in the objective space (`dist_f_avg`) and several hypervolume-based evolvability features. In contrast, for a larger budget (50 000), the first problem-dependent feature (`seed_n`) is ranked 8, and a number of distance- and hypervolume-based features characterising the evolvability and the ruggedness appear on top of the ranking.

To complement our study, we also construct another classifier based on a simple decision tree [4, 31], as illustrated in Fig. 4, for a search budget of 50 000 evaluations. Even with such a simple decision tree of depth four, the proposed features are able to distinguish between the different algorithms with an error of 22.58%. The root of the decision tree corresponds once again to `dist_f_max`. While MOEA/D-DE-DRA is the best algorithm overall for this budget, the classifier recommends selecting NSGA-II instead when sampled solutions are more scattered in the objective space. Below the root of the tree appear landscape features related to multimodality (`slo_n`), evolvability (`dist_f_dist_x_avg_neig`, `nhv_avg_neig`, `hvd_avg_neig`) and ruggedness (`hvd_cor_neig`, `sup_cor_neig`, `inc_cor_neig`, `nhv_cor_neig`).

5.2 Prediction Accuracy

We finally investigate the ability of a feature-based model to accurately select a suitable algorithm for *unseen* problems. We measure

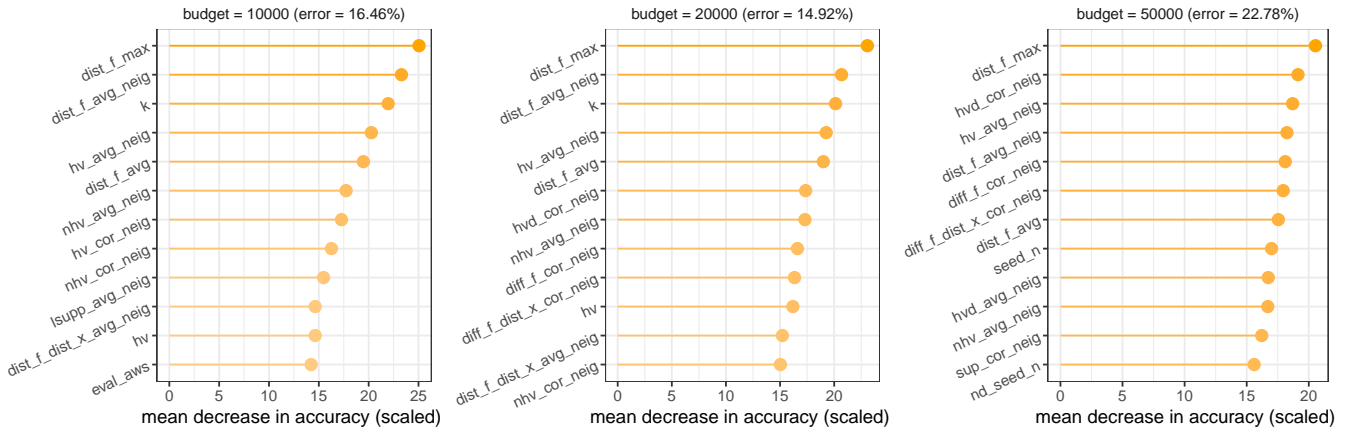


Figure 3: Relative importance of features for algorithm selection under various search budgets.

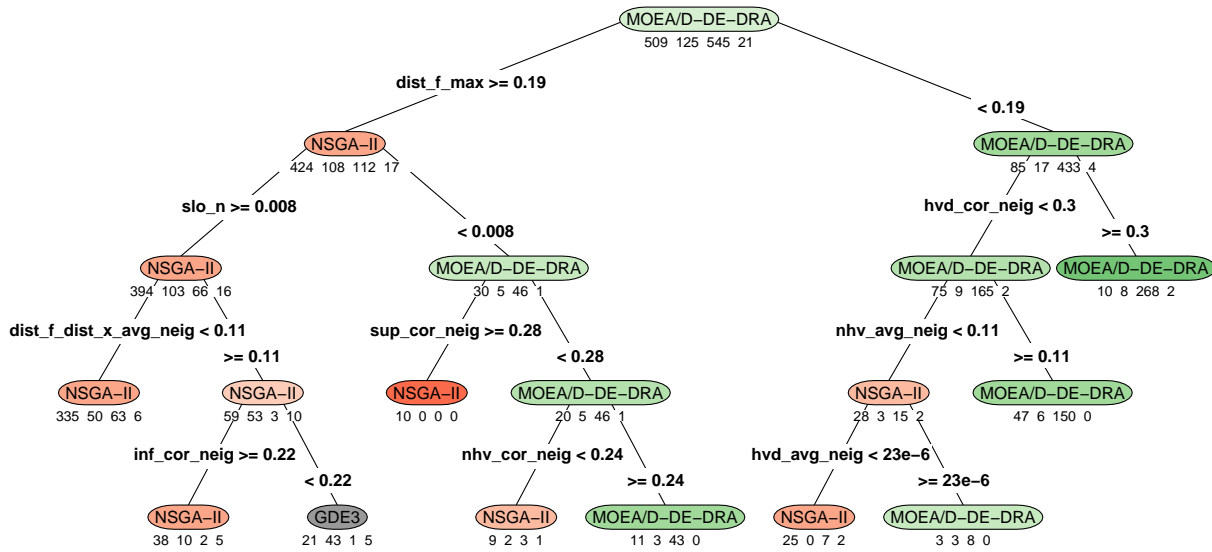


Figure 4: CART decision tree for algorithm selection under a search budget of 50 000. The nodes report the number of instances where NSGA-II, GDE3, MOEA/D-DE-DRA, and DECMO2++ performs better, on average, from left to right, respectively.

the accuracy of random forest classification models based on different subsets of features and different budgets allocated to their computation (i.e. the sampling size). In addition to the sample of $200 \cdot d$ solutions considered above, we investigate tighter values of 500 and 1 000. For each setting, results are from 50 independent executions of repeated random sub-sampling cross-validation [9] with a 80/20% split. Three error values are reported in Fig. 5: the error rate in predicting the algorithm with the best performance on average (top), the error rate in predicting an algorithm that is not statistically outperformed by any other, according to a Mann-Whitney test at a significance level of 0.05 with Bonferroni correction (middle), and the relative hypervolume deviation from the virtual best, an oracle that selects the best algorithm for any instance and budget (bottom).

Whatever the budget, a feature-based classifier is able to predict the best algorithm on average in more than 75% of the cases, and for one of the statistical best algorithms, in more than 85% of the cases. In terms of hypervolume, it is within 0.5% of the virtual best. This is significantly more accurate than a dummy (feature-less) classifier that always predicts the most-frequent best algorithm (NSGA-II for a budget of 10 000, and MOEA/D-DE-DRA for a budget of 20 000 and of 50 000), or than a classifier solely based on the problem dimension d , the only known problem-dependent feature in a black-box scenario. Interestingly, we do not observe any significant difference between a model based on landscape features only and a model based on (unknown) problem-dependent features. This means that the information extracted from a limited sampling of solutions allows a classification model to recommend

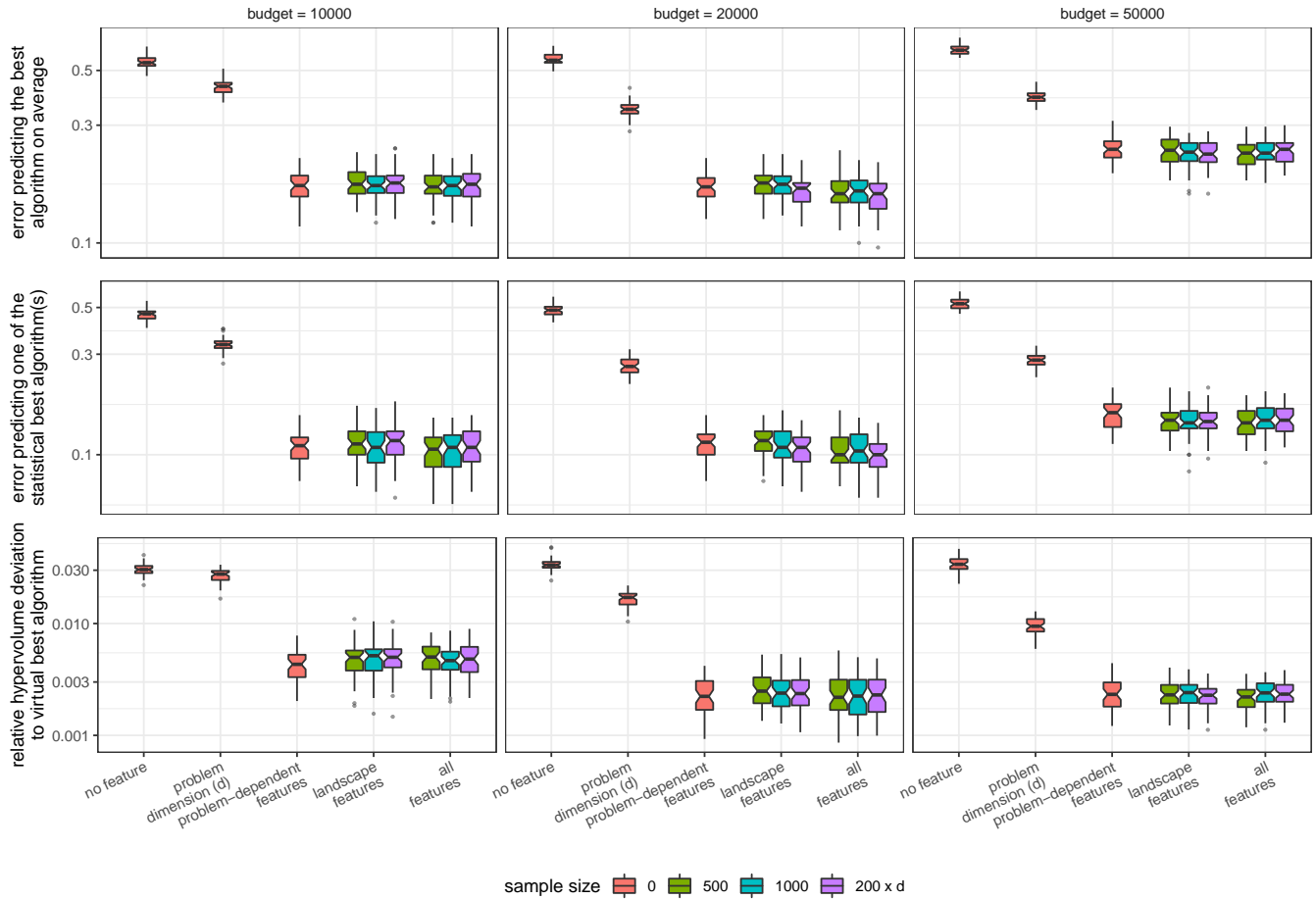


Figure 5: Cross-validated prediction accuracy for different subsets of features and sample sizes under various search budgets.

which algorithm to run on a blind instance with the same level of accuracy than it would do with deep problem knowledge, related to the random seeds and the interpolation function in the case of MO-ICOPs.

When analysing the three sample sizes considered for computing the landscape features, we observe that using fewer solutions does not significantly impact the prediction accuracy. This is of particular interest when confronting the budget allocated to the calculation of features with the budget allocated to the search process. Indeed, using a very limited sampling of 500 solutions, that is 1 to 5% of the search budget depending on the stopping condition, is enough to accurately discriminate between the different MOEAs. In comparison to a classifier that does not use any feature, the feature-based algorithm selection approach always reduces the prediction error by a factor greater than 2, and reduces the hypervolume relative deviation by a factor of 10.

6 CONCLUSIONS

In this paper, we proposed a set of landscape features for continuous multi-objective optimisation. Those features were evaluated

on a benchmark set of 1 200 multi-objective interpolated continuous optimisation problems. By means of correlation analysis and algorithm performance prediction, our analysis highlights that the features have a similar predictive power for the performance of all four tested algorithms. In addition, when combining them with a classification model for algorithm selection, the landscape features exhibit strong predictive power with an accuracy of over 85%, which is similar to the predictive power of problem-dependent features. This demonstrates the salience of the proposed features to characterising the landscape of multi-objective continuous optimisation problems, and to making viable recommendations for selecting a well-suited algorithm when faced with unseen problems.

In terms of benchmarks, we believe that MO-ICOPs provide a useful platform for generating diverse multi-objective continuous optimisation problems. It would be interesting to investigate problems with more than two objectives for assessing the scalability of landscape features. The framework proposed here also offers scope to compare our understanding of complexity for both combinatorial and continuous MOOPs. Finally, additional features could be considered and combined as well, including single- and multi-objective continuous landscape features [16, 17, 23, 24].

REFERENCES

- [1] H. Aguirre and K. Tanaka. 2007. Working principles, behavior, and performance of MOEAs on MNK-landscapes. *European Journal of Operational Research* 181, 3 (2007), 1670–1690.
- [2] N. Belkhir, J. Dréo, P. Savéant, and M. Schoenauer. 2017. Per instance algorithm configuration of CMA-ES with limited budget. In *Genetic and Evolutionary Computation Conference (GECCO 2017)*. ACM, Berlin, Germany, 681–688.
- [3] L. Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32.
- [4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and regression trees*. Wadsworth & Brooks.
- [5] R. Carnell. 2020. *lhs: Latin hypercube samples*. <https://CRAN.R-project.org/package=lhs> R package version 1.1.1.
- [6] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems* (second ed.). Springer.
- [7] K. Deb. 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons.
- [8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm : NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.
- [9] Werner Dubitzky, Martin Granzow, and Daniel P Berrar. 2007. *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media.
- [10] M. Ehrgott. 2005. *Multicriteria optimization* (second ed.). Springer.
- [11] D. Garrett and D. Dasgupta. 2009. Plateau connection structure and multiobjective metaheuristic performance. In *IEEE Congress on Evolutionary Computation (CEC 2009)*. 1281–1288.
- [12] C. Grimme, P. Kerschke, and H. Trautmann. 2019. Multimodality in multi-objective optimization – More boon than bane?. In *International Conference on Evolutionary Multi-Criterion Optimization (EMO 2019) (Lecture Notes in Computer Science, Vol. 11411)*. Springer, East Lansing, MI, USA, 126–138.
- [13] S. A. Kauffman. 1993. *The Origins of Order*. Oxford University Press.
- [14] P. Kerschke, H. Hoos, F. Neumann, and H. Trautmann. 2019. Automated algorithm selection: Survey and perspectives. *Evolutionary Computation* 27, 1 (2019), 3–45.
- [15] P. Kerschke and H. Trautmann. 2016. The R-Package FLACCO for exploratory landscape analysis with applications to multi-objective optimization problems. In *IEEE Congress on Evolutionary Computation (CEC 2016)*. 5262–5269.
- [16] P. Kerschke and H. Trautmann. 2019. Comprehensive feature-based landscape analysis of continuous and constrained optimization problems using the R-package flacco. In *Applications in Statistical Computing – From Music Data Analysis to Industrial Quality Improvement*. Springer, 93–123.
- [17] P. Kerschke, H. Wang, M. Preuss, C. Grimme, A. H. Deutz, H. Trautmann, and M. T. M. Emmerich. 2019. Search Dynamics on Multimodal Multiobjective Problems. *Evolutionary Computation* 27, 4 (2019), 577–609.
- [18] J. Knowles and D. Corne. 2003. Instance generators and test suites for the multiobjective quadratic assignment problem. In *Evolutionary Multi-Criterion Optimization (EMO 2003) (Lecture Notes in Computer Science, Vol. 2632)*. Springer, Faro, Portugal, 295–310.
- [19] S. Kukkonen and J. Lampinen. 2005. GDE3: The third evolution step of generalized differential evolution. In *IEEE Congress on Evolutionary Computation (CEC 2005)*. IEEE Press, 443–450.
- [20] A. Liaw and M. Wiener. 2002. Classification and regression by RandomForest. *R News* 2, 3 (2002), 18–22.
- [21] A. Liefvooghe, F. Daolio, S. Verel, B. Derbel, H. Aguirre, and K. Tanaka. 2020. Landscape-aware performance prediction for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation* 4, 9 (2020), 1063–1077.
- [22] K. Malan. 2021. A survey of advances in landscape analysis for optimisation. *Algorithms* 14, 2 (2021).
- [23] K. Malan and A. Engelbrecht. 2014. Fitness landscape analysis for metaheuristic performance prediction. In *Recent Advances in the Theory and Application of Fitness Landscapes*. Springer, 103–132.
- [24] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph. 2011. Exploratory landscape analysis. In *Genetic and Evolutionary Computation Conference (GECCO 2011)*. ACM, Dublin, Ireland, 829–836.
- [25] I. Moser, M. Gheorghita, and A. Aleti. 2017. Identifying features of fitness landscapes and relating them to problem difficulty. *Evolutionary computation* 25, 3 (2017), 407–437.
- [26] L. Paquete, T. Schiavinotto, and T. Stützle. 2007. On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research* 156, 1 (2007), 83–97.
- [27] J. R. Rice. 1976. The algorithm selection problem. *Advances in Computers* 15 (1976), 65–118.
- [28] Donald Shepard. 1968. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 23rd ACM national conference*. ACM, 517–524.
- [29] T. Smith, P. Husbands, and M. O’Shea. 2002. Fitness landscapes and evolvability. *Evolutionary Computation* 10, 1 (2002), 1–34.
- [30] K. Smith-Miles. 2008. Towards insightful algorithm selection for optimisation using meta-learning concepts. In *International Joint Conference on Neural Networks (IJCNN 2008)*. IEEE, Hong Kong, 4118–4124.
- [31] T. Therneau and B. Atkinson. 2018. *rpart: Recursive partitioning and regression trees*. R package version 4.1-13.
- [32] S. Verel, A. Liefvooghe, L. Jourdan, and C. Dhaenens. 2013. On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives. *European Journal of Operational Research* 227, 2 (2013), 331–342.
- [33] E. D. Weinberger. 1990. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics* 63, 5 (1990), 325–336.
- [34] A-C. Zăvoianu, B. Lacroix, and J. McCall. 2020. Comparative run-time performance of evolutionary algorithms on multi-objective interpolated continuous optimisation problems. In *Parallel Problem Solving from Nature (PPSN XVI)*. Springer, 287–300.
- [35] A-C. Zăvoianu, S. Saminger-Platz, E. Lughofer, and W. Amrhein. 2018. Two enhancements for improving the convergence speed of a robust multi-objective coevolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2018)*. ACM, 793–800.
- [36] Q. Zhang, W. Liu, and H. Li. 2009. *The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances*. Technical Report. School of CS & EE, University of Essex.
- [37] E. Zitzler. 1999. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Ph.D. Dissertation. Swiss Federal Institute of Technology.
- [38] E. Zitzler, M. Laumanns, and L. Thiele. 2002. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*. International Center for Numerical Methods in Engineering (CIMNE), 95–100.