Cost-vs-accuracy of sampling in multi-objective combinatorial exploratory landscape analysis

Raphaël Cosson, Bilel Derbel, Arnaud Liefooghe, Sébastien Vérel, Hernán E. Aguirre, Qingfu Zhang, Kiyoshi Tanaka

Cost-vs-Accuracy of Sampling in Multi-objective Combinatorial Exploratory Landscape Analysis

Raphaël Cosson
Univ. Lille, CNRS, Inria, CRIStAL
F-59000 Lille, France
raphael.cosson@univ-lille.fr

Bilel Derbel
Univ. Lille, CNRS, Inria, CRIStAL
F-59000 Lille, France
bilel.derbel@univ-lille.fr

Arnaud Liefooghe
Univ. Lille, CNRS, Inria, CRIStAL
F-59000 Lille, France
arnaud.liefooghe@univ-lille.fr

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

Hernan Aguirre
Shinshu University, Faculty of Engineering
Nagano, Japan
ahernan@shinshu-u.ac.jp

Qingfu Zhang
City University of Hong Kong, Kowloon Tong
Hong Kong
qingfu.zhang@cityu.edu.hk

Kiyoshi Tanaka
Shinshu University, Faculty of Engineering
Nagano, Japan
ktanaka@shinshu-u.ac.jp

ABSTRACT

The design of effective features enabling the development of automated landscape-aware techniques requires to address a number of inter-dependent issues. In this paper, we are interested in contrasting the amount of budget devoted to the computation of features with respect to: (i) the effectiveness of the features in grasping the characteristics of the landscape, and (ii) the gain in accuracy when solving an unknown problem instance by means of a feature-informed automated algorithm selection approach. We consider multi-objective combinatorial landscapes where, to the best of our knowledge, no in depth investigations have been conducted so far. We study simple cost-adjustable sampling strategies for extracting different state-of-the-art features. Based on extensive experiments, we report a comprehensive analysis on the impact of sampling on landscape feature values, and the subsequent automated algorithm selection task. In particular, we identify different global trends of feature values leading to non-trivial cost-vs-accuracy trade-off(s). Besides, we provide evidence that the sampling strategy can improve the prediction accuracy of automated algorithm selection. Interestingly, this holds independently of whether the sampling cost is taken into account or not in the overall solving budget.

CCS CONCEPTS

• **Applied computing** → *Multi-criterion optimization and decision-making*; • **Theory of computation** → *Evolutionary algorithms.*

KEYWORDS

Multi-objective optimization, landscape analysis, automated algorithm selection, NK-landscapes.

ACM Reference Format:
Raphaël Cosson, Bilel Derbel, Arnaud Liefooghe, Sébastien Verel, Hernan Aguirre, Qingfu Zhang, and Kiyoshi Tanaka. 2022. Cost-vs-Accuracy of Sampling in Multi-objective Combinatorial Exploratory Landscape Analysis. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512290.3528731

1 INTRODUCTION

Context and motivations. Since the seminal work of Rice [24], the design of high-level automated algorithm selection (AAS) approaches has received a growing attention. In the last decade, exploratory landscape analysis (ELA) [8, 10, 18] has emerged as a state-of-the-art methodology supporting the combination of landscape feature extraction with the development of automated recommendation systems integrating the so-considered features on the basis of statistical or machine learning (ML) models. Such a methodology is especially sound in the context of *blackbox* optimization problems, having unknown characteristics, and for which different algorithms can expose different performance profiles as a function of the problem instance being solved. In this context, fitness landscape analysis [7] allows one to compute meaningful features characterizing the search difficulty of the (blackbox) landscape underlying a given instance; e.g., in terms of difficulty, ruggedness, or multi-modality. Following a standard supervised ML methodology, a model can then be trained on the basis of extensive experiments on a set of known instances. The training phase leads to a mapping of (pre-computed) landscape feature values to the performance of (pre-executed) algorithms. Given a new unseen instance, features can thus be computed and provided as input to the trained model in order to obtain a prediction about the most suitable algorithm

for solving the instance. A key ingredient for the success of such an ELA approach relies on the ability to design cheap and meaningful features. In this paper, we are interested in studying the impact of feature extraction for *multi-objective combinatorial* optimization.

Independently of the nature of the optimization domain (i.e., continuous or discrete, single- or multi-objective), extracting blackbox features requires a particular *sampling* of solutions from the search space. This consists in evaluating the objective values of a carefully-chosen set of solutions, on the basis of which a numerical statistic, constituting the feature value, is computed. The sampling has two main critical implications. First, the feature value can be different depending on the *type* and the *size* of the sampling. This has a direct consequence on the quality of the ML model, and hence on the accuracy of an AAS approach. Second, one has to accommodate the cost of sampling, since the sample size impacts the overall budget, in terms of the number of evaluations affordable to solve a new unseen instance. Consequently, computing features that are both as informative as possible, and as cheap as possible, is a critically important issue. Our work aims at pushing a step towards a better understanding of the impact of the sampling type and cost on the design of a successful multi-objective combinatorial ELA approach.

Related work. In the *continuous* domain, eliciting the impact of sampling is relatively well studied. On the one hand, different sampling techniques such as uniform sampling, latin hypercube, or Sobol sequence were studied in the past [10, 16, 19]. Very recently, it was shown that, quoting the authors in [22, 23], "feature values are not absolute ... [and] cannot be interpreted as stand-alone measures" independently of the method being used for sampling. Such a statement was supported by an analysis of the high-level properties of features, such as their expressiveness and their robustness, with respect to sampling. On the other hand, the computational effort needed for feature computation was studied in the past [9, 14, 22, 23, 26]. For instance, recommendations for computing cheap features are given in [9]. Reviewing all the literature is out of the scope of this paper, since our focus is on multi-objective combinatorial optimization, but let us however remark that very few features can be interchangeably considered for continuous and discrete domains. In fact, sampling in discrete domains is fundamentally different from sampling in continuous domains, and so are the existing features.

In *combinatorial* domains, a fitness landscape is generally defined by specifying a neighborhood relation among solutions; e.g., bit-flip for bit-strings or swap for permutations. One can find different features defined by assuming a full enumeration of the search space. Although this allows for a more fundamental understanding of combinatorial landscapes, it can only be performed for small landscapes, and it is not a realistic option for AAS. Hence, relying on a sampling method is mandatory. The most common sampling technique is to perform some particular *walk* over the landscape. That is, starting from an initial solution, some neighbors are evaluated, a new current solution is chosen, and so on until some condition is satisfied. The solutions collected during a walk, and their neighbors, are used to define the landscape features. State-of-the-art features are typically based on solutions sampled by either random or adaptive walks [7, 25, 31]. Roughly speaking, this is intended to inform about the challenges that search algorithms have to face when exploring the landscape underlying a blackbox optimization problems, such as the landscape ruggedness or the distribution of local optima.

Although we can find a number of analyses [27–29] with respect to feature sampling in *single-objective* combinatorial optimization, very few lessons can be learnt when it comes to integrate those features in an effective AAS approach. In *multi-objective* combinatorial optimization, which is our main focus, few studies can be reported. In a state-of-the-art feature-based approach [12], the cost of sampling is shown to represent a relatively low proportion of the overall budget dedicated to running the automatically-selected algorithm. In [13], a similar observation is also reported. Despite such observations, the question of how the sampling influences the accuracy of a feature-based ELA approach remains open.

Contribution and methodology. In this paper, we provide the first in-depth investigations on the impact of sampling in multi-objective combinatorial landscape analysis. We rely on the fact that the computation of multi-objective landscape features are mainly influenced by: the type of the walk, the length of the walk, and the explored portion of the neighborhood. We hence propose to conduct a systematic empirical analysis of the impact of each factor, with a particular focus on the sampling cost issues. More precisely, our contribution can be summarized as follows:

- Considering two conventional sampling methods from the literature, we adjust the amount of budget devoted to the feature computation by: (i) constraining the number of neighbors to be evaluated along the walk, and (ii) controlling the length of the walk when the sampling method allows us to do so. Such settings are then analyzed using an extensive number of features taken from two state-of-the-art sets: dominance- and indicator-based features [12], and decomposition-based features [3].
- These cost-adjustable features are analyzed in a fine-grained manner by eliciting their correlation with global problem characteristics, and in a coarse-grained manner when integrated within an AAS approach. This allows us to contrast the amount of budget used for feature computation with respect to: (i) the ability of feature values to grasp the problem characteristics, and (ii) the gain in approximation quality when solving an unknown problem instance.
- We report a comprehensive study using a broad range of ρmnk-landscapes and a portfolio of three variants of the MOEA/D algorithm. We show that different global trends of feature values and non-trivial cost-vs-accuracy trade-off(s) can be distinguished. Depending on the feature under consideration, increasing the walk length or increasing the number of visited neighbors do not always lead to more informative feature values. Besides, controlling the cost of features computation can improve the accuracy of the AAS task. Interestingly, this holds independently of whether the sampling cost is taken into account in the available budget, or not.

Outline. In Section 2, we provide the necessary background. In Section 3, we discuss the sampling type and cost. In Section 4, we provide a descriptive analysis of feature values. In Section 5, we report the impact of sampling on high-level ELA tasks. In Section 6, we conclude the paper.

2 MULTI-OBJECTIVE LANDSCAPE ANALYSIS

2.1 Multi-objective Optimization

A multi-objective combinatorial optimization problem (MCOP) can be defined by a set of $\mathtt{m}$ objective functions $f = (f_1, \ldots, f_{\mathtt{m}})$, and a discrete set X of feasible solutions in the *decision space*. Let $Z = f(X) \subseteq \mathbb{R}^{\mathtt{m}}$ be the set of feasible outcome vectors in the *objective space*. To each solution $x \in X$ is assigned an objective vector $z \in Z$, on the basis of the vector function $f : X \to Z$. When maximizing, an objective vector $z \in Z$ is *dominated* by a vector $z' \in Z$ iff $\forall m \in \{1, \ldots, \mathtt{m}\}$, $z_m \leqslant z'_m$ and $\exists m \in \{1, \ldots, \mathtt{m}\}$ s.t. $z_m < z'_m$. A solution $x \in X$ is dominated by a solution $x' \in X$ iff $f(x)$ is dominated by $f(x')$. A solution $x^\star \in X$ is *Pareto optimal* if there does not exist any other solution $x \in X$ such that $x^\star$ is dominated by x. The set of all Pareto optimal solutions is the *Pareto set*. Its mapping in the objective space is the *Pareto front*. Computing a Pareto set approximation is a difficult task, for which multi-objective evolutionary algorithms (MOEAs) constitute an effective option [4].

2.2 Multi-objective Landscape Features

Characterizing the landscape of a MCOP and its impact on the performance of MOEAs is a well-established research area since several decades; see, e.g., [1, 6, 11, 20]. One can distinguish three general categories of landscape features: (i) features based on *problem-specific* knowledge, (ii) blackbox *global* feature requiring the full enumeration of the search space, and (iii) blackbox *local* features relying on sampling the search space. We are specifically interested in the latter, since it enables the effective development of high-level feature-based ELA approaches.

We consider an extensive number of blackbox local features taken from two recent state-of-the-art studies [3, 12]. These features are based on the specification of a neighborhood relation $\mathcal{N} : X \to 2^X$ among solutions. Features are defined on the basis of a *walk* on the landscape. More formally, a walk is an ordered sequence of *neighboring* solutions denoted $\mathcal{W}_{|\ell|} := (x^0, x^1, \cdots, x^\ell)$, such that $x^0 \in X$, and $x^t \in \mathcal{N}(x^{t-1})$ for every $t \in \{1, \cdots, \ell\}$ [7, 25, 31]. For the sake of clarity, the description of the different methods that we consider to perform a walk is delayed to Section 4. We first provide an overview of the considered features, as summarized in Table 1.

2.3 Decomposition-based Features

Decomposition-based landscape features [3] are defined by aggregating single-objective features obtained using objective space decomposition [32]. Given μ weight vectors in the objective space and a scalarizing function, such as the Chebychev function [5], μ scalarizing (single-objective) sub-problems are defined. For each sub-problem, a single-objective feature is first computed on the basis of some walk $\mathcal{W}_{|\ell|}$. For each single-objective feature, the so-obtained μ values are then aggregated to compute a statistic constituting the multi-objective feature value. Hence, one needs to specify how the single-objective features are defined and how their values are aggregated over sub-problems [3]. For aggregating, four simple statistics are considered: the mean (avg), the standard deviation (sd), the first (p1) and second (p2) coefficients of a second order polynomial regression to fit the single-objective features as a function of the ordered indices of sub-problem weight vectors. This is indicated using the letter *a* in the notation of Table 1.

Table 1: Summary of the 121 considered multi-objective landscape features ($s \in \{$avg, sd, r1, kr, sk$\}$, $a \in \{$avg, sd, p1, p2$\}$).

decomp. [3]	fv.*s*.*a*	scalar fitness values
	fd.*s*.*a*	fitness difference between a solution and its neighbors
	in.*s*.*a*	proportion of improving neighbors
	spd.*s*.*a*	maximum distance between improving sub-problems
dominance [12]	inf.*s*	proportion of dominated neighbors
	sup.*s*	proportion of dominating neighbors
	inc.*s*	proportion of incomparable neighbors
	lnd.*s*	proportion of locally non-dominated neighbors
	lsupp.*s*	proportion of supported locally non-dominated neighbors
indic. [12]	hv.*s*	solution's hypervolume
	hvd.*s*	neighborhood's hypervolume
	nhv.*s*	difference of hypervolume between a solution and its neighbors
[3, 12]	law	length of adaptive walk

Four categories of single-objective features with respect to each sub-problem are considered. The first one, denoted by fv. ⋆ .⋆ in Table 1, informs about the distribution of the scalarized single-objective *fitness values* along the walk $\mathcal{W}_{|\ell|}$. More precisely, the mean (avg), the standard deviation (sd), the first auto-correlation coefficient (r1) [7], the kurtosis (kr) and the skewness (sk) of the fitness values of solutions collected during the walk $\mathcal{W}_{|\ell|}$ is computed. This is indicated using the letter *s* in Table 1. For example, the feature fv.r1.avg refers to the average (over the sub-problems) of the first auto-correlation coefficient (over the sequence of solutions of the walk). The second category, denoted fd.⋆.⋆, computes for every solution $x^i \in \mathcal{W}_{|\ell|}$, the mean *fitness difference* with its neighbors $\mathcal{N}(x^i)$. The mean fitness difference values are then combined using the same $s \in$ {avg, sd, r1, kr, sk} statistics. Similarly, in the third category, denoted in. ⋆ .⋆ for *improving neighbors*, for every solution in the walk, the proportion of neighbors that improve the solution's scalarized fitness value (w.r.t. the considered single-objective sub-problem) is computed. The last category, denoted spd. ⋆ .⋆, computes the maximum *distance* (normalized by μ) between every two *sub-problems* improved by a solution in the neighborhood $\mathcal{N}(x^i)$ of every $x^i \in \mathcal{W}_{|\ell|}$. The interested reader is referred to [3] for more details. In total, we consider $4_{(\text{categories})} \times 5_{(s)} \times 4_{(a)} = 80$ decomposition-based landscape features.

2.4 Dominance- and Indicator-based Features

The second considered set contains landscape features based on the *dominance relation* or on the *hypervolume indicator* [12]. They are organized in 8 categories in Table 1. For each category, a measure is computed with respect to the solutions collected by means of a walk $\mathcal{W}_{|\ell|}$. Akin to decomposition-based features, the similar $s \in$ {avg, sd, r1, kr, sk} statistic is then applied to this measure in

order to obtain the feature value. The categories from [12] are as follows: the proportion of *dominated* (inf.⋆), *dominating* (sup.⋆), and *incomparable* (inc.⋆) neighbors with respect to the current solution, the proportion of *locally non-dominated* neighboring solutions (lnd.⋆), the proportion of *supported locally non-dominated* neighboring solutions (lsupp.⋆), the *hypervolume of a solution* (hv.⋆), the solution's *neighborhood hypervolume* (nhv.⋆), and the *difference of hypervolume* between a solution and its neighbors (hvd.⋆). The interested reader can refer [12] for a detailed description. All these features are based on computing the corresponding measure with respect to every solution in the input walk $\mathcal{W}_{|\ell|}$, and then combining the so-obtained measures using the s statistic, exactly as described previously. In total, we consider $8_{(\text{categories})} \times 5_{(s)} = 40$ dominance- and indicator-based landscape features.

Finally, let us notice that we include one feature which depends explicitly on the type of the walk, namely the length of an adaptive walk (law in Table 1) as will be discussed later. This feature is known to inform about the multi-modality of the landscape [12, 30].

3 COST-ADJUSTABLE FEATURES

3.1 Random and Adaptive Walks

Two methods are generally used in the literature [3, 12] in order to compute the input walk $\mathcal{W}_{|\ell|}$, namely *random walk* and *adaptive walk*. In a random walk, there is no particular criterion to pick the next neighboring solution. Starting with a first (randomly chosen) solution $x^{t=0}$, at each iteration $t \in \{0, \cdots, \ell - 1\}$, the next solution x^{t+1} is chosen uniformly at random among the neighbors $\mathcal{N}(x^t)$ of solution x^t. The walk length ℓ is a user-defined parameter.

In an adaptive walk, the next solution is selected among the *improving* neighbors in $\mathcal{N}(x^t)$. In a multi-objective setting, improving neighbors are considered with respect to the dominance relation [12, 30]. At each step t of the walk, x^{t+1} is picked among neighbors that dominate x^t. If no such a solution exists, then the walk ends. Besides, an adaptive walk requires to pick one dominating solution. A common strategy is to pick the *first* neighbor that dominates the current solution. Unlike a random walk, the length ℓ of an adaptive walk is determined by the number of steps required to fall into a Pareto local optimal solution x^ℓ. As commented previously, the actual value of ℓ is known to depend on landscape characteristics, and is considered as a feature on its own (law).

All features, with the exception of fv.⋆.⋆ and hv.⋆.⋆, depend *not only* on the sequence of solutions visited during the walk $\mathcal{W}_{|\ell|}$, but their computation also requires to evaluate the whole neighborhood $\mathcal{N}(x^i)$ for every $x^i \in \mathcal{W}_{|\ell|}$. Hence, the feature computation requires $\Theta(\ell \cdot |\mathcal{N}|)$ calls to the evaluation function, where $|\mathcal{N}|$ denotes the size of the neighborhood, and ℓ depends on the type of the walk as discussed previously.

3.2 Truncated Neighborhood-based Features

We are interested in studying the impact of the amount of budget devoted to feature computation. We apply simple modifications to the previous mechanisms in order to control more finely the feature cost. We consider two options: (i) to control the walk length ℓ, and (ii) to control the number of neighbors that need to be evaluated.

The first option is only possible when using a random walk, since the termination step of an adaptive walk cannot be controlled explicitly. This is however a natural option in order to adjust the overall budget which was not studied in a systematic manner. The second option applies for both random and adaptive walks. More specifically, for any integer value $r \leq |\mathcal{N}|$, let us call a truncated neighborhood, denoted $\widetilde{\mathcal{N}_r}$, a neighborhood obtained from the original neighborhood $\mathcal{N}$ by considering solely r neighbors, i.e., $\forall x \in X$, $\widetilde{\mathcal{N}_r}(x) = \{y^1, y^2, \cdots, y^r \mid \forall j \in [1..r], y^j \in \mathcal{N}(x)\}$. As such, we use a truncated neighborhood having r solutions sampled uniformly at random from $\mathcal{N}$. Then, we compute the different features on the basis of the so-defined truncated neighborhood.

On the one hand, these modifications do not affect the sequence of solutions of a random walk. By contrast, an adaptive walk using $\widetilde{\mathcal{N}_r}$ can stop earlier at any step t, if the r randomly-selected neighbors do not include any locally dominating solution, and even if there exist some solutions in $\mathcal{N}(x^t) \setminus \widetilde{\mathcal{N}_r}(x^t)$ that dominate x^t. On the other hand, independently of the type of the walk (i.e., random or adaptive), for every solution x^t in the so-computed walk, solely the r solutions of $\widetilde{\mathcal{N}_r}(x^t)$ are evaluated when computing the statistic required for the final feature value. In other words, although we follow exactly the same feature specification as discussed previously, by replacing the (full) neighborhood by its r-truncated variant, the so-computed feature values can be different.

Additionally, the previous simple modification reduces the feature budget to $\Theta(\ell \cdot r)$. However, it is not clear how this impacts the feature values, nor it is clear how much the so-extracted values are still meaningful. Besides, one may wonder whether a specific choice of the budget is to be preferred to accurately characterize the underlying landscapes. In fact, different combinations of r and ℓ values could be considered, while still constraining the overall budget to the same pre-fixed value.

4 A DESCRIPTIVE PER-FEATURE ANALYSIS

4.1 Experimental Setup

4.1.1 Benchmarks. Following previous works [3, 12], we use ρmnk-landscapes [30] as a problem-independent model for constructing diverse multi-objective multi-modal benchmarks with objective correlation. The function vector $f = (f_1, \ldots, f_i, \ldots, f_\mathtt{m})$ is defined as $f\colon \{0,1\}^\mathtt{n} \mapsto [0,1]^\mathtt{m}$ such that each objective f_i is to be maximized. The objective value $f_i(x)$ of a solution $x = (x_1, \ldots, x_j, \ldots, x_\mathtt{n})$ is an average value of the contributions associated with each variable x_j. Given objective f_i, and variable x_j, a contribution function $f_{ij}\colon \{0,1\}^{\mathtt{k}+1} \mapsto [0,1]$ assigns a real value for every combination of x_j and its $\mathtt{k}$ *epistatic interactions* $\{x_{j_1}, \ldots, x_{j_\mathtt{k}}\}$. The functions are defined by:

$$f_i(x) = \frac{1}{\mathtt{n}} \sum_{j=1}^{\mathtt{n}} f_{ij}(x_j, x_{j_1}, \ldots, x_{j_\mathtt{k}}), \forall i \in \{1, \ldots, \mathtt{m}\}$$

The $\mathtt{k}$ interactions are set uniformly at random among the $(\mathtt{n} - 1)$ variables other than x_j. By increasing the value of $\mathtt{k}$, problem instances can be tuned from smooth to rugged. The f_{ij}-values follow a multivariate uniform distribution of dimension $\mathtt{m}$, defined by an $\mathtt{m} \times \mathtt{m}$ positive-definite symmetric covariance matrix (c_{pq}) s.t. $c_{pp} = 1$ and $c_{pq} = \rho$ for all $p \neq q$ where $\rho > \frac{-1}{\mathtt{m}-1}$ defines the correlation among the objectives.

4.1.2 Parameter setting. We use standard latin hypercube sampling to generate a set of 1 000 bi-objective instances spanning the following ranges: $\mathsf{n} \in \{50, 51, \ldots, 200\}$, $\mathsf{k} \in \{0, 1, 2, \ldots, 8\}$ and $\rho \in]-1, 1]$. A standard bit-flip neighborhood $\mathcal{N}$ is considered, i.e., two solutions are neighbors if their Hamming distance is 1. We consider a truncated neighborhood $\widetilde{\mathcal{N}}_r$ of size $r = \alpha \cdot \mathsf{n}$, where $\alpha \in \{0.05, 0.1, 0.25, 0.5, 1\}$. Notice that $\alpha = 1$ corresponds to the full original neighborhood, whereas for $\alpha = 0.05$ a sample of 5% of the original neighborhood is explored. The two types of walks are experimented. For a random walk, the length is set in the range $\ell \in \{5, 10, 25, 50, 100\}$. For each instance, we thus obtain $1_{(\text{rnd. walk})} \times 5_{(r)} \times 5_{(\ell)} + 1_{(\text{adapt. walk})} \times 5_{(r)} = 30$ possible values for each of the 121 features.

4.2 Random Walk Analysis

We start by studying the impact of the different sampling configurations when *a random walk* is considered. We use two measures to elicit the relationship between the setting of ℓ and r, and the behavior of the so-computed feature values. Firstly, we compute the deviation of feature values with respect to the highest budget-demanding setting of the sampling. More precisely, for every instance, we compute the value $V_{\max}$ for every feature f, obtained when $\alpha = 1$ and $\ell = 100$. Then, we compute for every other value $V_{r,\ell}(\mathsf{f})$ of feature f, obtained with other settings of r and ℓ, the relative deviation to $V_{\max}(\mathsf{f})$, that is, $|V_{\max}(\mathsf{f}) - V_{r,\ell}(\mathsf{f})|/V_{\max}(\mathsf{f})$. Secondly, for each feature (in each sampling setting), we compute the Spearman correlation between the feature value and the value of ρ and k over the considered $\rho\mathsf{mnk}$-landscapes .

4.2.1 Preliminaries. Before going into further details, we found that 30% of features are neither correlated to ρ nor k, hence being of limited interest in the following analysis, given the extensive number of considered configurations. They mostly consist of features relating to the first auto-correlation coefficient ⋆.r1 or the kurtosis statistic ⋆.kr. Among the remaining features, we are able to elicit 4 classes where we observe specific trends of feature values and correlation as a function of the setting of ℓ and r. In the first and second classes, containing respectively 20% and 15% of the overall features, feature values depend exclusively on either ℓ or r. In the third and fourth classes, containing respectively 15% and 20% of the features, they expose a more complex dependency on the values of ℓ and r. These four classes are illustrated in Fig. 1, reporting the average values of the relative deviation over the different instances (top), and the evolution of the Spearman correlation coefficients (bottom), as a function of the sampling cost. Due to space restrictions, only one selected feature from each class is shown in Fig. 1. Two features are taken from the decomposition-based set [3], namely the average maximal distance between improving sub-problems *spd.avg.avg* and the average proportion of improving neighbors for each sub-problem *in.avg.p2*. The two other features are taken from the dominance- and indicator-based set [12], namely the average number of incomparable solutions *inc.avg* and the standard deviation of the neighborhood hypervolume *nhv.sd*. This is discussed in more details below.

4.2.2 First and second feature classes. In the first class, illustrated by feature *nhv.sd* in Fig. 1 (first column), the feature values are mostly impacted by the length of the walk ℓ, while being insensitive to the value of r. More precisely, the higher the value of ℓ, the smaller the relative deviation, independently of r. This indicates that the features converge to some fixed values as a function of ℓ. However, the proportion of evaluated neighbors r has almost *no* impact on the feature values, while implying a substantially higher costs. For instance, computing *nhv.sd* using a 5%-truncated neighborhood and $\ell = 100$ achieves a relative deviation lower than 0.05%, while being 20 times less expensive than a full neighborhood exploration. Besides, our observations with respect to the relative deviation values stay consistent when looking at the Spearman correlations. For instance, the *nhv.sd* feature is mostly correlated with the parameter ρ of the considered landscapes. The correlation changes in a consistent manner with respect to the relative feature deviation. In particular, the larger ℓ, the higher the correlation with ρ. However, the neighborhood proportion has almost no impact. Interestingly, exactly the opposite behavior can be reported for other features, which constitute our second class of features represented by *inc.avg* in Fig. 1 (second column). In this second class, feature values appear to converge to some value as a function of the neighborhood proportion r. However, longer walks have a very small impact on the feature values while leading to a substantially higher cost. For instance, computing the *inc.avg* feature using the full neighborhood and a very small random walk of size $\ell = 5$ achieves less then 2.5% deviation compared to the most expensive setting, while requiring 20 times less evaluations. We also observe that the evolution of the correlation coefficient with respect to ρ follows the same trend as a function of ℓ and r.

4.2.3 Third and fourth feature classes. In the third class of features, represented by *in.avg.p2* in Fig. 1 (third column), both ℓ and r influences substantially the feature value. Roughly speaking, a larger budget, i.e., larger values of ℓ and r, leads to lower relative deviations. However, this trend is not linear with ℓ and r. For instance, for a small neighborhood proportion $\alpha \in \{0.1, 0.05\}$, a clear gap is observed with the other values of $\alpha \in \{0.25, 0.5, 1\}$. A similar observation can be made for relatively small walks of length $\ell \in \{5, 10\}$. However, combining a random walk of length $\ell = 25$ and a neighborhood proportion of $r = 25\%$, is enough to achieve a relative deviation lower than 3% while using 16 times less evaluations compared against the most expensive setting. Notice that the correlation coefficient increases consistently as commented previously for the other classes. Finally, in the fourth class of features, represented by spd.avg.avg in Fig. 1 (forth column), the relative deviation decreases consistently with the cost of sampling. However, the correlation coefficient does not follow the same trend in the sense that intermediate settings can provide substantially higher correlations. For instance, computing the *spd.avg.avg* feature using a small walk of length $\ell \in \{5, 10\}$ and the full neighborhood shows significantly larger correlation values compared against the most expensive setting, while requiring a significantly lower budget.

4.2.4 Discussion. From the previous observations, we can say that there is a complex interaction between the feature values and the sampling configuration. In particular, a higher budget does not systematically lead to more consistent values, independently of the considered feature. Moreover, the settings of ℓ and r can lead to seemingly different features exposing different trade-offs both in

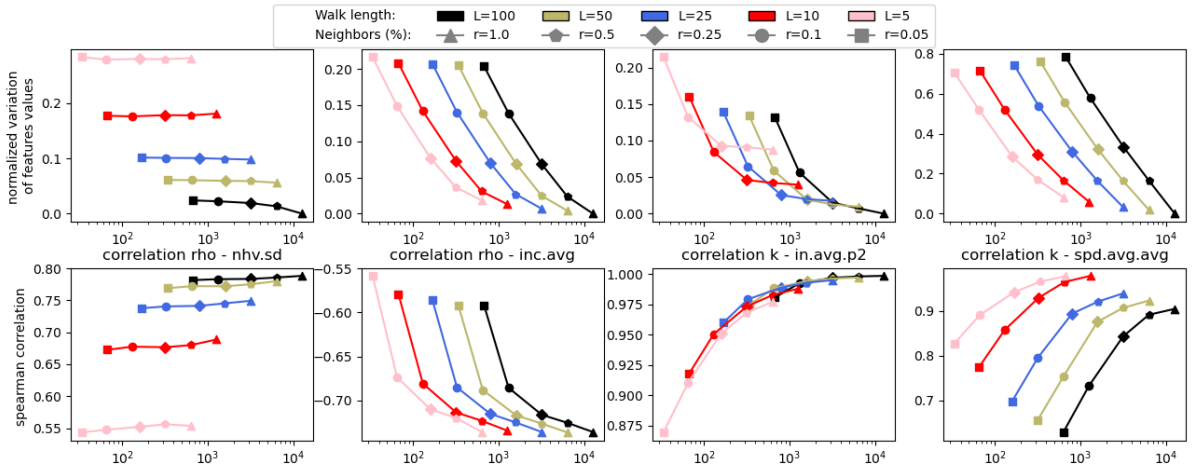

Figure 1: Average feature relative deviation (top), and feature correlation (bottom) with ρ (first and second subfigure) and k (third and fourth subfigures) for random walks. From left to right: nhv.sd, inc.avg, in.avg.p2 and spd.avg.avg. The *x-axis* (in log-scale) refers the number of evaluations of the sampling.

terms of feature values, feature cost, and feature ability to grasp the global characteristics of the underlying landscape. Hopefully, our analysis suggests that there could be some setting of ℓ and r leading to reasonably cheap and accurate features. This will be studied in more details in Section 5. In the following, we first complement our analysis with respect to the adaptive walk.

4.3 Adaptive Walk Analysis

4.3.1 Feature values. Firstly, we report in Fig. 2 (left) the relative deviation of features with respect to the values obtained using an adaptive walk with the largest budget ($\ell = 100$ and $r = 100\%$). We observe that the relative deviation spans a wide range (from 2% to more than 60%), as a function of r. Actually, the largest deviations are with respect to very small r values. This is to contrast with random walks where, although r was found to have a significant impact, the range of the relative deviation is relatively small, which indicates that the size of the r-truncated neighborhood is critical for an adaptive walk. In general, the feature values converge as r increases. However, some features are found to be less sensitive to r, as illustrated by the in.avg.p2 and the spd.avg features, for which a small value of r allows to obtain a significantly lower deviation range. Overall, an adaptive walk visiting 25% of the neighborhood provides feature values which are relatively close to a full neighborhood (up to 5%), while requiring a significantly lower budget. One should however keep in mind that given the range of feature values, even such a small deviation is not necessarily negligible.

4.3.2 Correlation values. Secondly, in Fig. 2 (middle and right), we report the evolution of the correlation between feature values and the benchmark parameters ρ and k. In the situations where a feature shows a significant (positive or negative) correlation with either ρ (e.g., rhv.sd or inc.avg) or k (e.g., spd.avg) when using a full neighborhood, the strength of the correlation decreases when extracting the feature with decreasing values of r. For the other situations, no general tendency can be reported. From our collected data, we can state that computing features with an adaptive walk using the original neighborhood seems to expose the strongest correlation with the benchmark characteristics for the most part.

5 HIGH-LEVEL ELA TASKS

In the second part of our analysis, we consider two high-level ELA tasks: (i) predicting the global benchmark parameters ρ and k, and (ii) selecting the (per-instance) best performing algorithm in a portfolio. For both tasks, we use the exact same set of ρmnk-landscapes described previously.

5.1 Task #1: Predicting Benchmark Parameters

5.1.1 Task description. This task informs about the power of a predictive model as a function of the sampling. We train one model with each possible configuration of the sampling in order to fit respectively the k and ρ parameters of the input benchmarks. As in [3], we use a random forest [2] classification model for k, and a random forest regression model for ρ. Each model is trained on the basis of the input feature values using a default value of 100 trees [21]. We then focus on the models' R^2 values, as summarized in Fig. 3. The higher the R^2 value of a model, the more explainable the variance, in the benchmark parameter ρ or k, using the underlying sampling.

5.1.2 Results and Discussion. From Fig. 3, we observe that in comparison to a random walk, an adaptive walk provides a poor trade-off in terms of the R^2 value and the implied budget for k. We also notice that the R^2 has a global tendency to increase as the budget underlying the sampling increases. However, reasonably high R^2 values are obtained at lower costs by adjusting the sampling parameters. For instance, an adaptive walk with half of the neighborhood ($r = 50\%$) obtains almost the same R^2 values (0.94 for ρ and 0.84 for k) while reducing the feature cost by half in average. Looking more carefully at the random walk configurations, we find a number of settings of ℓ and r that obtain the same high level of accuracy, but

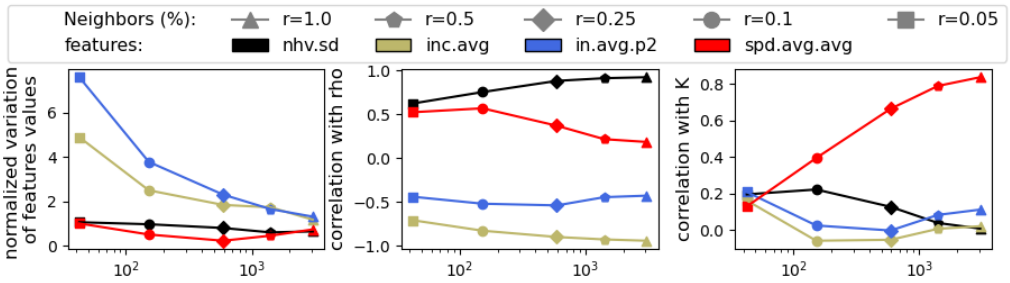

Figure 2: Average feature relative deviation (left), and feature correlation with ρ (middle) and k (right) for adaptive walks. The *x-axis* (in log-scale) refers the number of evaluations of the sampling.

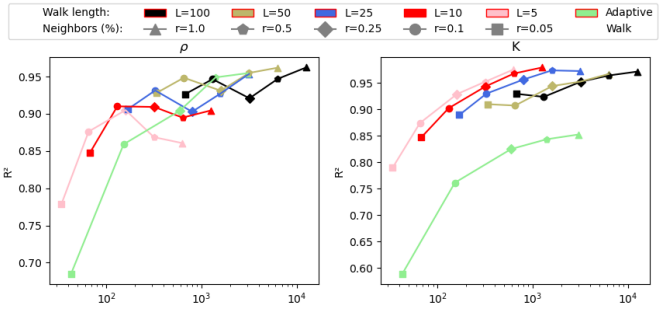

Figure 3: Average R^2 (over 50 repetitions) of random forest models for predicting ρ (left) and k (right). The *x-axis* (in log-scale) refers the number of evaluations.

with a drastic saving in the budget when compared against the most expensive setting. Interestingly, we found that the configuration providing the best trade-offs are different depending on the target prediction task. For k, a very small random walk ($\ell = 5$) and the full neighborhood exploration ($r = 100\%$) provides among the highest R^2 values (0.97) while requiring 20 times less budget than the most expensive setting. In fact, it appears that the most important sampling parameter for predicting k is the proportion r of the neighborhood. For ρ, a configuration using a walk length $\ell = 50$ and very small proportion of the neighborhood $r = 10\%$ provides among the highest R^2 values (0.94) while requiring 20 times less budget than the most expensive setting. Actually, the R^2 value with respect to ρ is mostly impacted by the length of the walk, provided that the proportion of evaluated neighbors is at least $r = 10\%$.

5.2 Task #2: Automated Algorithm Selection

5.2.1 Task description. The second and more sophisticated task aims at studying the impact of the sampling cost when tackling the AAS problem [24] using a feature-informed ELA approach. For this purpose, we consider the exact same portfolio of algorithms as in [3]. The reader is referred there for more details. For completeness, we comment that this portfolio is composed of three variants of the state-of-the-art MOEA/D algorithm that are shown to have different performance profiles depending on the considered ρmnk-landscapes [17]. We measure algorithm performance in terms of hypervolume relative deviation [33] to the best-known Pareto front collected over all runs for every instance. Following a standard supervised ML approach, we split the 1000 ρmnk instances into a training set $\mathcal{T}$ and a testing set $\mathcal{I}$. We then train a multi-output random forest regression model [2] in order to fit the relative hypervolume deviation of the different algorithms on the basis of the feature extracted with respect to each instance in the training set $\mathcal{T}$. Afterwards, given an unseen instance from the testing set $\mathcal{I}$, the features are first computed, and the performance of each algorithm is then predicted on the basis of the so-trained model. The algorithm with the best predicted performance is selected as the recommended one. Since extracting the features has a cost, we distinguish two scenarios for the purpose of our analysis. In the first scenario, the selected algorithm is run with the maximum budget B, hence not counting the sampling cost. In the second more realistic scenario, the algorithm is run with a budget of B − FB, where FB is the number of evaluations used to compute the features.

For performance assessment, we consider an adaptation of the merit measure [15]. Let $\mathrm{rhv}(A, i)$ be the average relative hypervolume deviation of a given algorithm A using the maximum allowed budget B on an instance i. Similarly, let $\mathrm{rhvc}(A, i)$ be the average relative hypervolume deviation of algorithm A when subtracting the cost of feature computation, that is when running the algorithm with a budget of B − FB. Let $\widetilde{\mathrm{rhvc}}(A, J)$ and $\overline{\mathrm{rhv}}(A, J)$ be the average over a given set of instances J of the relative hypervolume deviation, respectively when the feature cost is or not taken into account. The so-called *single best solver* (SBS) is the algorithm with the best $\overline{\mathrm{rhv}}$ value on the training set $\mathcal{T}$; i.e., $SBS = \arg\min_A \overline{\mathrm{rhv}}(A, \mathcal{T})$. Notice that the SBS does not need any feature computation. The *virtual best solver* (VBS) is the oracle providing the best $\mathrm{rhv}(\cdot, i, \mathrm{B})$ value for each testing instance $i \in \mathcal{I}$. Let us call the Automated Solver (AS) the algorithm selected by the trained model. The *merit* measures the gain in quality of AS relatively to SBS and VBS. For the purpose of our analysis, we consider two variants of the merit where the sampling cost is either taken into account (m) or not (m′):

$$\mathrm{m} = \frac{\widetilde{\mathrm{rhvc}}(\mathrm{AS}, \mathcal{I}) - \overline{\mathrm{rhv}}(\mathrm{VBS}, \mathcal{I})}{\overline{\mathrm{rhv}}(\mathrm{SBS}, \mathcal{I}) - \overline{\mathrm{rhv}}(\mathrm{VBS}, \mathcal{I})}; \mathrm{m}' = \frac{\overline{\mathrm{rhv}}(\mathrm{AS}, \mathcal{I}) - \overline{\mathrm{rhv}}(\mathrm{VBS}, \mathcal{I})}{\overline{\mathrm{rhv}}(\mathrm{SBS}, \mathcal{I}) - \overline{\mathrm{rhv}}(\mathrm{VBS}, \mathcal{I})}$$

A merit of $m = 0$ corresponds to the perfect oracle. A merit $\mathrm{m} < 1$ (respectively $\mathrm{m} > 1$) indicates that using the model performs better (respectively worse) than SBS.

Table 2: Merit average values for the different sampling configurations.

	Merit with feature cost (m)						Merit without feature cost (m′)					
	Random Walk					Adaptive	Random Walk					Adaptive
r	$\ell = 100$	$\ell = 50$	$\ell = 25$	$\ell = 10$	$\ell = 5$	Walk	$\ell = 100$	$\ell = 50$	$\ell = 25$	$\ell = 10$	$\ell = 5$	Walk
100%	0.537	0.529	0.562	0.577	0.576	0.624	0.513	0.517	0.555	0.573	0.573	0.617
50%	0.521	0.552	0.580	0.620	0.577	0.610	0.507	0.544	0.575	0.616	0.574	0.606
25%	**0.515**	0.566	0.569	0.640	0.634	0.629	**0.507**	0.561	0.566	0.637	0.631	0.626
10%	0.544	0.583	0.566	0.609	0.585	0.672	0.538	0.579	0.563	0.606	0.583	0.670
5%	0.551	0.575	0.552	0.639	0.646	0.692	0.547	0.572	0.550	0.637	0.644	0.690

The maximum budget is set to 10^6 evaluations. The parameters of the three MOEA/D algorithms are exactly the same as in [3]. Each algorithm is executed for 20 independent runs on each instance. We adopt a standard repeated random hold-out strategy with a 90/10% split for training/testing, and we report the average merit value over 100 independent folds.

5.2.2 Results and Discussion. In Table 2, we report the merit values obtained for the different configurations. First, we observe that the merit value m, taking into account the feature cost, is always worse than m′. With no surprise, this indicates that reducing the budget devoted to feature computation can help improving the overall performance. More importantly, the merit m is always lower than 1. This means that a feature-informed AAS approach is always better than SBS, independently of the experimented sampling configuration. This is especially interesting for the cheapest configuration ($\ell = 5, r = 5\%$), implying an extremely constrained feature budget of at most 50 evaluations. Besides, a random walk provides better merit values than an adaptive walk. The merit increases when the sampling size decreases. More precisely, the loss of accuracy is slightly more important when the length of random walk is reduced. We also find that using the most expensive sampling configuration ($\ell = 100, r = 100\%$) is not the most efficient in terms of merit. In fact, a random walk with $\ell = 100$ and $r = 25\%$ requires four time less budget while providing the best merit values. Surprisingly, this holds independently of whether the cost of sampling is taken into account or not. This gain can be attributed to the fact that a relatively small budget (i.e., 0.5% of the global budget) still allows to compute informative features leading to accurate predictions, while leaving more chance for the algorithm to converge. Which is consistent with the results from our first task, where we were able to elicit some cost-vs-accuracy trade-offs when varying the sampling parameters ℓ and r for predicting the values of k and ρ.

Finally, we use the mean Gini impurity measure to extract the importance of each feature from the trained random forest models. We then compute the importance rank of each feature normalized in $[0, 1]$, with 1 being the most important feature. In Fig. 4, we report the 10 most important features when using the most expensive setting ($\ell = 100, r = 100\%$, left) and the sampling providing the best merit ($\ell = 100, r = 25\%$, right). We clearly see that the ten most important features are the same in both settings. This indicates that using a restricted budget does not necessary lead to a major change in feature importance. However, we can observe small variations in the respective ranks of features. Besides, we found that five features fall into the first category elicited in the analysis of Section 4. In particular, the values of the four most important features hvd.sd, lsupp.sd, inc.sd, hv.sd are highly impacted by the length of the walk ℓ. One feature fd.avg.p2 falls into the second category where the proportion of neighbors r is found to have the greatest impact on this feature value. Another feature inf.sd belongs to the third category, increasing its accuracy according to the sampling size. Finally, three features, namely hv.r1, in.sd.avg, and fv.r1.avg fall into the fourth category, where the highest correlation with the landscape characteristics is observed with a reduced sampling size.

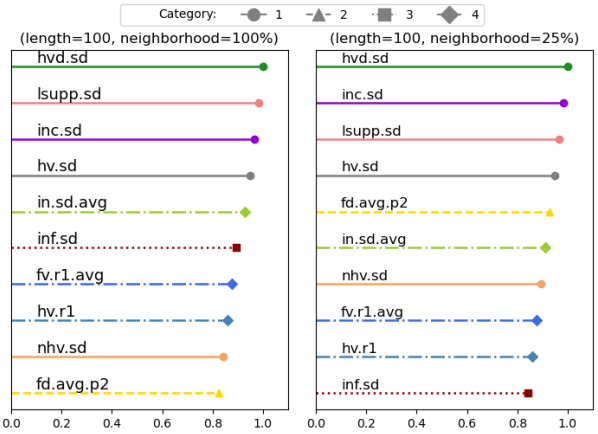

Figure 4: Top 10 most important features. Line styles and shapes refer to the feature categories.

6 CONCLUSION

In this paper, we conducted the first systematic analysis on the impact of sampling on the extraction of multi-objective combinatorial landscape features, and their integration into feature-informed performance prediction and algorithm selection approaches. From extensive experiments using state-of-the-art features, we provide evidence that feature values, and their correlation with unknown benchmark characteristics, expose complex dependencies with the sampling type and cost. In particular, we showed that a random walk using a reasonably small proportion of neighbors leads to cheap and informative feature values. Different questions are however left open. For instance, it would be interesting to extend our analysis to other algorithms, combinatorial domains and objective space dimensions, eventually including real-world problems, where the development of successful landscape-aware approaches is still not fully addressed.

REFERENCES

[1] Hernan Aguirre and Kiyoshi Tanaka. 2007. Working principles, behavior, and performance of MOEAs on MNK-landscapes. *European Journal of Operational Research* 181 (09 2007), 1670–1690. https://doi.org/10.1016/j.ejor.2006.08.004

[2] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32.

[3] Cosson, Derbel, Liefooghe, Aguirre, Tanaka, and Zhang. 2021. Decomposition-Based Multi-objective Landscape Features and Automated Algorithm Selection. In *Evolutionary Computation in Combinatorial Optimization*, Christine Zarges and Sébastien Verel (Eds.). Springer International Publishing, 34–50.

[4] K. Deb. 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. Cambridge University Press. 409–410 pages.

[5] Matthias Ehrgott. 2005. *Multicriteria Optimization (2. ed.)*. Springer. https://doi.org/10.1007/3-540-27659-9

[6] Deon Garrett and Dipankar Dasgupta. 2007. Multiobjective Landscape Analysis and the Generalized Assignment Problem. 110–124. https://doi.org/10.1007/978-3-540-92695-5_9

[7] S. A. Kauffman. 1993. *The Origins of Order*. Oxford University Press.

[8] Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. 2019. Automated Algorithm Selection: Survey and Perspectives. *Evol. Comput.* 27, 1 (2019), 3–45.

[9] Pascal Kerschke, Mike Preuss, Simon Wessing, and Heike Trautmann. 2016. Low-Budget Exploratory Landscape Analysis on Multiple Peaks Models. In *The on Genetic and Evolutionary Computation Conference GECCO*. ACM, 229–236. https://doi.org/10.1145/2908812.2908845

[10] Pascal Kerschke and Heike Trautmann. 2019. Automated Algorithm Selection on Continuous Black-Box Problems by Combining Exploratory Landscape Analysis and Machine Learning. *Evol. Comput.* 27, 1 (2019), 99–127. https://doi.org/10.1162/evco_a_00236

[11] Joshua Knowles and David Corne. 2003. Instance Generators and Test Suites for the Multiobjective Quadratic Assignment Problem. In *Evolutionary Multi-Criterion Optimization*, Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Lothar Thiele, and Kalyanmoy Deb (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 295–310.

[12] Arnaud Liefooghe, Fabio Daolio, Sébastien Verel, Bilel Derbel, Hernan Aguirre, and Kiyoshi Tanaka. 2020. Landscape-aware performance prediction for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation* (2020).

[13] Arnaud Liefooghe, Sébastien Vérel, Bilel Derbel, Hernán E. Aguirre, and Kiyoshi Tanaka. 2020. Dominance, Indicator and Decomposition Based Search for Multi-objective QAP: Landscape Analysis and Automated Algorithm Selection. In *Parallel Problem Solving from Nature - PPSN XVI - 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12269)*, Thomas Bäck, Mike Preuss, André H. Deutz, Hao Wang, Carola Doerr, Michael T. M. Emmerich, and Heike Trautmann (Eds.). Springer, 33–47. https://doi.org/10.1007/978-3-030-58112-1_3

[14] Arnaud Liefooghe, Sébastien Vérel, Benjamin Lacroix, Alexandru-Ciprian Zavoianu, and John A. W. McCall. 2021. Landscape features and automated algorithm selection for multi-objective interpolated continuous optimisation problems. In *Genetic and Evolutionary Computation (GECCO 2021)*. ACM, Lille, France, 421–429. https://doi.org/10.1145/3449639.3459353

[15] Marius Lindauer, Jan N. van Rijn, and Lars Kotthoff. 2018. The Algorithm Selection Competition Series 2015-17. *CoRR* abs/1805.01214 (2018). arXiv:1805.01214

[16] Katherine Mary Malan. 2021. A Survey of Advances in Landscape Analysis for Optimisation. *Algorithms* 14, 2 (2021), 40. https://doi.org/10.3390/a14020040

[17] Gauvain Marquet, Bilel Derbel, Arnaud Liefooghe, and El-Ghazali Talbi. 2014. Shake Them All! - Rethinking Selection and Replacement in MOEA/D. In *PPSN XIII*. 641–651.

[18] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. 2011. Exploratory Landscape Analysis. In *GECCO*. 829–836.

[19] Rachael Morgan and Marcus Gallagher. 2014. Sampling Techniques and Distance Metrics in High Dimensional Continuous Landscape Analysis: Limitations and Improvements. *IEEE Transactions on Evolutionary Computation* 18, 3 (2014), 456–461. https://doi.org/10.1109/TEVC.2013.2281521

[20] Luis Paquete, Tommaso Schiavinotto, and Thomas Stützle. 2007. On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research* 156 (09 2007), 83–97. https://doi.org/10.1007/s10479-007-0230-0

[21] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[22] Quentin Renau, Carola Doerr, Johann Dréo, and Benjamin Doerr. 2020. Exploratory Landscape Analysis is Strongly Sensitive to the Sampling Strategy. In *Parallel Problem Solving from Nature - PPSN XVI - 16th International Conference, PPSN (Lecture Notes in Computer Science, Vol. 12270)*. Springer, 139–153. https://doi.org/10.1007/978-3-030-58115-2_10

[23] Quentin Renau, Johann Dréo, Carola Doerr, and Benjamin Doerr. 2019. Expressiveness and robustness of landscape features. In *The Genetic and Evolutionary Computation Conference Companion, GECCO*. ACM, 2048–2051. https://doi.org/10.1145/3319619.3326913

[24] John R. Rice. 1976. The Algorithm Selection Problem. *Adv Comput* 15 (1976), 65–118.

[25] Hendrik Richter and Andries Engelbrecht (Eds.). 2014. *Recent Advances in the Theory and Application of Fitness Landscapes*. Springer.

[26] Sobia Saleem, Marcus Gallagher, and Ian Wood. 2019. Direct Feature Evaluation in Black-Box Optimization Using Problem Transformations. *Evolutionary Computation* 27, 1 (03 2019), 75–98. https://doi.org/10.1162/evco_a_00247 arXiv:https://direct.mit.edu/evco/article-pdf/27/1/75/1553218/evco_a_00247.pdf

[27] Sarah L. Thomson, Gabriela Ochoa, and Sébastien Vérel. 2019. Clarifying the Difference in Local Optima Network Sampling Algorithms. In *Evolutionary Computation in Combinatorial Optimization - 19th European Conference, EvoCOP (Lecture Notes in Computer Science, Vol. 11452)*. Springer, 163–178. https://doi.org/10.1007/978-3-030-16711-0_11

[28] Sarah L. Thomson, Gabriela Ochoa, Sébastien Vérel, and Nadarajen Veerapen. 2020. Inferring Future Landscapes: Sampling the Local Optima Level. *Evol. Comput.* 28, 4 (2020), 621–641. https://doi.org/10.1162/evco_a_00271

[29] Leonardo Vanneschi, Sébastien Vérel, Marco Tomassini, and Philippe Collard. 2009. NK Landscapes Difficulty and Negative Slope Coefficient: How Sampling Influences the Results. In *Applications of Evolutionary Computing, EvoWorkshops (Lecture Notes in Computer Science, Vol. 5484)*. Springer, 645–654. https://doi.org/10.1007/978-3-642-01129-0_74

[30] Sébastien Verel, Arnaud Liefooghe, Laetitia Jourdan, and Clarisse 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.

[31] E. D. Weinberger. 1990. Correlated and uncorrelatated fitness landscapes and how to tell the difference. *Biol Cybern* 63, 5 (1990), 325–336.

[32] Qingfu Zhang and Hui Li. 2008. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE TEVC* 11 (01 2008), 712 – 731.

[33] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. 2003. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE TEVC* 7, 2 (2003), 117–132.